NTNU – Trondheim
Norwegian University of
Science and Technology

Computer-aided optimization of an offshore jacket for a wind turbine with a simplified load model

Ole Edvard Bakkom

Civil and Environmental Engineering
Submission date: March 2015
Supervisor: Michael Muskulus, BAT

Norwegian University of Science and Technology
Department of Civil and Transport Engineering

PREFACE

This report is my graduate thesis for the degree Master of Science, Structural Engineering. The thesis is written for the Department of Civil and Transport Engineering at The Norwegian University of Technology and Science on the subject of offshore wind turbines. The assignment is given and supervised by associate professor Michael Muskulus.

This work done for this thesis was primarily executed in the spring of 2014. The thesis was interrupted for some time and finished in early 2015. The work done in 2015 mainly consists of writing of this report and some analysis of the results, while the models and simulations where done earlier.

The subject of the report is optimisation and structural redundancy for a jacket structure for an offshore wind turbine. Structural redundancy is added later from the original task description.

Thanks is given to supervisor Michael Muskulus for the task, feedback, discussions and the patience. Also to PhD stud. Daniel Zwick who have provided knowledgeable technical support and the time series' that form the basis for the load cases used for all analyses in this report.

ABSTRACT

In this report the jacket structure of a 5MW reference offshore wind turbine has been explored in terms of structural redundancy and for optimisation of the node locations. This has been done by the creation and utilisation of fully parametric scripts that creates input and post processes results from Abaqus CAE. The dynamic nature of offshore wind turbine forms a basis for complex loading of the jacket structure. This makes the analysis work needed to fully understand such structures substantial. Fatigue may be the limiting factor for structure life time, and this has been the evaluating factor for results for both redundancy and optimisation. Five different load cases are employed to evaluate different weather, and give a fuller picture of what the structure may experience. Differences in the evaluated configurations show a nonlinear performance variation in the load cases, which complicates a clear optimal configuration. A further study of the lifetime expected distribution of load cases could be done to better understand this, but only the findings for the individual load cases has been presented in this report. For structural redundancy, the loss of a structural member shows disparity between members in the legs or braces, where the legs shows greater contribution to the stiffness and strength of the tower.

SAMMENDRAG

Denne oppgaven ser på optimalisering av støttestrukturen til en 5MW offshore vindturbin. Støttestrukturen har blitt analysert med hensyn på nodeplassering og med tanke på strukturell redundans. Det har blitt gjort ved hjelp av flere parametriske skript som genererer og postprosesserer data til og fra programvaren Abaqus CAE.

Offshore vindturbiner opplever kompliserte lastsituasjoner der bølger og vind skaper store dynamiske variasjoner. Dette gjør at analysearbeidet som skal til for å fullt forstå disse konstruksjonene i mange tilfeller er betydelig. Utmatting vil i mange tilfeller være det som begrenser levetiden til konstruksjonen og i denne oppgaven har utmatting blitt brukt som den primære variablene for å måle ytelse for både optimalisering og redundans. I alt fem lasttilfeller har blitt brukt for å analysere ulike deler av livsløpet til vindturbinen. Ulineariteter i responsen mellom lasttilfellene gjør at en entydig optimal løsning ikke nødvendigvis er enkel eller mulig å finne. En mer detaljert gjennomgang av fordelingen av lastsituasjonen gjennom levetiden kunne hjulpet forståelsen av dette, men kun resultater for individuelle lastsituasjoner er presentert her. For strukturell redundans så skiller det seg klart mellom krysstag og de primære beina til strukturen, hvor krysstagene bidrar klart mindre til stivhet og styrke.

TABLE OF CONTENTS

List of Figures

LIST OF TABLES

LIST OF APPENDICES

1 INTRODUCTION

This report is an attempt to explore methods of optimisation for the jacket structure of an offshore wind turbine. Offshore jacket structures experience harsh environmental loading and must carry the loads transferred from the turbine. The motivation for optimisation of the jacket is primarily a pursuit for better economic performance with respect to initial cost or by increasing the lifetime and reducing the chance of structural failure.

Analyses are done in the commercial finite element software Abaqus CAE, and a large number of configurations are included with the help of scripting. The report is in part an introduction to the use of the Abaqus scripting interface, partly on the application to the jacket structure and a summary of the findings. Most of the work behind this report is in the development of the scripts used for result generation, and in the understanding of these mechanisms. But in hours, the running of the scripts are the most substantial. This is due to the highly dynamic nature of the environment in which offshore wind turbines operate, that results in substantial load cases, and inefficiencies in the algorithms used. As a result, a large amount of data has been generated in the making of this report, not all aspects of this have been analysed or represented in this report.

Results are split into structural redundancy checks and node coordinate optimisation. Structural redundancy is explored by the removal of structural members, while node locations are optimised by an iterative approach.

2 THE OC4

The OC4 (Offshore Code Comparison Collaboration Continuation) is a joint research project on a baseline wind turbine. It is designed to be benchmark research in the field of offshore wind turbine. Both operating under the International Energy Agency, the OC4 proceeds the OC3 (Offshore Code Comparison Collaboration) in this research. In OC3, a monopile and a tripod support structure was used as a basis for a fixed-bottom wind turbine (1) (2).

For the OC4, a jacket support structure is used. This jacket is fixed at the ocean floor at 50m water depth and the rotor-nacelle assembly is attached to the jacket through a tubular steel pile (3).

The jacket is grouted to the ocean floor in a configuration shown in figure 1 and 2. This is assumed stiff in both translation and rotation for all calculations.

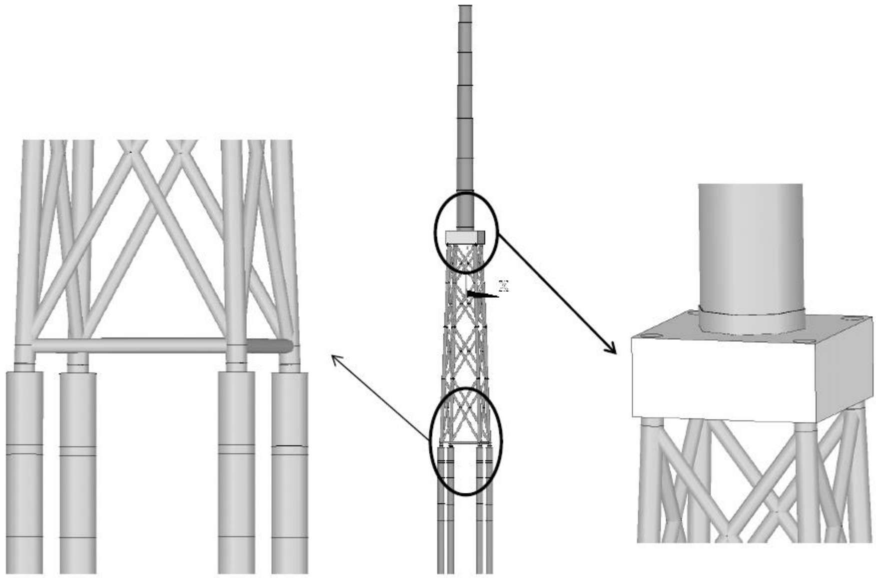

Figure 1 - OC4-configuration

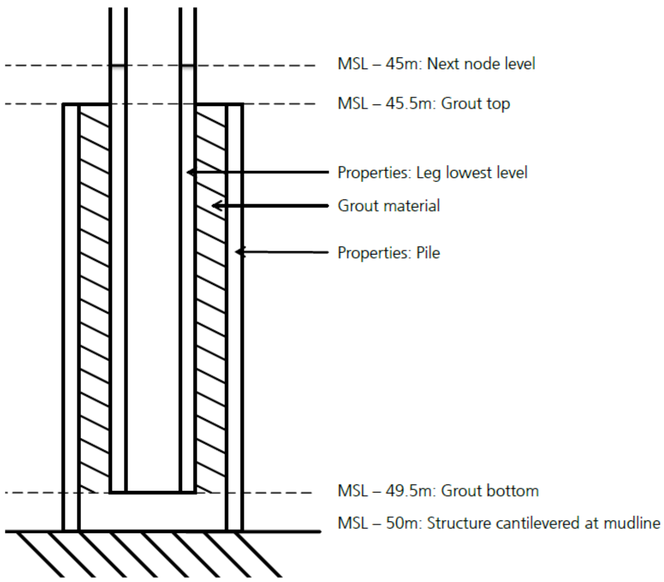

Figure 2 - OC4 grouted connection

The jacket is a steel truss made from several different tubular cross sections. Four legs run the whole height of the jacket supported by a range of braces.

The jacket structure is finely tuned to the rotational speed of the turbine. The turbine has a rotational speed corresponding to a frequency between 0.115 and 0.202Hz, this range is donated 1P and resonance can occur if any eigenfrequency of the jacket where to be in this range. For a 3 bladed turbine, the frequency range which corresponds to the triple frequency, i.e. the combined excitation from every blade (denoted 3P) is also of similar concern.

This gives three possible design philosophies for the tower. Either a design which is stiffer than both 1P and 3P, i.e. a stiff-stiff design, a design that is softer than both, soft-soft. Or a design which is between 1P and 3P, a soft-stiff design. The OC4 jacket is of the third design with a first eigenfrequency of approximately 0.31Hz (4). Using a 10% safety factor on the frequencies puts the structure in the upper range of the allowable range for a soft-stiff design (5). This is illustrated in figure 3.

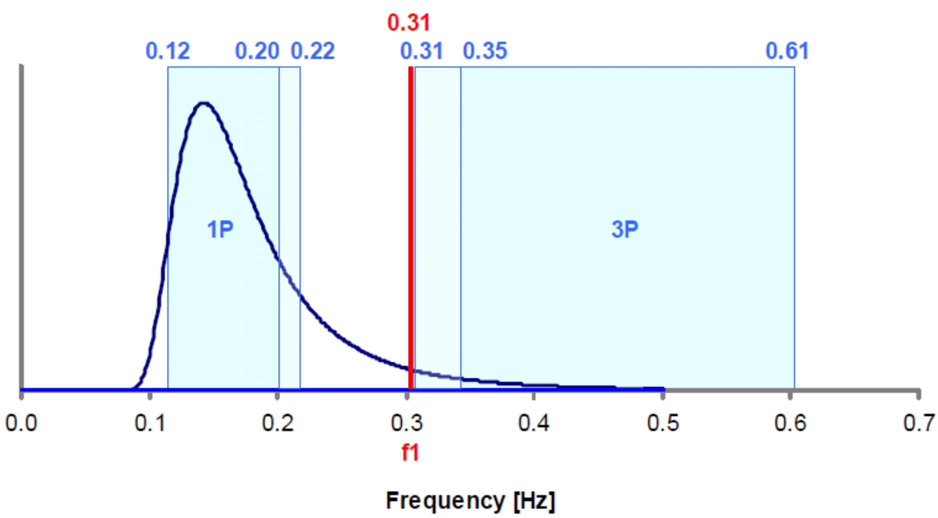

Figure 3 - OC4 rotor frequency (4).

Throughout this report, the structural members of the jacket truss are defined as either part of the legs or braces. In figure 4 the legs are coloured and the braces grey.

The node coordinates are given in appendix 1 and the definition of every member in table 1, the cross section parameters is found in table 2 with indication of use marked in figure 4.

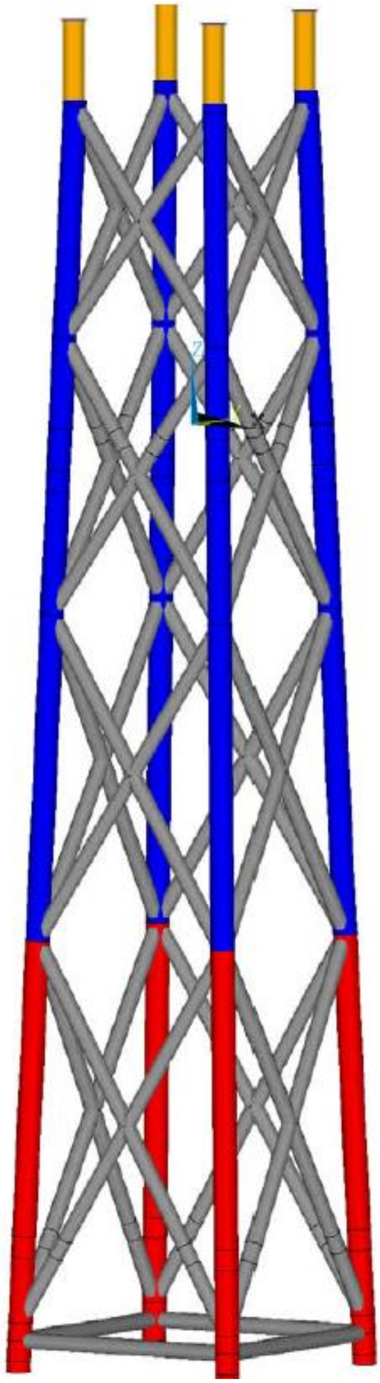

Figure 4 - OC4 member configuration

Description	Name	Position
Jacket leg 1	L1	x > 0; y > 0
Jacket leg 2	L2	x < 0; y > 0
Jacket leg 3	L3	x < 0; y < 0
Jacket leg 4	L4	x > 0; y < 0
Jacket side 1	S1	leg 1 to leg 2
Jacket side 2	S2	leg 2 to leg 3
Jacket side 3	S3	leg 3 to leg 4
Jacket side 4	S4	leg 4 to leg 1
Intersection level at upper Y-Joints	YUp	z = 15.615m
Intersection level at highest X-Joints	X1	z = 10.262m
Intersection level at 2 X-Joints	X2	z = - 1.958m
Intersection level at 3 X-Joints	X3	z = - 16.371m
Intersection level at 4 X-Joints	X4	z = - 33.373m
Intersection level at highest K-Joints	K1	z = 4.378m
Intersection level at middle K-Joints	K2	z = - 8.922m
Intersection level at lower K-Joints	K3	z = - 24.614m
Intersection level at lower Y-Joints	YBottom	z = - 43.127m
Intersection level at mud brace	Mud brace	z = - 44.001m

Table 1 - OC4 jacket definition

Property set	*Component*	*Color in figure 4*	*Other diameter [m]*	*Thickness [mm]*
1	X- and mud braces	Grey	0.8	20
2	Leg at lowest level	Red	1.2	50
3	Leg 2 to 4 level	Blue	1.2	35
4	Leg crossing TP	Orange	1.2	40
5	Pile	Not shown	2.082	60

Table 2 - OC4 jacket cross section properties

3 ABAQUS CAE AND THE SCRIPTING INTERFACE

For this chapter, reference is made to the Abaqus CAE 6.13-4 Documentation (6). The documentation contains a brief introduction to all functions used, this is further expanded in the book *Python Scripts for Abaqus* (7) which also contains examples for common tasks.

The Abaqus CAE scripting interface is an extension for the programming language Python, to run Abaqus functions by the use of a script. This gives additional abilities for analysis run in batch or parametric studies, and can for example be a quicker approach if several materials or similar geometries should be run in succession. Some knowledge of Python is recommended, but the Abaqus documentation should be a sufficient reference base for many applications. A brief review of the use in in conjugation with this report is given below. The review is for the structure and ability of Abaqus CAE, but primarily for the running of Abaqus through a script, but no introduction to Python is given.

Some commands and phrases will be repeatedly used in this report for the Abaqus scripting interface. These are stored in variables to increase readability of the script.

These variables will be used thorough this report:

```
MODEL=mdb.models['Model-1']
PART=parts['Part-1']
```

'mdb.models['Model-1']' is referencing a model with the name 'Model-1'. This is the standard name for the first model created in an Abaqus CAE project. In the report only one model is referenced, and a variable with the name 'MODEL' is introduced for simplicity. The same is true for the phrase 'parts['Part-1']', only one part are included and will be named

‘PART’. Models and part form a hierarchy, that is, a model can include several parts, but one part can be used in several models. Following this, a model must be referred to before a part to complete the address.

An example of this is:

```
MODEL.PART=mdb.models['Model-1'].parts['Part-
1']
```

This is the complete reference to ‘Part-1’ in ‘Model-1’.

3.1 PART MODULE

Geometry is handled in Abaqus CAE by the part module. Geometry may either be created or imported to form part of or the complete model. Three-dimensional geometry may be created with solid, shell, beam, truss or membrane elements, two-dimensional geometry can be created with either two-dimensional continuum solid elements or truss or beam elements. Axisymmetric models may also be created with axisymmetric solid continuum elements or axisymmetric shell elements.

For this report, three-dimensional geometry is utilized with primarily beam elements. Beam elements are added to wire geometry, a geometry described by a line in two or three dimensions. The simplest form of wire geometry is a straight line between two points and this is adequate for modelling most truss structures since straight members tend to be beneficial in structures. Wire geometry is easily handled in

Abaqus by creating two points followed by a wire between, more complex wires may be created by using more points.

For the Abaqus scripting interface, geometric points may be created with the following code, where X, Y and Z indicates the respective coordinates in numerical value, which may be stored in a variable:

```
MODEL.PART.DatumPointByCoordinate(coords=(X,Y,
Z))
```

The wire may then be created by referencing two points (datums):

```
MODEL.PART.WirePolyLine(mergeWire=OFF,
     meshable=ON,points=((
MODEL.PART.datums[1],
MODEL.PART.datums[2]), ))
```

Or alternatively by directly referencing coordinates (X,Y,Z):

```
MODEL.PART.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((X1,Y1,Z1),(X2,Y2,
Z2)), ))
```

3.2 PROPERTY MODULE

The property module is used to assign properties to a part that is not given by the geometry. This includes defining material settings, composite layups and shell thicknesses etc.

For a simple linear model, a steel material can be defined by the following lines of code:

```
MODEL.PART.Material(name='Steel')
MODEL.PART.materials['Steel'].Elastic(table=
((207000000000.0, 0.3), ))
MODEL.PART.materials['Steel'].Density(table=
((7850.0, ), ))
MODEL.PART.materials['Steel'].Damping(alpha=
0.01)
```

The first line creates the material named *'Steel'*, the second line defines the elastic properties; the elastic modulus (207GPa in consistent units) followed by the Poisson-ratio (0.3). The third line is density (7850kg/m^3) and the forth the dampening ratio (not required for static analyses). Line 2-4 is referencing the created material by the phrase *'materials['Steel']'*. Materials is directly below the part in the hierarchy, and is therefore referenced directly after the part.

For wire geometry, section properties and beam orientation must be assigned in the property module. All different section profiles supported by Abaqus CAE may be created with the Scripting interface. For example tubular pipes, used on the OC4:

```
MODEL.PART.PipeProfile(name='Profile',
r=radius, t=thickness)
```

Where *'radius'* is a variable containing a numerical value for the radius to the outside edge of the profile, and *'thickness'* containing the wall thickness.

A profile is the geometric region used to define mechanical properties to a section. A section is a profile combined with other properties including material. An example is given:

```
MODEL.PART.BeamSection(consistentMassMatrix=
False, integration=DURING_ANALYSIS,
material='Steel', name='SECTION',
poissonRatio=0.3, profile='PROFILE',
temperatureVar=LINEAR)
```

The material for the section along with the profile is referenced. Input for the integration scheme, the Poisson-ratio and the temperature variation (optional) of the section are also included.

Additional properties may also be added to a section, this includes properties related to fluid inertia and several other referenced in the Abaqus CAE documentation. For submerged or partially submerged structures subjected to dynamic loading, the fluid related properties may have an effect on the behaviour. A section utilizing fluid related properties can be defined in the following way:

```
MODEL.PART.BeamSection(consistentMassMatrix=
False,crossSectionRadius=radius,
fluidMassDensity=DENSITY,
integration=DURING_ANALYSIS,
material='Steel', name='SECTION_SUBMERGED',
poissonRatio=0.3, profile='PROFILE',
temperatureVar=LINEAR, useFluidInertia=ON)
```

Where 'DENSITY' is the density of the fluid in consistent units. The keyword *'useFluidInertia'* implies a submerged section, for only partially submerged section, the argument *‘submerged=half’* may be included, this reduces the added inertia per length to half the calculated value. This means that it can be more correct to model half submerged structures in such a way that the boundary between fluid and non-fluid areas matches that of section properties.

Note that the sections are not assigned to geometry. This must be done separately:

```
MODEL.PART.SectionAssignment(offset=0.0,
offsetField='', offsetType=MIDDLE_SURFACE,
region=Region(edges=MODEL.PART.edges.findAt
(((X,Y,Z),)),sectionName='SECTION',
thicknessAssignment=FR
OM_SECTION)
```

Here geometry is referenced with the *‘findAt’*-method. *‘findAt’* will look for entities that lay within a given distance (default 10^{-6}) of one or more sets of coordinates. An edge with a point at or near the set of coordinates (X,Y,Z) is used as an example. For straight wire geometry, no other coordinates than wire ends will usually be available, and end-points may be shared between wires. Using *‘findAt’* on such a point will return all wires connected to the point. The alternative is to mathematically define another point on the wire, for example the mid-point.

A *set* may be created to organize regions of geometry.

```
MODEL.PART.Set(edges=MODEL.PART.edges.findAt((
(X
,Y,Z),)),name='SET_NAME')
```

This means that for later uses the following way of referencing the wire:

```
region=Region(edges=MODEL.PART.edges.findAt((X
,Y
,Z),))
```

Can be replaced with:

```
region=MODEL.PART.sets['SET_NAME']
```

Even with tubular sections, the orientation of the beam has to be defined on the geometry.

```
MODEL.PART.assignBeamSectionOrientation(method
=
N1_COSINES,n1=((x,y,z))region=
MODEL.PART.sets['SET_NAME'])
```

Where *x*, *y* and *z* in *n1* is components used to define the 1-direction (n_1) of the cross-section in respect to the global coordinate system. I.e. n1=(0,1,0) would align the 1-direction of the cross-section to the global y-axis. This may be approximate since *t* will always be tangential to the beam, and *n1* will be defined such that the relative angle to the defined vector is the smallest possible. This will uniquely describe *n1* unless the vector is exactly parallel to the tangent of the beam (*t*).

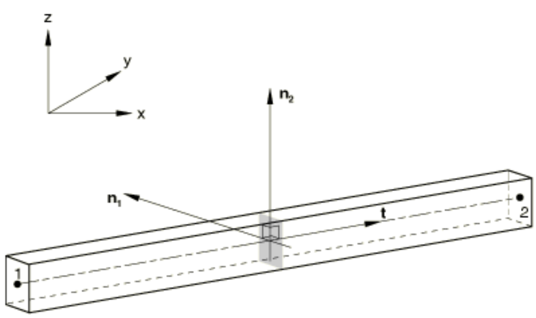

Figure 5 - Abaqus beam orientation

3.3 ASSEMBLY MODULE

An Abaqus model contains a main assembly which again can consist of one or more instances of one or more parts. Part instances may be dependent or independent. A dependent instance is linked to the original part, an independent instance is a copy of the part. This means that individual operations like meshing must be done for independent instances, but may not be done on dependent instances where such attributes are added the part level.

For this model, only one part and one instance is used, therefore not much attention is paid to the assembly module. However every model needs to have an assembly, for a single instance assembly this can be done in the following way in the Abaqus scripting interface:

```
MODEL.rootAssembly.DatumCsysByDefault(CARTESIA
N)
```

```
MODEL.rootAssembly.Instance(dependent=ON,
name='Part-1-1',  part=MODEL.PART)
```

The first line defines the coordinate system in the assembly, which here is a Cartesian system. Note that the assembly is referenced as *rootAssembly*.

The second line creates the instance *Part-1-1*, which is standard name for the first instance in the first part of the model, and the part *MODEL.PART* is the part mentioned thorough this report.

3.4 STEP MODULE

The main purposes of the step module is to control the changes in loads and boundary conditions, to define the output of the analysis and control adaptive meshing and solver settings.

An example is given for the OC4-model:

```
MODEL.ImplicitDynamicsStep(initialInc=0.025,
maxNumInc=24001, name='Dynamic', noStop=OFF,
nohaf=OFF, previous='Initial',
timeIncrementationMethod=FIXED,
timePeriod=600.0)
```

This creates the step *'Dynamic'*, an implicit dynamic step with increments of 0.025 seconds to a total time of 600 seconds, this corresponds to 600/0.025+1=24001 increments. For this model this is a fixed configuration to match the loading which is given for every 0.025 seconds. To arrange the steps in order, all new steps are created with a reference to the previous step, which here is *'Initial'*. '*Initial'* is always a step 0 corresponding to a time 0 and zero loading.

A static step may be created with:

```
MODEL.StaticStep(name='Static',
previous='Initial')
```

And an eigenfrequency extraction step (for the first 10 eigenfrequencys) with:

```
MODEL.FrequencyStep(maxEigen=10.0,
name='Frequency', previous='Initial')
```

Output request are also handled by the step module. Abaqus has two types of outputs, field and history, where field outputs is quantities distributed over the whole model. Stress- and displacement fields would be examples of this. History output is output related to the whole model, for example mass properties, or output data from specified nodes or elements.

Field output is specified in the scripting interface with the following:

```
MODEL.fieldOutputRequests['F-Output-
1'].setValues(variables=('S', 'U'))
```

This is an output request for the variables '*S'* and *'U'*, respectively stress and displacement outputs.

An example for a history output is also given:

```
MODEL.historyOutputRequests['H-Output-
1'].setValues(frequency=1
    , rebar=EXCLUDE,
region=model.rootAssembly.allInstances['Part
-1-1'].sets['All_set']
    , sectionPoints=DEFAULT, variables=('S11',
'S22', 'S33', 'S12', 'S13',
    'S23', 'SP', 'TRESC', 'PRESS', 'INV3',
'U1',
'U2', 'U3', 'UR1', 'UR2',
    'UR3'))
```

Here outputs are stored for every frame in the analysis (frequency=1). The documentation for Abaqus CAE should be referenced for the variable definitions.

3.5 INTERACTION MODULE

The interaction module governs all mechanical and thermal interactions. This includes constraints between regions, contacts etc. Additionally, effects such as inertia is coupled to the model here.

A coupling between nodes is used several places in the OC4-model. For example, the concrete block is modelled as a point mass and coupled to the steel jacket:

```
MODEL.Coupling(controlPoint=Region(vertices=
MODEL.rootAssembly.instances['Part-1-
1'].vertices.findAt([POINT1])),
couplingType=KINEMATIC,
influenceRadius=WHOLE_SURFACE,
localCsys=None,    name='Constraint-2',
surface=MODEL.rootAssembly.sets[
'JACKET_CONNECTION'], u1=ON, u2=ON, u3=ON,
ur1=ON, ur2=ON, ur3=ON)
```

'POINT1' is the center of the concrete block, and *'JACKET_CONNECTION'* is a set of nodes on the jacket which the concrete block is attached to. The coupling restricts all three translations (*u1-3*) and rotations (*ur1-3*).

It is worth noting that meshed geometry must be connected to every node used in the coupling. If a single point have to be coupled to the model, a workaround can be to attach a very small beam to the point. This beams needs to be assigned a section and material etc. like every elsewhere used in the model, but it don't need to serve any purpose other than to enable the coupling of the point.

A point mass can be created at a node and assigned inertia in the following way:

```
MODEL.rootAssembly.engineeringFeatures.PointMa
ss
Inertia(alpha=0.0, composite=0.0,
i11=INERTIA1, i22=INERTIA2,
i33=INERTIA3, mass=MASS, name='Inertia-1',
region=Region(vertices=MODEL.rootAssembly.
instances['Part-1-1'].vertices.
findAt([COORDINATE])))
```

Where '*COORDINATE'* is the coordinates of the node, '*INERTIA1-3'* is the inertia about the three primary axis' in the global coordinate system and '*MASS'* is the added mass. *Alpha* and *composite* is dampening properties.

3.6 LOAD MODULE

The load module handles the addition of loads and boundary conditions to the model.

For loads applied with complex dynamics, a user defined load amplitude should be applied. An amplitude describes a load pattern over time. Some built in patterns exists in Abaqus CAE, but in this example, tabular data is used. Tabular data may be best handled by the graphical user interface (GUI) of Abaqus CAE, since operations such as copy and pasting can be done on tabular form. For the Abaqus scripting interface, tabular data is supported in the following way:

```
MODEL.TabularAmplitude(data=((x1,y1), (x2,y2),
(x3,y3)),
name='AMPLITUDE',smooth=SOLVER_DEFAULT,
timeSpan=STEP)
```

This defines the amplitude *'AMPLITUDE'* with three data points, where '*x'* is the step time for the input and *'y'* is the input value to be multiplied with the load value. For example the values *(0,0)*, *(0.5,1)* and *(1,0)* would apply the load fully *(1)* halfway in the step, with complete unloading at the end of the step *(0)*, illustrated in figure 6.

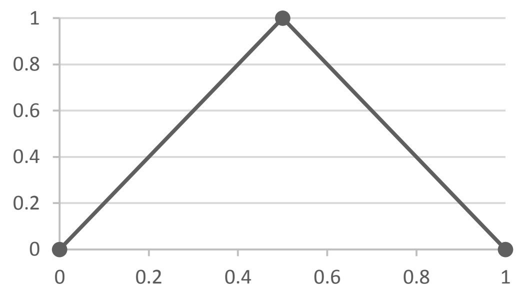

Figure 6 - Amplitude example

Similarly, more complex load patterns may be produced as seen in figure 7:

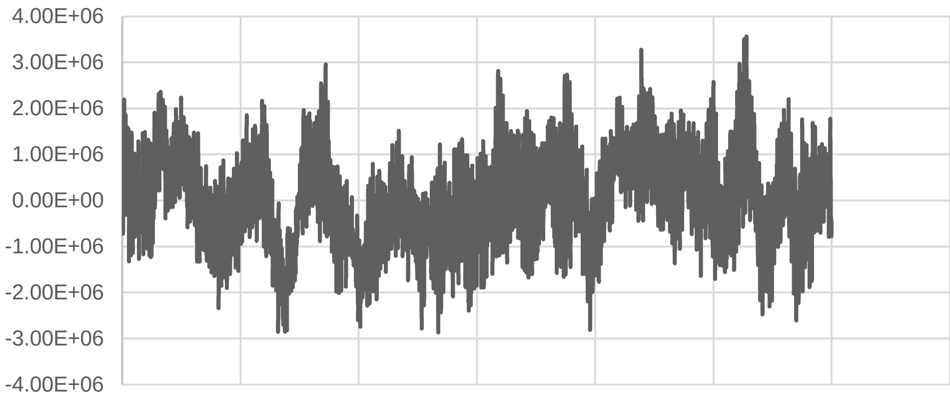

Figure 7 - Amplitude example 2

In combination with an amplitude, a number of different loads may be applied, for example a concentrated force in the 1-direction (*cf1* is a concentrated force along the x-axis, *cf2* along the y-axis etc.) can be added with the previous amplitude:

```
MODEL.ConcentratedForce(amplitude='AMPLITUDE',
cf1=1.0,createStepName='Dynamic',
distributionType=UNIFORM, field='',
localCsys=None, name='FORCE',
region=Region(vertices=model.rootAssembly.in
stances['Part-1-1'].
vertices.findAt([COORDINATE])))
```

Or a moment about the same axis added at the same point (*‘COORDINATE’*) with a second amplitude:

```
MODEL.Moment(amplitude='AMPLITUDE2', cm1=1.0,
createStepName='Dynamic',
distributionType=UNIFORM, field='',
localCsys=None, name='MOMENT',
region=Region(vertices=model.rootAssembly.in
stances['Part-1-1'].
vertices.findAt([COORDINATE])))
```

3.7 MESH MODULE

The mesh module assigns and controls the finite element mesh on the model. For dependent instances, a mesh must be assigned to the parent part and would then be similar for all instances. An independent instance must be assigned a unique mesh.

The mesh may consist of several different element types.

In the Abaqus scripting interface, elements may be assigned to geometry regions. An example is as follows:

```
MODEL.PART.setElementType(elemTypes=
(ElemType(elemCode=B31,
elemLibrary=STANDARD), ),
regions=(MODEL.PART.edges.
getSequenceFromMask(('[#ffffffff:3
#ff ]', ), ), ))
```

This assigns B31-elements to the model. B31 are beam elements in 3D-space with linear interpolation formula. The naming convention for beam elements in Abaqus is described in figure 8. The region reference shown in the example is a cryptic display of the interaction between the Abaqus GUI and the Abaqus scripting interface. It is a reference to all geometry regions in the model. This means that the whole model will be assigned B31-elements.

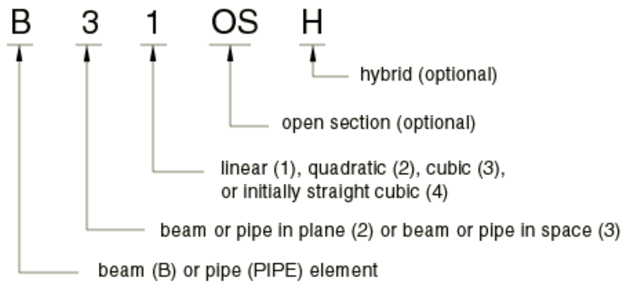

Figure 8 - Mesh element definition

Along with being assigned an element type, geometry must be seeded before a mesh ca be generated. This controls the distribution of mesh elements on the model.

```
MODEL.PART.seedEdgeByNumber(constraint=FINER,
edges= MODEL.PART.edges.findAt(
[COORDINATE]),
number=5)
```

Here one or several edges including a given coordinate is seeded with 5 elements. Wire geometry is considered an edge and this is an appropriate method for seeding beam elements.

The meshing process must be completed by generation of the mesh, this is done with this command:

```
MODEL.PART.generateMesh()
```

3.8 JOB MODULE

The Job module lets you define and run the analysis job. Several jobs can be run and monitored at the same time.

In the Abaqus scripting interface, a job simply named “JOB” can be created in the following way?

```
mdb.Job(atTime=None, contactPrint=OFF,
description='', echoPrint=OFF,
     explicitPrecision=SINGLE,
getMemoryFromAnalysis=True,
historyPrint=OFF,
     memory=90, memoryUnits=PERCENTAGE,
model='Model-1', modelPrint=OFF,
     multiprocessingMode=DEFAULT, name='JOB',
nodalOutputPrecision=SINGLE,
     numCpus=1, numGPUs=1, queue=None,
scratch='', type=ANALYSIS,
     userSubroutine='', waitHours=0,
waitMinutes=0)
```

Note that a large number of parameters can be set for control.

To submit a job for analysis, the submit command is used:

```
mdb.jobs['Job'].submit()
```

If the script requires the results, for example post processing, Abaqus can be asked to wait to results are produced:

```
mdb.jobs['Job'].waitForCompletion()
```

3.9 ABAQUS AQUA

The Aqua routines in Abaqus CAE is used to simulate hydrodynamic loading. Several different wave models, including custom models, can be included with Abaqus Aqua along with steady currents and wind loads.

These routines are not implemented in the graphical user interface and must be controlled by the use of the input file (.inp). Abaqus GUI and the scripting interface both create input files to run an analysis, these are just different approaches to create input files. A more powerful approach is to set up the analysis by directly writing the input file.

The disconnection of Aqua and the GUI, is also true for Aqua and the scripting interface. This means that there are no built in functions available in the scripting interface for Abaqus Aqua. This is problematic if these two are to be combined as they are in this project, since all configurations related to Aqua must be done through the input file. Manipulation of keyword blocks in the input file is possible from the scripting interface and this gives complete access to manipulation of the input file, a similar approach to what is found in the Abaqus GUI with an insert keyword command.

For this project, wave loading is applied to the model by the following keywords in the input file:

```
*AQUA
elevation of seabed, elevation of sea water
surface, gravitational constant, density of
fluid

*WAVE, TYPE=STOKES
wave height, wave period, phase angle,
direction of travel cosines
```

In values in consistent units this corresponds to for example:

```
*AQUA
-50, 0, 9.81, 1021
*WAVE, TYPE=STOKES
2.615 ,6.850, 0, 1, 0
```

This is applied through the script with the following input:

```
MODEL.keywordBlock.insert(596, """*AQUA\n
-50, 0,9.81, 1021\n
*WAVE, TYPE=STOKES\n
2.615,6.850, 0, 1, 0""")
```

Where *596* is the placement in the input file for insertion of the keywords. This is not an elegant way of doing this, since the value has to be manually found. *\n* indicates a line shift. It is possible for the script to read the input file directly, and then place the keywords using this information. This has not been done successfully for this project with iterative analyses, and the manual approach has proven more robust.

3.10 IO-OPERATIONS

Input and output of files are handled easily by a script, since such operations are supported by Python by default. Some additional functions are needed to read an Abaqus output database (.odb). The following line can be used to open the odb:

```
odb=openOdb(path=filename.odb)
```

An Abaqus output database is divided into steps, which in turn is divided into frames and frames contain output data.

For example to extract the stresses in the last frame (frame number -1) of *Step-1*, the following line can be included in the script:

```
stress=odb.steps['Step-1'].frames[-
1].fieldOutputs['S']
```

This is effective for most application where element stresses are needed, but if average node stress is of interest it is necessary to insert additional steps. An approach included below has proven successful for such operations, it extracts the average stress for all frames for a set of nodes and store the results in the file *file.txt*.

```
output = open("file.txt", 'w')
nodelist=(N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N
12
,N13,N14,N15,N16,N17,N18,N19,N20,N21,N22,N23
,N24,N25,N26,N27,N28,N29,N30,N31,N32,N33,N34
,N35,N36,N37,N38,N39,N40,N41,N42,N43,N44,N45
,N46,N47,N48,N49,N50,N51,N52,N53,N54,N55,N56
)
numframes=len(odb.steps['Step-1'].frames)

for i in xrange(numframes):
     for j in xrange(len(nodelist)):
          node=((nodelist[j]),)
          path=session.Path(name='NODES',
               type=POINT_LIST,
expression=node)        stress=(('S',
INTEGRATION_POINT,
               ((INVARIANT, 'Mises' ), )),)
          session.paths['NODES']
          session.viewports['Viewport: 1'].
               setValues(displayedObject=odb)
          stressdata=session.XYDataFromPath(
               name='Von-Mises',path=path,
               includeIntersections=False,
               shape=UNDEFORMED,
               labelType=TRUE_DISTANCE,
               step=0, frame=i,
variable=stress)
     first=stressdata.data[0]
          value=first[1]
          output.write(str(value) + "\t")
     output.write("\n")
output.close()
```

The first line above assigns a filename for writing ("w"), and stores the file in the variable *output*. The obtained information is later written to the file by lines including the phrase *output.write()*. *n* is, as before, a line shift, and *t* is a TAB or indent. Indents are here used as separator for post processing.

4 STRUCTURAL OPTIMISATION

Some degree of structural optimisation is needed for applications where both cost and structural performance are of concern. It is relevant when the extra cost in engineering design can offset the reduced cost of a better design.

Mathematically, structural optimisation is no different than other kinds of optimisation. That is, it is the procedure to minimise or maximise a function to a set of constraints.

If we define this function the objective function *f* and denotes the design variables *x* and the state variables *y.* The objective function is then a function that evaluates the performance of the structure, i.e. the value used to determine the greatness of the design. The design variables are the variables that are to be changed during the optimisation, this could be related to geometry, material parameters etc. State variables are structural responses, this could include stresses, strains, displacements etc.

The structural optimisation would then be on the form

Minimise/maximise *f(x,y).*

In form:

f has behavioural constraints by *y*

f has design constraints by *x*

Structural equilibrium must be in place

There is no limit on the number of objective functions, design and state variables. This means that some problems could be infinitely complex with an infinite number of solutions. For a set with multiple objective functions, usually a new set of constraints must be set on the objective functions to be able to find a global optimum.

The setup of this equation set is usually complex since a structural analysis can consist of thousands and millions of degrees of freedom, and the task of defining the criteria for what is the optimal structure can be non-trivial.

In general, three types of structural optimisation is defined:

- Shape optimisation – optimisation of a beam cross section along the length of the beam etc. with the shape as variable.
- Size optimisation – optimisation of thickness of plates and truss cross sections etc.
- Topology optimisation – optimisation of plate thickness, truss member cross section etc. by assigning local values between a finite value and 0. Only the design space of the structure is included in the optimisation to ensure that it is possible to extract structure that conforms to the ideal load paths from a much heavier structure. This is different from shape optimisation, as shape optimisation only can optimise a given topology.

Structural optimisation problems can be defined in several ways, but in general a relationship between stiffness, displacement and external force on the form

$$K(x)u = F(x)$$

Where *K* is the stiffness matrix, *u* is a displacement vector and *F* is a force vector. In general the stiffness matrix especially, but also the force vector, can be sensitive to changes due to the optimisation process (8).

5 LOAD CASES

For the OC4 project, a set of load cases are described for reference use (9). A different set of five load cases has been used in this project, which do not correspond to the OC4 cases. These load cases have been created with a OC4-model in the software FEDEM Windpower to generate time series for forces, moments, rotations and displacements at the node x=0, y=0, z=20.15m. Values are stored for a total time of 600 seconds for load case 1-4 and 690 seconds for load case 5 at 0.025 second interval. The extra time for load case 5 is to facilitate the nature of the emergency shutdown situation where stabilisation of the analysis is wanted, both before and after the event. An overview of the used load cases is shown in table 3.

Load case	Description	Wind speed [m/s	Turbulence intensity [%]	Wave amplitude [m]	Wave period [s]
1	Power production	9	15.6	1.395	5.705
2	Power production	14	14.2	1.910	6.070
3	Power production	19	13.5	2.615	6.850
4	Extreme case	42	12.0	8.100	10.090
5	Emergency shutdown	14	14.2	1.910	6.070

Table 3 - Load case overview

An example from load case 5 is given in figure 9. This is a plot of the load on the transition piece on the wind direction axis and represents the behaviour for the emergency shutdown.

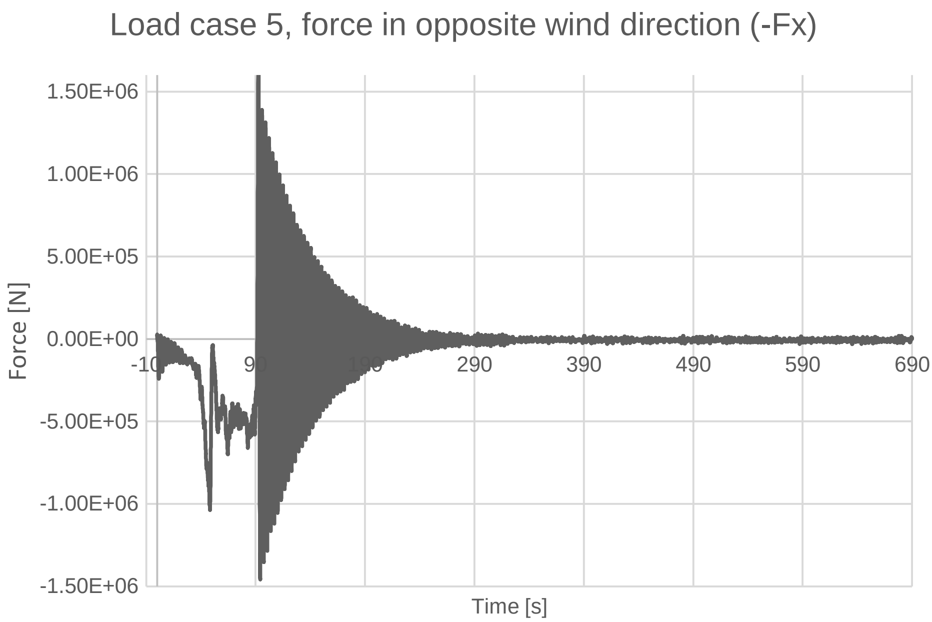

Figure 9 - Load case 5 example

6 STRUCTURAL BASE MODEL

Five base models are created. All similar except for the load case used. Since Abaqus AQUA is involved for all load cases, it has been found that a separate model for each load case has been beneficial.

The model of the OC4 jacket is set up by using the features of Abaqus CAE and the scripting interface. From Abaqus the modules shown in table 4 are imported for use in the script:

Imported programming modules:

From Abaqus:	Part
	Material
	Section
	Assembly
	Step
	Interaction
	Load
	Mesh
	Job
	Sketch
	Visualization
	ConnectionBehavior
	RegionToolset
From Python:	Random

Table 4 - Python modules

6.1 Material model

A linear elastic material model for steel is included with the properties given in table 5.

Young's modulus	207 GPa
Poisson ratio	0.3
Density	7850 kg/m^3
Damping ratio	0.01

Table 5 - Material model

6.2 Geometry

The structure is modelled with wire geometry and point masses. This is the basis for all mass and inertia distributions. The grouted connection, the transition piece, nacelle and rotor is not modelled, but all except the grouted connection is represented with coupled point masses. Both the jacket and the tower is modelled with beam elements. Point masses must be assigned to nodes, and since both the nacelle-rotor assembly and the concrete transition piece are not modelled with any elements, no non-rigid elements are added in these areas. An easy fix to this problem is to add a beam element between the point mass and any other coordinate that is not a part of the system (free air). This ensures that no stiffness is added to the structure, the length or material density should be kept at a minimum so that the mass added can be neglected. These "dummy" beam elements are used several places in the model for this purpose.

The transition piece is modelled with a single node in its simplified center of mass. This node is coupled to both the jacket and the tower. The rotor and the nacelle is modelled in a similar way, both coupled to the top of the tower. Point masses representing additional masses on the tower, previously described, are added to the tower at the respective nodes. Marine growth is not included in the model.

6.3 MESH

Each wire beam is seeded with 5 elements of equal size to generate the mesh. All beams are meshed with elements designed B31. The beams are assigned section properties submerged, semi-submerged or not submerged depending on the elevation in respect to mean sea level. A set of if-sentences are used to assign these properties. Section properties is added corresponding to table 2.

6.4 LOADS AND CONSTRAINTS

Loads are taken from the load cases and added to the node in the connection between the transition piece and the tower at coordinate x=0, y=0, z=20.15. Time series are used for the forces F_x, F_y and F_z, and moments M_x, M_y and M_z in addition to a static gravity load for the complete structure, but data related to position and rotation is discarded. AQUA is used to add wave loading corresponding to the load case. The grouted connections is modelled with fixed constraints at node 1, 6, 11 and 16.

7 DAMAGE ACCUMULATION AND FATIGUE USING RAINFLOW COUNTING

7.1 FATIGUE

For mild steel structures subjected to stresses well below the yield stress and where no instabilities occur, a significant lifetime can be expected. However the loading and stress history of the structure can be of great importance for the length of the lifetime. This is due to fatigue. Imperfections are inescapable in a large structure and some crack growth should be counted on over time. This means that a static analysis of a structure which show no yielding, do not guarantee an infinite lifetime. Therefore, the stress cycles can be of greater importance than the amplitude of the stresses. An example of this is structural bolts, which often are preloaded to great stresses to remove most of the stress variation.

For the OC4 jacket, structural soundness of the structure is taken as a given and no elementary static check are done. Instead, special attention has been given to fatigue in the structure. Especially the joints which are expected to be governing in damage accumulation. Other areas of stress concentrations may be of importance, such as areas where parts, lugs etc. may be welded to the structure. These effects are not taken into account.

7.2 RAINFLOW COUNTING

To evaluate fatigue and damage accumulation, one must have an overview of the stress history and how these stresses affect damage accumulation. For a finite element analysis, stress histories are readily output since this is a common job for such analysis. Some cases will have elegant responses with defined and repeated stress histories which can be easily evaluated, but for most combinations of structures and loads this is not the truth. The response of the structure may be out of sync with the loading and the stress history may be chaotic and not accurately definable by some mathematical function that is not of great complexity.

For the OC4 jacket, which is subject to both dynamic wind and wave loading, this is what is to be expected. Therefore it is necessary to find some way of evaluating and simplify the stress history for the fatigue calculation.

Rainflow counting is such a method, a method for counting the amplitudes and the magnitude of the amplitude. The idea is to identify the large amplitudes from the small ones (10).

The rainflow algorithm can be described with the following steps:

1. Turn the semi-continuous stress or strain history 90 degrees clockwise, so that the earliest time point is the most upper point.
2. Pretend this signal is a pagoda roof.
3. Let “rain” or “water” flow from different sources (see figure 10):
    a. The point of earliest time (origin).
    b. The top of each “roof”.

4. Running down a roof, the streams either runs into another stream falling from above or falls over a roof edge. In the first case a half-cycle is completed.
5. For positive and negative direction (i.e. flowing along the positive y-direction or negative y-direction, typically tensile and compressive components), a half-cycle is completed if either:
    a. The stream reaches the end of the time history.
    b. The stream merges with a stream from an earlier peak.
    c. The stream reaches a point in time where it is opposite a value of equal direction and larger magnitude.
6. Each half-cycle's magnitude is equal to the magnitude difference between start and termination point for the stream.
7. Two half-cycles of opposite direction, but with the same magnitude constitutes a complete cycle.

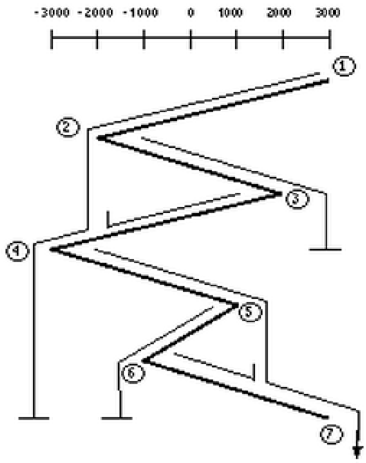

Figure 10 - Rainflow principle

A mathematical description of the rainflow counting algorithm is found in the literature (11) and shown in figure 11:

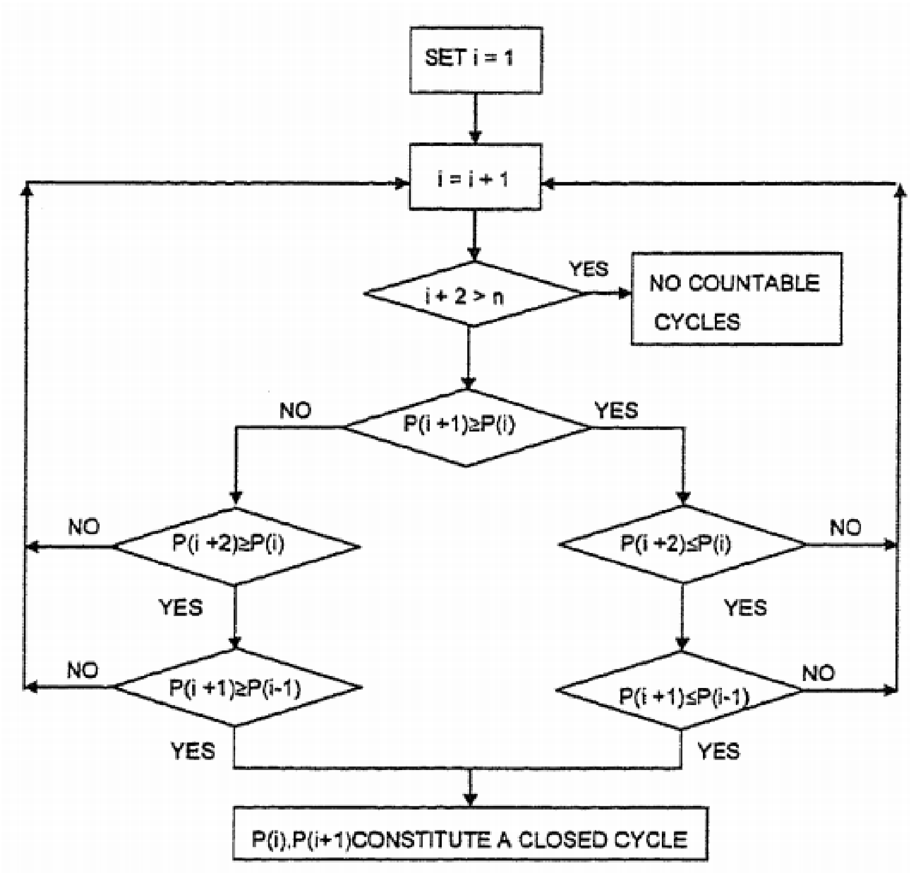

Figure 11 - Rainflow algorithm

8 STRUCTURAL REDUNDANCY OF THE OC4-JACKET

8.1 INPUT

A use for a parametric model is to check a large set of configurations. An example of this is checks for structural redundancy, i.e. to evaluate the structural soundness of a structure if one or more members were to be removed or damaged. This is relevant for an offshore jacket were incidents that can result in large local damage, such as a ship collision, is a possibility.

For this project, a single member has been removed from the five base models, and the analysis rerun for every load case. 47 different members have been removed, including symmetry, this totals to most possible configurations. This equals to (47x5) 235 analyses, each with approximately 60 minutes run time. Another 6 minutes per analysis is required to create the post processed output file.

Members (numbered) removed for redundancy check are shown in table 6:

0	3	4	7	8	17	18	19	20	21	22	23	24	33	34	36	37	40	41	45	46	47	48	53
56	57	60	61	62	63	64	69	72	73	76	77	78	79	80	85	88	89	92	93	94	95	96	

Table 6 - Members for structural redundancy

8.2 Output

Each analysis creates an output database of approximately 1.24GB (for load case 1-4). These output databases' has been worked into condensed output files for further post processing in MATLAB. This is done by a separate script since this has been found to be a more robust solution. The result handling is heavy on memory usage, and a lack of memory will stop the analyses from running.

The model needs some time to conform to the loads acting the structure and only the 100 last seconds of the analyses has been output. This is also helping in limiting memory usage and is believed to be sufficient to describe the behaviour of the models.

8.3 POST PROCESSING

Post processing of the output is done and programming language and computing language MATLAB, in the form of version R2014a. This is personal choice, and the same procedure can be duplicated using other tools, including Python. In this project, the implementation of the Rainflow counting algorithm is the most critical component in post processing. This is available for MATLAB through the MathWorks File Exchange (12).

By reading the condensed output files made from the Python script, the information can be stored in a matrix and be used as input for Rainflow evaluation. For accurate damage calculations, several parameters must be specified and calibrated according to the fatigue curve corresponding to the material and geometry. No attention has been paid to these parameters in this project.

A MATLAB-file used for fatigue calculations is included below. Note that use of functions related to Rainflow counting means these functions must be downloaded.

```
clear all
clc
li=[0,3,4,7,8,17,18,19,20,21,22,23,24,33,34,36
,37,40,41,45,46,47,48,53,56,57,60,61,62,63,64,
69,72,73,76,77,78,79,80,85,88,89,92,93,94,95,9
6];
len=length(li);
damage=zeros(len,56);
for j=1:len
    member=li(j);
    memberstr=num2str(member);
    loadcase='Job1_2';
    name=strcat(loadcase,'-
',memberstr,'.txt');
    a=strcat(name);
    Input =dlmread(a, '\t',1,0);

    for i=1:56
        Var=Input(:,i);
        % S-N curve parameters:
        sigaf=355000000;   % endurance limit
        m=3;               % slope of the curve
        Nk=1e7;            % number of cycle for
knee
                           point
        To=100;            % length in second of the
                           time history
        tp=sig2ext(Var);    % turning points
        rf=rainflow(tp);     % rainflow
        CycleRate=rf(3,:);   % number of
cycles
        siga=rf(1,:);        % cycle
amplitudes
        % calculation of the damage:
       damage(j,i)=(sum((CycleRate/Nk).*
            ((siga/sigaf).^m)));
    end
end
```

9 OPTIMISATION

With a fully parametric model implemented in a scripting interface, an optimization routine can be included.

The simplified optimisation procedure used here can be illustrated (figure 12) in the following way (13):

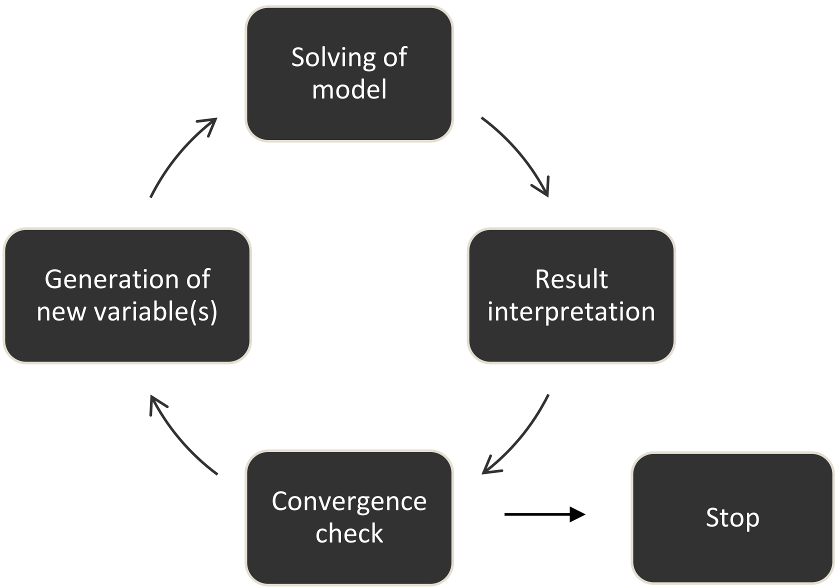

Figure 12 - Optimisation principle

Solving of the model is done by the use of Abaqus/CAE-solvers, as previously explained, and all features including Aqua can be included. This is important since built in optimisation tools tend to lack the ability to use some features.

Result interpretation is done by asking Abaqus to wait for results, and then to load the output database (*.odb). This enables the reading and manipulation of the results. An algorithm for detecting an improvement should then be deployed to evaluate the results.

A convergence or stop criteria should be included to disallow infinite loops. This can be an algorithm for detecting a converged result or simply a given number of iteration the optimisation should run for.

If no stop criteria has been reached, a new set of variables must be created to enable a new iteration. The way the variable or variables is changed is very important for the speed of convergence and the ability to overcome local optima.

This ends the cycle, and a new iteration can be started with the solving of the problem for the new set of variables. This will continue until the stop criteria has been reached.

For the OC4 jacket, a simple optimisation has been tested in this project to evaluate the use on such structures.

The inclusion of wave loading in addition to the time series loading, ensures that the Abaqus model for the OC4 is fairly computationally heavy. This, combined with the highly dynamic nature of the problem makes it unsuitable for typical integrated optimisation modules in finite element codes.

With the assumption that the node stresses should be a function of the stresses on the truss members, and that the nodes are the critical areas for fatigue, the stresses are only checked in the nodes. For a wire model, a node is a mathematical point, and the stress is checked at this point. The maximum of all node stresses are checked against the reference of the previous lowest maximum to evaluate an iteration. If the stress is lower than this maximum, the iteration is an improvement and this new value is stored as the new reference. This ensures that every iteration that is not an improvement is discarded. This guarantees convergence towards a local or global optima.

The optimisation is set to run for a given number of iterations, and new variable sets are generated by randomly changing on of the previous variables by a given delta. This is a simple, but very ineffective algorithm. No memory of previous iterations, other than results are stored, and the optimisation is in essence blind, were every iterations that gives an improvement is by random chance.

Several variables are included in the optimisation, all related to the coordinates of nodes, these are shown in table 7.

Variable	Start value	Final value
B1	6	5.62
H3	-44.001	-43.770
H4	-43.127	-42.130
H5	-33.373	-33.970
H6	-24.614	-27.790
H7	-16.371	-14.480
H8	-8.922	-4.902
H9	-1.958	-1.108
H10	4.378	3.798
H11	10.262	12.540
H12	15.651	15.930

Table 7 - Optimisation variables

Additional variables related to the beam cross section could have been included if wanted. Optimisation of other parameters related to coordinates, along with mesh sizing, load and boundary condition etc. are also easily implemented. It is also possible to vary the number of nodes but this will require an algorithm for generating a new set of beams, since the number of beams no longer will be constant.

To reduce the number of variables optimised, one width variable is included in addition to all node heights that is not related to either connections to the sea bed or the concrete transition block. A set of constraint is added to control the complete node coordinates, referenced in table 8.

Constraints
H1=-45.50000
H2=-45.00000
H13=16.15000
H14=20.15000
B12=4.00000
B2=(4-B1)*H3/61.15+4-16.15*(4-B1)/61.15
B3=(4-B1)*H4/61.15+4-16.15*(4-B1)/61.15
B4=(4-B1)*H5/61.15+4-16.15*(4-B1)/61.15
B5=(4-B1)*H6/61.15+4-16.15*(4-B1)/61.15
B6=(4-B1)*H7/61.15+4-16.15*(4-B1)/61.15
B7=(4-B1)*H8/61.15+4-16.15*(4-B1)/61.15
B8=(4-B1)*H9/61.15+4-16.15*(4-B1)/61.15
B9=(4-B1)*H10/61.15+4-16.15*(4-B1)/61.15
B10=(4-B1)*H11/61.15+4-16.15*(4-B1)/61.15
B11=(4-B1)*H12/61.15+4-16.15*(4-B1)/61.15

Table 8 - Optimisation constraints

These constraints fixes the width of the tower at the top and the heights of the connection nodes. In addition, a set of constraints is ensuring that the four legs of the tower are straight. These last constraints work by mathematically assigning the nodes' x- and y- coordinates value as a function of the z-coordinates.

For this project, the optimisation is run without dynamic loading to significantly reduce calculation time. The dynamic model with full load history (0.025 second steps at 600 seconds for load case 1-4 and 690 seconds for load case 5) has a runtime of in excess of 1 hour on an ordinary desktop computer per iteration. This makes iterating with an ineffective algorithm problematic. Instead the model is run with a static load. This enables far more iterations to be done at a similar time span since an iteration can be completed in few seconds. It would be beneficial to employ more computational power it the model should be optimised using the full time series'.

10 RESULTS

10.1 BASE MODEL

The base model has an estimated lowest eigenfrequency of 0.30901Hz, this can be rounded to 0.31Hz which is the expected frequency for the OC4. This indicates that no large deviations in respect to benchmark models is found. The model is therefore assumed to have a large probability of giving a fairly accurate representation of the OC4.

10.2 STRUCTURAL REDUNDANCY

Structural redundancy have been explored for the members indicated in table 6. To allow for stabilisation of the model, only the last 100 seconds of the time series' for load cases 1, 3 and 4 have been included. For load case 2, the complete time series of 600 seconds have been used, and 690 seconds for load case 5. All load cases except number 2 have been evaluated using output for every time step. For load case 2, output has been taken from every 10th time step. This is not intentional and is also the reason for the difference in time length, it has not been corrected due to time constraints on the project.

Fatigue has been evaluated from Von Mises-stresses. This is not optimal since no distinction between compressive or tensile stresses are made, but the magnitude of the stresses are kept. Along with the fact that no calibration of the damage accumulation curves have been made, this means that no value are attributed to the magnitude of the fatigue evaluations. Instead, the result are normalised to the results from the base model with load case 1 (I.e. this results are taken to have the value 1, and all other results are scaled with the same factor).

One check for the model, is to evaluate the eigenfrequencies for the different configurations to find any areas where problems can be expected. When structural members are removed from the model, the stiffness along with the total mass is expected to change, this means that the effect on natural frequencies of the system is not given.

The eigenfrequencies for the configurations has been evaluated using modal analysis in Abaqus and the results can be seen in figure 13 where the first frequency is plotted for different members removed (member 0 equals no members removed, i.e. the base model).

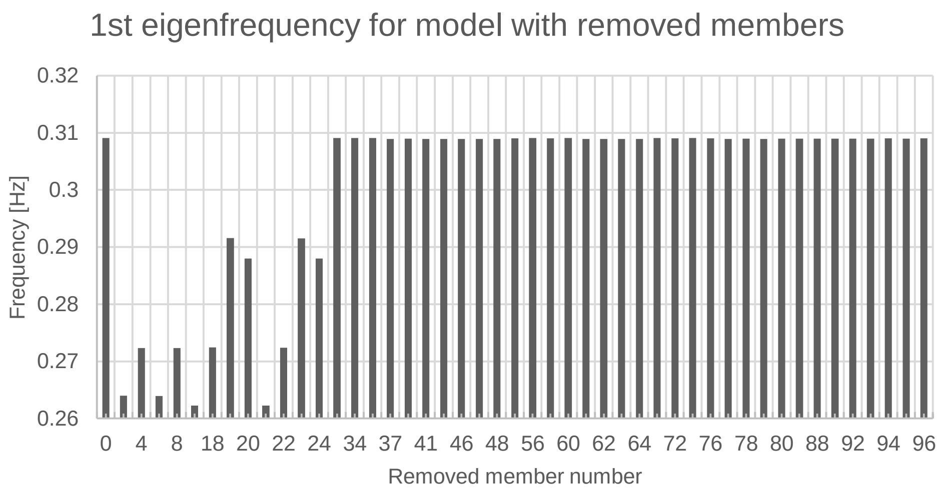

Figure 13 - 1st eigenfrequencies for structural redundancy

The results show that all configurations are within the interval for a soft-stiff structure (figure 3). Significant changes to the response are only seen with the removal of members 3, 4, 7, 8, 17, 18, 19, 20, 21, 22, 23 and 24. These members are all parts of the legs of the jacket. The indicated members and their symmetric counter parts is shown in red in figure 14. These members are the members with the largest cross sectional areas and should be involved in the primary load paths of the structure. Removal of the braces has negligible impact on the first frequency.

The results for the fatigue calculations for all load cases are given in figure 15-19. These results show the effect of removing one structural member on the total damage accumulation for each load case. A combined plot for the results for load case 1-4 is shown in figure 20. Load case 5 is here removed due to the large difference in magnitude of the results.

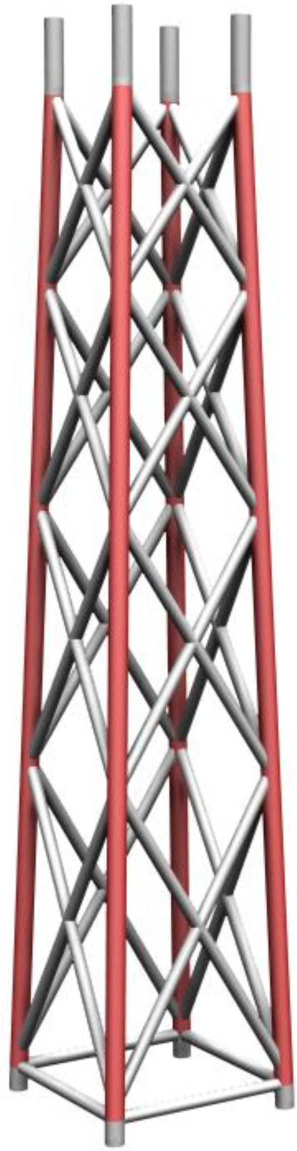

Figure 14 - OC4 jacket

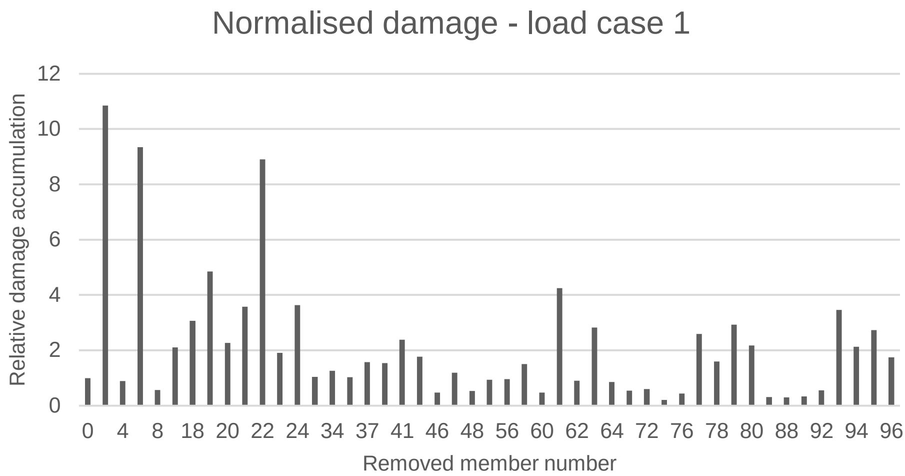

Figure 15 - Normalised damage load case 1

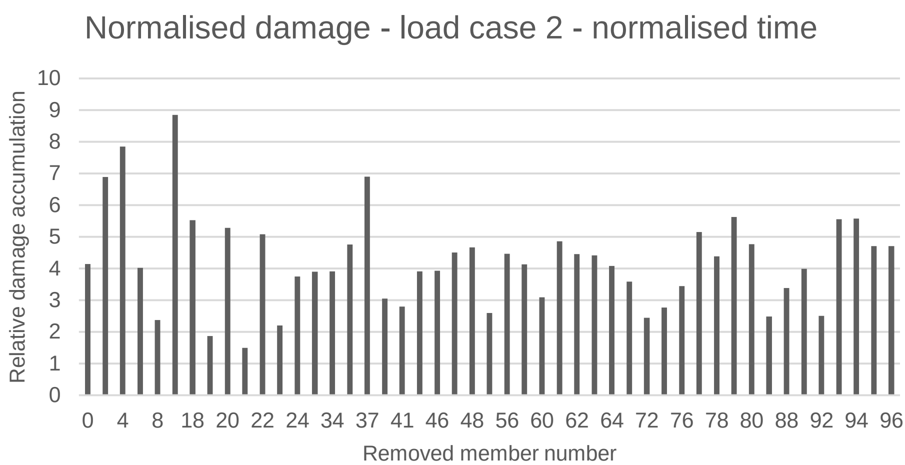

Figure 16 - Normalised damage load case 2

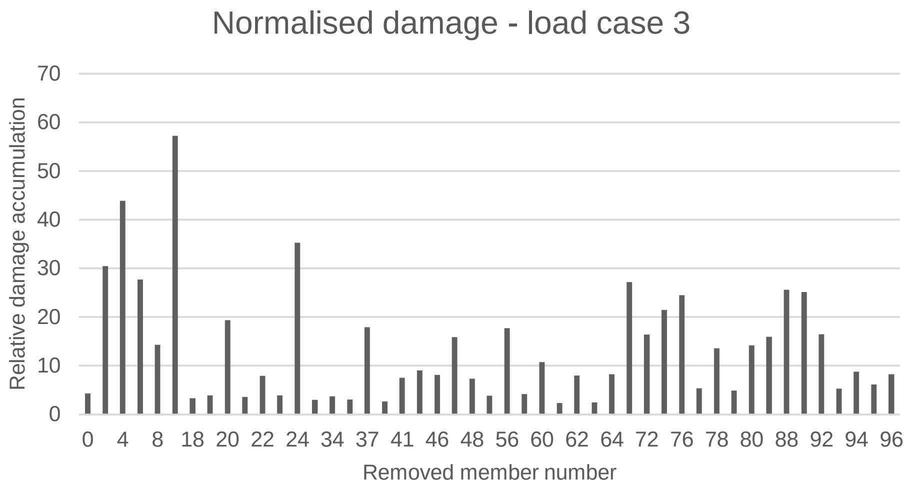

Figure 17 - Normalised damage load case 3

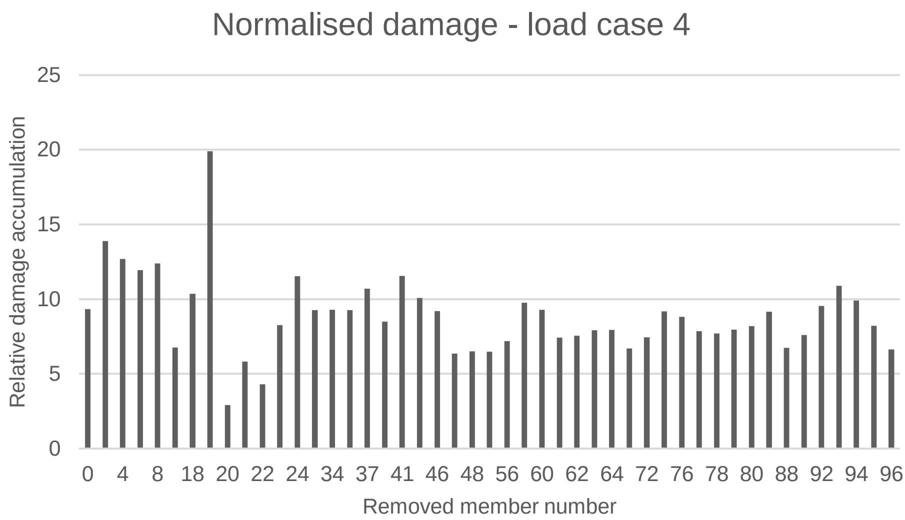

Figure 18 - Normalised damage load case 4

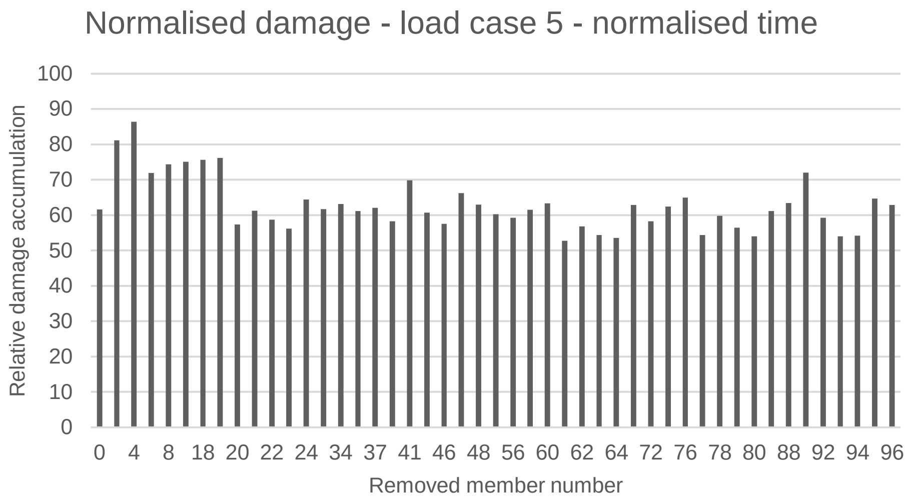

Figure 19 - Normalised damage load case 5

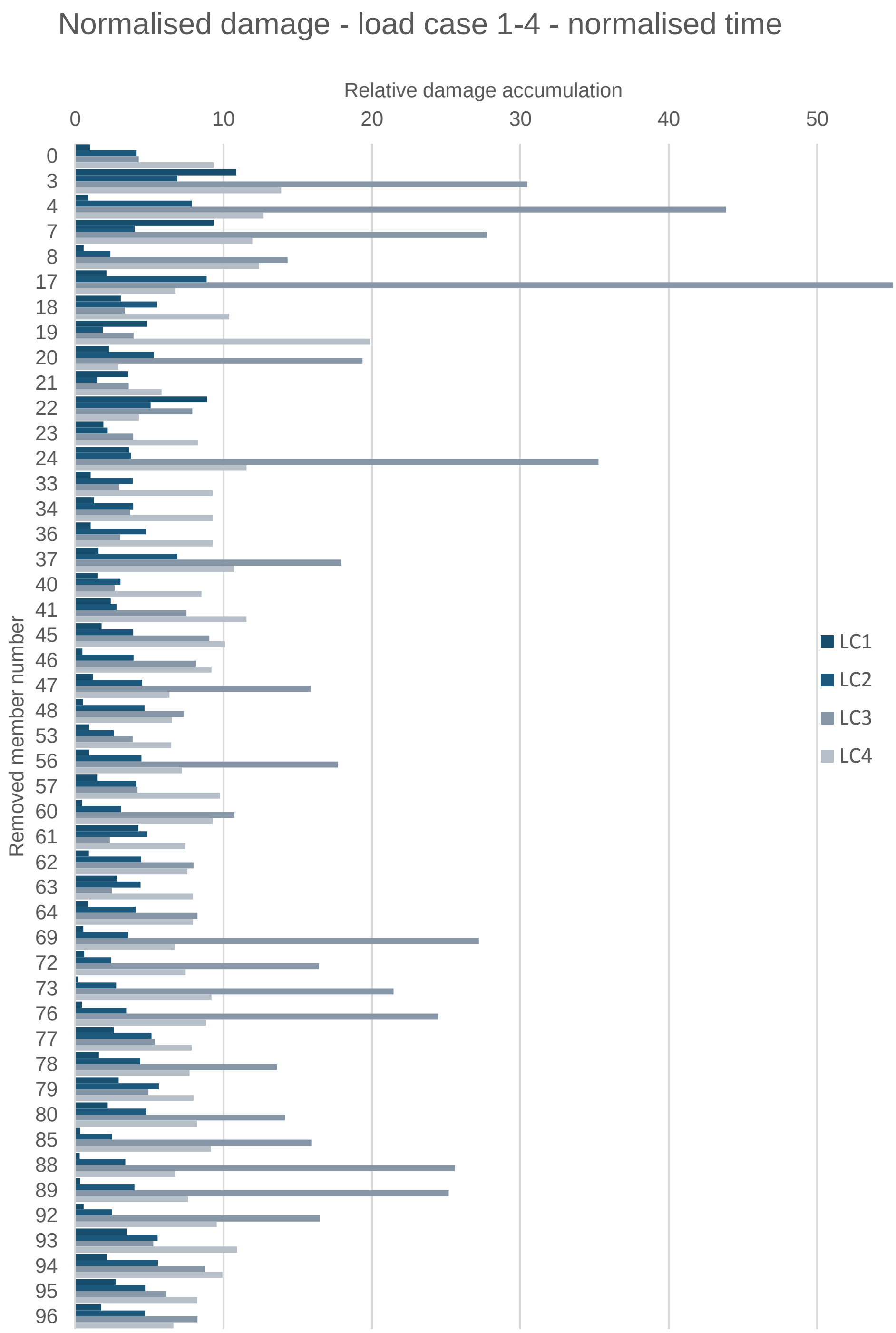

Figure 20 - Combined results structural redundancy

10.3 OPTIMISATION

1500 iterations were completed for the optimisation, 1200 with a step length of 0.1m followed by 300 iterations with a step of 0.01m.

That gave the following run history (figure 21):

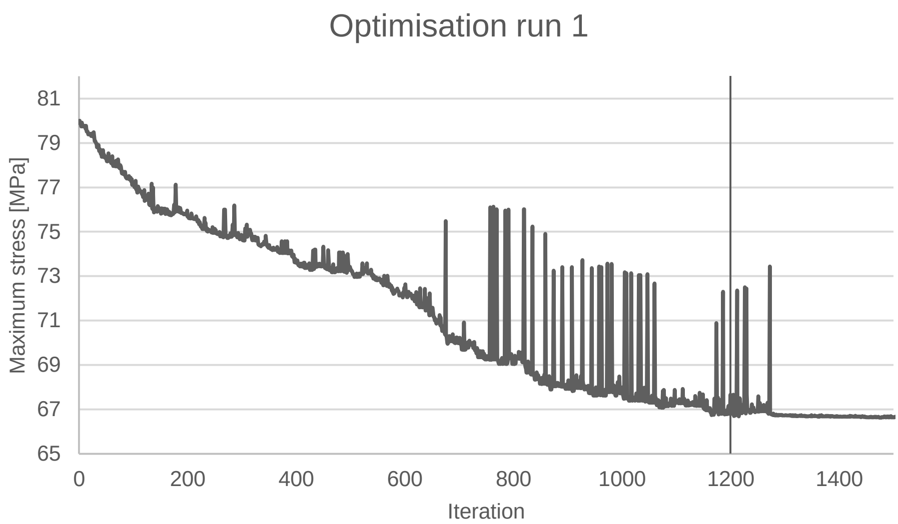

Figure 21 - Optimisation run 1

The slow, but steady, crawl towards a converged result can be seen. It can also be seen that model gets increasingly more sensitive to poor variable choices, as is illustrated by the spikes in the curve. The variables are all both increasing and decreasing during the iteration process, and it stands to reason that it is encountered some configurations of variables that are more unstable than others. Worth noticing is also that the region of most rapid improvement is found after approximately 600 iterations, a testimony to the algorithms poor ability to find the right choices.

The variable history can be illustrated with the following plot (figure 22):

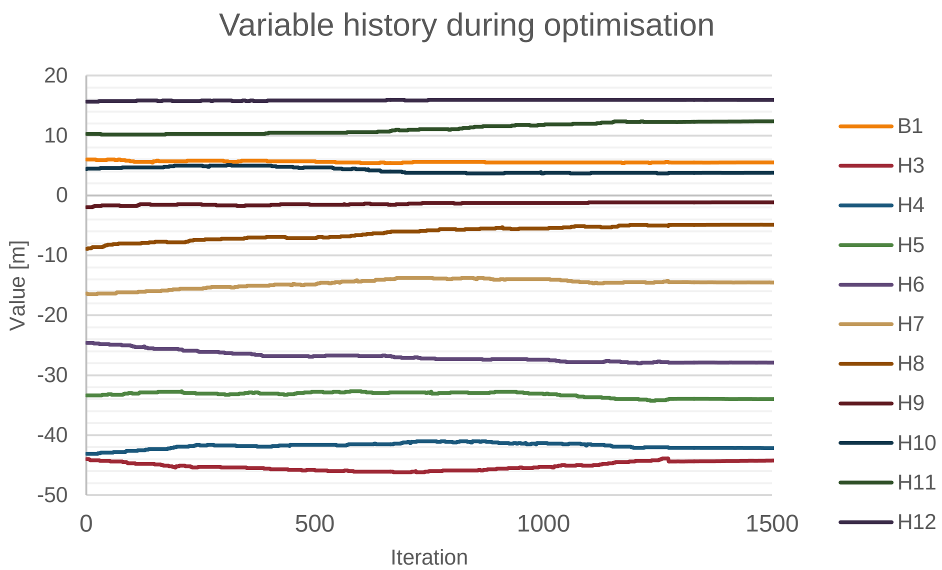

Figure 22 - Optimisation variable history

The nonlinear paths some of the variables follow is visible. It is discovered that it is very difficult for the optimisation to find improvements if too large a step is taken in a single iteration, this is evidence that the variables are influencing each other, as may be expected.

The final value of the variables is shown in table 9:

Variable	Start value	Final value
B1	6	5.62
H3	-44.001	-43.770
H4	-43.127	-42.130
H5	-33.373	-33.970
H6	-24.614	-27.790
H7	-16.371	-14.480
H8	-8.922	-4.902
H9	-1.958	-1.108
H10	4.378	3.798
H11	10.262	12.540
H12	15.651	15.930

Table 9 - Optimisation variables start and end value

Plotting the node coordinates for the side view of the OC4 tower, the optimised configuration can be clearly illustrated (figure 23 and 24). It is interesting to note that only a small change in the total width of the tower has taken place, compared to the relative changes in node elevations.

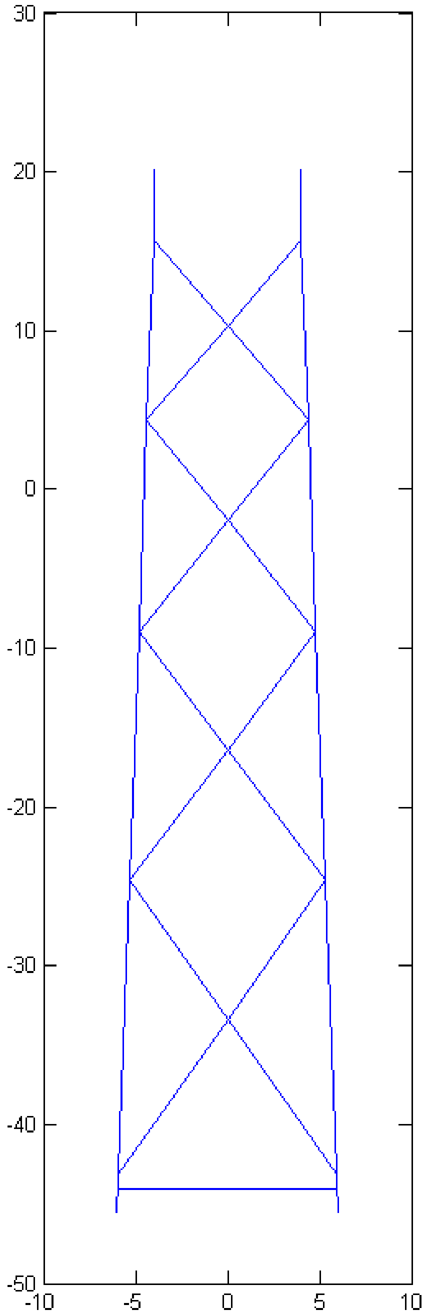

Figure 24 - OC4 original design

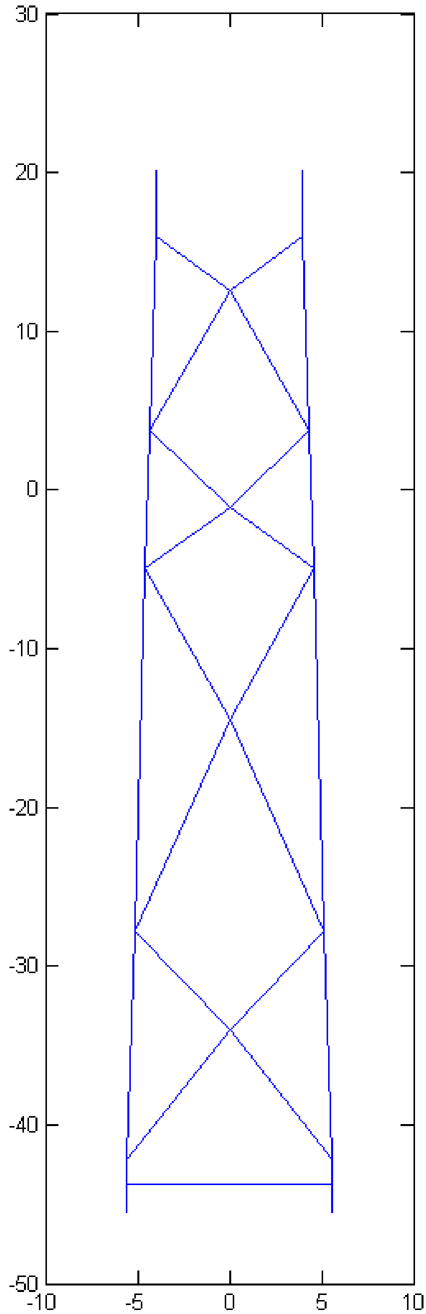

Figure 23 - OC4 optimised design

Plotting all iterations in the same plot gives a view of the total movement of the nodes (figure 25).

This again shows the stability of tower width variable, where some height variables changes significantly.

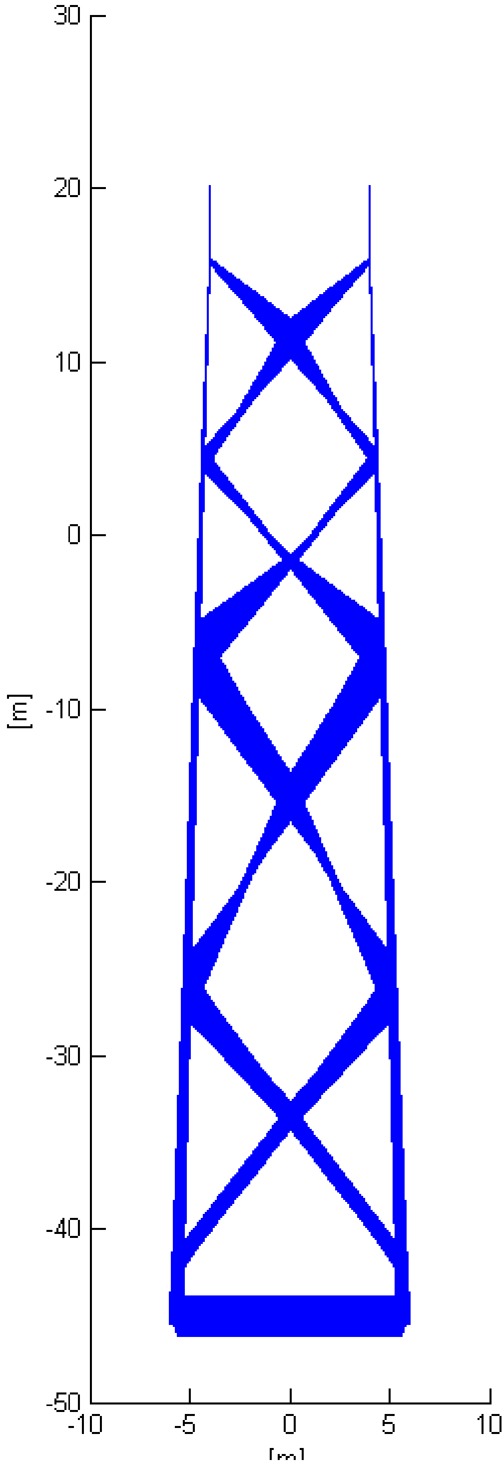

Figure 25 - Plot of all node coordinates seen

By looking at the stress response for the different load cases the two jacket configurations can be evaluated. In figure (26-30) fragments of these responses is shown together with the similar fragment from the original design. 15 second fragments from near the end of the analysis for load case 1-4 and a 60 second fragment near the shutdown incident for load case 5 is chosen. This is because the end of the analysis is expected to show greater accuracy, but for load case 5 the shutdown incident is the area of importance. Node 11 is chosen as an example, since this node generally shows poor performance.

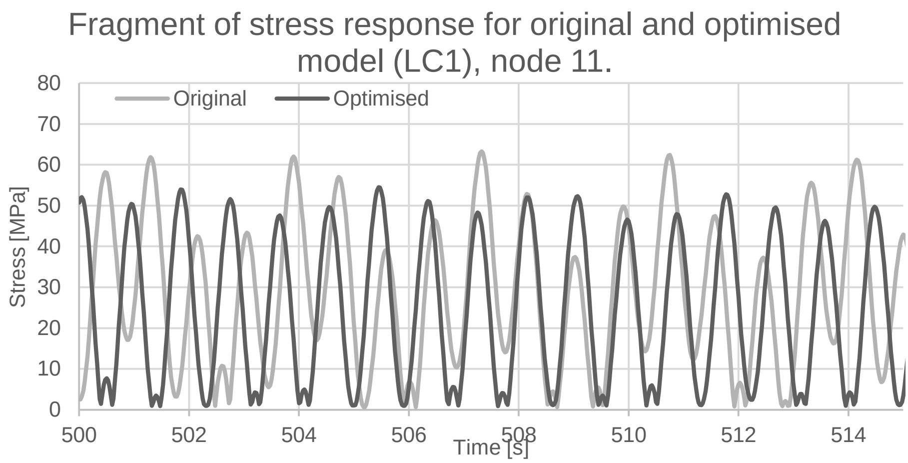

Figure 26 - Stress response fragment LC1

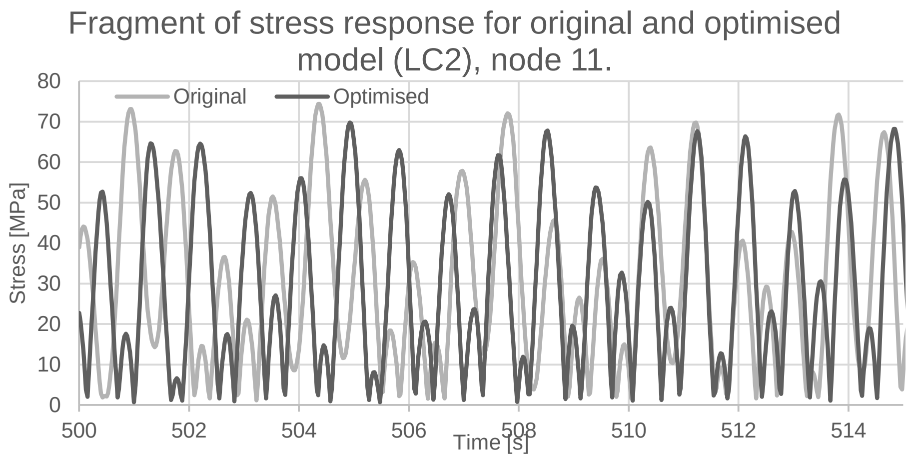

Figure 27 - Stress response fragment LC2

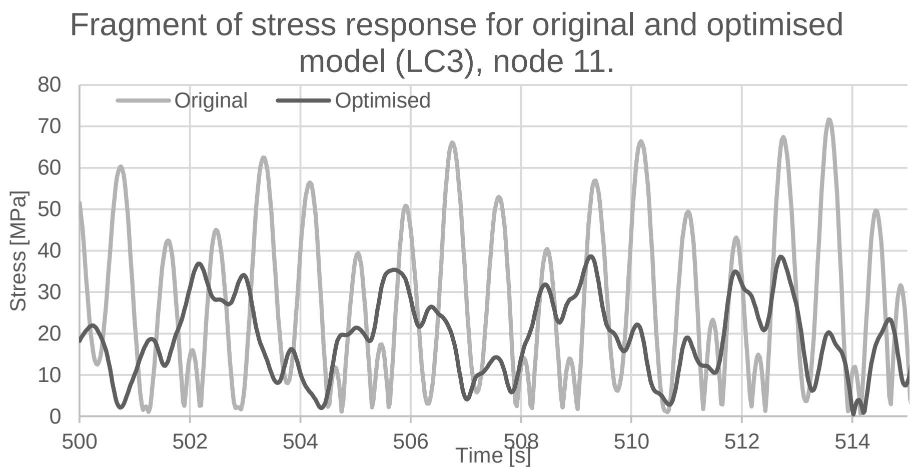

Figure 28 - Stress response fragment LC3

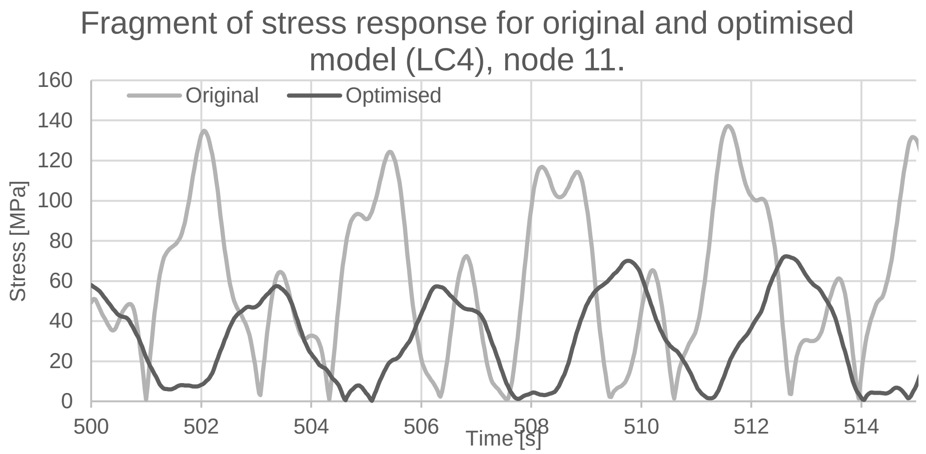

Figure 29 - Stress response fragment LC4

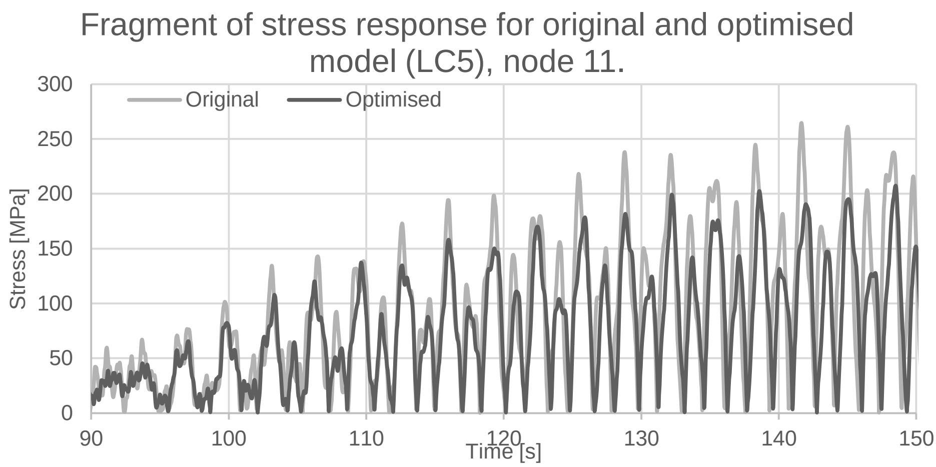

Figure 30 - Stress response fragment LC5

By running a modal analysis on both configurations, the natural frequencies can be investigated. All eigenfrequencies below 10Hz is shown in figure 31. It is clear that the optimised model shows slightly softer tendencies. The difference ranges from approximately 0.8% for the lowest frequencies to near 20% for higher frequencies.

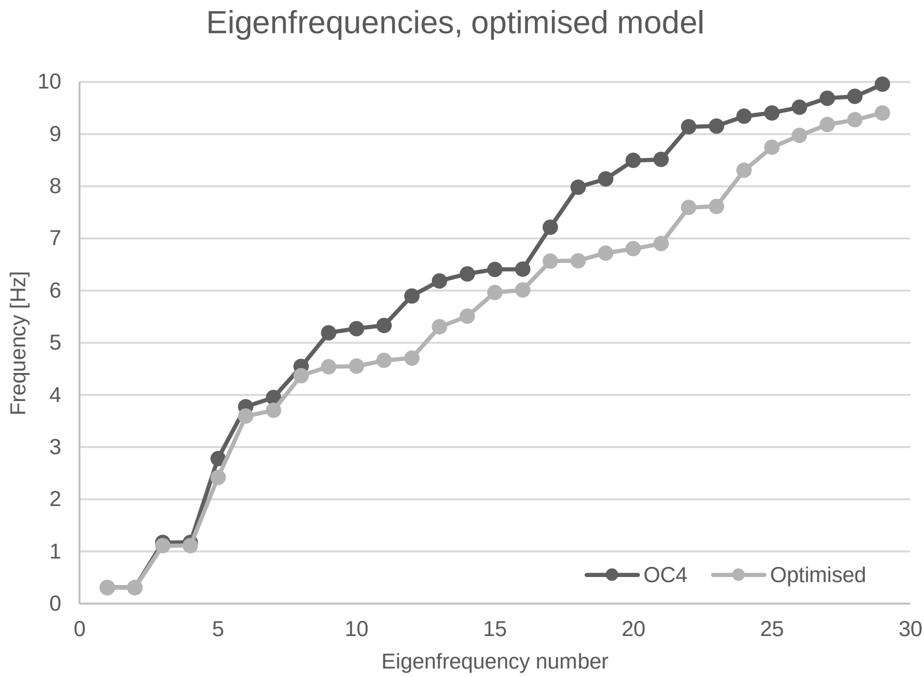

Figure 31 - Model comparison, eigenfrequencies

By running the same fatigue calculations as earlier shown, the results for the optimised model can be compared to the original results. In figure 32 this is shown for the worst node at both configurations and all load cases. The results are normalised by a magnitude equal to that of load case 1 for the original model.

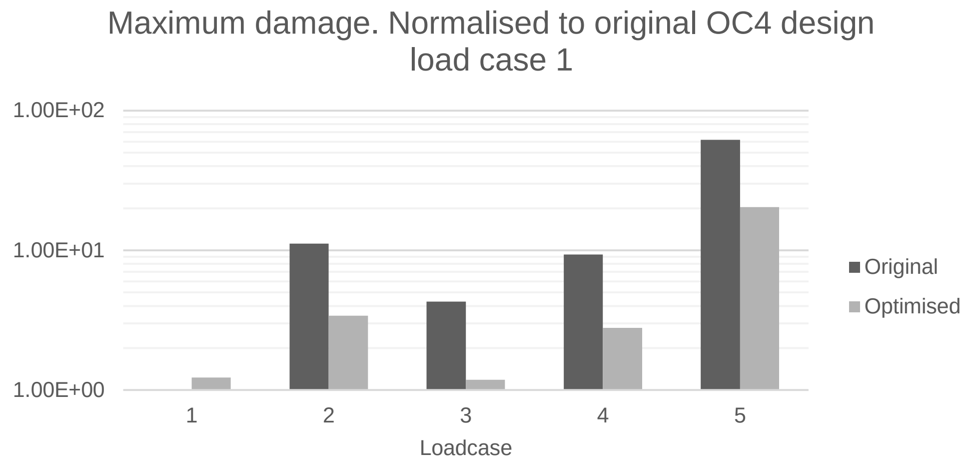

Figure 32 - Model comparison, maximum damage

The optimised model shows better results for load case 2-5, but slightly worse for load case 1. The plot is logarithmic so that any visual differences may be deceiving. The difference in the results compared to the original is better illustrated in figure 33 where maximum damage ratio between original and optimised model is plotted.

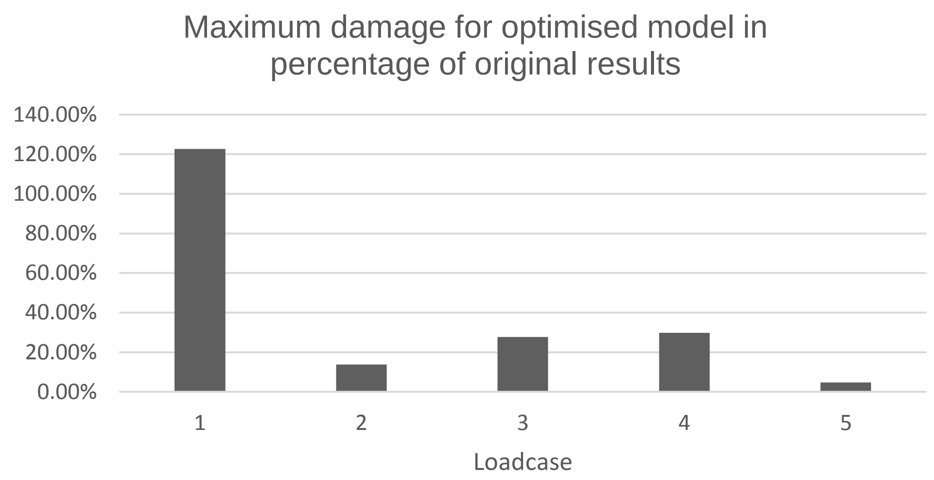

Figure 33 - Model comparison, maximum damage vs original

Looking further into the results, they can be divided into individual nodes. The results are again normalised (maximum damage for original model LC1 node 11 is equal to 1) and are shown in figure 34 to 38.

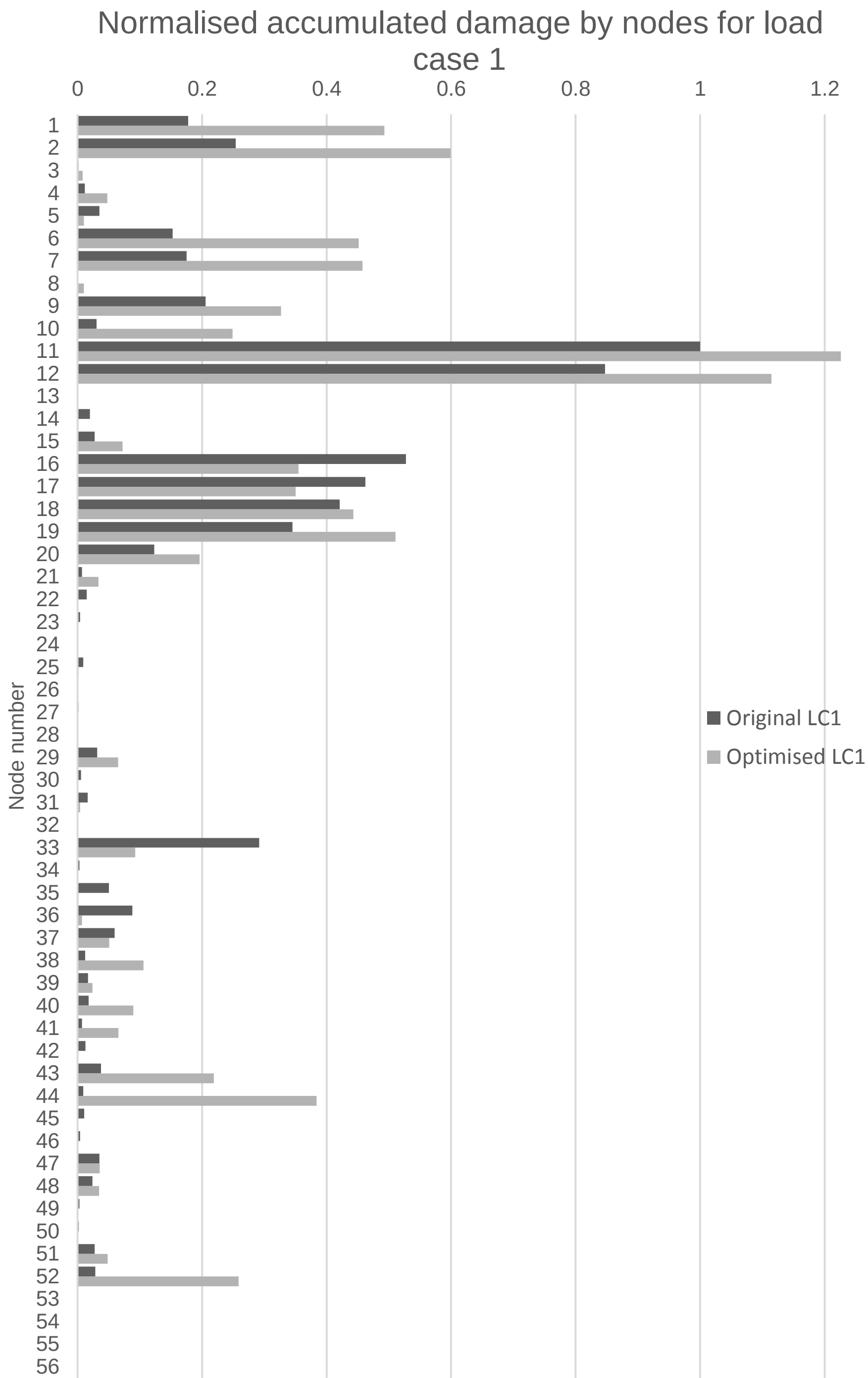

Figure 34 - Accumulated damage by nodes LC1

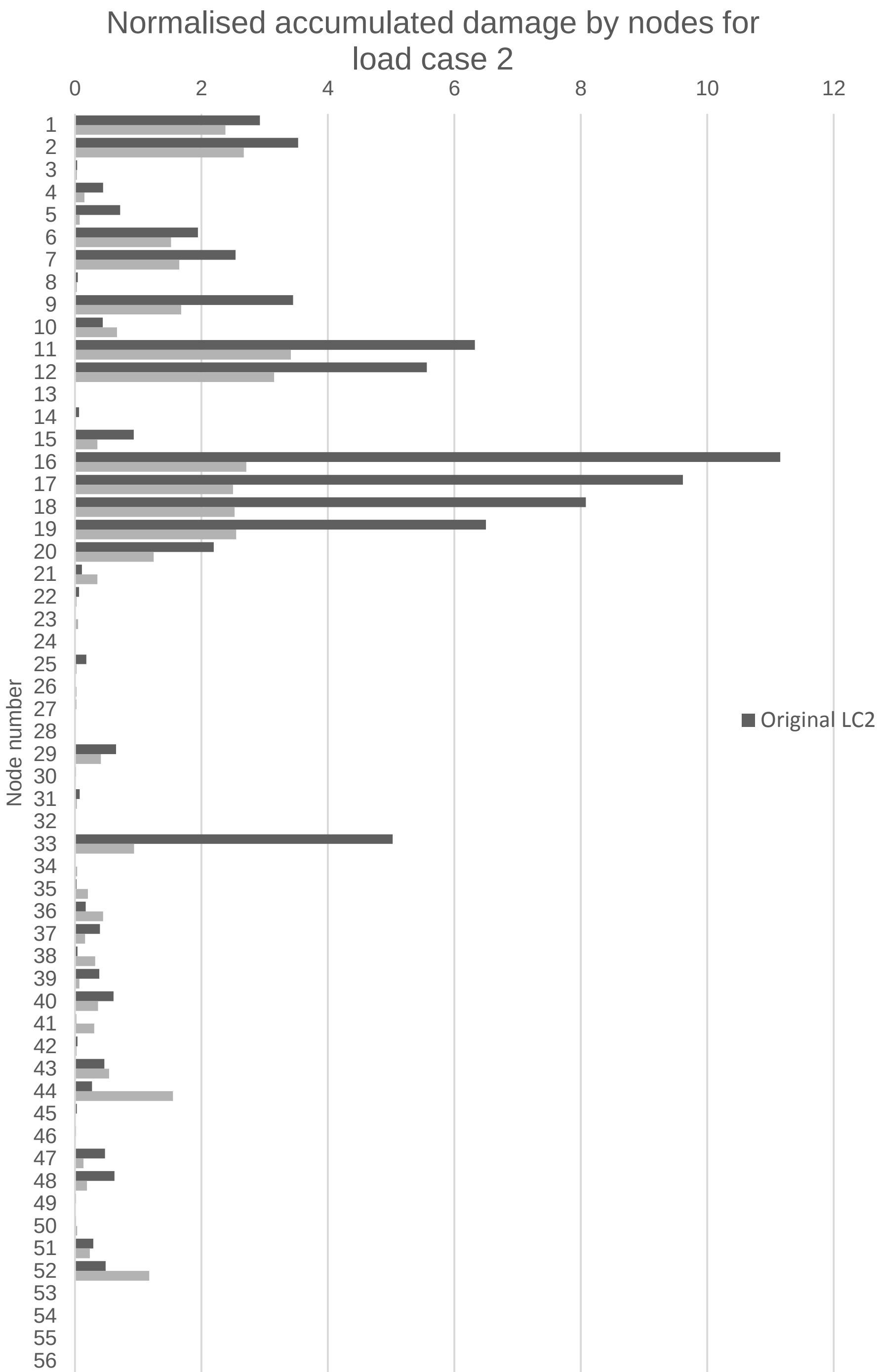

Figure 35 - Accumulated damage by nodes LC2

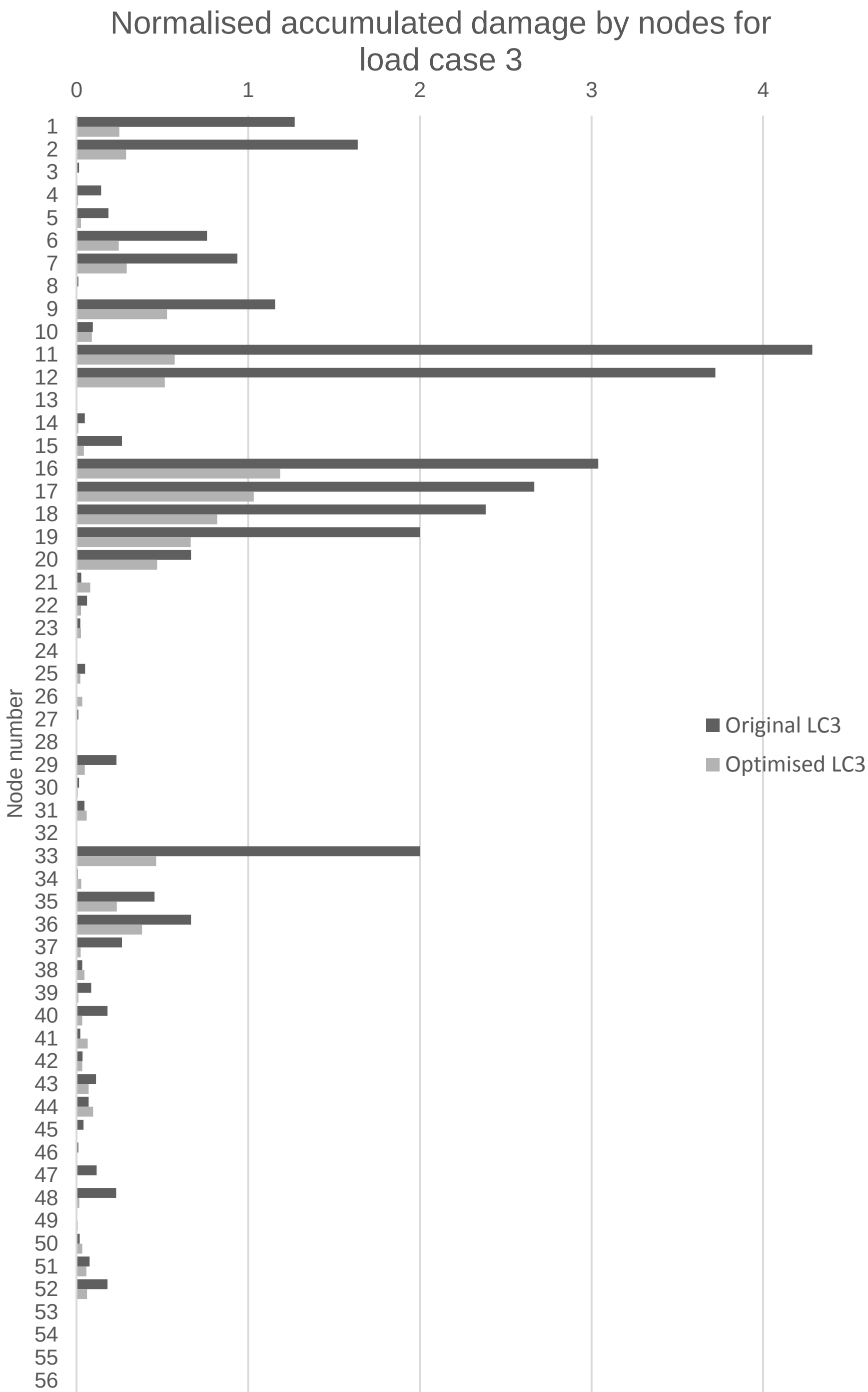

Figure 36 - Accumulated damage by nodes LC3

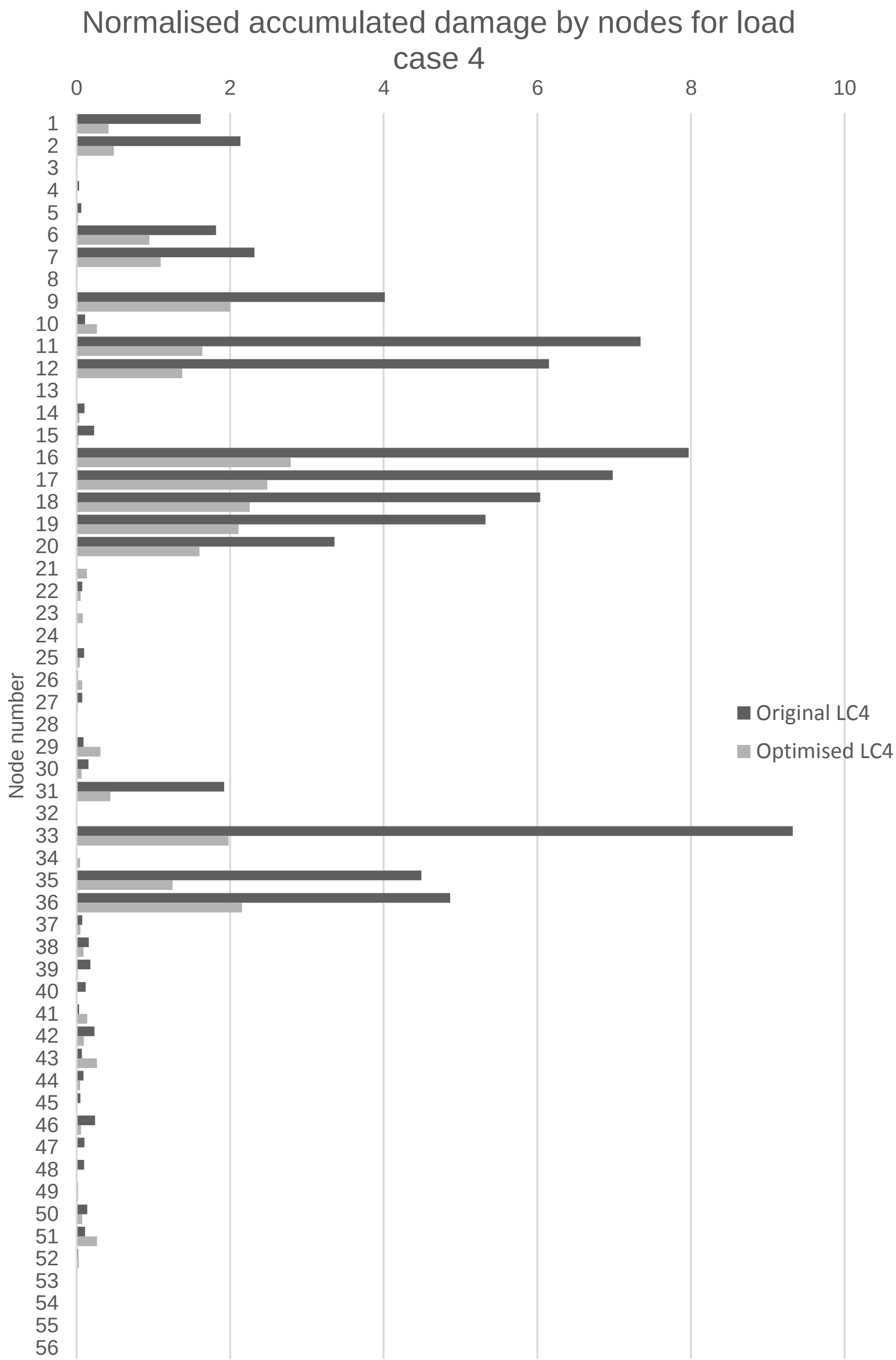

Figure 37 - Accumulated damage by nodes LC4

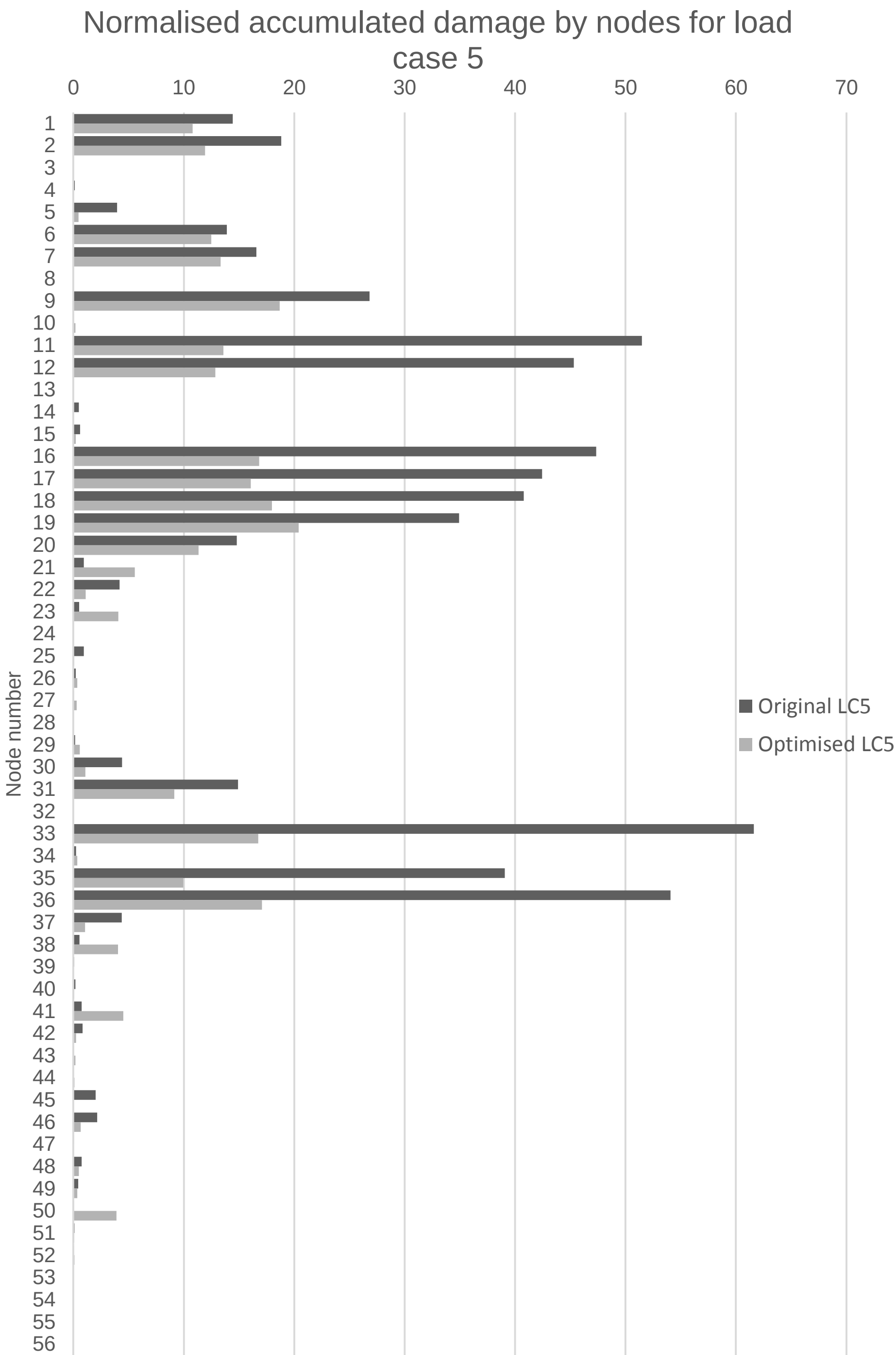

Figure 38 - Accumulated damage by nodes LC5

11 ANALYSIS

11.1 GENERAL

From the results, few definitive conclusion can be made. An idealised model, as used here, is not able to accurately evaluate the stress situation in structural joints where several other parameters are in effect. These are parameters related to machining, welding, fastening etc.

The dynamic nature of the system makes the relation between wind- and wave loading and structural response complex (14). This makes any interpolation and extrapolation between and outside load cases difficult, and the results should be considered representative for the given situations only. In addition, no check towards a reference is done, other than assessing if the results seems probable and reasonable. Still, some indications of the validity of the model exists. For example is the natural frequency of the base model within the expected area and large errors in model setup is therefore less probable, but not excluded.

Only one wind and wave direction is checked (positive x), and different combinations and directions may yield other results. The symmetric nature of the structure makes sure that rotations of 90 degrees is valid. The results do not take this into account, but it can be extrapolated with confidence.

All analysis of output stress for this projects is Von Mises, which is not particularly suited for fatigue calculations since no attention is paid to the compressive or tensile nature of the stresses. This means that for Rainflow counting, two full cycles are counted for every real cycle if the real stress is changing sign during the cycle.

Load case 5 (emergency shutdown) is a highly dynamic event and the model shows some difficulty keeping up with this. This load case should therefore be explored further.

11.2 STRUCTURAL REDUNDANCY

Most uncertainties for structural redundancy are reduced by not changing the model in any other way than the removal of one member at the time.

From the results, it can be seen that the removal of one member has drastically different effects for the different load cases and that the removal of no member should result in immediate collapse. No single point of stress larger than the yield stress is observed.

The removal of leg members generally has a larger influence than the removal of brace members, this is further confirmed by looking at the effect on the natural frequency.

Structural redundancy is about consequence management since an unlikely event must occur for a structural member to fail. Such an event would also be expected to alter the remaining structure in a significant way. That is, an event that causes a member to fail may also be expected to put significant stress on the joints which this member was connected to, and the rest of the structure. This is not taken into consideration.

11.3 OPTIMISATION

It is clear that it is possible to design a jacket structure that has increased structural performance compared to the original OC4 design but further constraints are taken into the account for the original design.

The optimisation routine is made as a proof of concept. The original design is assumed to be sufficient in respect to life time and structural strength. The optimised design do not use less steel and would not be cheaper to manufacture, and is therefore expected to be more expensive. The benefit of an optimisation is in further advancements to create cheaper and safer structures, longer life times and generally increased economic performance. This is outside of the scope of this project, but installation and material costs can be significant for a project and a reduction in these may be beneficial on the cost of increased design time (14).

The optimised design have not been checked against any design codes, and constraints related to buckling safety factors etc. should be included for a verification of the design. It is also noted from the results, the performance of the optimised structure is worse than the original in load case 1, but significantly improved in load case 4 and 5. The full life cycle of the wind turbine is not known, it is therefore not given how the performance over the lifetime is changed due to the optimisation.

A tendency for the optimisation to move structural joints towards mean sea level is observed. This is expected to be in relation to the load application from the Abaqus Aqua.

12 CONCLUSION

In this project the support jacket of an offshore wind turbine has been explored. Both the properties related to structural redundancy, by the loss of structural members, and shape optimisation of the structure have been evaluated using the scripting interface of the commercial finite element analysis software Abaqus CAE.

In structural redundancy, five different load cases have been evaluated with a large set of different members removed. These analysis have formed the basis for fatigue estimations. Fatigue has been evaluated at structural joints. No general conclusion can be drawn from these results alone, but the tendency shows that the removal of members part of the jacket legs have the largest impact on the damage accumulation. For all load cases, there exist configurations where members are removed and the performance are improved, but no member can be removed and improve performance for all load cases. Load case 5 seems to be the least critical, with minimal deviation between the configurations.

The optimisation done in this report, has mostly been a proof of concept. Again, fatigue and the five load cases have been considered. 1500 iterations have been performed to find new coordinates for the structural joints of the jacket. This has resulted in a configuration that performs better than the original for four out of the five load cases.

13 FURTHER WORK

This work can be improved upon and further evolved by several means.

The structural base model in the scripting interface are set up for new analyses. This means that a range of new load situations, jacket configurations etc. can be evaluated. Closest to the work done in this report is possibly the full-height lattice tower, where optimisation can constitute even larger benefits. In addition to further work on the same structure, the model can be adapted for use on other structures. This is rooted in the very intention of the procedure used, since the time savings primarily are expected with continued use of the analysis.

In structural redundancy, only loading due to wind in x-direction is included. This would not necessarily be the only possible situation, nor the worst for all members. Additionally, no structural design code checks have been included and no attention have been paid to stress concentration areas in the structural joint. The latter is a limitation in the model (15).

For general optimisation much work is needed to get something close to an optimal structure. Here, an approach for optimisation of node placement is explored, but only a simple algorithm has been used for the optimisation process. This means there is a large potential to increase the efficiency of the optimisation. Optimisation of other parameters have not been included, and this should take part in a proper optimisation. These parameters include member sizing and further constraints of the structure in respect to response, safety and cost.

14 References

1. Fischer T, de Vries W, Schmidt B. Upwind Design Basis (WP4: Offshore Foundations and Support Structures). Project Description. Stuttgart: Universität Stuttgart, Endowed Chair of Wind Energy (SWE) at the Institute of Aircraft Design; 2010.

2. Robertson A, Jonkman J, Musial W, Vorpahl F, Popko W. Offshore Code Comparison Collaboration, Continuation: Phase II Results of a Floating Semisubmersible Wind System. In EWEA Offshore; 2013; Frankfurt. p. 10.

3. Vorpahl F, Popko , Kaufer D. Description of a basic model of the `UpWind reference jacket' for code comparison in the OC4 project under IEA Wind Annex 30. Bremerhaven: Fraunhofer Institute for Wind Energy and Energy System Technology, Department of Turbine Simulation, Software Development and Aerodynamics; 2013.

4. International Society of Offshore and Polar Engineers. Offshore Code Comparison Collaboration Continuation (OC4), Phase I – Results of Coupled Simulations of an Offshore Wind Turbine with Jacket Support Structure. In International Offshore and Polar Engineering Conference; 2012; Rhodes. p. 10.

5. Kühn M. Soft or stiff - A fundamental question for designers of offshore wind energy converters. In European Wind Energy Conference; 1997; Dublin. p. 4.

6. Dassault Systèmes. Abaqus 6.13 Online Documentation. 2013..

7. Puri GM. Python Scripts for Abaqus. 1st ed. Charleston; 2011.

8. Christensen PW, Klarbring A. An Introduction to Structural Optimization Gladwell GML, editor. Linköping: Springer; 2009.

9. Vorpahl F, Popko. Description of the Load Cases and Output Sensors to be Simulated in the OC4 Project under IEA Wind Annex 30. Technical Report. Bremerhaven: Fraunhofer Institute for Wind Energy and Energy System Technology IWES, Department of Turbine Simulation, Software Development and Aerodynamics; 2013.

10. Downing SD, Socie DF. Simple rainflow counting algorithms. International Journal of Fatigue. 1982 January; 4(1).

11. Singh KL, Ranganath VR. Cycle counting using rainflow algorithm for fatigue analysis. Bangalore: NAL, Structural Integrity Divison.

12. Nieslony A. Mathworks fileexchange. [Online].; 2010 [cited 2015 February 24. Available from: http://www.mathworks.com/matlabcentral/fileexchange/3026-rainflow-counting-algorithm.

13. Muskulus M, Schafhirt S. Design Optimization of Wind Turbine Support Structures—A Review. Journal of Ocean and Wind Energy. 2014 February; 1(1).

14. Hau E. Wind Turbines - Fundamentals, Technologies, Application, Economics. 3rd ed. Munich: Springer; 2013.

15. Dubois J, Muskulus M, Schaumann P. Advanced representation of tubular joints in jacket models for offshore wind turbine simulation. In DeepWind; 2013; Trondheim. p. 10.

APPENDIX 1 – NODE COORDINATES

Node	x [m]	y [m]	z [m]	Node	x [m]	y [m]	z [m]
1	6	6	-45.5	29	-4.82	-4.82	-8.922
2	6	6	-45	30	-4.385	-4.385	4.378
3	5.967	5.967	-44.001	31	-4.016	-4.016	15.651
4	5.939	5.939	-43.127	32	-4	-4	16.15
5	5.333	5.333	-24.614	33	4.82	-4.82	-8.922
6	-6	6	-45.5	34	4.385	-4.385	4.378
7	-6	6	-45	35	4.016	-4.016	15.651
8	-5.967	5.967	-44.001	36	4	-4	16.15
9	-5.939	5.939	-43.127	41	5.62	0	-33.373
10	-5.333	5.333	-24.614	42	5.62	0	-33.373
11	-6	-6	-45.5	43	0	5.62	-33.373
12	-6	-6	-45	44	0	5.62	-33.373
13	-5.967	-5.967	-44.001	45	5.064	0	-16.371
14	-5.939	-5.939	-43.127	46	5.064	0	-16.371
15	-5.333	-5.333	-24.614	47	0	5.064	-16.371
16	6	-6	-45.5	48	0	5.064	-16.371
17	6	-6	-45	49	4.592	0	-1.958
18	5.967	-5.967	-44.001	53	4.592	0	-1.958
19	5.939	-5.939	-43.127	51	0	4.592	-1.958
20	5.333	-5.333	-24.614	52	0	4.592	-1.958
21	4.82	4.82	-8.922	53	4.193	0	10.262
22	4.385	4.385	4.378	54	4.193	0	10.262
23	4.016	4.016	15.651	55	0	4.193	10.262
24	4	4	16.15	56	0	4.193	10.262
25	-4.82	4.82	-8.922	57	4	4	20.15

26	-4.385	4.385	4.378	58	4	4	20.15
27	-4.016	4.016	15.651	59	4	4	20.15
28	-4	4	16.15	60	4	4	20.15

APPENDIX 2 – STRUCTURAL MEMBERS DEFINITIONS

Member	Node 1	Node 2	Property set	Member	Node 1	Node 2	Property set
1	1	2	2	57	5	45	1
2	2	3	2	58	45	33	1
3	3	4	2	59	20	45	1
4	4	5	2	60	45	21	1
5	6	7	2	61	10	46	1
6	7	8	2	62	46	29	1
7	8	9	2	63	15	46	1
8	9	10	2	64	46	25	1
9	11	12	2	65	5	47	1
10	12	13	2	66	47	25	1
11	13	14	2	67	10	47	1
12	14	15	2	68	47	21	1
13	16	17	2	69	20	48	1
14	17	18	2	70	48	29	1
15	18	19	2	71	15	48	1
16	19	20	2	72	48	33	1
17	5	21	3	73	21	49	1
18	21	22	3	74	49	34	1
19	22	23	3	75	33	49	1
20	23	24	3	76	49	22	1
21	10	25	3	77	25	50	1
22	25	26	3	78	50	30	1
23	26	27	3	79	29	50	1
24	27	28	3	80	50	26	1

25	15	29	3	81	21	51	1
26	29	30	3	82	51	26	1
27	30	31	3	83	25	51	1
28	31	32	3	84	51	22	1
29	20	33	3	85	33	52	1
30	33	34	3	86	52	30	1
31	34	35	3	87	29	52	1
32	35	36	3	88	52	34	1
37	3	8	1	89	22	53	1
38	8	13	1	90	53	35	1
39	13	18	1	91	34	53	1
40	18	3	1	92	53	23	1
41	4	41	1	93	26	54	1
42	41	20	1	94	54	31	1
43	19	41	1	95	30	54	1
44	41	5	1	96	54	27	1
45	9	42	1	97	22	55	1
46	42	15	1	98	55	27	1
47	14	42	1	99	26	55	1
48	42	10	1	100	55	23	1
49	4	43	1	101	34	56	1
50	43	10	1	102	56	31	1
51	9	43	1	103	30	56	1
52	43	5	1	104	56	35	1
53	19	44	1	105	24	57	4
54	44	15	1	106	28	58	4
55	14	44	1	107	32	60	4
56	44	20	1	108	36	59	4

APPENDIX 3 – PYTHON SCRIPT FOR STRUCTURAL REDUNDANCY CHECK IN ABAQUS (LC1, LOADCASE EXCLUDED FOR SIZE CONSIDERATIONS)

```
# -*- coding: mbcs -*-
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
from regionToolset import *

import random


half_member_list=[0,3,4,7,8,17,18,19,20,21,22,23,24,33,34,3
6,37,40,41,45,46,47,48,53,56,57,60,61,62,63,64,69,72,73,76,
77,78,79,80,85,88,89,92,93,94,95,96]
count=0


for q in half_member_list:

    iterations=1
    time=1
    numinc=10
    increment=0.1
    rotation=3.14159265359
    load_amplitude=1000

    count=count+1
    jobname="Job1-"+str(q)

    mdb.Job(atTime=None, contactPrint=OFF, description='',
echoPrint=OFF,
```

```
        explicitPrecision=SINGLE,
getMemoryFromAnalysis=True, historyPrint=OFF,
        memory=90, memoryUnits=PERCENTAGE, model='Model-1',
modelPrint=OFF,
        multiprocessingMode=DEFAULT, name=jobname,
nodalOutputPrecision=SINGLE,
        numCpus=1, numGPUs=1, queue=None, scratch='',
type=ANALYSIS,
        userSubroutine='', waitHours=0, waitMinutes=0)

    H1=-45.50000
    H2=-45.00000
    H3=-44.00100
    H4=-43.12700
    H5=-33.37300
    H6=-24.61400
    H7=-16.37100
    H8=-8.92200
    H9=-1.95800
    H10=4.37800
    H11=10.26200
    H12=15.65100
    H13=16.15000
    H14=20.15000
    B1=6.00000
    B2=5.96700
    B3=5.93900
    B4=5.62000
    B5=5.33300
    B6=5.06400
    B7=4.82000
    B8=4.59200
    B9=4.38500
    B10=4.19300
    B11=4.01600
    B12=4.00000


    CC=[0,0,18.15]
    concrete_mass=666000
    concrete_w=9.6
    concrete_h=4.0
    concrete_Ih=(1.0/12)*concrete_mass*(2*(concrete_w)**2)

concrete_Iw=(1.0/12)*concrete_mass*((concrete_h)**2+(concre
te_w)**2)
    load_point=[0,0,20.15]
    ##
```

```
V0=[H2,0,0]
V1=[H3,0,0]
V2=[H4,0,0]
V3=[H5,0,0]
V4=[H6,0,0]
V5=[H7,0,0]
V6=[H8,0,0]
V7=[H9,0,0]
V8=[H10,0,0]
V9=[H11,0,0]
V10=[H12,0,0]
V11=[H13,0,0]
V12=[B1,0,0]
V13=[B2,0,0]
V14=[B3,0,0]
V15=[B4,0,0]
V16=[B5,0,0]
V17=[B6,0,0]
V18=[B7,0,0]
V19=[B8,0,0]
V20=[B9,0,0]
V21=[B10,0,0]
V22=[B11,0,0]
V23=[B12,0,0]


##
towerh1=20.15
towerh2=21.15
towerh3=32.15
towerh4=42.15
towerh5=54.15
towerh6=64.15
towerh7=74.15
towerh8=83.15
towerh9=88.15
towerconnect=[0,0,20.15]

point_mass_1=(0,0,20.15)
point_mass_2=(0,0,54.15)
point_mass_3=(0,0,88.15)


nacelle_overhang=1.90
hub_overhang=5.0/1

s_overhang=sin(rotation)
c_overhang=cos(rotation)
```

```
    nacelle_cg=(-nacelle_overhang*c_overhang,-
nacelle_overhang*s_overhang,towerh9+1.75)

hub_cg=(hub_overhang*c_overhang,hub_overhang*s_overhang,tow
erh9+2.40)

    nacelle_dummy_wire=(-nacelle_overhang*c_overhang,-
nacelle_overhang*s_overhang,towerh9+1.75+0.05)

hub_dummy_wire=(hub_overhang*c_overhang,hub_overhang*s_over
hang,towerh9+2.40+0.05)

    variables=[0 for i in xrange(iterations)]
    variables[0]=[-45.00000,-44.00100,-43.12700,-33.37300,-
24.61400,-16.37100,-8.92200,-
1.95800,4.37800,10.26200,15.65100,16.15000,6.00000,5.96700,
5.93900,5.62000,5.33300,5.06400,4.82000,4.59200,4.38500,4.1
9300,4.01600,4.00000]
    num_var=len(variables[0])

    max_stress=[0]*iterations
    max_stress_element_label=[0]*iterations
    max_stress_object_index=[0]*iterations

    t=0

    ############## Job

    num_var=len(variables[0])

    store=[0 for i in xrange(iterations)]
    store[0]=tuple(variables[0])

    file_open=open("script_run.txt", "w")



    ############## Amplitudes

    ## Initial displacement amplitude
    mdb.models['Model-1'].TabularAmplitude(data=((0.0,
1.0), (0.025, 0.0), (600.0,
        0.0)), name='BC_amp', smooth=SOLVER_DEFAULT,
timeSpan=STEP)

    ##  LC1-fy:
    mdb.models['Model-1'].TabularAmplitude(data=((0.0, -
5273.01758), (0.025,
```

```
            -4964.41846), (0.05, -4390.28418), ..., (600.0,
            23448.4375)), name='LC1-fy',
smooth=SOLVER_DEFAULT, timeSpan=STEP)
    ##  LC1-fz:
    mdb.models['Model-1'].TabularAmplitude(data=(...),
name='LC1-fz',
        smooth=SOLVER_DEFAULT, timeSpan=STEP)
    ##  LC1-mx:
    mdb.models['Model-1'].TabularAmplitude(data=(...),
name='LC1-mx',
        smooth=SOLVER_DEFAULT, timeSpan=STEP)
    ##  LC1-my:
    mdb.models['Model-1'].TabularAmplitude(data=(...),
name='LC1-my',
        smooth=SOLVER_DEFAULT, timeSpan=STEP)
    ##  LC1-mz:
    mdb.models['Model-1'].TabularAmplitude(data=(...),
name='LC1-mz', smooth=SOLVER_DEFAULT, timeSpan=STEP)


    rand=int(ceil((num_var)*random.random()-1))
    sign=random.choice([-1,1])

    H2=variables[t][0]
    H3=variables[t][1]
    H4=variables[t][2]
    H5=variables[t][3]
    H6=variables[t][4]
    H7=variables[t][5]
    H8=variables[t][6]
    H9=variables[t][7]
    H10=variables[t][8]
    H11=variables[t][9]
    H12=variables[t][10]
    H13=variables[t][11]
    B1=variables[t][12]
    B2=variables[t][13]
    B3=variables[t][14]
    B4=variables[t][15]
    B5=variables[t][16]
    B6=variables[t][17]
    B7=variables[t][18]
    B8=variables[t][19]
    B9=variables[t][20]
    B10=variables[t][21]
    B11=variables[t][22]
    B12=variables[t][23]

    model=mdb.models['Model-1']
```

```
############## Material

model.Material(name='Steel')

model.materials['Steel'].Elastic(table=((207000000000.0,
0.3),
        ))
    model.materials['Steel'].Density(table=((7850.0, ), ))
    model.materials['Steel'].Damping(alpha=0.01)


############## Nodes

N1=(B1,B1,H1)
N2=(B1,B1,H2)
N3=(B2,B2,H3)
N4=(B3,B3,H4)
N5=(B5,B5,H6)
N6=(-B1,B1,H1)
N7=(-B1,B1,H2)
N8=(-B2,B2,H3)
N9=(-B3,B3,H4)
N10=(-B5,B5,H6)
N11=(-B1,-B1,H1)
N12=(-B1,-B1,H2)
N13=(-B2,-B2,H3)
N14=(-B3,-B3,H4)
N15=(-B5,-B5,H6)
N16=(B1,-B1,H1)
N17=(B1,-B1,H2)
N18=(B2,-B2,H3)
N19=(B3,-B3,H4)
N20=(B5,-B5,H6)
N21=(B7,B7,H8)
N22=(B9,B9,H10)
N23=(B11,B11,H12)
N24=(B12,B12,H13)
N25=(-B7,B7,H8)
N26=(-B9,B9,H10)
N27=(-B11,B11,H12)
N28=(-B12,B12,H13)
N29=(-B7,-B7,H8)
N30=(-B9,-B9,H10)
N31=(-B11,-B11,H12)
N32=(-B12,-B12,H13)
N33=(B7,-B7,H8)
N34=(B9,-B9,H10)
```

```
N35=(B11,-B11,H12)
N36=(B12,-B12,H13)
N37=(B4,0,H5)
N38=(-B4,0,H5)
N39=(0,B4,H5)
N40=(0,-B4,H5)
N41=(B6,0,H7)
N42=(-B6,0,H7)
N43=(0,B6,H7)
N44=(0,-B6,H7)
N45=(B8,0,H9)
N46=(-B8,0,H9)
N47=(0,B8,H9)
N48=(0,-B8,H9)
N49=(B10,0,H11)
N50=(-B10,0,H11)
N51=(0,B10,H11)
N52=(0,-B10,H11)
N53=(B12,B12,H14)
N54=(-B12,B12,H14)
N55=(B12,-B12,H14)
N56=(-B12,-B12,H14)

Nmid1=(B1/2,B1/2,H1/2)
Nmid2=(B1/2,B1/2,H2/2)
Nmid3=(B2/2,B2/2,H3/2)
Nmid4=(B3/2,B3/2,H4/2)
Nmid5=(B5/2,B5/2,H6/2)
Nmid6=(-B1/2,B1/2,H1/2)
Nmid7=(-B1/2,B1/2,H2/2)
Nmid8=(-B2/2,B2/2,H3/2)
Nmid9=(-B3/2,B3/2,H4/2)
Nmid10=(-B5/2,B5/2,H6/2)
Nmid11=(-B1/2,-B1/2,H1/2)
Nmid12=(-B1/2,-B1/2,H2/2)
Nmid13=(-B2/2,-B2/2,H3/2)
Nmid14=(-B3/2,-B3/2,H4/2)
Nmid15=(-B5/2,-B5/2,H6/2)
Nmid16=(B1/2,-B1/2,H1/2)
Nmid17=(B1/2,-B1/2,H2/2)
Nmid18=(B2/2,-B2/2,H3/2)
Nmid19=(B3/2,-B3/2,H4/2)
Nmid20=(B5/2,-B5/2,H6/2)
Nmid21=(B7/2,B7/2,H8/2)
Nmid22=(B9/2,B9/2,H10/2)
Nmid23=(B11/2,B11/2,H12/2)
Nmid24=(B12/2,B12/2,H13/2)
Nmid25=(-B7/2,B7/2,H8/2)
Nmid26=(-B9/2,B9/2,H10/2)
```

```
Nmid27=(-B11/2,B11/2,H12/2)
Nmid28=(-B12/2,B12/2,H13/2)
Nmid29=(-B7/2,-B7/2,H8/2)
Nmid30=(-B9/2,-B9/2,H10/2)
Nmid31=(-B11/2,-B11/2,H12/2)
Nmid32=(-B12/2,-B12/2,H13/2)
Nmid33=(B7/2,-B7/2,H8/2)
Nmid34=(B9/2,-B9/2,H10/2)
Nmid35=(B11/2,-B11/2,H12/2)
Nmid36=(B12/2,-B12/2,H13/2)
Nmid37=(B4/2,0/2,H5/2)
Nmid38=(-B4/2,0/2,H5/2)
Nmid39=(0/2,B4/2,H5/2)
Nmid40=(0/2,-B4/2,H5/2)
Nmid41=(B6/2,0/2,H7/2)
Nmid42=(-B6/2,0/2,H7/2)
Nmid43=(0/2,B6/2,H7/2)
Nmid44=(0/2,-B6/2,H7/2)
Nmid45=(B8/2,0/2,H9/2)
Nmid46=(-B8/2,0/2,H9/2)
Nmid47=(0/2,B8/2,H9/2)
Nmid48=(0/2,-B8/2,H9/2)
Nmid49=(B10/2,0/2,H11/2)
Nmid50=(-B10/2,0/2,H11/2)
Nmid51=(0/2,B10/2,H11/2)
Nmid52=(0/2,-B10/2,H11/2)
Nmid53=(B12/2,B12/2,H14/2)
Nmid54=(-B12/2,B12/2,H14/2)
Nmid55=(B12/2,-B12/2,H14/2)
Nmid56=(-B12/2,-B12/2,H14/2)


M1=(N1,N2)
M2=(N2,N3)
M3=(N3,N4)
M4=(N4,N5)
M5=(N6,N7)
M6=(N7,N8)
M7=(N8,N9)
M8=(N9,N10)
M9=(N11,N12)
M10=(N12,N13)
M11=(N13,N14)
M12=(N14,N15)
M13=(N16,N17)
M14=(N17,N18)
M15=(N18,N19)
M16=(N19,N20)
M17=(N5,N21)
```

```
M18=(N21,N22)
M19=(N22,N23)
M20=(N23,N24)
M21=(N10,N25)
M22=(N25,N26)
M23=(N26,N27)
M24=(N27,N28)
M25=(N15,N29)
M26=(N29,N30)
M27=(N30,N31)
M28=(N31,N32)
M29=(N20,N33)
M30=(N33,N34)
M31=(N34,N35)
M32=(N35,N36)
M33=(N3,N8)
M34=(N8,N13)
M35=(N13,N18)
M36=(N18,N3)
M37=(N4,N37)
M38=(N37,N20)
M39=(N19,N37)
M40=(N37,N5)
M41=(N9,N38)
M42=(N38,N15)
M43=(N14,N38)
M44=(N38,N10)
M45=(N4,N39)
M46=(N39,N10)
M47=(N9,N39)
M48=(N39,N5)
M49=(N19,N40)
M50=(N40,N15)
M51=(N14,N40)
M52=(N40,N20)
M53=(N5,N41)
M54=(N41,N33)
M55=(N20,N41)
M56=(N41,N21)
M57=(N10,N42)
M58=(N42,N29)
M59=(N15,N42)
M60=(N42,N25)
M61=(N5,N43)
M62=(N43,N25)
M63=(N10,N43)
M64=(N43,N21)
M65=(N20,N44)
M66=(N44,N29)
```

```
M67=(N15,N44)
M68=(N44,N33)
M69=(N21,N45)
M70=(N45,N34)
M71=(N33,N45)
M72=(N45,N22)
M73=(N25,N46)
M74=(N46,N30)
M75=(N29,N46)
M76=(N46,N26)
M77=(N21,N47)
M78=(N47,N26)
M79=(N25,N47)
M80=(N47,N22)
M81=(N33,N48)
M82=(N48,N30)
M83=(N29,N48)
M84=(N48,N34)
M85=(N22,N49)
M86=(N49,N35)
M87=(N34,N49)
M88=(N49,N23)
M89=(N26,N50)
M90=(N50,N31)
M91=(N30,N50)
M92=(N50,N27)
M93=(N22,N51)
M94=(N51,N27)
M95=(N26,N51)
M96=(N51,N23)
M97=(N34,N52)
M98=(N52,N31)
M99=(N30,N52)
M100=(N52,N35)
M101=(N24,N53)
M102=(N28,N54)
M103=(N32,N56)
M104=(N36,N55)
```

```
M=[[N1,N2],[N2,N3],[N3,N4],[N4,N5],[N6,N7],[N7,N8],[N8,N9],
[N9,N10],[N11,N12],[N12,N13],[N13,N14],[N14,N15],[N16,N17],

[N17,N18],[N18,N19],[N19,N20],[N5,N21],[N21,N22],[N22,N23],
[N23,N24],[N10,N25],[N25,N26],[N26,N27],[N27,N28],[N15,N29]
,

[N29,N30],[N30,N31],[N31,N32],[N20,N33],[N33,N34],[N34,N35]
```

```
,[N35,N36],[N3,N8],[N8,N13],[N13,N18],[N18,N3],[N4,N37],[N3
7,N20],

[N19,N37],[N37,N5],[N9,N38],[N38,N15],[N14,N38],[N38,N10],[
N4,N39],[N39,N10],[N9,N39],[N39,N5],[N19,N40],[N40,N15],[N1
4,N40],

[N40,N20],[N5,N41],[N41,N33],[N20,N41],[N41,N21],[N10,N42],
[N42,N29],[N15,N42],[N42,N25],[N5,N43],[N43,N25],[N10,N43],
[N43,N21],

[N20,N44],[N44,N29],[N15,N44],[N44,N33],[N21,N45],[N45,N34]
,[N33,N45],[N45,N22],[N25,N46],[N46,N30],[N29,N46],[N46,N26
],

[N21,N47],[N47,N26],[N25,N47],[N47,N22],[N33,N48],[N48,N30]
,[N29,N48],[N48,N34],[N22,N49],[N49,N35],[N34,N49],[N49,N23
],

[N26,N50],[N50,N31],[N30,N50],[N50,N27],[N22,N51],[N51,N27]
,[N26,N51],[N51,N23],[N34,N52],[N52,N31],[N30,N52],[N52,N35
],
       [N24,N53],[N28,N54],[N32,N56],[N36,N55]]


Mmid=[[Nmid1,Nmid2],[Nmid2,Nmid3],[Nmid3,Nmid4],[Nmid4,Nmid
5],[Nmid6,Nmid7],[Nmid7,Nmid8],[Nmid8,Nmid9],[Nmid9,Nmid10]
,

[Nmid11,Nmid12],[Nmid12,Nmid13],[Nmid13,Nmid14],[Nmid14,Nmi
d15],[Nmid16,Nmid17],[Nmid17,Nmid18],[Nmid18,Nmid19],

[Nmid19,Nmid20],[Nmid5,Nmid21],[Nmid21,Nmid22],[Nmid22,Nmid
23],[Nmid23,Nmid24],[Nmid10,Nmid25],[Nmid25,Nmid26],

[Nmid26,Nmid27],[Nmid27,Nmid28],[Nmid15,Nmid29],[Nmid29,Nmi
d30],[Nmid30,Nmid31],[Nmid31,Nmid32],[Nmid20,Nmid33],

[Nmid33,Nmid34],[Nmid34,Nmid35],[Nmid35,Nmid36],[Nmid3,Nmid
8],[Nmid8,Nmid13],[Nmid13,Nmid18],[Nmid18,Nmid3],[Nmid4,Nmi
d37],

[Nmid37,Nmid20],[Nmid19,Nmid37],[Nmid37,Nmid5],[Nmid9,Nmid3
8],[Nmid38,Nmid15],[Nmid14,Nmid38],[Nmid38,Nmid10],[Nmid4,N
mid39],

[Nmid39,Nmid10],[Nmid9,Nmid39],[Nmid39,Nmid5],[Nmid19,Nmid4
0],[Nmid40,Nmid15],[Nmid14,Nmid40],[Nmid40,Nmid20],[Nmid5,N
mid41],
```

```
[Nmid41,Nmid33],[Nmid20,Nmid41],[Nmid41,Nmid21],[Nmid10,Nmi
d42],[Nmid42,Nmid29],[Nmid15,Nmid42],[Nmid42,Nmid25],

[Nmid5,Nmid43],[Nmid43,Nmid25],[Nmid10,Nmid43],[Nmid43,Nmid
21],[Nmid20,Nmid44],[Nmid44,Nmid29],[Nmid15,Nmid44],[Nmid44
,Nmid33],

[Nmid21,Nmid45],[Nmid45,Nmid34],[Nmid33,Nmid45],[Nmid45,Nmi
d22],[Nmid25,Nmid46],[Nmid46,Nmid30],[Nmid29,Nmid46],[Nmid4
6,Nmid26],

[Nmid21,Nmid47],[Nmid47,Nmid26],[Nmid25,Nmid47],[Nmid47,Nmi
d22],[Nmid33,Nmid48],[Nmid48,Nmid30],[Nmid29,Nmid48],[Nmid4
8,Nmid34],

[Nmid22,Nmid49],[Nmid49,Nmid35],[Nmid34,Nmid49],[Nmid49,Nmi
d23],[Nmid26,Nmid50],[Nmid50,Nmid31],[Nmid30,Nmid50],[Nmid5
0,Nmid27],

[Nmid22,Nmid51],[Nmid51,Nmid27],[Nmid26,Nmid51],[Nmid51,Nmi
d23],[Nmid34,Nmid52],[Nmid52,Nmid31],[Nmid30,Nmid52],[Nmid5
2,Nmid35],

[Nmid24,Nmid53],[Nmid28,Nmid54],[Nmid32,Nmid56],[Nmid36,Nmi
d55]]


    ############## Datum points

    model.Part(dimensionality=THREE_D, name='Part-1',
type=DEFORMABLE_BODY)

    trusspart=model.parts['Part-1']

    trusspart.DatumPointByCoordinate(coords=(B1, B1, H1))
    trusspart.DatumPointByCoordinate(coords=(B1, B1, H2))
    trusspart.DatumPointByCoordinate(coords=(B2, B2, H3))
    trusspart.DatumPointByCoordinate(coords=(B3, B3, H4))
    trusspart.DatumPointByCoordinate(coords=(B5, B5, H6))
    trusspart.DatumPointByCoordinate(coords=(-B1, B1, H1))
    trusspart.DatumPointByCoordinate(coords=(-B1, B1, H2))
    trusspart.DatumPointByCoordinate(coords=(-B2, B2, H3))
    trusspart.DatumPointByCoordinate(coords=(-B3, B3, H4))
    trusspart.DatumPointByCoordinate(coords=(-B5, B5, H6))
    trusspart.DatumPointByCoordinate(coords=(-B1, -B1, H1))
    trusspart.DatumPointByCoordinate(coords=(-B1, -B1, H2))
    trusspart.DatumPointByCoordinate(coords=(-B2, -B2, H3))
    trusspart.DatumPointByCoordinate(coords=(-B3, -B3, H4))
```

```
    trusspart.DatumPointByCoordinate(coords=(-B5, -B5, H6))
    trusspart.DatumPointByCoordinate(coords=(B1, -B1, H1))
    trusspart.DatumPointByCoordinate(coords=(B1, -B1, H2))
    trusspart.DatumPointByCoordinate(coords=(B2, -B2, H3))
    trusspart.DatumPointByCoordinate(coords=(B3, -B3, H4))
    trusspart.DatumPointByCoordinate(coords=(B5, -B5, H6))
    trusspart.DatumPointByCoordinate(coords=(B7, B7, H8))
    trusspart.DatumPointByCoordinate(coords=(B9, B9, H10))
    trusspart.DatumPointByCoordinate(coords=(B11, B11,
H12))
    trusspart.DatumPointByCoordinate(coords=(B12, B12,
H13))
    trusspart.DatumPointByCoordinate(coords=(-B7, B7, H8))
    trusspart.DatumPointByCoordinate(coords=(-B9, B9, H10))
    trusspart.DatumPointByCoordinate(coords=(-B11, B11,
H12))
    trusspart.DatumPointByCoordinate(coords=(-B12, B12,
H13))
    trusspart.DatumPointByCoordinate(coords=(-B7, -B7, H8))
    trusspart.DatumPointByCoordinate(coords=(-B9, -B9,
H10))
    trusspart.DatumPointByCoordinate(coords=(-B11, -B11,
H12))
    trusspart.DatumPointByCoordinate(coords=(-B12, -B12,
H13))
    trusspart.DatumPointByCoordinate(coords=(B7, -B7, H8))
    trusspart.DatumPointByCoordinate(coords=(B9, -B9, H10))
    trusspart.DatumPointByCoordinate(coords=(B11, -B11,
H12))
    trusspart.DatumPointByCoordinate(coords=(B12, -B12,
H13))
    trusspart.DatumPointByCoordinate(coords=(B4, 0, H5))
    trusspart.DatumPointByCoordinate(coords=(-B4, 0, H5))
    trusspart.DatumPointByCoordinate(coords=(0, B4, H5))
    trusspart.DatumPointByCoordinate(coords=(0, -B4, H5))
    trusspart.DatumPointByCoordinate(coords=(B6, 0, H7))
    trusspart.DatumPointByCoordinate(coords=(-B6, 0, H7))
    trusspart.DatumPointByCoordinate(coords=(0, B6, H7))
    trusspart.DatumPointByCoordinate(coords=(0, -B6, H7))
    trusspart.DatumPointByCoordinate(coords=(B8, 0, H9))
    trusspart.DatumPointByCoordinate(coords=(-B8, 0, H9))
    trusspart.DatumPointByCoordinate(coords=(0, B8, H9))
    trusspart.DatumPointByCoordinate(coords=(0, -B8, H9))
    trusspart.DatumPointByCoordinate(coords=(B10, 0, H11))
    trusspart.DatumPointByCoordinate(coords=(-B10, 0, H11))
    trusspart.DatumPointByCoordinate(coords=(0, B10, H11))
    trusspart.DatumPointByCoordinate(coords=(0, -B10, H11))
    trusspart.DatumPointByCoordinate(coords=(B12, B12,
H14))
```

```
    trusspart.DatumPointByCoordinate(coords=(-B12, B12,
H14))
    trusspart.DatumPointByCoordinate(coords=(B12, -B12,
H14))
    trusspart.DatumPointByCoordinate(coords=(-B12, -B12,
H14))


concrete_center=trusspart.DatumPointByCoordinate(coords=(CC
))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh1))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh2))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh3))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh4))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh5))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh6))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh7))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh8))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh9))


rotor_center=trusspart.DatumPointByCoordinate(coords=(hub_c
g))

nacelle_center=trusspart.DatumPointByCoordinate(coords=(nac
elle_cg))

rotor_center=trusspart.DatumPointByCoordinate(coords=(hub_d
ummy_wire))

nacelle_center=trusspart.DatumPointByCoordinate(coords=(nac
elle_dummy_wire))


    CC_id=concrete_center.id


    ############## Nodenames
```

```
    trusspart.features.changeKey(fromName='Datum pt-1',toName='N1')
    trusspart.features.changeKey(fromName='Datum pt-2',toName='N2')
    trusspart.features.changeKey(fromName='Datum pt-3',toName='N3')
    trusspart.features.changeKey(fromName='Datum pt-4',toName='N4')
    trusspart.features.changeKey(fromName='Datum pt-5',toName='N5')
    trusspart.features.changeKey(fromName='Datum pt-6',toName='N6')
    trusspart.features.changeKey(fromName='Datum pt-7',toName='N7')
    trusspart.features.changeKey(fromName='Datum pt-8',toName='N8')
    trusspart.features.changeKey(fromName='Datum pt-9',toName='N9')
    trusspart.features.changeKey(fromName='Datum pt-10',toName='N10')
    trusspart.features.changeKey(fromName='Datum pt-11',toName='N11')
    trusspart.features.changeKey(fromName='Datum pt-12',toName='N12')
    trusspart.features.changeKey(fromName='Datum pt-13',toName='N13')
    trusspart.features.changeKey(fromName='Datum pt-14',toName='N14')
    trusspart.features.changeKey(fromName='Datum pt-15',toName='N15')
    trusspart.features.changeKey(fromName='Datum pt-16',toName='N16')
    trusspart.features.changeKey(fromName='Datum pt-17',toName='N17')
    trusspart.features.changeKey(fromName='Datum pt-18',toName='N18')
    trusspart.features.changeKey(fromName='Datum pt-19',toName='N19')
    trusspart.features.changeKey(fromName='Datum pt-20',toName='N20')
    trusspart.features.changeKey(fromName='Datum pt-21',toName='N21')
    trusspart.features.changeKey(fromName='Datum pt-22',toName='N22')
    trusspart.features.changeKey(fromName='Datum pt-23',toName='N23')
    trusspart.features.changeKey(fromName='Datum pt-24',toName='N24')
```

```
    trusspart.features.changeKey(fromName='Datum pt-
25',toName='N25')
    trusspart.features.changeKey(fromName='Datum pt-
26',toName='N26')
    trusspart.features.changeKey(fromName='Datum pt-
27',toName='N27')
    trusspart.features.changeKey(fromName='Datum pt-
28',toName='N28')
    trusspart.features.changeKey(fromName='Datum pt-
29',toName='N29')
    trusspart.features.changeKey(fromName='Datum pt-
30',toName='N30')
    trusspart.features.changeKey(fromName='Datum pt-
31',toName='N31')
    trusspart.features.changeKey(fromName='Datum pt-
32',toName='N32')
    trusspart.features.changeKey(fromName='Datum pt-
33',toName='N33')
    trusspart.features.changeKey(fromName='Datum pt-
34',toName='N34')
    trusspart.features.changeKey(fromName='Datum pt-
35',toName='N35')
    trusspart.features.changeKey(fromName='Datum pt-
36',toName='N36')
    trusspart.features.changeKey(fromName='Datum pt-
37',toName='N37')
    trusspart.features.changeKey(fromName='Datum pt-
38',toName='N38')
    trusspart.features.changeKey(fromName='Datum pt-
39',toName='N39')
    trusspart.features.changeKey(fromName='Datum pt-
40',toName='N40')
    trusspart.features.changeKey(fromName='Datum pt-
41',toName='N41')
    trusspart.features.changeKey(fromName='Datum pt-
42',toName='N42')
    trusspart.features.changeKey(fromName='Datum pt-
43',toName='N43')
    trusspart.features.changeKey(fromName='Datum pt-
44',toName='N44')
    trusspart.features.changeKey(fromName='Datum pt-
45',toName='N45')
    trusspart.features.changeKey(fromName='Datum pt-
46',toName='N47')
    trusspart.features.changeKey(fromName='Datum pt-
47',toName='N48')
    trusspart.features.changeKey(fromName='Datum pt-
48',toName='N46')
```

```
    trusspart.features.changeKey(fromName='Datum pt-
49',toName='N49')
    trusspart.features.changeKey(fromName='Datum pt-
50',toName='N50')
    trusspart.features.changeKey(fromName='Datum pt-
51',toName='N51')
    trusspart.features.changeKey(fromName='Datum pt-
52',toName='N52')
    trusspart.features.changeKey(fromName='Datum pt-
53',toName='N53')
    trusspart.features.changeKey(fromName='Datum pt-
54',toName='N54')
    trusspart.features.changeKey(fromName='Datum pt-
55',toName='N55')
    trusspart.features.changeKey(fromName='Datum pt-
56',toName='N56')




############## Wires

    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[1],trusspart.datums[2
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[2],trusspart.datums[3
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[3],trusspart.datums[4
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[4],trusspart.datums[5
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[6],trusspart.datums[7
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[7],trusspart.datums[8
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[8],trusspart.datums[9
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[9],trusspart.datums[1
0]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[11],trusspart.datums[
12]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[12],trusspart.datums[
13]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[13],trusspart.datums[
14]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[14],trusspart.datums[
15]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[16],trusspart.datums[
17]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[17],trusspart.datums[
18]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[18],trusspart.datums[
19]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[19],trusspart.datums[
20]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[5],trusspart.datums[2
1]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[21],trusspart.datums[
22]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[22],trusspart.datums[
23]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[23],trusspart.datums[
24]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[10],trusspart.datums[
25]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[25],trusspart.datums[
26]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[26],trusspart.datums[
27]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[27],trusspart.datums[
28]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[15],trusspart.datums[
29]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[29],trusspart.datums[
30]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[30],trusspart.datums[
31]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[31],trusspart.datums[
32]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[20],trusspart.datums[
33]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[33],trusspart.datums[
34]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[34],trusspart.datums[
35]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[35],trusspart.datums[
36]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[3],trusspart.datums[8
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[8],trusspart.datums[1
3]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[13],trusspart.datums[
18]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[18],trusspart.datums[
3]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[4],trusspart.datums[3
7]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[37],trusspart.datums[
20]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[19],trusspart.datums[
37]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[37],trusspart.datums[
5]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[9],trusspart.datums[3
8]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[38],trusspart.datums[
15]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[14],trusspart.datums[
38]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[38],trusspart.datums[
10]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[4],trusspart.datums[3
9]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[39],trusspart.datums[
10]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[9],trusspart.datums[3
9]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[39],trusspart.datums[
5]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[19],trusspart.datums[
40]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[40],trusspart.datums[
15]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[14],trusspart.datums[
40]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[40],trusspart.datums[
20]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[5],trusspart.datums[4
1]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[41],trusspart.datums[
33]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[20],trusspart.datums[
41]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[41],trusspart.datums[
21]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[10],trusspart.datums[
42]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[42],trusspart.datums[
29]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[15],trusspart.datums[
42]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[42],trusspart.datums[
25]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[5],trusspart.datums[4
3]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[43],trusspart.datums[
25]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[10],trusspart.datums[
43]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[43],trusspart.datums[
21]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[20],trusspart.datums[
44]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[44],trusspart.datums[
29]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[15],trusspart.datums[
44]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[44],trusspart.datums[
33]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[21],trusspart.datums[
45]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[45],trusspart.datums[
34]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[33],trusspart.datums[
45]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[45],trusspart.datums[
22]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[25],trusspart.datums[
46]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[46],trusspart.datums[
30]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[29],trusspart.datums[
46]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[46],trusspart.datums[
26]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[21],trusspart.datums[
47]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[47],trusspart.datums[
26]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[25],trusspart.datums[
47]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[47],trusspart.datums[
22]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[33],trusspart.datums[
48]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[48],trusspart.datums[
30]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[29],trusspart.datums[
48]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[48],trusspart.datums[
34]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[22],trusspart.datums[
49]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[49],trusspart.datums[
35]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[34],trusspart.datums[
49]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[49],trusspart.datums[
23]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[26],trusspart.datums[
50]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[50],trusspart.datums[
31]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[30],trusspart.datums[
50]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[50],trusspart.datums[
27]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[22],trusspart.datums[
51]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[51],trusspart.datums[
27]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[26],trusspart.datums[
51]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[51],trusspart.datums[
23]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[34],trusspart.datums[
52]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[52],trusspart.datums[
31]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[30],trusspart.datums[
52]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[52],trusspart.datums[
35]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[24],trusspart.datums[
53]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[28],trusspart.datums[
54]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[32],trusspart.datums[
56]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[36],trusspart.datums[
55]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh1), (0.0, 0.0,
towerh2)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh2), (0.0, 0.0,
towerh3)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh3), (0.0, 0.0,
towerh4)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh4), (0.0, 0.0,
towerh5)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh5), (0.0, 0.0,
towerh6)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh6), (0.0, 0.0,
towerh7)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh7), (0.0, 0.0,
towerh8)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh8), (0.0, 0.0,
towerh9)), ))

    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((-nacelle_overhang*c_overhang,-
nacelle_overhang*s_overhang,towerh9+1.75), (-
nacelle_overhang*c_overhang,-
nacelle_overhang*s_overhang,towerh9+1.75+.05)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((hub_overhang*c_overhang,hub_overhang*
s_overhang,towerh9+2.40),
(hub_overhang*c_overhang,hub_overhang*s_overhang,towerh9+2.
40+0.05)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0,0,18.15), (0,0,18.16)), ))


    ############## Wire set creation


f=[[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[
N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],
[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1]
,[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1
],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N
1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[
```

```
N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],
[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1]
,[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],]
    f[0][0]=tuple(map(sum,zip(Mmid[0][0],Mmid[0][1])))
    f[1][0]=tuple(map(sum,zip(Mmid[1][0],Mmid[1][1])))
    f[2][0]=tuple(map(sum,zip(Mmid[2][0],Mmid[2][1])))
    f[3][0]=tuple(map(sum,zip(Mmid[3][0],Mmid[3][1])))
    f[4][0]=tuple(map(sum,zip(Mmid[4][0],Mmid[4][1])))
    f[5][0]=tuple(map(sum,zip(Mmid[5][0],Mmid[5][1])))
    f[6][0]=tuple(map(sum,zip(Mmid[6][0],Mmid[6][1])))
    f[7][0]=tuple(map(sum,zip(Mmid[7][0],Mmid[7][1])))
    f[8][0]=tuple(map(sum,zip(Mmid[8][0],Mmid[8][1])))
    f[9][0]=tuple(map(sum,zip(Mmid[9][0],Mmid[9][1])))
    f[10][0]=tuple(map(sum,zip(Mmid[10][0],Mmid[10][1])))
    f[11][0]=tuple(map(sum,zip(Mmid[11][0],Mmid[11][1])))
    f[12][0]=tuple(map(sum,zip(Mmid[12][0],Mmid[12][1])))
    f[13][0]=tuple(map(sum,zip(Mmid[13][0],Mmid[13][1])))
    f[14][0]=tuple(map(sum,zip(Mmid[14][0],Mmid[14][1])))
    f[15][0]=tuple(map(sum,zip(Mmid[15][0],Mmid[15][1])))
    f[16][0]=tuple(map(sum,zip(Mmid[16][0],Mmid[16][1])))
    f[17][0]=tuple(map(sum,zip(Mmid[17][0],Mmid[17][1])))
    f[18][0]=tuple(map(sum,zip(Mmid[18][0],Mmid[18][1])))
    f[19][0]=tuple(map(sum,zip(Mmid[19][0],Mmid[19][1])))
    f[20][0]=tuple(map(sum,zip(Mmid[20][0],Mmid[20][1])))
    f[21][0]=tuple(map(sum,zip(Mmid[21][0],Mmid[21][1])))
    f[22][0]=tuple(map(sum,zip(Mmid[22][0],Mmid[22][1])))
    f[23][0]=tuple(map(sum,zip(Mmid[23][0],Mmid[23][1])))
    f[24][0]=tuple(map(sum,zip(Mmid[24][0],Mmid[24][1])))
    f[25][0]=tuple(map(sum,zip(Mmid[25][0],Mmid[25][1])))
    f[26][0]=tuple(map(sum,zip(Mmid[26][0],Mmid[26][1])))
    f[27][0]=tuple(map(sum,zip(Mmid[27][0],Mmid[27][1])))
    f[28][0]=tuple(map(sum,zip(Mmid[28][0],Mmid[28][1])))
    f[29][0]=tuple(map(sum,zip(Mmid[29][0],Mmid[29][1])))
    f[30][0]=tuple(map(sum,zip(Mmid[30][0],Mmid[30][1])))
    f[31][0]=tuple(map(sum,zip(Mmid[31][0],Mmid[31][1])))
    f[32][0]=tuple(map(sum,zip(Mmid[32][0],Mmid[32][1])))
    f[33][0]=tuple(map(sum,zip(Mmid[33][0],Mmid[33][1])))
    f[34][0]=tuple(map(sum,zip(Mmid[34][0],Mmid[34][1])))
    f[35][0]=tuple(map(sum,zip(Mmid[35][0],Mmid[35][1])))
    f[36][0]=tuple(map(sum,zip(Mmid[36][0],Mmid[36][1])))
    f[37][0]=tuple(map(sum,zip(Mmid[37][0],Mmid[37][1])))
    f[38][0]=tuple(map(sum,zip(Mmid[38][0],Mmid[38][1])))
    f[39][0]=tuple(map(sum,zip(Mmid[39][0],Mmid[39][1])))
    f[40][0]=tuple(map(sum,zip(Mmid[40][0],Mmid[40][1])))
    f[41][0]=tuple(map(sum,zip(Mmid[41][0],Mmid[41][1])))
    f[42][0]=tuple(map(sum,zip(Mmid[42][0],Mmid[42][1])))
    f[43][0]=tuple(map(sum,zip(Mmid[43][0],Mmid[43][1])))
    f[44][0]=tuple(map(sum,zip(Mmid[44][0],Mmid[44][1])))
    f[45][0]=tuple(map(sum,zip(Mmid[45][0],Mmid[45][1])))
```

```
f[46][0]=tuple(map(sum,zip(Mmid[46][0],Mmid[46][1])))
f[47][0]=tuple(map(sum,zip(Mmid[47][0],Mmid[47][1])))
f[48][0]=tuple(map(sum,zip(Mmid[48][0],Mmid[48][1])))
f[49][0]=tuple(map(sum,zip(Mmid[49][0],Mmid[49][1])))
f[50][0]=tuple(map(sum,zip(Mmid[50][0],Mmid[50][1])))
f[51][0]=tuple(map(sum,zip(Mmid[51][0],Mmid[51][1])))
f[52][0]=tuple(map(sum,zip(Mmid[52][0],Mmid[52][1])))
f[53][0]=tuple(map(sum,zip(Mmid[53][0],Mmid[53][1])))
f[54][0]=tuple(map(sum,zip(Mmid[54][0],Mmid[54][1])))
f[55][0]=tuple(map(sum,zip(Mmid[55][0],Mmid[55][1])))
f[56][0]=tuple(map(sum,zip(Mmid[56][0],Mmid[56][1])))
f[57][0]=tuple(map(sum,zip(Mmid[57][0],Mmid[57][1])))
f[58][0]=tuple(map(sum,zip(Mmid[58][0],Mmid[58][1])))
f[59][0]=tuple(map(sum,zip(Mmid[59][0],Mmid[59][1])))
f[60][0]=tuple(map(sum,zip(Mmid[60][0],Mmid[60][1])))
f[61][0]=tuple(map(sum,zip(Mmid[61][0],Mmid[61][1])))
f[62][0]=tuple(map(sum,zip(Mmid[62][0],Mmid[62][1])))
f[63][0]=tuple(map(sum,zip(Mmid[63][0],Mmid[63][1])))
f[64][0]=tuple(map(sum,zip(Mmid[64][0],Mmid[64][1])))
f[65][0]=tuple(map(sum,zip(Mmid[65][0],Mmid[65][1])))
f[66][0]=tuple(map(sum,zip(Mmid[66][0],Mmid[66][1])))
f[67][0]=tuple(map(sum,zip(Mmid[67][0],Mmid[67][1])))
f[68][0]=tuple(map(sum,zip(Mmid[68][0],Mmid[68][1])))
f[69][0]=tuple(map(sum,zip(Mmid[69][0],Mmid[69][1])))
f[70][0]=tuple(map(sum,zip(Mmid[70][0],Mmid[70][1])))
f[71][0]=tuple(map(sum,zip(Mmid[71][0],Mmid[71][1])))
f[72][0]=tuple(map(sum,zip(Mmid[72][0],Mmid[72][1])))
f[73][0]=tuple(map(sum,zip(Mmid[73][0],Mmid[73][1])))
f[74][0]=tuple(map(sum,zip(Mmid[74][0],Mmid[74][1])))
f[75][0]=tuple(map(sum,zip(Mmid[75][0],Mmid[75][1])))
f[76][0]=tuple(map(sum,zip(Mmid[76][0],Mmid[76][1])))
f[77][0]=tuple(map(sum,zip(Mmid[77][0],Mmid[77][1])))
f[78][0]=tuple(map(sum,zip(Mmid[78][0],Mmid[78][1])))
f[79][0]=tuple(map(sum,zip(Mmid[79][0],Mmid[79][1])))
f[80][0]=tuple(map(sum,zip(Mmid[80][0],Mmid[80][1])))
f[81][0]=tuple(map(sum,zip(Mmid[81][0],Mmid[81][1])))
f[82][0]=tuple(map(sum,zip(Mmid[82][0],Mmid[82][1])))
f[83][0]=tuple(map(sum,zip(Mmid[83][0],Mmid[83][1])))
f[84][0]=tuple(map(sum,zip(Mmid[84][0],Mmid[84][1])))
f[85][0]=tuple(map(sum,zip(Mmid[85][0],Mmid[85][1])))
f[86][0]=tuple(map(sum,zip(Mmid[86][0],Mmid[86][1])))
f[87][0]=tuple(map(sum,zip(Mmid[87][0],Mmid[87][1])))
f[88][0]=tuple(map(sum,zip(Mmid[88][0],Mmid[88][1])))
f[89][0]=tuple(map(sum,zip(Mmid[89][0],Mmid[89][1])))
f[90][0]=tuple(map(sum,zip(Mmid[90][0],Mmid[90][1])))
f[91][0]=tuple(map(sum,zip(Mmid[91][0],Mmid[91][1])))
f[92][0]=tuple(map(sum,zip(Mmid[92][0],Mmid[92][1])))
f[93][0]=tuple(map(sum,zip(Mmid[93][0],Mmid[93][1])))
f[94][0]=tuple(map(sum,zip(Mmid[94][0],Mmid[94][1])))
```

```
    f[95][0]=tuple(map(sum,zip(Mmid[95][0],Mmid[95][1])))
    f[96][0]=tuple(map(sum,zip(Mmid[96][0],Mmid[96][1])))
    f[97][0]=tuple(map(sum,zip(Mmid[97][0],Mmid[97][1])))
    f[98][0]=tuple(map(sum,zip(Mmid[98][0],Mmid[98][1])))
    f[99][0]=tuple(map(sum,zip(Mmid[99][0],Mmid[99][1])))

f[100][0]=tuple(map(sum,zip(Mmid[100][0],Mmid[100][1])))

f[101][0]=tuple(map(sum,zip(Mmid[101][0],Mmid[101][1])))

f[102][0]=tuple(map(sum,zip(Mmid[102][0],Mmid[102][1])))

f[103][0]=tuple(map(sum,zip(Mmid[103][0],Mmid[103][1])))



    ############## Sets

P=[2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,3,3,
3,3,3,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,4,4,4,4]
    length=len(P)
    len2=length+1


trusspart.Set(edges=trusspart.edges.getSequenceFromMask(('[
#ffffffff:3 #ff ]', ), ), name='All_set')
    for i in range(1,len2):
        trusspart.Set(edges=trusspart.edges.findAt(((f[i-
1][0]),),), name='Wire-'+str(i)+'-Set-1')
        wire_region=trusspart.edges.findAt(((f[i-1][0]),),)
        globals()['wire%s' % i] = wire_region
    trusspart.Set(edges=trusspart.edges.findAt([CC]),
name='Concrete_help_wire')



    ############## Sections
    waterdensity=1000

    radius1=0.4
    radius2=0.6
    radius3=0.6
    radius4=0.6

    thickness1=0.02
    thickness2=0.05
    thickness3=0.035
```

```
    thickness4=0.04

    model.PipeProfile(name='Dim1', r=radius1, t=thickness1)
    model.PipeProfile(name='Dim2', r=radius2, t=thickness2)
    model.PipeProfile(name='Dim3', r=radius3, t=thickness3)
    model.PipeProfile(name='Dim4', r=radius4, t=thickness4)


    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Braces', poissonRatio=0.3, profile='Dim1',
temperatureVar=LINEAR)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius1, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Braces_submerged', poissonRatio=0.3, profile='Dim1',
temperatureVar=LINEAR, useFluidInertia=ON)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius1, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Braces_half_submerged', poissonRatio=0.3,
profile='Dim1', submerged=HALF, temperatureVar=LINEAR,
useFluidInertia=ON)

    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Lower_leg', poissonRatio=0.3, profile='Dim2',
temperatureVar=LINEAR)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius2, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Lower_leg_submerged', poissonRatio=0.3,
profile='Dim2', temperatureVar=LINEAR, useFluidInertia=ON)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius2, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Lower_leg_half_submerged', poissonRatio=0.3,
profile='Dim2', submerged=HALF, temperatureVar=LINEAR,
useFluidInertia=ON)

    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Upper_leg', poissonRatio=0.3, profile='Dim3',
temperatureVar=LINEAR)
```

```
model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius3, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Upper_leg_submerged', poissonRatio=0.3,
profile='Dim3', temperatureVar=LINEAR, useFluidInertia=ON)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius3, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Upper_leg_half_submerged', poissonRatio=0.3,
profile='Dim3', submerged=HALF, temperatureVar=LINEAR,
useFluidInertia=ON)

    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel', name='Top',
poissonRatio=0.3, profile='Dim4', temperatureVar=LINEAR)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius4, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Top_submerged', poissonRatio=0.3, profile='Dim4',
temperatureVar=LINEAR, useFluidInertia=ON)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius4, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Top_half__submerged', poissonRatio=0.3,
profile='Dim4', submerged=HALF, temperatureVar=LINEAR,
useFluidInertia=ON)


    ################ Tower sections

    ##model.PipeProfile(name='Tower1', r=5.600/2, t=0.032)
    model.PipeProfile(name='Tower2', r=5.577/2, t=0.032)
    model.PipeProfile(name='Tower3', r=5.318/2, t=0.030)
    model.PipeProfile(name='Tower4', r=5.082/2, t=0.028)
    model.PipeProfile(name='Tower5', r=4.800/2, t=0.024)
    model.PipeProfile(name='Tower6', r=4.565/2, t=0.024)
    model.PipeProfile(name='Tower7', r=4.329/2, t=0.022)
    model.PipeProfile(name='Tower8', r=4.118/2, t=0.030)
    model.PipeProfile(name='Tower9', r=4.000/2, t=0.030)


    ##model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
```

```
name='Tower_section_1', poissonRatio=0.3, profile='Tower1',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_2', poissonRatio=0.3, profile='Tower2',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_3', poissonRatio=0.3, profile='Tower3',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_4', poissonRatio=0.3, profile='Tower4',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_5', poissonRatio=0.3, profile='Tower5',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_6', poissonRatio=0.3, profile='Tower6',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_7', poissonRatio=0.3, profile='Tower7',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_8', poissonRatio=0.3, profile='Tower8',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_9', poissonRatio=0.3, profile='Tower9',
temperatureVar=LINEAR)


    towermid0=(0,0,(18.15+towerh1)/2)
    towermid1=(0,0,(towerh1+towerh2)/2)
    towermid2=(0,0,(towerh2+towerh3)/2)
    towermid3=(0,0,(towerh3+towerh4)/2)
    towermid4=(0,0,(towerh4+towerh5)/2)
    towermid5=(0,0,(towerh5+towerh6)/2)
    towermid6=(0,0,(towerh6+towerh7)/2)
    towermid7=(0,0,(towerh7+towerh8)/2)
    towermid8=(0,0,(towerh8+towerh9)/2)


    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
```

```
region=Region(edges=trusspart.edges.findAt([towermid1])),
sectionName='Tower_section_2',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid2])),
sectionName='Tower_section_3',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid3])),
sectionName='Tower_section_4',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid4])),
sectionName='Tower_section_5',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid5])),
sectionName='Tower_section_6',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid6])),
sectionName='Tower_section_7',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid7])),
sectionName='Tower_section_8',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid8])),
sectionName='Tower_section_9',thicknessAssignment=FROM_SECT
ION)



trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid1])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid2])))
```

```
trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid3])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid4])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid5])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid6])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid7])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid8])))


    ################ Dummy sections

    Concrete_center=[0,0,18.155]
    model.PipeProfile(name='Dummy', r=0.1, t=0.010)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Dummy_section', poissonRatio=0.3, profile='Dummy',
temperatureVar=LINEAR)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([nacelle_cg])),
sectionName='Dummy_section',thicknessAssignment=FROM_SECTIO
N)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([hub_cg])),
sectionName='Dummy_section',thicknessAssignment=FROM_SECTIO
N)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([Concrete_center
])),
```

```
sectionName='Dummy_section',thicknessAssignment=FROM_SECTIO
N)

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([nacelle_cg])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([hub_cg])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([Concrete_center
])))

    ################ Steps

    model.ImplicitDynamicsStep(initialInc=0.025,
maxNumInc=24001, name='Dynamic', noStop=OFF, nohaf=OFF,
previous='Initial', timeIncrementationMethod=FIXED,
timePeriod=600.0)


    ############### Properties

    S=[0]*len(P)
    member_sets=[0]*len(P)

    for i in xrange(len(P)):
        if P[i-1]==1:

            if M[i-1][0][2]>=0 and M[i-1][1][2]>=0:
                S[i-1]='Braces'
            elif M[i-1][0][2]<0 and M[i-1][1][2]<0:
                S[i-1]='Braces_submerged'
            else:
                S[i-1]='Braces_half_submerged'

        elif P[i-1]==2:
            if M[i-1][0][2]>=0 and M[i-1][1][2]>=0:
                S[i-1]='Lower_leg'
            elif M[i-1][0][2]<0 and M[i-1][1][2]<0:
                S[i-1]='Lower_leg_submerged'
            else:
                S[i-1]='Lower_leg_half_submerged'

        elif P[i-1]==3:
            if M[i-1][0][2]>=0 and M[i-1][1][2]>=0:
```

```
                S[i-1]='Upper_leg'
            elif M[i-1][0][2]<0 and M[i-1][1][2]<0:
                S[i-1]='Upper_leg_submerged'
            else:
                S[i-1]='Upper_leg_half_submerged'

        else:
            if M[i-1][0][2]>=0 and M[i-1][1][2]>=0:
                S[i-1]='Top'
            elif M[i-1][0][2]<0 and M[i-1][1][2]<0:
                S[i-1]='Top_submerged'
            else:
                S[i-1]='Top_half_submerged'

        member_sets[i-1]='Wire-'+str(i)+'-Set-1'
        member_sets[len(P)-1]='Wire-'+str(len(P))+'-Set-1'



    ############## Section assignment

    for i in xrange(len(P)):

trusspart.SectionAssignment(offset=0.0,offsetField='',
offsetType=MIDDLE_SURFACE,
region=trusspart.sets[member_sets[i]], sectionName=S[i],
thicknessAssignment=FROM_SECTION)


    ############## Mesh


trusspart.setElementType(elemTypes=(ElemType(elemCode=B31,
elemLibrary=STANDARD), ),
regions=(trusspart.edges.getSequenceFromMask(('[#ffffffff:3
#ff ]', ), ), ))

    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([CC]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([nacelle_cg]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([hub_cg]), number=5)
    for i in xrange(len(P)+1):
            trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt(f[i-1]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid1]), number=5)
```

```
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid2]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid3]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid4]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid5]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid6]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid7]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid8]), number=5)

    trusspart.generateMesh()


    ############## Assembly

    model.rootAssembly.DatumCsysByDefault(CARTESIAN)
    model.rootAssembly.Instance(dependent=ON, name='Part-1-
1',  part=trusspart)

    model.rootAssembly.regenerate()

    model.rootAssembly.Set(name='Rotor_nacelle',
vertices=model.rootAssembly.instances['Part-1-
1'].vertices.findAt([nacelle_cg],[hub_cg]))

model.rootAssembly.Set(name='Concrete_jacket_connection',
vertices=model.rootAssembly.instances['Part-1-
1'].vertices.findAt([N24],[N28],[N32],[N36],[CC]))


model.Coupling(controlPoint=Region(vertices=model.rootAssem
bly.instances['Part-1-1'].vertices.findAt([point_mass_1])),
couplingType=KINEMATIC, influenceRadius=WHOLE_SURFACE,
localCsys=None,
        name='Constraint-2',
surface=model.rootAssembly.sets['Concrete_jacket_connection
'], u1=ON, u2=ON, u3=ON, ur1=ON, ur2=ON, ur3=ON)


model.Coupling(controlPoint=Region(vertices=model.rootAssem
bly.instances['Part-1-1'].vertices.findAt([point_mass_3])),
couplingType=KINEMATIC, influenceRadius=WHOLE_SURFACE,
localCsys=None,
```

```
        name='Constraint-3',
surface=model.rootAssembly.sets['Rotor_nacelle'], u1=ON,
u2=ON, u3=ON, ur1=ON, ur2=ON, ur3=ON)


    ############## Point masses


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=concrete_Iw,
i22=concrete_Iw, i33=concrete_Ih, mass=concrete_mass, name=
        'Inertia-1',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([CC])))


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=0, i22=0, i33=0, mass=1900,
name=
        'Pointmass-1',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([point_mass_1])))


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=0, i22=0, i33=0, mass=1400,
name=
        'Pointmass-2',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([point_mass_2])))


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=0, i22=0, i33=0, mass=1000,
name=
        'Pointmass-3',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([point_mass_3])))


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=0, i22=0, i33=0,
mass=240000, name=
```

```
        'Pointmass-4',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([nacelle_cg])))


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=0, i22=0, i33=0,
mass=110000, name=
        'Pointmass-5',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([hub_cg])))


    ############## Load
    model.ConcentratedForce(amplitude='lc1-fz', cf1=1.0,
        createStepName='Dynamic', distributionType=UNIFORM,
field='', localCsys=None
        , name='lc1-fx',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([load_point])))
    model.ConcentratedForce(amplitude='lc1-fy', cf2=-1.0,
        createStepName='Dynamic', distributionType=UNIFORM,
field='', localCsys=None
        , name='lc1-fy',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([load_point])))


    model.Moment(amplitude='LC1-mx', cm1=1.0,
createStepName=
        'Dynamic', distributionType=UNIFORM, field='',
localCsys=None, name=
        'LC1-mx',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([load_point])))
    model.Moment(amplitude='LC1-my', cm2=1.0,
createStepName=
        'Dynamic', distributionType=UNIFORM, field='',
localCsys=None, name=
        'LC1-my',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([load_point])))
    model.Moment(amplitude='LC1-mz', cm3=1.0,
createStepName=
        'Dynamic', distributionType=UNIFORM, field='',
localCsys=None, name=
```

```
        'LC1-mz',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([load_point])))
    model.Gravity(comp3=-9.81, createStepName='Dynamic',
        distributionType=UNIFORM, field='', name='Gravity')

    ############## BC

    model.rootAssembly.Set(name='BC_set',
vertices=model.rootAssembly.instances['Part-1-
1'].vertices.findAt([N1],[N6],[N11],[N16]))
    model.DisplacementBC(amplitude=UNSET,
createStepName='Dynamic',distributionType=UNIFORM,
fieldName='', fixed=OFF, localCsys=None, name='BC-1',
        region=model.rootAssembly.sets['BC_set'], u1=0.0,
u2=0.0, u3=0.0, ur1=0.0, ur2=0.0, ur3=0.0)


    ############## Beam orientations


    beamvectors=[0]*len(P)
    beamnormals=[0]*len(P)
    for i in xrange(len(P)):
            beamvectors[i]=[0]*3
            beamnormals[i]=[0]*3
    for i in xrange(len(P)+1):
        for j in xrange(1,4):
            beamvectors[i-1][j-1]=M[i-1][1][j-1]-M[i-
1][0][j-1]

    for i in xrange(len(P)):
        if beamvectors[i][0]==0:
            beamnormals[i][0]=1
            beamnormals[i][1]=0
            beamnormals[i][2]=0
        elif beamvectors[i][1]==0:
            beamnormals[i][0]=0
            beamnormals[i][1]=1
            beamnormals[i][2]=0
        else:
            beamnormals[i][1]=1
            beamnormals[i][2]=1
            beamnormals[i][0]=-
(beamvectors[i][1]*beamnormals[i][1]+beamvectors[i][2]*beam
normals[i][2])/(beamvectors[i][0])


    for i in xrange(len(P)):
```

```
trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(beamnormals[i]), region=trusspart.sets['Wire-
'+str(i+1)+'-Set-1'])

    ############## Outputs
    model.fieldOutputRequests['F-Output-
1'].setValues(variables=('S', 'U'),frequency=1)
    mdb.models['Model-
1'].HistoryOutputRequest(createStepName='Dynamic',
frequency=
        LAST_INCREMENT, name='H-Output-1',
variables=('MASS', ))

    mdb.models['Model-
1'].keywordBlock.synchVersions(storeNodesAndElements=False)
    keywordblocks=model.keywordBlock.sieBlocks
    stepblock='*Step, name=Step-1, nlgeom=NO'
    if count==1:
        model.keywordBlock.insert(595, """*AQUA\n-50, 0,
9.81, 1021\n*WAVE, TYPE=STOKES\n1.395, 5.705, 0, 1, 0""")

    if q>0:
        wiredel="Wire-"+str(q)
        del mdb.models['Model-1'].parts['Part-
1'].features[wiredel]


    ############## Run job
    mdb.jobs[jobname].submit()
    mdb.jobs[jobname].waitForCompletion()
```

APPENDIX 4 – PYTHON SCRIPT FOR OUTPUT OF STRUCTURAL REDUNDANCY RESULTS (LC1)

```
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
from regionToolset import *
from odbAccess import *
from abaqusConstants import *


loadcase="Job1"

half_member_list=[half_member_list=[0,3,4,7,8,17,18,19,20,21,22,23,24,33,34,36,37,40,41,45,46,47,48,53,56,57,60,61,62,63,64,69,72,73,76,77,78,79,80,85,88,89,92,93,94,95,96]]


count=0


for q in half_member_list:



    H1=-45.50000
    H2=-45.00000
    H3=-44.00100
    H4=-43.12700
    H5=-33.37300
    H6=-24.61400
    H7=-16.37100
    H8=-8.92200
    H9=-1.95800
    H10=4.37800
    H11=10.26200
```

```
H12=15.65100
H13=16.15000
H14=20.15000
B1=6.00000
B2=5.96700
B3=5.93900
B4=5.62000
B5=5.33300
B6=5.06400
B7=4.82000
B8=4.59200
B9=4.38500
B10=4.19300
B11=4.01600
B12=4.00000

N1=(B1,B1,H1)
N2=(B1,B1,H2)
N3=(B2,B2,H3)
N4=(B3,B3,H4)
N5=(B5,B5,H6)
N6=(-B1,B1,H1)
N7=(-B1,B1,H2)
N8=(-B2,B2,H3)
N9=(-B3,B3,H4)
N10=(-B5,B5,H6)
N11=(-B1,-B1,H1)
N12=(-B1,-B1,H2)
N13=(-B2,-B2,H3)
N14=(-B3,-B3,H4)
N15=(-B5,-B5,H6)
N16=(B1,-B1,H1)
N17=(B1,-B1,H2)
N18=(B2,-B2,H3)
N19=(B3,-B3,H4)
N20=(B5,-B5,H6)
N21=(B7,B7,H8)
N22=(B9,B9,H10)
N23=(B11,B11,H12)
N24=(B12,B12,H13)
N25=(-B7,B7,H8)
N26=(-B9,B9,H10)
N27=(-B11,B11,H12)
N28=(-B12,B12,H13)
N29=(-B7,-B7,H8)
N30=(-B9,-B9,H10)
N31=(-B11,-B11,H12)
N32=(-B12,-B12,H13)
N33=(B7,-B7,H8)
```

```
    N34=(B9,-B9,H10)
    N35=(B11,-B11,H12)
    N36=(B12,-B12,H13)
    N37=(B4,0,H5)
    N38=(-B4,0,H5)
    N39=(0,B4,H5)
    N40=(0,-B4,H5)
    N41=(B6,0,H7)
    N42=(-B6,0,H7)
    N43=(0,B6,H7)
    N44=(0,-B6,H7)
    N45=(B8,0,H9)
    N46=(-B8,0,H9)
    N47=(0,B8,H9)
    N48=(0,-B8,H9)
    N49=(B10,0,H11)
    N50=(-B10,0,H11)
    N51=(0,B10,H11)
    N52=(0,-B10,H11)
    N53=(B12,B12,H14)
    N54=(-B12,B12,H14)
    N55=(B12,-B12,H14)
    N56=(-B12,-B12,H14)


    jobname=loadcase+"-"+str(q)
    jobpath=str(jobname)+".odb"
    odb=openOdb(path=jobpath)

    lastFrame = odb.steps['Dynamic'].frames[-1]
    numframes=len(odb.steps['Dynamic'].frames)



    name=loadcase+"-"+str(q)+".txt"


    output = open(name, 'w')



nodelist=(N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N1
5,N16,N17,N18,N19,N20,N21,N22,N23,N24,N25,N26,

N27,N28,N29,N30,N31,N32,N33,N34,N35,N36,N37,N38,N39,N40,N41
,N42,N43,N44,N45,N46,N47,N48,N49,N50,N51,N52,N53,N54,N55,N5
6)

    numframes=len(odb.steps['Dynamic'].frames)
```

```
    for i in xrange(numframes):
    for j in xrange(len(nodelist)):
        node=((nodelist[j]),)
        path=session.Path(name='NODES', type=POINT_LIST,
expression=node)
        stress=(('S', INTEGRATION_POINT, (INVARIANT,
'Mises' ), ),)
        session.paths['NODES']
        session.viewports['Viewport:
1'].setValues(displayedObject=odb)
        stressdata=session.XYDataFromPath(name='Von-
Mises',path=path, includeIntersections=False,
shape=UNDEFORMED, labelType=TRUE_DISTANCE, step=0, frame=i,
variable=stress)
        first=stressdata.data[0]
        value=first[1]
        output.write(str(value) + "\t")
        output.write("\n")
    output.close()

    # END ==============
```

APPENDIX 5 – PYTHON SCRIPT FOR NODE COORDINATE OPTIMISATION

```
# -*- coding: mbcs -*-
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
from regionToolset import *


import random



iterations=2000
time=1
numinc=10
increment=0.01
rotation=3.14159265359
load_amplitude=1000

jobname="Job_op"

mdb.Job(atTime=None, contactPrint=OFF, description='',
echoPrint=OFF,
    explicitPrecision=SINGLE, getMemoryFromAnalysis=True,
historyPrint=OFF,
    memory=90, memoryUnits=PERCENTAGE, model='Model-1',
modelPrint=OFF,
    multiprocessingMode=DEFAULT, name=jobname,
nodalOutputPrecision=SINGLE,
    numCpus=1, numGPUs=1, queue=None, scratch='',
type=ANALYSIS,
    userSubroutine='', waitHours=0, waitMinutes=0)

H1=-45.50000
```

```
H2=-45.00000
H13=16.15000
H14=20.15000
B12=4.00000


B1=8.00000
H3=-44.00100
H4=-43.12700
H5=-33.37300
H6=-24.61400
H7=-16.37100
H8=-8.92200
H9=-1.95800
H10=4.37800
H11=10.26200
H12=15.65100

B2=(4-B1)*H3/61.15+4-16.15*(4-B1)/61.15
B3=(4-B1)*H4/61.15+4-16.15*(4-B1)/61.15
B4=(4-B1)*H5/61.15+4-16.15*(4-B1)/61.15
B5=(4-B1)*H6/61.15+4-16.15*(4-B1)/61.15
B6=(4-B1)*H7/61.15+4-16.15*(4-B1)/61.15
B7=(4-B1)*H8/61.15+4-16.15*(4-B1)/61.15
B8=(4-B1)*H9/61.15+4-16.15*(4-B1)/61.15
B9=(4-B1)*H10/61.15+4-16.15*(4-B1)/61.15
B10=(4-B1)*H11/61.15+4-16.15*(4-B1)/61.15
B11=(4-B1)*H12/61.15+4-16.15*(4-B1)/61.15



CC=[0,0,18.15]
concrete_mass=666000
concrete_w=9.6
concrete_h=4.0
concrete_Ih=(1.0/12)*concrete_mass*(2*(concrete_w)**2)
concrete_Iw=(1.0/12)*concrete_mass*((concrete_h)**2+(concre
te_w)**2)
load_point=[0,0,20.15]
##


V0=[H2,0,0]
V1=[H3,0,0]
V2=[H4,0,0]
V3=[H5,0,0]
V4=[H6,0,0]
V5=[H7,0,0]
V6=[H8,0,0]
```

```
V7=[H9,0,0]
V8=[H10,0,0]
V9=[H11,0,0]
V10=[H12,0,0]
V11=[H13,0,0]
V12=[B1,0,0]
V13=[B2,0,0]
V14=[B3,0,0]
V15=[B4,0,0]
V16=[B5,0,0]
V17=[B6,0,0]
V18=[B7,0,0]
V19=[B8,0,0]
V20=[B9,0,0]
V21=[B10,0,0]
V22=[B11,0,0]
V23=[B12,0,0]


##
towerh1=20.15
towerh2=21.15
towerh3=32.15
towerh4=42.15
towerh5=54.15
towerh6=64.15
towerh7=74.15
towerh8=83.15
towerh9=88.15
towerconnect=[0,0,20.15]

point_mass_1=(0,0,20.15)
point_mass_2=(0,0,54.15)
point_mass_3=(0,0,88.15)


nacelle_overhang=1.90
hub_overhang=5.0/1

s_overhang=sin(rotation)
c_overhang=cos(rotation)

nacelle_cg=(-nacelle_overhang*c_overhang,-
nacelle_overhang*s_overhang,towerh9+1.75)
hub_cg=(hub_overhang*c_overhang,hub_overhang*s_overhang,tow
erh9+2.40)

nacelle_dummy_wire=(-nacelle_overhang*c_overhang,-
nacelle_overhang*s_overhang,towerh9+1.75+0.05)
```

```
hub_dummy_wire=(hub_overhang*c_overhang,hub_overhang*s_over
hang,towerh9+2.40+0.05)


variables=[0 for i in xrange(iterations)]

variables[0]=[6,-44.001,-43.12700,-33.37300,-24.61400,-
16.37100,-8.92200,-1.95800,4.37800,10.26200,15.65100]


num_var=len(variables[0])

max_stress=[0]*iterations
max_stress_element_label=[0]*iterations
max_stress_object_index=[0]*iterations


t=0
count=1


############## Job

num_var=len(variables[0])

store=[0 for i in xrange(iterations)]
store[0]=tuple(variables[0])

file_open=open("script_run.txt", "w")


############## Amplitudes

mdb.models['Model-1'].TabularAmplitude(data=((0.0, 1.0),
(0.025, 0.0), (600.0,
    0.0)), name='BC_amp', smooth=SOLVER_DEFAULT,
timeSpan=STEP)

for t in range(iterations):

    rand=int(ceil((num_var)*random.random()-1))
    sign=random.choice([-1,1])


    if t!=0:
       variables[t]=[0]*len(variables[t-1])
       for u in range(len(variables[t-1])):
           variables[t][u]=float(variables[t-1][u])
```

```
        store[t]=tuple(variables[t])
        variables[t][rand]=variables[t][rand]+sign*increment
        for z in range(t-1):
            if variables[t]==variables[z]:

variables[t][rand]=variables[t][rand]+sign*increment
                z=0
            else:
                variables[t]=variables[t]
    else:
        variables[t]=variables[t]

    B1=variables[t][0]
    H3=variables[t][1]
    H4=variables[t][2]
    H5=variables[t][3]
    H6=variables[t][4]
    H7=variables[t][5]
    H8=variables[t][6]
    H9=variables[t][7]
    H10=variables[t][8]
    H11=variables[t][9]
    H12=variables[t][10]

    B2=(4-B1)*H3/61.15+4-16.15*(4-B1)/61.15
    B3=(4-B1)*H4/61.15+4-16.15*(4-B1)/61.15
    B4=(4-B1)*H5/61.15+4-16.15*(4-B1)/61.15
    B5=(4-B1)*H6/61.15+4-16.15*(4-B1)/61.15
    B6=(4-B1)*H7/61.15+4-16.15*(4-B1)/61.15
    B7=(4-B1)*H8/61.15+4-16.15*(4-B1)/61.15
    B8=(4-B1)*H9/61.15+4-16.15*(4-B1)/61.15
    B9=(4-B1)*H10/61.15+4-16.15*(4-B1)/61.15
    B10=(4-B1)*H11/61.15+4-16.15*(4-B1)/61.15
    B11=(4-B1)*H12/61.15+4-16.15*(4-B1)/61.15


    model=mdb.models['Model-1']


    ############## Material

    model.Material(name='Steel')

model.materials['Steel'].Elastic(table=((207000000000.0,
0.3),
        ))
    model.materials['Steel'].Density(table=((7850.0, ), ))
    model.materials['Steel'].Damping(alpha=0.01)
```

```
############## Nodes

N1=(B1,B1,H1)
N2=(B1,B1,H2)
N3=(B2,B2,H3)
N4=(B3,B3,H4)
N5=(B5,B5,H6)
N6=(-B1,B1,H1)
N7=(-B1,B1,H2)
N8=(-B2,B2,H3)
N9=(-B3,B3,H4)
N10=(-B5,B5,H6)
N11=(-B1,-B1,H1)
N12=(-B1,-B1,H2)
N13=(-B2,-B2,H3)
N14=(-B3,-B3,H4)
N15=(-B5,-B5,H6)
N16=(B1,-B1,H1)
N17=(B1,-B1,H2)
N18=(B2,-B2,H3)
N19=(B3,-B3,H4)
N20=(B5,-B5,H6)
N21=(B7,B7,H8)
N22=(B9,B9,H10)
N23=(B11,B11,H12)
N24=(B12,B12,H13)
N25=(-B7,B7,H8)
N26=(-B9,B9,H10)
N27=(-B11,B11,H12)
N28=(-B12,B12,H13)
N29=(-B7,-B7,H8)
N30=(-B9,-B9,H10)
N31=(-B11,-B11,H12)
N32=(-B12,-B12,H13)
N33=(B7,-B7,H8)
N34=(B9,-B9,H10)
N35=(B11,-B11,H12)
N36=(B12,-B12,H13)
N37=(B4,0,H5)
N38=(-B4,0,H5)
N39=(0,B4,H5)
N40=(0,-B4,H5)
N41=(B6,0,H7)
N42=(-B6,0,H7)
N43=(0,B6,H7)
N44=(0,-B6,H7)
N45=(B8,0,H9)
```

```
N46=(-B8,0,H9)
N47=(0,B8,H9)
N48=(0,-B8,H9)
N49=(B10,0,H11)
N50=(-B10,0,H11)
N51=(0,B10,H11)
N52=(0,-B10,H11)
N53=(B12,B12,H14)
N54=(-B12,B12,H14)
N55=(B12,-B12,H14)
N56=(-B12,-B12,H14)

Nmid1=(B1/2,B1/2,H1/2)
Nmid2=(B1/2,B1/2,H2/2)
Nmid3=(B2/2,B2/2,H3/2)
Nmid4=(B3/2,B3/2,H4/2)
Nmid5=(B5/2,B5/2,H6/2)
Nmid6=(-B1/2,B1/2,H1/2)
Nmid7=(-B1/2,B1/2,H2/2)
Nmid8=(-B2/2,B2/2,H3/2)
Nmid9=(-B3/2,B3/2,H4/2)
Nmid10=(-B5/2,B5/2,H6/2)
Nmid11=(-B1/2,-B1/2,H1/2)
Nmid12=(-B1/2,-B1/2,H2/2)
Nmid13=(-B2/2,-B2/2,H3/2)
Nmid14=(-B3/2,-B3/2,H4/2)
Nmid15=(-B5/2,-B5/2,H6/2)
Nmid16=(B1/2,-B1/2,H1/2)
Nmid17=(B1/2,-B1/2,H2/2)
Nmid18=(B2/2,-B2/2,H3/2)
Nmid19=(B3/2,-B3/2,H4/2)
Nmid20=(B5/2,-B5/2,H6/2)
Nmid21=(B7/2,B7/2,H8/2)
Nmid22=(B9/2,B9/2,H10/2)
Nmid23=(B11/2,B11/2,H12/2)
Nmid24=(B12/2,B12/2,H13/2)
Nmid25=(-B7/2,B7/2,H8/2)
Nmid26=(-B9/2,B9/2,H10/2)
Nmid27=(-B11/2,B11/2,H12/2)
Nmid28=(-B12/2,B12/2,H13/2)
Nmid29=(-B7/2,-B7/2,H8/2)
Nmid30=(-B9/2,-B9/2,H10/2)
Nmid31=(-B11/2,-B11/2,H12/2)
Nmid32=(-B12/2,-B12/2,H13/2)
Nmid33=(B7/2,-B7/2,H8/2)
Nmid34=(B9/2,-B9/2,H10/2)
Nmid35=(B11/2,-B11/2,H12/2)
Nmid36=(B12/2,-B12/2,H13/2)
Nmid37=(B4/2,0/2,H5/2)
```

```
Nmid38=(-B4/2,0/2,H5/2)
Nmid39=(0/2,B4/2,H5/2)
Nmid40=(0/2,-B4/2,H5/2)
Nmid41=(B6/2,0/2,H7/2)
Nmid42=(-B6/2,0/2,H7/2)
Nmid43=(0/2,B6/2,H7/2)
Nmid44=(0/2,-B6/2,H7/2)
Nmid45=(B8/2,0/2,H9/2)
Nmid46=(-B8/2,0/2,H9/2)
Nmid47=(0/2,B8/2,H9/2)
Nmid48=(0/2,-B8/2,H9/2)
Nmid49=(B10/2,0/2,H11/2)
Nmid50=(-B10/2,0/2,H11/2)
Nmid51=(0/2,B10/2,H11/2)
Nmid52=(0/2,-B10/2,H11/2)
Nmid53=(B12/2,B12/2,H14/2)
Nmid54=(-B12/2,B12/2,H14/2)
Nmid55=(B12/2,-B12/2,H14/2)
Nmid56=(-B12/2,-B12/2,H14/2)


M1=(N1,N2)
M2=(N2,N3)
M3=(N3,N4)
M4=(N4,N5)
M5=(N6,N7)
M6=(N7,N8)
M7=(N8,N9)
M8=(N9,N10)
M9=(N11,N12)
M10=(N12,N13)
M11=(N13,N14)
M12=(N14,N15)
M13=(N16,N17)
M14=(N17,N18)
M15=(N18,N19)
M16=(N19,N20)
M17=(N5,N21)
M18=(N21,N22)
M19=(N22,N23)
M20=(N23,N24)
M21=(N10,N25)
M22=(N25,N26)
M23=(N26,N27)
M24=(N27,N28)
M25=(N15,N29)
M26=(N29,N30)
M27=(N30,N31)
M28=(N31,N32)
```

```
M29=(N20,N33)
M30=(N33,N34)
M31=(N34,N35)
M32=(N35,N36)
M33=(N3,N8)
M34=(N8,N13)
M35=(N13,N18)
M36=(N18,N3)
M37=(N4,N37)
M38=(N37,N20)
M39=(N19,N37)
M40=(N37,N5)
M41=(N9,N38)
M42=(N38,N15)
M43=(N14,N38)
M44=(N38,N10)
M45=(N4,N39)
M46=(N39,N10)
M47=(N9,N39)
M48=(N39,N5)
M49=(N19,N40)
M50=(N40,N15)
M51=(N14,N40)
M52=(N40,N20)
M53=(N5,N41)
M54=(N41,N33)
M55=(N20,N41)
M56=(N41,N21)
M57=(N10,N42)
M58=(N42,N29)
M59=(N15,N42)
M60=(N42,N25)
M61=(N5,N43)
M62=(N43,N25)
M63=(N10,N43)
M64=(N43,N21)
M65=(N20,N44)
M66=(N44,N29)
M67=(N15,N44)
M68=(N44,N33)
M69=(N21,N45)
M70=(N45,N34)
M71=(N33,N45)
M72=(N45,N22)
M73=(N25,N46)
M74=(N46,N30)
M75=(N29,N46)
M76=(N46,N26)
M77=(N21,N47)
```

```
M78=(N47,N26)
M79=(N25,N47)
M80=(N47,N22)
M81=(N33,N48)
M82=(N48,N30)
M83=(N29,N48)
M84=(N48,N34)
M85=(N22,N49)
M86=(N49,N35)
M87=(N34,N49)
M88=(N49,N23)
M89=(N26,N50)
M90=(N50,N31)
M91=(N30,N50)
M92=(N50,N27)
M93=(N22,N51)
M94=(N51,N27)
M95=(N26,N51)
M96=(N51,N23)
M97=(N34,N52)
M98=(N52,N31)
M99=(N30,N52)
M100=(N52,N35)
M101=(N24,N53)
M102=(N28,N54)
M103=(N32,N56)
M104=(N36,N55)
```

```
M=[[N1,N2],[N2,N3],[N3,N4],[N4,N5],[N6,N7],[N7,N8],[N8,N9],
[N9,N10],[N11,N12],[N12,N13],[N13,N14],[N14,N15],[N16,N17],

[N17,N18],[N18,N19],[N19,N20],[N5,N21],[N21,N22],[N22,N23],
[N23,N24],[N10,N25],[N25,N26],[N26,N27],[N27,N28],[N15,N29]
,

[N29,N30],[N30,N31],[N31,N32],[N20,N33],[N33,N34],[N34,N35]
,[N35,N36],[N3,N8],[N8,N13],[N13,N18],[N18,N3],[N4,N37],[N3
7,N20],

[N19,N37],[N37,N5],[N9,N38],[N38,N15],[N14,N38],[N38,N10],[
N4,N39],[N39,N10],[N9,N39],[N39,N5],[N19,N40],[N40,N15],[N1
4,N40],

[N40,N20],[N5,N41],[N41,N33],[N20,N41],[N41,N21],[N10,N42],
[N42,N29],[N15,N42],[N42,N25],[N5,N43],[N43,N25],[N10,N43],
[N43,N21],

[N20,N44],[N44,N29],[N15,N44],[N44,N33],[N21,N45],[N45,N34]
```

```
,[N33,N45],[N45,N22],[N25,N46],[N46,N30],[N29,N46],[N46,N26
],

[N21,N47],[N47,N26],[N25,N47],[N47,N22],[N33,N48],[N48,N30]
,[N29,N48],[N48,N34],[N22,N49],[N49,N35],[N34,N49],[N49,N23
],

[N26,N50],[N50,N31],[N30,N50],[N50,N27],[N22,N51],[N51,N27]
,[N26,N51],[N51,N23],[N34,N52],[N52,N31],[N30,N52],[N52,N35
],
       [N24,N53],[N28,N54],[N32,N56],[N36,N55]]


Mmid=[[Nmid1,Nmid2],[Nmid2,Nmid3],[Nmid3,Nmid4],[Nmid4,Nmid
5],[Nmid6,Nmid7],[Nmid7,Nmid8],[Nmid8,Nmid9],[Nmid9,Nmid10]
,

[Nmid11,Nmid12],[Nmid12,Nmid13],[Nmid13,Nmid14],[Nmid14,Nmi
d15],[Nmid16,Nmid17],[Nmid17,Nmid18],[Nmid18,Nmid19],

[Nmid19,Nmid20],[Nmid5,Nmid21],[Nmid21,Nmid22],[Nmid22,Nmid
23],[Nmid23,Nmid24],[Nmid10,Nmid25],[Nmid25,Nmid26],

[Nmid26,Nmid27],[Nmid27,Nmid28],[Nmid15,Nmid29],[Nmid29,Nmi
d30],[Nmid30,Nmid31],[Nmid31,Nmid32],[Nmid20,Nmid33],

[Nmid33,Nmid34],[Nmid34,Nmid35],[Nmid35,Nmid36],[Nmid3,Nmid
8],[Nmid8,Nmid13],[Nmid13,Nmid18],[Nmid18,Nmid3],[Nmid4,Nmi
d37],

[Nmid37,Nmid20],[Nmid19,Nmid37],[Nmid37,Nmid5],[Nmid9,Nmid3
8],[Nmid38,Nmid15],[Nmid14,Nmid38],[Nmid38,Nmid10],[Nmid4,N
mid39],

[Nmid39,Nmid10],[Nmid9,Nmid39],[Nmid39,Nmid5],[Nmid19,Nmid4
0],[Nmid40,Nmid15],[Nmid14,Nmid40],[Nmid40,Nmid20],[Nmid5,N
mid41],

[Nmid41,Nmid33],[Nmid20,Nmid41],[Nmid41,Nmid21],[Nmid10,Nmi
d42],[Nmid42,Nmid29],[Nmid15,Nmid42],[Nmid42,Nmid25],

[Nmid5,Nmid43],[Nmid43,Nmid25],[Nmid10,Nmid43],[Nmid43,Nmid
21],[Nmid20,Nmid44],[Nmid44,Nmid29],[Nmid15,Nmid44],[Nmid44
,Nmid33],

[Nmid21,Nmid45],[Nmid45,Nmid34],[Nmid33,Nmid45],[Nmid45,Nmi
d22],[Nmid25,Nmid46],[Nmid46,Nmid30],[Nmid29,Nmid46],[Nmid4
6,Nmid26],
```

```
[Nmid21,Nmid47],[Nmid47,Nmid26],[Nmid25,Nmid47],[Nmid47,Nmid22],[Nmid33,Nmid48],[Nmid48,Nmid30],[Nmid29,Nmid48],[Nmid48,Nmid34],

[Nmid22,Nmid49],[Nmid49,Nmid35],[Nmid34,Nmid49],[Nmid49,Nmid23],[Nmid26,Nmid50],[Nmid50,Nmid31],[Nmid30,Nmid50],[Nmid50,Nmid27],

[Nmid22,Nmid51],[Nmid51,Nmid27],[Nmid26,Nmid51],[Nmid51,Nmid23],[Nmid34,Nmid52],[Nmid52,Nmid31],[Nmid30,Nmid52],[Nmid52,Nmid35],

[Nmid24,Nmid53],[Nmid28,Nmid54],[Nmid32,Nmid56],[Nmid36,Nmid55]]


    ############## Datum points

    model.Part(dimensionality=THREE_D, name='Part-1',
type=DEFORMABLE_BODY)

    trusspart=model.parts['Part-1']



    trusspart.DatumPointByCoordinate(coords=(B1, B1, H1))
    trusspart.DatumPointByCoordinate(coords=(B1, B1, H2))
    trusspart.DatumPointByCoordinate(coords=(B2, B2, H3))
    trusspart.DatumPointByCoordinate(coords=(B3, B3, H4))
    trusspart.DatumPointByCoordinate(coords=(B5, B5, H6))
    trusspart.DatumPointByCoordinate(coords=(-B1, B1, H1))
    trusspart.DatumPointByCoordinate(coords=(-B1, B1, H2))
    trusspart.DatumPointByCoordinate(coords=(-B2, B2, H3))
    trusspart.DatumPointByCoordinate(coords=(-B3, B3, H4))
    trusspart.DatumPointByCoordinate(coords=(-B5, B5, H6))
    trusspart.DatumPointByCoordinate(coords=(-B1, -B1, H1))
    trusspart.DatumPointByCoordinate(coords=(-B1, -B1, H2))
    trusspart.DatumPointByCoordinate(coords=(-B2, -B2, H3))
    trusspart.DatumPointByCoordinate(coords=(-B3, -B3, H4))
    trusspart.DatumPointByCoordinate(coords=(-B5, -B5, H6))
    trusspart.DatumPointByCoordinate(coords=(B1, -B1, H1))
    trusspart.DatumPointByCoordinate(coords=(B1, -B1, H2))
    trusspart.DatumPointByCoordinate(coords=(B2, -B2, H3))
    trusspart.DatumPointByCoordinate(coords=(B3, -B3, H4))
    trusspart.DatumPointByCoordinate(coords=(B5, -B5, H6))
    trusspart.DatumPointByCoordinate(coords=(B7, B7, H8))
    trusspart.DatumPointByCoordinate(coords=(B9, B9, H10))
```

```
    trusspart.DatumPointByCoordinate(coords=(B11, B11,
H12))
    trusspart.DatumPointByCoordinate(coords=(B12, B12,
H13))
    trusspart.DatumPointByCoordinate(coords=(-B7, B7, H8))
    trusspart.DatumPointByCoordinate(coords=(-B9, B9, H10))
    trusspart.DatumPointByCoordinate(coords=(-B11, B11,
H12))
    trusspart.DatumPointByCoordinate(coords=(-B12, B12,
H13))
    trusspart.DatumPointByCoordinate(coords=(-B7, -B7, H8))
    trusspart.DatumPointByCoordinate(coords=(-B9, -B9,
H10))
    trusspart.DatumPointByCoordinate(coords=(-B11, -B11,
H12))
    trusspart.DatumPointByCoordinate(coords=(-B12, -B12,
H13))
    trusspart.DatumPointByCoordinate(coords=(B7, -B7, H8))
    trusspart.DatumPointByCoordinate(coords=(B9, -B9, H10))
    trusspart.DatumPointByCoordinate(coords=(B11, -B11,
H12))
    trusspart.DatumPointByCoordinate(coords=(B12, -B12,
H13))
    trusspart.DatumPointByCoordinate(coords=(B4, 0, H5))
    trusspart.DatumPointByCoordinate(coords=(-B4, 0, H5))
    trusspart.DatumPointByCoordinate(coords=(0, B4, H5))
    trusspart.DatumPointByCoordinate(coords=(0, -B4, H5))
    trusspart.DatumPointByCoordinate(coords=(B6, 0, H7))
    trusspart.DatumPointByCoordinate(coords=(-B6, 0, H7))
    trusspart.DatumPointByCoordinate(coords=(0, B6, H7))
    trusspart.DatumPointByCoordinate(coords=(0, -B6, H7))
    trusspart.DatumPointByCoordinate(coords=(B8, 0, H9))
    trusspart.DatumPointByCoordinate(coords=(-B8, 0, H9))
    trusspart.DatumPointByCoordinate(coords=(0, B8, H9))
    trusspart.DatumPointByCoordinate(coords=(0, -B8, H9))
    trusspart.DatumPointByCoordinate(coords=(B10, 0, H11))
    trusspart.DatumPointByCoordinate(coords=(-B10, 0, H11))
    trusspart.DatumPointByCoordinate(coords=(0, B10, H11))
    trusspart.DatumPointByCoordinate(coords=(0, -B10, H11))
    trusspart.DatumPointByCoordinate(coords=(B12, B12,
H14))
    trusspart.DatumPointByCoordinate(coords=(-B12, B12,
H14))
    trusspart.DatumPointByCoordinate(coords=(B12, -B12,
H14))
    trusspart.DatumPointByCoordinate(coords=(-B12, -B12,
H14))
```

```
concrete_center=trusspart.DatumPointByCoordinate(coords=(CC))
    trusspart.DatumPointByCoordinate(coords=(0, 0, towerh1))
    trusspart.DatumPointByCoordinate(coords=(0, 0, towerh2))
    trusspart.DatumPointByCoordinate(coords=(0, 0, towerh3))
    trusspart.DatumPointByCoordinate(coords=(0, 0, towerh4))
    trusspart.DatumPointByCoordinate(coords=(0, 0, towerh5))
    trusspart.DatumPointByCoordinate(coords=(0, 0, towerh6))
    trusspart.DatumPointByCoordinate(coords=(0, 0, towerh7))
    trusspart.DatumPointByCoordinate(coords=(0, 0, towerh8))
    trusspart.DatumPointByCoordinate(coords=(0, 0, towerh9))


rotor_center=trusspart.DatumPointByCoordinate(coords=(hub_cg))

nacelle_center=trusspart.DatumPointByCoordinate(coords=(nacelle_cg))

rotor_center=trusspart.DatumPointByCoordinate(coords=(hub_dummy_wire))

nacelle_center=trusspart.DatumPointByCoordinate(coords=(nacelle_dummy_wire))


    CC_id=concrete_center.id


    ############## Nodenames

    trusspart.features.changeKey(fromName='Datum pt-1',toName='N1')
    trusspart.features.changeKey(fromName='Datum pt-2',toName='N2')
    trusspart.features.changeKey(fromName='Datum pt-3',toName='N3')
    trusspart.features.changeKey(fromName='Datum pt-4',toName='N4')
```

```
    trusspart.features.changeKey(fromName='Datum pt-
5',toName='N5')
    trusspart.features.changeKey(fromName='Datum pt-
6',toName='N6')
    trusspart.features.changeKey(fromName='Datum pt-
7',toName='N7')
    trusspart.features.changeKey(fromName='Datum pt-
8',toName='N8')
    trusspart.features.changeKey(fromName='Datum pt-
9',toName='N9')
    trusspart.features.changeKey(fromName='Datum pt-
10',toName='N10')
    trusspart.features.changeKey(fromName='Datum pt-
11',toName='N11')
    trusspart.features.changeKey(fromName='Datum pt-
12',toName='N12')
    trusspart.features.changeKey(fromName='Datum pt-
13',toName='N13')
    trusspart.features.changeKey(fromName='Datum pt-
14',toName='N14')
    trusspart.features.changeKey(fromName='Datum pt-
15',toName='N15')
    trusspart.features.changeKey(fromName='Datum pt-
16',toName='N16')
    trusspart.features.changeKey(fromName='Datum pt-
17',toName='N17')
    trusspart.features.changeKey(fromName='Datum pt-
18',toName='N18')
    trusspart.features.changeKey(fromName='Datum pt-
19',toName='N19')
    trusspart.features.changeKey(fromName='Datum pt-
20',toName='N20')
    trusspart.features.changeKey(fromName='Datum pt-
21',toName='N21')
    trusspart.features.changeKey(fromName='Datum pt-
22',toName='N22')
    trusspart.features.changeKey(fromName='Datum pt-
23',toName='N23')
    trusspart.features.changeKey(fromName='Datum pt-
24',toName='N24')
    trusspart.features.changeKey(fromName='Datum pt-
25',toName='N25')
    trusspart.features.changeKey(fromName='Datum pt-
26',toName='N26')
    trusspart.features.changeKey(fromName='Datum pt-
27',toName='N27')
    trusspart.features.changeKey(fromName='Datum pt-
28',toName='N28')
```

```
    trusspart.features.changeKey(fromName='Datum pt-
29',toName='N29')
    trusspart.features.changeKey(fromName='Datum pt-
30',toName='N30')
    trusspart.features.changeKey(fromName='Datum pt-
31',toName='N31')
    trusspart.features.changeKey(fromName='Datum pt-
32',toName='N32')
    trusspart.features.changeKey(fromName='Datum pt-
33',toName='N33')
    trusspart.features.changeKey(fromName='Datum pt-
34',toName='N34')
    trusspart.features.changeKey(fromName='Datum pt-
35',toName='N35')
    trusspart.features.changeKey(fromName='Datum pt-
36',toName='N36')
    trusspart.features.changeKey(fromName='Datum pt-
37',toName='N37')
    trusspart.features.changeKey(fromName='Datum pt-
38',toName='N38')
    trusspart.features.changeKey(fromName='Datum pt-
39',toName='N39')
    trusspart.features.changeKey(fromName='Datum pt-
40',toName='N40')
    trusspart.features.changeKey(fromName='Datum pt-
41',toName='N41')
    trusspart.features.changeKey(fromName='Datum pt-
42',toName='N42')
    trusspart.features.changeKey(fromName='Datum pt-
43',toName='N43')
    trusspart.features.changeKey(fromName='Datum pt-
44',toName='N44')
    trusspart.features.changeKey(fromName='Datum pt-
45',toName='N45')
    trusspart.features.changeKey(fromName='Datum pt-
46',toName='N47')
    trusspart.features.changeKey(fromName='Datum pt-
47',toName='N48')
    trusspart.features.changeKey(fromName='Datum pt-
48',toName='N46')
    trusspart.features.changeKey(fromName='Datum pt-
49',toName='N49')
    trusspart.features.changeKey(fromName='Datum pt-
50',toName='N50')
    trusspart.features.changeKey(fromName='Datum pt-
51',toName='N51')
    trusspart.features.changeKey(fromName='Datum pt-
52',toName='N52')
```

```
    trusspart.features.changeKey(fromName='Datum pt-
53',toName='N53')
    trusspart.features.changeKey(fromName='Datum pt-
54',toName='N54')
    trusspart.features.changeKey(fromName='Datum pt-
55',toName='N55')
    trusspart.features.changeKey(fromName='Datum pt-
56',toName='N56')



############## Wires

    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[1],trusspart.datums[2
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[2],trusspart.datums[3
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[3],trusspart.datums[4
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[4],trusspart.datums[5
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[6],trusspart.datums[7
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[7],trusspart.datums[8
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[8],trusspart.datums[9
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[9],trusspart.datums[1
0]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[11],trusspart.datums[
12]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[12],trusspart.datums[
13]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[13],trusspart.datums[
14]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[14],trusspart.datums[
15]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[16],trusspart.datums[
17]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[17],trusspart.datums[
18]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[18],trusspart.datums[
19]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[19],trusspart.datums[
20]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[5],trusspart.datums[2
1]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[21],trusspart.datums[
22]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[22],trusspart.datums[
23]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[23],trusspart.datums[
24]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[10],trusspart.datums[
25]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[25],trusspart.datums[
26]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[26],trusspart.datums[
27]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[27],trusspart.datums[
28]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[15],trusspart.datums[
29]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[29],trusspart.datums[
30]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[30],trusspart.datums[
31]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[31],trusspart.datums[
32]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[20],trusspart.datums[
33]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[33],trusspart.datums[
34]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[34],trusspart.datums[
35]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[35],trusspart.datums[
36]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[3],trusspart.datums[8
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[8],trusspart.datums[1
3]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[13],trusspart.datums[
18]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[18],trusspart.datums[
3]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[4],trusspart.datums[3
7]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[37],trusspart.datums[
20]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[19],trusspart.datums[
37]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[37],trusspart.datums[
5]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[9],trusspart.datums[3
8]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[38],trusspart.datums[
15]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[14],trusspart.datums[
38]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[38],trusspart.datums[
10]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[4],trusspart.datums[3
9]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[39],trusspart.datums[
10]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[9],trusspart.datums[3
9]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[39],trusspart.datums[
5]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[19],trusspart.datums[
40]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[40],trusspart.datums[
15]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[14],trusspart.datums[
40]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[40],trusspart.datums[
20]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[5],trusspart.datums[4
1]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[41],trusspart.datums[
33]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[20],trusspart.datums[
41]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[41],trusspart.datums[
21]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[10],trusspart.datums[
42]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[42],trusspart.datums[
29]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[15],trusspart.datums[
42]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[42],trusspart.datums[
25]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[5],trusspart.datums[4
3]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[43],trusspart.datums[
25]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[10],trusspart.datums[
43]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[43],trusspart.datums[
21]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[20],trusspart.datums[
44]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[44],trusspart.datums[
29]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[15],trusspart.datums[
44]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[44],trusspart.datums[
33]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[21],trusspart.datums[
45]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[45],trusspart.datums[
34]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[33],trusspart.datums[
45]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[45],trusspart.datums[
22]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[25],trusspart.datums[
46]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[46],trusspart.datums[
30]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[29],trusspart.datums[
46]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[46],trusspart.datums[
26]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[21],trusspart.datums[
47]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[47],trusspart.datums[
26]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[25],trusspart.datums[
47]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[47],trusspart.datums[
22]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[33],trusspart.datums[
48]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[48],trusspart.datums[
30]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[29],trusspart.datums[
48]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[48],trusspart.datums[
34]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[22],trusspart.datums[
49]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[49],trusspart.datums[
35]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[34],trusspart.datums[
49]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[49],trusspart.datums[
23]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[26],trusspart.datums[
50]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[50],trusspart.datums[
31]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[30],trusspart.datums[
50]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[50],trusspart.datums[
27]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[22],trusspart.datums[
51]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[51],trusspart.datums[
27]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[26],trusspart.datums[
51]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[51],trusspart.datums[
23]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[34],trusspart.datums[
52]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[52],trusspart.datums[
31]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[30],trusspart.datums[
52]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[52],trusspart.datums[
35]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[24],trusspart.datums[
53]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[28],trusspart.datums[
54]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[32],trusspart.datums[
56]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[36],trusspart.datums[
55]), ))

    ##trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[CC_id], (0.0, 0.0,
towerh1)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh1), (0.0, 0.0,
towerh2)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh2), (0.0, 0.0,
towerh3)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh3), (0.0, 0.0,
towerh4)), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh4), (0.0, 0.0,
towerh5)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh5), (0.0, 0.0,
towerh6)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh6), (0.0, 0.0,
towerh7)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh7), (0.0, 0.0,
towerh8)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh8), (0.0, 0.0,
towerh9)), ))



    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((-nacelle_overhang*c_overhang,-
nacelle_overhang*s_overhang,towerh9+1.75), (-
nacelle_overhang*c_overhang,-
nacelle_overhang*s_overhang,towerh9+1.75+.05)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((hub_overhang*c_overhang,hub_overhang*
s_overhang,towerh9+2.40),
(hub_overhang*c_overhang,hub_overhang*s_overhang,towerh9+2.
40+0.05)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0,0,18.15), (0,0,18.16)), ))


    ############## Wire set creation


f=[[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[
N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],
[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1]
,[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1
],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N
1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[
N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],
[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1]
,[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],]
    f[0][0]=tuple(map(sum,zip(Mmid[0][0],Mmid[0][1])))
    f[1][0]=tuple(map(sum,zip(Mmid[1][0],Mmid[1][1])))
    f[2][0]=tuple(map(sum,zip(Mmid[2][0],Mmid[2][1])))
```

```
f[3][0]=tuple(map(sum,zip(Mmid[3][0],Mmid[3][1])))
f[4][0]=tuple(map(sum,zip(Mmid[4][0],Mmid[4][1])))
f[5][0]=tuple(map(sum,zip(Mmid[5][0],Mmid[5][1])))
f[6][0]=tuple(map(sum,zip(Mmid[6][0],Mmid[6][1])))
f[7][0]=tuple(map(sum,zip(Mmid[7][0],Mmid[7][1])))
f[8][0]=tuple(map(sum,zip(Mmid[8][0],Mmid[8][1])))
f[9][0]=tuple(map(sum,zip(Mmid[9][0],Mmid[9][1])))
f[10][0]=tuple(map(sum,zip(Mmid[10][0],Mmid[10][1])))
f[11][0]=tuple(map(sum,zip(Mmid[11][0],Mmid[11][1])))
f[12][0]=tuple(map(sum,zip(Mmid[12][0],Mmid[12][1])))
f[13][0]=tuple(map(sum,zip(Mmid[13][0],Mmid[13][1])))
f[14][0]=tuple(map(sum,zip(Mmid[14][0],Mmid[14][1])))
f[15][0]=tuple(map(sum,zip(Mmid[15][0],Mmid[15][1])))
f[16][0]=tuple(map(sum,zip(Mmid[16][0],Mmid[16][1])))
f[17][0]=tuple(map(sum,zip(Mmid[17][0],Mmid[17][1])))
f[18][0]=tuple(map(sum,zip(Mmid[18][0],Mmid[18][1])))
f[19][0]=tuple(map(sum,zip(Mmid[19][0],Mmid[19][1])))
f[20][0]=tuple(map(sum,zip(Mmid[20][0],Mmid[20][1])))
f[21][0]=tuple(map(sum,zip(Mmid[21][0],Mmid[21][1])))
f[22][0]=tuple(map(sum,zip(Mmid[22][0],Mmid[22][1])))
f[23][0]=tuple(map(sum,zip(Mmid[23][0],Mmid[23][1])))
f[24][0]=tuple(map(sum,zip(Mmid[24][0],Mmid[24][1])))
f[25][0]=tuple(map(sum,zip(Mmid[25][0],Mmid[25][1])))
f[26][0]=tuple(map(sum,zip(Mmid[26][0],Mmid[26][1])))
f[27][0]=tuple(map(sum,zip(Mmid[27][0],Mmid[27][1])))
f[28][0]=tuple(map(sum,zip(Mmid[28][0],Mmid[28][1])))
f[29][0]=tuple(map(sum,zip(Mmid[29][0],Mmid[29][1])))
f[30][0]=tuple(map(sum,zip(Mmid[30][0],Mmid[30][1])))
f[31][0]=tuple(map(sum,zip(Mmid[31][0],Mmid[31][1])))
f[32][0]=tuple(map(sum,zip(Mmid[32][0],Mmid[32][1])))
f[33][0]=tuple(map(sum,zip(Mmid[33][0],Mmid[33][1])))
f[34][0]=tuple(map(sum,zip(Mmid[34][0],Mmid[34][1])))
f[35][0]=tuple(map(sum,zip(Mmid[35][0],Mmid[35][1])))
f[36][0]=tuple(map(sum,zip(Mmid[36][0],Mmid[36][1])))
f[37][0]=tuple(map(sum,zip(Mmid[37][0],Mmid[37][1])))
f[38][0]=tuple(map(sum,zip(Mmid[38][0],Mmid[38][1])))
f[39][0]=tuple(map(sum,zip(Mmid[39][0],Mmid[39][1])))
f[40][0]=tuple(map(sum,zip(Mmid[40][0],Mmid[40][1])))
f[41][0]=tuple(map(sum,zip(Mmid[41][0],Mmid[41][1])))
f[42][0]=tuple(map(sum,zip(Mmid[42][0],Mmid[42][1])))
f[43][0]=tuple(map(sum,zip(Mmid[43][0],Mmid[43][1])))
f[44][0]=tuple(map(sum,zip(Mmid[44][0],Mmid[44][1])))
f[45][0]=tuple(map(sum,zip(Mmid[45][0],Mmid[45][1])))
f[46][0]=tuple(map(sum,zip(Mmid[46][0],Mmid[46][1])))
f[47][0]=tuple(map(sum,zip(Mmid[47][0],Mmid[47][1])))
f[48][0]=tuple(map(sum,zip(Mmid[48][0],Mmid[48][1])))
f[49][0]=tuple(map(sum,zip(Mmid[49][0],Mmid[49][1])))
f[50][0]=tuple(map(sum,zip(Mmid[50][0],Mmid[50][1])))
f[51][0]=tuple(map(sum,zip(Mmid[51][0],Mmid[51][1])))
```

```
f[52][0]=tuple(map(sum,zip(Mmid[52][0],Mmid[52][1])))
f[53][0]=tuple(map(sum,zip(Mmid[53][0],Mmid[53][1])))
f[54][0]=tuple(map(sum,zip(Mmid[54][0],Mmid[54][1])))
f[55][0]=tuple(map(sum,zip(Mmid[55][0],Mmid[55][1])))
f[56][0]=tuple(map(sum,zip(Mmid[56][0],Mmid[56][1])))
f[57][0]=tuple(map(sum,zip(Mmid[57][0],Mmid[57][1])))
f[58][0]=tuple(map(sum,zip(Mmid[58][0],Mmid[58][1])))
f[59][0]=tuple(map(sum,zip(Mmid[59][0],Mmid[59][1])))
f[60][0]=tuple(map(sum,zip(Mmid[60][0],Mmid[60][1])))
f[61][0]=tuple(map(sum,zip(Mmid[61][0],Mmid[61][1])))
f[62][0]=tuple(map(sum,zip(Mmid[62][0],Mmid[62][1])))
f[63][0]=tuple(map(sum,zip(Mmid[63][0],Mmid[63][1])))
f[64][0]=tuple(map(sum,zip(Mmid[64][0],Mmid[64][1])))
f[65][0]=tuple(map(sum,zip(Mmid[65][0],Mmid[65][1])))
f[66][0]=tuple(map(sum,zip(Mmid[66][0],Mmid[66][1])))
f[67][0]=tuple(map(sum,zip(Mmid[67][0],Mmid[67][1])))
f[68][0]=tuple(map(sum,zip(Mmid[68][0],Mmid[68][1])))
f[69][0]=tuple(map(sum,zip(Mmid[69][0],Mmid[69][1])))
f[70][0]=tuple(map(sum,zip(Mmid[70][0],Mmid[70][1])))
f[71][0]=tuple(map(sum,zip(Mmid[71][0],Mmid[71][1])))
f[72][0]=tuple(map(sum,zip(Mmid[72][0],Mmid[72][1])))
f[73][0]=tuple(map(sum,zip(Mmid[73][0],Mmid[73][1])))
f[74][0]=tuple(map(sum,zip(Mmid[74][0],Mmid[74][1])))
f[75][0]=tuple(map(sum,zip(Mmid[75][0],Mmid[75][1])))
f[76][0]=tuple(map(sum,zip(Mmid[76][0],Mmid[76][1])))
f[77][0]=tuple(map(sum,zip(Mmid[77][0],Mmid[77][1])))
f[78][0]=tuple(map(sum,zip(Mmid[78][0],Mmid[78][1])))
f[79][0]=tuple(map(sum,zip(Mmid[79][0],Mmid[79][1])))
f[80][0]=tuple(map(sum,zip(Mmid[80][0],Mmid[80][1])))
f[81][0]=tuple(map(sum,zip(Mmid[81][0],Mmid[81][1])))
f[82][0]=tuple(map(sum,zip(Mmid[82][0],Mmid[82][1])))
f[83][0]=tuple(map(sum,zip(Mmid[83][0],Mmid[83][1])))
f[84][0]=tuple(map(sum,zip(Mmid[84][0],Mmid[84][1])))
f[85][0]=tuple(map(sum,zip(Mmid[85][0],Mmid[85][1])))
f[86][0]=tuple(map(sum,zip(Mmid[86][0],Mmid[86][1])))
f[87][0]=tuple(map(sum,zip(Mmid[87][0],Mmid[87][1])))
f[88][0]=tuple(map(sum,zip(Mmid[88][0],Mmid[88][1])))
f[89][0]=tuple(map(sum,zip(Mmid[89][0],Mmid[89][1])))
f[90][0]=tuple(map(sum,zip(Mmid[90][0],Mmid[90][1])))
f[91][0]=tuple(map(sum,zip(Mmid[91][0],Mmid[91][1])))
f[92][0]=tuple(map(sum,zip(Mmid[92][0],Mmid[92][1])))
f[93][0]=tuple(map(sum,zip(Mmid[93][0],Mmid[93][1])))
f[94][0]=tuple(map(sum,zip(Mmid[94][0],Mmid[94][1])))
f[95][0]=tuple(map(sum,zip(Mmid[95][0],Mmid[95][1])))
f[96][0]=tuple(map(sum,zip(Mmid[96][0],Mmid[96][1])))
f[97][0]=tuple(map(sum,zip(Mmid[97][0],Mmid[97][1])))
f[98][0]=tuple(map(sum,zip(Mmid[98][0],Mmid[98][1])))
f[99][0]=tuple(map(sum,zip(Mmid[99][0],Mmid[99][1])))
```

```
f[100][0]=tuple(map(sum,zip(Mmid[100][0],Mmid[100][1])))

f[101][0]=tuple(map(sum,zip(Mmid[101][0],Mmid[101][1])))

f[102][0]=tuple(map(sum,zip(Mmid[102][0],Mmid[102][1])))

f[103][0]=tuple(map(sum,zip(Mmid[103][0],Mmid[103][1])))

    ############## Sets

P=[2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,3,3,
3,3,3,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,4,4,4,4]
    length=len(P)
    len2=length+1


trusspart.Set(edges=trusspart.edges.getSequenceFromMask(('[
#ffffffff:3 #ff ]', ), ), name='All_set')
    for i in range(1,len2):
        trusspart.Set(edges=trusspart.edges.findAt(((f[i-
1][0]),),), name='Wire-'+str(i)+'-Set-1')
        wire_region=trusspart.edges.findAt(((f[i-1][0]),),)
        globals()['wire%s' % i] = wire_region
    trusspart.Set(edges=trusspart.edges.findAt([CC]),
name='Concrete_help_wire')

    ############## Member rename


    ############## Sections
    waterdensity=1000

    radius1=0.4
    radius2=0.6
    radius3=0.6
    radius4=0.6

    thickness1=0.02
    thickness2=0.05
    thickness3=0.035
    thickness4=0.04

    model.PipeProfile(name='Dim1', r=radius1, t=thickness1)
    model.PipeProfile(name='Dim2', r=radius2, t=thickness2)
    model.PipeProfile(name='Dim3', r=radius3, t=thickness3)
```

```
    model.PipeProfile(name='Dim4', r=radius4, t=thickness4)


    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Braces', poissonRatio=0.3, profile='Dim1',
temperatureVar=LINEAR)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius1, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Braces_submerged', poissonRatio=0.3, profile='Dim1',
temperatureVar=LINEAR, useFluidInertia=ON)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius1, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Braces_half_submerged', poissonRatio=0.3,
profile='Dim1', submerged=HALF, temperatureVar=LINEAR,
useFluidInertia=ON)

    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Lower_leg', poissonRatio=0.3, profile='Dim2',
temperatureVar=LINEAR)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius2, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Lower_leg_submerged', poissonRatio=0.3,
profile='Dim2', temperatureVar=LINEAR, useFluidInertia=ON)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius2, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Lower_leg_half_submerged', poissonRatio=0.3,
profile='Dim2', submerged=HALF, temperatureVar=LINEAR,
useFluidInertia=ON)

    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Upper_leg', poissonRatio=0.3, profile='Dim3',
temperatureVar=LINEAR)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius3, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Upper_leg_submerged', poissonRatio=0.3,
profile='Dim3', temperatureVar=LINEAR, useFluidInertia=ON)
```

```
model.BeamSection(consistentMassMatrix=False,crossSectionRadius=radius3, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Upper_leg_half_submerged', poissonRatio=0.3,
profile='Dim3', submerged=HALF, temperatureVar=LINEAR,
useFluidInertia=ON)

    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel', name='Top',
poissonRatio=0.3, profile='Dim4', temperatureVar=LINEAR)

model.BeamSection(consistentMassMatrix=False,crossSectionRadius=radius4, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Top_submerged', poissonRatio=0.3, profile='Dim4',
temperatureVar=LINEAR, useFluidInertia=ON)

model.BeamSection(consistentMassMatrix=False,crossSectionRadius=radius4, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Top_half__submerged', poissonRatio=0.3,
profile='Dim4', submerged=HALF, temperatureVar=LINEAR,
useFluidInertia=ON)



    ################ Tower sections

    model.PipeProfile(name='Tower2', r=5.577/2, t=0.032)
    model.PipeProfile(name='Tower3', r=5.318/2, t=0.030)
    model.PipeProfile(name='Tower4', r=5.082/2, t=0.028)
    model.PipeProfile(name='Tower5', r=4.800/2, t=0.024)
    model.PipeProfile(name='Tower6', r=4.565/2, t=0.024)
    model.PipeProfile(name='Tower7', r=4.329/2, t=0.022)
    model.PipeProfile(name='Tower8', r=4.118/2, t=0.030)
    model.PipeProfile(name='Tower9', r=4.000/2, t=0.030)


    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_2', poissonRatio=0.3, profile='Tower2',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_3', poissonRatio=0.3, profile='Tower3',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
```

```
name='Tower_section_4', poissonRatio=0.3, profile='Tower4',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_5', poissonRatio=0.3, profile='Tower5',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_6', poissonRatio=0.3, profile='Tower6',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_7', poissonRatio=0.3, profile='Tower7',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_8', poissonRatio=0.3, profile='Tower8',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_9', poissonRatio=0.3, profile='Tower9',
temperatureVar=LINEAR)


    towermid0=(0,0,(18.15+towerh1)/2)
    towermid1=(0,0,(towerh1+towerh2)/2)
    towermid2=(0,0,(towerh2+towerh3)/2)
    towermid3=(0,0,(towerh3+towerh4)/2)
    towermid4=(0,0,(towerh4+towerh5)/2)
    towermid5=(0,0,(towerh5+towerh6)/2)
    towermid6=(0,0,(towerh6+towerh7)/2)
    towermid7=(0,0,(towerh7+towerh8)/2)
    towermid8=(0,0,(towerh8+towerh9)/2)


    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid1])),
sectionName='Tower_section_2',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid2])),
sectionName='Tower_section_3',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid3])),
```

```
sectionName='Tower_section_4',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid4])),
sectionName='Tower_section_5',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid5])),
sectionName='Tower_section_6',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid6])),
sectionName='Tower_section_7',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid7])),
sectionName='Tower_section_8',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid8])),
sectionName='Tower_section_9',thicknessAssignment=FROM_SECT
ION)



trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid1])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid2])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid3])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid4])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid5])))
```

```
trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid6])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid7])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid8])))


    ################ Dummy sections
    Concrete_center=[0,0,18.155]
    model.PipeProfile(name='Dummy', r=0.1, t=0.010)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Dummy_section', poissonRatio=0.3, profile='Dummy',
temperatureVar=LINEAR)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([nacelle_cg])),
sectionName='Dummy_section',thicknessAssignment=FROM_SECTIO
N)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([hub_cg])),
sectionName='Dummy_section',thicknessAssignment=FROM_SECTIO
N)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([Concrete_center
])),
sectionName='Dummy_section',thicknessAssignment=FROM_SECTIO
N)

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([nacelle_cg])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([hub_cg])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
```

```
region=Region(edges=trusspart.edges.findAt([Concrete_center
])))

    ################ Steps

    model.StaticStep(name='Step-1', previous='Initial')


    ############## Properties

    S=[0]*len(P)
    member_sets=[0]*len(P)

    for i in xrange(len(P)):
        if P[i-1]==1:

            if M[i-1][0][2]>=0 and M[i-1][1][2]>=0:
                S[i-1]='Braces'
            elif M[i-1][0][2]<0 and M[i-1][1][2]<0:
                S[i-1]='Braces_submerged'
            else:
                S[i-1]='Braces_half_submerged'

        elif P[i-1]==2:
            if M[i-1][0][2]>=0 and M[i-1][1][2]>=0:
                S[i-1]='Lower_leg'
            elif M[i-1][0][2]<0 and M[i-1][1][2]<0:
                S[i-1]='Lower_leg_submerged'
            else:
                S[i-1]='Lower_leg_half_submerged'

        elif P[i-1]==3:
            if M[i-1][0][2]>=0 and M[i-1][1][2]>=0:
                S[i-1]='Upper_leg'
            elif M[i-1][0][2]<0 and M[i-1][1][2]<0:
                S[i-1]='Upper_leg_submerged'
            else:
                S[i-1]='Upper_leg_half_submerged'

        else:
            if M[i-1][0][2]>=0 and M[i-1][1][2]>=0:
                S[i-1]='Top'
            elif M[i-1][0][2]<0 and M[i-1][1][2]<0:
                S[i-1]='Top_submerged'
            else:
                S[i-1]='Top_half_submerged'

        member_sets[i-1]='Wire-'+str(i)+'-Set-1'
```

```
        member_sets[len(P)-1]='Wire-'+str(len(P))+'-Set-1'


    ############## Section assignment

    for i in xrange(len(P)):

trusspart.SectionAssignment(offset=0.0,offsetField='',
offsetType=MIDDLE_SURFACE,
region=trusspart.sets[member_sets[i]], sectionName=S[i],
thicknessAssignment=FROM_SECTION)


    ############## Mesh


trusspart.setElementType(elemTypes=(ElemType(elemCode=B31,
elemLibrary=STANDARD), ),
regions=(trusspart.edges.getSequenceFromMask(('[#ffffffff:3
#ff ]', ), ), ))

    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([CC]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([nacelle_cg]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([hub_cg]), number=5)
    for i in xrange(len(P)+1):
            trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt(f[i-1]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid1]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid2]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid3]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid4]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid5]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid6]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid7]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid8]), number=5)
```

```
    trusspart.generateMesh()

    ############## Assembly

    model.rootAssembly.DatumCsysByDefault(CARTESIAN)
    model.rootAssembly.Instance(dependent=ON, name='Part-1-
1',  part=trusspart)

    model.rootAssembly.regenerate()

    model.rootAssembly.Set(name='Rotor_nacelle',
vertices=model.rootAssembly.instances['Part-1-
1'].vertices.findAt([nacelle_cg],[hub_cg]))

model.rootAssembly.Set(name='Concrete_jacket_connection',
vertices=model.rootAssembly.instances['Part-1-
1'].vertices.findAt([N24],[N28],[N32],[N36],[CC]))


model.Coupling(controlPoint=Region(vertices=model.rootAssem
bly.instances['Part-1-1'].vertices.findAt([point_mass_1])),
couplingType=KINEMATIC, influenceRadius=WHOLE_SURFACE,
localCsys=None,
        name='Constraint-2',
surface=model.rootAssembly.sets['Concrete_jacket_connection
'], u1=ON, u2=ON, u3=ON, ur1=ON, ur2=ON, ur3=ON)


model.Coupling(controlPoint=Region(vertices=model.rootAssem
bly.instances['Part-1-1'].vertices.findAt([point_mass_3])),
couplingType=KINEMATIC, influenceRadius=WHOLE_SURFACE,
localCsys=None,
        name='Constraint-3',
surface=model.rootAssembly.sets['Rotor_nacelle'], u1=ON,
u2=ON, u3=ON, ur1=ON, ur2=ON, ur3=ON)


    ############## Point masses


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=concrete_Iw,
i22=concrete_Iw, i33=concrete_Ih, mass=concrete_mass, name=
        'Inertia-1',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([CC])))
```

```
model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=0, i22=0, i33=0, mass=1900,
name=
        'Pointmass-1',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([point_mass_1])))


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=0, i22=0, i33=0, mass=1400,
name=
        'Pointmass-2',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([point_mass_2])))


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=0, i22=0, i33=0, mass=1000,
name=
        'Pointmass-3',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([point_mass_3])))


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=0, i22=0, i33=0,
mass=240000, name=
        'Pointmass-4',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([nacelle_cg])))


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=0, i22=0, i33=0,
mass=110000, name=
        'Pointmass-5',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([hub_cg])))


    ############## Load

    model.ConcentratedForce(cf1=5600000,
```

```
        createStepName='Step-1', distributionType=UNIFORM,
field='', localCsys=None
        , name='LC1-fx',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([load_point])))
    model.Moment(cm2=48420000, createStepName=
        'Step-1', distributionType=UNIFORM, field='',
localCsys=None, name=
        'LC1-my',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([load_point])))
    model.Gravity(comp3=-9.81, createStepName='Step-1',
        distributionType=UNIFORM, field='', name='Gravity')


    ############## BC

    model.rootAssembly.Set(name='BC_set',
vertices=model.rootAssembly.instances['Part-1-
1'].vertices.findAt([N1],[N6],[N11],[N16]))
    model.DisplacementBC(amplitude=UNSET,
createStepName='Step-1',distributionType=UNIFORM,
fieldName='', fixed=OFF, localCsys=None, name='BC-1',
        region=model.rootAssembly.sets['BC_set'], u1=0.0,
u2=0.0, u3=0.0, ur1=0.0, ur2=0.0, ur3=0.0)


    ############## Beam orientations


    beamvectors=[0]*len(P)
    beamnormals=[0]*len(P)
    for i in xrange(len(P)):
            beamvectors[i]=[0]*3
            beamnormals[i]=[0]*3
    for i in xrange(len(P)+1):
        for j in xrange(1,4):
            beamvectors[i-1][j-1]=M[i-1][1][j-1]-M[i-
1][0][j-1]

    for i in xrange(len(P)):
        if beamvectors[i][0]==0:
            beamnormals[i][0]=1
            beamnormals[i][1]=0
            beamnormals[i][2]=0
        elif beamvectors[i][1]==0:
            beamnormals[i][0]=0
            beamnormals[i][1]=1
            beamnormals[i][2]=0
```

```
        else:
            beamnormals[i][1]=1
            beamnormals[i][2]=1
            beamnormals[i][0]=-
(beamvectors[i][1]*beamnormals[i][1]+beamvectors[i][2]*beam
normals[i][2])/(beamvectors[i][0])


    for i in xrange(len(P)):

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(beamnormals[i]), region=trusspart.sets['Wire-
'+str(i+1)+'-Set-1'])

    ############## Outputs
    model.fieldOutputRequests['F-Output-
1'].setValues(variables=('S', 'U'),frequency=1)
    mdb.models['Model-
1'].HistoryOutputRequest(createStepName='Step-1',
frequency=
        LAST_INCREMENT, name='H-Output-1',
variables=('MASS', ))



    ############## Run job
    mdb.jobs[jobname].submit()
    mdb.jobs[jobname].waitForCompletion()




############################################################
###########################################################

############################################################
###########################################################
    ############## Output

############################################################
###########################################################

############################################################
###########################################################


    jobpath=str(jobname)+".odb"
    odb=openOdb(path=jobpath)
```

```
    lastFrame = odb.steps['Step-1'].frames[-1]

    displacement=lastFrame.fieldOutputs['U']
    fieldValues=displacement.values



length=(len(lastFrame.fieldOutputs['S'].values[0].instance.
elements))
    stresses=[0]*length

    for k in range(length):

stresses[k]=lastFrame.fieldOutputs['S'].values[k].mises
       if stresses[k]>max_stress[t]:
           max_stress[t]=stresses[k]

max_stress_element_label[t]=lastFrame.fieldOutputs['S'].val
ues[k].elementLabel
           max_stress_object_index[t]=k

    maxstore=max_stress[t]
    if t>0:
        num=t-1
        for i in xrange(num):
            if max_stress[i]<maxstore:
                maxstore=max_stress[i]

    if t==0:
       variables[1]=variables[0]
       best=max_stress[0]

    elif 0<t and t<(iterations-1):
       if max_stress[t]<best:
            variables[t]=variables[t]
            best=max_stress[t]
       else:
            variables[t]=store[t]
            best=max_stress[t-1]

    else:
       variables[t]=variables[t]

    file_open.write(str(max_stress[t]) + "\t")
    for i in range(num_var):
        temp=str(variables[t][i])
        file_open.write(temp + "\t")
    file_open.write("\n")
```

```
    odb.close()
file_open.close()
```

APPENDIX 6 – PYTHON SCRIP FOR RESULT OUTPUT FOR OPTIMISED MODEL (LC1, LOAD CASE EXCLUDED)

```
# -*- coding: mbcs -*-
from part import *
from material import *
from section import *
from assembly import *
from step import *
from interaction import *
from load import *
from mesh import *
from optimization import *
from job import *
from sketch import *
from visualization import *
from connectorBehavior import *
from regionToolset import *


import random

half_member_list=[0,3,4,7,8,17,18,19,20,21,22,23,24,33,34,3
6,37,40,41,45,46,47,48,53,56,57,60,61,62,63,64,69,72,73,76,
77,78,79,80,85,88,89,92,93,94,95,96]
count=0


for q in half_member_list:

    iterations=1
    time=1
    numinc=10
    increment=0.1
    rotation=3.14159265359
    load_amplitude=1000

    count=count+1
    jobname="01-"+str(q)

    mdb.Job(atTime=None, contactPrint=OFF, description='',
echoPrint=OFF,
        explicitPrecision=SINGLE,
getMemoryFromAnalysis=True, historyPrint=OFF,
        memory=90, memoryUnits=PERCENTAGE, model='Model-1',
modelPrint=OFF,
```

```
        multiprocessingMode=DEFAULT, name=jobname,
nodalOutputPrecision=SINGLE,
        numCpus=1, numGPUs=1, queue=None, scratch='',
type=ANALYSIS,
        userSubroutine='', waitHours=0, waitMinutes=0)

    H1=-45.50000
    H2=-45.00000
    H13=16.15000
    H14=20.15000
    B12=4.00000

    B1=5.620
    H3=-43.771
    H4=-42.137
    H5=-33.973
    H6=-27.794
    H7=-14.481
    H8=-4.902
    H9=-1.108
    H10=3.798
    H11=12.542
    H12=15.931

    B2=(4-B1)*H3/61.15+4-16.15*(4-B1)/61.15
    B3=(4-B1)*H4/61.15+4-16.15*(4-B1)/61.15
    B4=(4-B1)*H5/61.15+4-16.15*(4-B1)/61.15
    B5=(4-B1)*H6/61.15+4-16.15*(4-B1)/61.15
    B6=(4-B1)*H7/61.15+4-16.15*(4-B1)/61.15
    B7=(4-B1)*H8/61.15+4-16.15*(4-B1)/61.15
    B8=(4-B1)*H9/61.15+4-16.15*(4-B1)/61.15
    B9=(4-B1)*H10/61.15+4-16.15*(4-B1)/61.15
    B10=(4-B1)*H11/61.15+4-16.15*(4-B1)/61.15
    B11=(4-B1)*H12/61.15+4-16.15*(4-B1)/61.15



    CC=[0,0,18.15]
    concrete_mass=666000
    concrete_w=9.6
    concrete_h=4.0
    concrete_Ih=(1.0/12)*concrete_mass*(2*(concrete_w)**2)

concrete_Iw=(1.0/12)*concrete_mass*((concrete_h)**2+(concre
te_w)**2)
    load_point=[0,0,20.15]
    ##
```

```
V0=[H2,0,0]
V1=[H3,0,0]
V2=[H4,0,0]
V3=[H5,0,0]
V4=[H6,0,0]
V5=[H7,0,0]
V6=[H8,0,0]
V7=[H9,0,0]
V8=[H10,0,0]
V9=[H11,0,0]
V10=[H12,0,0]
V11=[H13,0,0]
V12=[B1,0,0]
V13=[B2,0,0]
V14=[B3,0,0]
V15=[B4,0,0]
V16=[B5,0,0]
V17=[B6,0,0]
V18=[B7,0,0]
V19=[B8,0,0]
V20=[B9,0,0]
V21=[B10,0,0]
V22=[B11,0,0]
V23=[B12,0,0]


##
towerh1=20.15
towerh2=21.15
towerh3=32.15
towerh4=42.15
towerh5=54.15
towerh6=64.15
towerh7=74.15
towerh8=83.15
towerh9=88.15
towerconnect=[0,0,20.15]

point_mass_1=(0,0,20.15)
point_mass_2=(0,0,54.15)
point_mass_3=(0,0,88.15)


nacelle_overhang=1.90
hub_overhang=5.0/1

s_overhang=sin(rotation)
c_overhang=cos(rotation)
```

```
    nacelle_cg=(-nacelle_overhang*c_overhang,-
nacelle_overhang*s_overhang,towerh9+1.75)

hub_cg=(hub_overhang*c_overhang,hub_overhang*s_overhang,tow
erh9+2.40)

    nacelle_dummy_wire=(-nacelle_overhang*c_overhang,-
nacelle_overhang*s_overhang,towerh9+1.75+0.05)

hub_dummy_wire=(hub_overhang*c_overhang,hub_overhang*s_over
hang,towerh9+2.40+0.05)


    max_stress=[0]*iterations
    max_stress_element_label=[0]*iterations
    max_stress_object_index=[0]*iterations






    t=0

    ############## Amplitudes

    mdb.models['Model-1'].TabularAmplitude(data=((0.0,
1.0), (0.025, 0.0), (600.0,
        0.0)), name='BC_amp', smooth=SOLVER_DEFAULT,
timeSpan=STEP)

    ##  LC1-fy:
    mdb.models['Model-1'].TabularAmplitude(data=(...),
name='LC1-fy',
        smooth=SOLVER_DEFAULT, timeSpan=STEP)
    ##  LC1-fz:
    mdb.models['Model-1'].TabularAmplitude(data=(...),
name='LC1-fz',
        smooth=SOLVER_DEFAULT, timeSpan=STEP)
    ##  LC1-mx:
    mdb.models['Model-1'].TabularAmplitude(data=(...),
name='LC1-mx',
        smooth=SOLVER_DEFAULT, timeSpan=STEP)
    ##  LC1-my:
    mdb.models['Model-1'].TabularAmplitude(data=(...),
name='LC1-my',
        smooth=SOLVER_DEFAULT, timeSpan=STEP)
    ##  LC1-mz:
```

```
    mdb.models['Model-1'].TabularAmplitude(data=(...),
name='LC1-mz',
        smooth=SOLVER_DEFAULT, timeSpan=STEP)


    model=mdb.models['Model-1']


    ############## Material

    model.Material(name='Steel')

model.materials['Steel'].Elastic(table=((207000000000.0,
0.3),
        ))
    model.materials['Steel'].Density(table=((7850.0, ), ))
    model.materials['Steel'].Damping(alpha=0.01)


    ############## Nodes

    N1=(B1,B1,H1)
    N2=(B1,B1,H2)
    N3=(B2,B2,H3)
    N4=(B3,B3,H4)
    N5=(B5,B5,H6)
    N6=(-B1,B1,H1)
    N7=(-B1,B1,H2)
    N8=(-B2,B2,H3)
    N9=(-B3,B3,H4)
    N10=(-B5,B5,H6)
    N11=(-B1,-B1,H1)
    N12=(-B1,-B1,H2)
    N13=(-B2,-B2,H3)
    N14=(-B3,-B3,H4)
    N15=(-B5,-B5,H6)
    N16=(B1,-B1,H1)
    N17=(B1,-B1,H2)
    N18=(B2,-B2,H3)
    N19=(B3,-B3,H4)
    N20=(B5,-B5,H6)
    N21=(B7,B7,H8)
    N22=(B9,B9,H10)
    N23=(B11,B11,H12)
    N24=(B12,B12,H13)
    N25=(-B7,B7,H8)
    N26=(-B9,B9,H10)
    N27=(-B11,B11,H12)
    N28=(-B12,B12,H13)
```

```
N29=(-B7,-B7,H8)
N30=(-B9,-B9,H10)
N31=(-B11,-B11,H12)
N32=(-B12,-B12,H13)
N33=(B7,-B7,H8)
N34=(B9,-B9,H10)
N35=(B11,-B11,H12)
N36=(B12,-B12,H13)
N37=(B4,0,H5)
N38=(-B4,0,H5)
N39=(0,B4,H5)
N40=(0,-B4,H5)
N41=(B6,0,H7)
N42=(-B6,0,H7)
N43=(0,B6,H7)
N44=(0,-B6,H7)
N45=(B8,0,H9)
N46=(-B8,0,H9)
N47=(0,B8,H9)
N48=(0,-B8,H9)
N49=(B10,0,H11)
N50=(-B10,0,H11)
N51=(0,B10,H11)
N52=(0,-B10,H11)
N53=(B12,B12,H14)
N54=(-B12,B12,H14)
N55=(B12,-B12,H14)
N56=(-B12,-B12,H14)

Nmid1=(B1/2,B1/2,H1/2)
Nmid2=(B1/2,B1/2,H2/2)
Nmid3=(B2/2,B2/2,H3/2)
Nmid4=(B3/2,B3/2,H4/2)
Nmid5=(B5/2,B5/2,H6/2)
Nmid6=(-B1/2,B1/2,H1/2)
Nmid7=(-B1/2,B1/2,H2/2)
Nmid8=(-B2/2,B2/2,H3/2)
Nmid9=(-B3/2,B3/2,H4/2)
Nmid10=(-B5/2,B5/2,H6/2)
Nmid11=(-B1/2,-B1/2,H1/2)
Nmid12=(-B1/2,-B1/2,H2/2)
Nmid13=(-B2/2,-B2/2,H3/2)
Nmid14=(-B3/2,-B3/2,H4/2)
Nmid15=(-B5/2,-B5/2,H6/2)
Nmid16=(B1/2,-B1/2,H1/2)
Nmid17=(B1/2,-B1/2,H2/2)
Nmid18=(B2/2,-B2/2,H3/2)
Nmid19=(B3/2,-B3/2,H4/2)
Nmid20=(B5/2,-B5/2,H6/2)
```

```
Nmid21=(B7/2,B7/2,H8/2)
Nmid22=(B9/2,B9/2,H10/2)
Nmid23=(B11/2,B11/2,H12/2)
Nmid24=(B12/2,B12/2,H13/2)
Nmid25=(-B7/2,B7/2,H8/2)
Nmid26=(-B9/2,B9/2,H10/2)
Nmid27=(-B11/2,B11/2,H12/2)
Nmid28=(-B12/2,B12/2,H13/2)
Nmid29=(-B7/2,-B7/2,H8/2)
Nmid30=(-B9/2,-B9/2,H10/2)
Nmid31=(-B11/2,-B11/2,H12/2)
Nmid32=(-B12/2,-B12/2,H13/2)
Nmid33=(B7/2,-B7/2,H8/2)
Nmid34=(B9/2,-B9/2,H10/2)
Nmid35=(B11/2,-B11/2,H12/2)
Nmid36=(B12/2,-B12/2,H13/2)
Nmid37=(B4/2,0/2,H5/2)
Nmid38=(-B4/2,0/2,H5/2)
Nmid39=(0/2,B4/2,H5/2)
Nmid40=(0/2,-B4/2,H5/2)
Nmid41=(B6/2,0/2,H7/2)
Nmid42=(-B6/2,0/2,H7/2)
Nmid43=(0/2,B6/2,H7/2)
Nmid44=(0/2,-B6/2,H7/2)
Nmid45=(B8/2,0/2,H9/2)
Nmid46=(-B8/2,0/2,H9/2)
Nmid47=(0/2,B8/2,H9/2)
Nmid48=(0/2,-B8/2,H9/2)
Nmid49=(B10/2,0/2,H11/2)
Nmid50=(-B10/2,0/2,H11/2)
Nmid51=(0/2,B10/2,H11/2)
Nmid52=(0/2,-B10/2,H11/2)
Nmid53=(B12/2,B12/2,H14/2)
Nmid54=(-B12/2,B12/2,H14/2)
Nmid55=(B12/2,-B12/2,H14/2)
Nmid56=(-B12/2,-B12/2,H14/2)


M1=(N1,N2)
M2=(N2,N3)
M3=(N3,N4)
M4=(N4,N5)
M5=(N6,N7)
M6=(N7,N8)
M7=(N8,N9)
M8=(N9,N10)
M9=(N11,N12)
M10=(N12,N13)
M11=(N13,N14)
```

```
M12=(N14,N15)
M13=(N16,N17)
M14=(N17,N18)
M15=(N18,N19)
M16=(N19,N20)
M17=(N5,N21)
M18=(N21,N22)
M19=(N22,N23)
M20=(N23,N24)
M21=(N10,N25)
M22=(N25,N26)
M23=(N26,N27)
M24=(N27,N28)
M25=(N15,N29)
M26=(N29,N30)
M27=(N30,N31)
M28=(N31,N32)
M29=(N20,N33)
M30=(N33,N34)
M31=(N34,N35)
M32=(N35,N36)
M33=(N3,N8)
M34=(N8,N13)
M35=(N13,N18)
M36=(N18,N3)
M37=(N4,N37)
M38=(N37,N20)
M39=(N19,N37)
M40=(N37,N5)
M41=(N9,N38)
M42=(N38,N15)
M43=(N14,N38)
M44=(N38,N10)
M45=(N4,N39)
M46=(N39,N10)
M47=(N9,N39)
M48=(N39,N5)
M49=(N19,N40)
M50=(N40,N15)
M51=(N14,N40)
M52=(N40,N20)
M53=(N5,N41)
M54=(N41,N33)
M55=(N20,N41)
M56=(N41,N21)
M57=(N10,N42)
M58=(N42,N29)
M59=(N15,N42)
M60=(N42,N25)
```

```
M61=(N5,N43)
M62=(N43,N25)
M63=(N10,N43)
M64=(N43,N21)
M65=(N20,N44)
M66=(N44,N29)
M67=(N15,N44)
M68=(N44,N33)
M69=(N21,N45)
M70=(N45,N34)
M71=(N33,N45)
M72=(N45,N22)
M73=(N25,N46)
M74=(N46,N30)
M75=(N29,N46)
M76=(N46,N26)
M77=(N21,N47)
M78=(N47,N26)
M79=(N25,N47)
M80=(N47,N22)
M81=(N33,N48)
M82=(N48,N30)
M83=(N29,N48)
M84=(N48,N34)
M85=(N22,N49)
M86=(N49,N35)
M87=(N34,N49)
M88=(N49,N23)
M89=(N26,N50)
M90=(N50,N31)
M91=(N30,N50)
M92=(N50,N27)
M93=(N22,N51)
M94=(N51,N27)
M95=(N26,N51)
M96=(N51,N23)
M97=(N34,N52)
M98=(N52,N31)
M99=(N30,N52)
M100=(N52,N35)
M101=(N24,N53)
M102=(N28,N54)
M103=(N32,N56)
M104=(N36,N55)
```

```
M=[[N1,N2],[N2,N3],[N3,N4],[N4,N5],[N6,N7],[N7,N8],[N8,N9],
[N9,N10],[N11,N12],[N12,N13],[N13,N14],[N14,N15],[N16,N17],
```

```
[N17,N18],[N18,N19],[N19,N20],[N5,N21],[N21,N22],[N22,N23],
[N23,N24],[N10,N25],[N25,N26],[N26,N27],[N27,N28],[N15,N29]
,

[N29,N30],[N30,N31],[N31,N32],[N20,N33],[N33,N34],[N34,N35]
,[N35,N36],[N3,N8],[N8,N13],[N13,N18],[N18,N3],[N4,N37],[N3
7,N20],

[N19,N37],[N37,N5],[N9,N38],[N38,N15],[N14,N38],[N38,N10],[
N4,N39],[N39,N10],[N9,N39],[N39,N5],[N19,N40],[N40,N15],[N1
4,N40],

[N40,N20],[N5,N41],[N41,N33],[N20,N41],[N41,N21],[N10,N42],
[N42,N29],[N15,N42],[N42,N25],[N5,N43],[N43,N25],[N10,N43],
[N43,N21],

[N20,N44],[N44,N29],[N15,N44],[N44,N33],[N21,N45],[N45,N34]
,[N33,N45],[N45,N22],[N25,N46],[N46,N30],[N29,N46],[N46,N26
],

[N21,N47],[N47,N26],[N25,N47],[N47,N22],[N33,N48],[N48,N30]
,[N29,N48],[N48,N34],[N22,N49],[N49,N35],[N34,N49],[N49,N23
],

[N26,N50],[N50,N31],[N30,N50],[N50,N27],[N22,N51],[N51,N27]
,[N26,N51],[N51,N23],[N34,N52],[N52,N31],[N30,N52],[N52,N35
],
       [N24,N53],[N28,N54],[N32,N56],[N36,N55]]


Mmid=[[Nmid1,Nmid2],[Nmid2,Nmid3],[Nmid3,Nmid4],[Nmid4,Nmid
5],[Nmid6,Nmid7],[Nmid7,Nmid8],[Nmid8,Nmid9],[Nmid9,Nmid10]
,

[Nmid11,Nmid12],[Nmid12,Nmid13],[Nmid13,Nmid14],[Nmid14,Nmi
d15],[Nmid16,Nmid17],[Nmid17,Nmid18],[Nmid18,Nmid19],

[Nmid19,Nmid20],[Nmid5,Nmid21],[Nmid21,Nmid22],[Nmid22,Nmid
23],[Nmid23,Nmid24],[Nmid10,Nmid25],[Nmid25,Nmid26],

[Nmid26,Nmid27],[Nmid27,Nmid28],[Nmid15,Nmid29],[Nmid29,Nmi
d30],[Nmid30,Nmid31],[Nmid31,Nmid32],[Nmid20,Nmid33],

[Nmid33,Nmid34],[Nmid34,Nmid35],[Nmid35,Nmid36],[Nmid3,Nmid
8],[Nmid8,Nmid13],[Nmid13,Nmid18],[Nmid18,Nmid3],[Nmid4,Nmi
d37],

[Nmid37,Nmid20],[Nmid19,Nmid37],[Nmid37,Nmid5],[Nmid9,Nmid3
```

```
8],[Nmid38,Nmid15],[Nmid14,Nmid38],[Nmid38,Nmid10],[Nmid4,N
mid39],

[Nmid39,Nmid10],[Nmid9,Nmid39],[Nmid39,Nmid5],[Nmid19,Nmid4
0],[Nmid40,Nmid15],[Nmid14,Nmid40],[Nmid40,Nmid20],[Nmid5,N
mid41],

[Nmid41,Nmid33],[Nmid20,Nmid41],[Nmid41,Nmid21],[Nmid10,Nmi
d42],[Nmid42,Nmid29],[Nmid15,Nmid42],[Nmid42,Nmid25],

[Nmid5,Nmid43],[Nmid43,Nmid25],[Nmid10,Nmid43],[Nmid43,Nmid
21],[Nmid20,Nmid44],[Nmid44,Nmid29],[Nmid15,Nmid44],[Nmid44
,Nmid33],

[Nmid21,Nmid45],[Nmid45,Nmid34],[Nmid33,Nmid45],[Nmid45,Nmi
d22],[Nmid25,Nmid46],[Nmid46,Nmid30],[Nmid29,Nmid46],[Nmid4
6,Nmid26],

[Nmid21,Nmid47],[Nmid47,Nmid26],[Nmid25,Nmid47],[Nmid47,Nmi
d22],[Nmid33,Nmid48],[Nmid48,Nmid30],[Nmid29,Nmid48],[Nmid4
8,Nmid34],

[Nmid22,Nmid49],[Nmid49,Nmid35],[Nmid34,Nmid49],[Nmid49,Nmi
d23],[Nmid26,Nmid50],[Nmid50,Nmid31],[Nmid30,Nmid50],[Nmid5
0,Nmid27],

[Nmid22,Nmid51],[Nmid51,Nmid27],[Nmid26,Nmid51],[Nmid51,Nmi
d23],[Nmid34,Nmid52],[Nmid52,Nmid31],[Nmid30,Nmid52],[Nmid5
2,Nmid35],

[Nmid24,Nmid53],[Nmid28,Nmid54],[Nmid32,Nmid56],[Nmid36,Nmi
d55]]


    ############## Datum points

    model.Part(dimensionality=THREE_D, name='Part-1',
type=DEFORMABLE_BODY)

    trusspart=model.parts['Part-1']



    trusspart.DatumPointByCoordinate(coords=(B1, B1, H1))
    trusspart.DatumPointByCoordinate(coords=(B1, B1, H2))
    trusspart.DatumPointByCoordinate(coords=(B2, B2, H3))
    trusspart.DatumPointByCoordinate(coords=(B3, B3, H4))
    trusspart.DatumPointByCoordinate(coords=(B5, B5, H6))
    trusspart.DatumPointByCoordinate(coords=(-B1, B1, H1))
```

```
    trusspart.DatumPointByCoordinate(coords=(-B1, B1, H2))
    trusspart.DatumPointByCoordinate(coords=(-B2, B2, H3))
    trusspart.DatumPointByCoordinate(coords=(-B3, B3, H4))
    trusspart.DatumPointByCoordinate(coords=(-B5, B5, H6))
    trusspart.DatumPointByCoordinate(coords=(-B1, -B1, H1))
    trusspart.DatumPointByCoordinate(coords=(-B1, -B1, H2))
    trusspart.DatumPointByCoordinate(coords=(-B2, -B2, H3))
    trusspart.DatumPointByCoordinate(coords=(-B3, -B3, H4))
    trusspart.DatumPointByCoordinate(coords=(-B5, -B5, H6))
    trusspart.DatumPointByCoordinate(coords=(B1, -B1, H1))
    trusspart.DatumPointByCoordinate(coords=(B1, -B1, H2))
    trusspart.DatumPointByCoordinate(coords=(B2, -B2, H3))
    trusspart.DatumPointByCoordinate(coords=(B3, -B3, H4))
    trusspart.DatumPointByCoordinate(coords=(B5, -B5, H6))
    trusspart.DatumPointByCoordinate(coords=(B7, B7, H8))
    trusspart.DatumPointByCoordinate(coords=(B9, B9, H10))
    trusspart.DatumPointByCoordinate(coords=(B11, B11,
H12))
    trusspart.DatumPointByCoordinate(coords=(B12, B12,
H13))
    trusspart.DatumPointByCoordinate(coords=(-B7, B7, H8))
    trusspart.DatumPointByCoordinate(coords=(-B9, B9, H10))
    trusspart.DatumPointByCoordinate(coords=(-B11, B11,
H12))
    trusspart.DatumPointByCoordinate(coords=(-B12, B12,
H13))
    trusspart.DatumPointByCoordinate(coords=(-B7, -B7, H8))
    trusspart.DatumPointByCoordinate(coords=(-B9, -B9,
H10))
    trusspart.DatumPointByCoordinate(coords=(-B11, -B11,
H12))
    trusspart.DatumPointByCoordinate(coords=(-B12, -B12,
H13))
    trusspart.DatumPointByCoordinate(coords=(B7, -B7, H8))
    trusspart.DatumPointByCoordinate(coords=(B9, -B9, H10))
    trusspart.DatumPointByCoordinate(coords=(B11, -B11,
H12))
    trusspart.DatumPointByCoordinate(coords=(B12, -B12,
H13))
    trusspart.DatumPointByCoordinate(coords=(B4, 0, H5))
    trusspart.DatumPointByCoordinate(coords=(-B4, 0, H5))
    trusspart.DatumPointByCoordinate(coords=(0, B4, H5))
    trusspart.DatumPointByCoordinate(coords=(0, -B4, H5))
    trusspart.DatumPointByCoordinate(coords=(B6, 0, H7))
    trusspart.DatumPointByCoordinate(coords=(-B6, 0, H7))
    trusspart.DatumPointByCoordinate(coords=(0, B6, H7))
    trusspart.DatumPointByCoordinate(coords=(0, -B6, H7))
    trusspart.DatumPointByCoordinate(coords=(B8, 0, H9))
    trusspart.DatumPointByCoordinate(coords=(-B8, 0, H9))
```

```
    trusspart.DatumPointByCoordinate(coords=(0, B8, H9))
    trusspart.DatumPointByCoordinate(coords=(0, -B8, H9))
    trusspart.DatumPointByCoordinate(coords=(B10, 0, H11))
    trusspart.DatumPointByCoordinate(coords=(-B10, 0, H11))
    trusspart.DatumPointByCoordinate(coords=(0, B10, H11))
    trusspart.DatumPointByCoordinate(coords=(0, -B10, H11))
    trusspart.DatumPointByCoordinate(coords=(B12, B12,
H14))
    trusspart.DatumPointByCoordinate(coords=(-B12, B12,
H14))
    trusspart.DatumPointByCoordinate(coords=(B12, -B12,
H14))
    trusspart.DatumPointByCoordinate(coords=(-B12, -B12,
H14))


concrete_center=trusspart.DatumPointByCoordinate(coords=(CC
))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh1))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh2))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh3))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh4))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh5))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh6))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh7))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh8))
    trusspart.DatumPointByCoordinate(coords=(0, 0,
towerh9))


rotor_center=trusspart.DatumPointByCoordinate(coords=(hub_c
g))

nacelle_center=trusspart.DatumPointByCoordinate(coords=(nac
elle_cg))

rotor_center=trusspart.DatumPointByCoordinate(coords=(hub_d
ummy_wire))

nacelle_center=trusspart.DatumPointByCoordinate(coords=(nac
elle_dummy_wire))
```

```
    CC_id=concrete_center.id

    ############## Nodenames
    trusspart.features.changeKey(fromName='Datum pt-1',toName='N1')
    trusspart.features.changeKey(fromName='Datum pt-2',toName='N2')
    trusspart.features.changeKey(fromName='Datum pt-3',toName='N3')
    trusspart.features.changeKey(fromName='Datum pt-4',toName='N4')
    trusspart.features.changeKey(fromName='Datum pt-5',toName='N5')
    trusspart.features.changeKey(fromName='Datum pt-6',toName='N6')
    trusspart.features.changeKey(fromName='Datum pt-7',toName='N7')
    trusspart.features.changeKey(fromName='Datum pt-8',toName='N8')
    trusspart.features.changeKey(fromName='Datum pt-9',toName='N9')
    trusspart.features.changeKey(fromName='Datum pt-10',toName='N10')
    trusspart.features.changeKey(fromName='Datum pt-11',toName='N11')
    trusspart.features.changeKey(fromName='Datum pt-12',toName='N12')
    trusspart.features.changeKey(fromName='Datum pt-13',toName='N13')
    trusspart.features.changeKey(fromName='Datum pt-14',toName='N14')
    trusspart.features.changeKey(fromName='Datum pt-15',toName='N15')
    trusspart.features.changeKey(fromName='Datum pt-16',toName='N16')
    trusspart.features.changeKey(fromName='Datum pt-17',toName='N17')
    trusspart.features.changeKey(fromName='Datum pt-18',toName='N18')
    trusspart.features.changeKey(fromName='Datum pt-19',toName='N19')
    trusspart.features.changeKey(fromName='Datum pt-20',toName='N20')
    trusspart.features.changeKey(fromName='Datum pt-21',toName='N21')
```

```
    trusspart.features.changeKey(fromName='Datum pt-
22',toName='N22')
    trusspart.features.changeKey(fromName='Datum pt-
23',toName='N23')
    trusspart.features.changeKey(fromName='Datum pt-
24',toName='N24')
    trusspart.features.changeKey(fromName='Datum pt-
25',toName='N25')
    trusspart.features.changeKey(fromName='Datum pt-
26',toName='N26')
    trusspart.features.changeKey(fromName='Datum pt-
27',toName='N27')
    trusspart.features.changeKey(fromName='Datum pt-
28',toName='N28')
    trusspart.features.changeKey(fromName='Datum pt-
29',toName='N29')
    trusspart.features.changeKey(fromName='Datum pt-
30',toName='N30')
    trusspart.features.changeKey(fromName='Datum pt-
31',toName='N31')
    trusspart.features.changeKey(fromName='Datum pt-
32',toName='N32')
    trusspart.features.changeKey(fromName='Datum pt-
33',toName='N33')
    trusspart.features.changeKey(fromName='Datum pt-
34',toName='N34')
    trusspart.features.changeKey(fromName='Datum pt-
35',toName='N35')
    trusspart.features.changeKey(fromName='Datum pt-
36',toName='N36')
    trusspart.features.changeKey(fromName='Datum pt-
37',toName='N37')
    trusspart.features.changeKey(fromName='Datum pt-
38',toName='N38')
    trusspart.features.changeKey(fromName='Datum pt-
39',toName='N39')
    trusspart.features.changeKey(fromName='Datum pt-
40',toName='N40')
    trusspart.features.changeKey(fromName='Datum pt-
41',toName='N41')
    trusspart.features.changeKey(fromName='Datum pt-
42',toName='N42')
    trusspart.features.changeKey(fromName='Datum pt-
43',toName='N43')
    trusspart.features.changeKey(fromName='Datum pt-
44',toName='N44')
    trusspart.features.changeKey(fromName='Datum pt-
45',toName='N45')
```

```
    trusspart.features.changeKey(fromName='Datum pt-
46',toName='N47')
    trusspart.features.changeKey(fromName='Datum pt-
47',toName='N48')
    trusspart.features.changeKey(fromName='Datum pt-
48',toName='N46')
    trusspart.features.changeKey(fromName='Datum pt-
49',toName='N49')
    trusspart.features.changeKey(fromName='Datum pt-
50',toName='N50')
    trusspart.features.changeKey(fromName='Datum pt-
51',toName='N51')
    trusspart.features.changeKey(fromName='Datum pt-
52',toName='N52')
    trusspart.features.changeKey(fromName='Datum pt-
53',toName='N53')
    trusspart.features.changeKey(fromName='Datum pt-
54',toName='N54')
    trusspart.features.changeKey(fromName='Datum pt-
55',toName='N55')
    trusspart.features.changeKey(fromName='Datum pt-
56',toName='N56')




############## Wires

    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[1],trusspart.datums[2
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[2],trusspart.datums[3
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[3],trusspart.datums[4
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[4],trusspart.datums[5
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[6],trusspart.datums[7
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[7],trusspart.datums[8
]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[8],trusspart.datums[9
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[9],trusspart.datums[1
0]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[11],trusspart.datums[
12]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[12],trusspart.datums[
13]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[13],trusspart.datums[
14]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[14],trusspart.datums[
15]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[16],trusspart.datums[
17]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[17],trusspart.datums[
18]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[18],trusspart.datums[
19]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[19],trusspart.datums[
20]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[5],trusspart.datums[2
1]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[21],trusspart.datums[
22]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[22],trusspart.datums[
23]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[23],trusspart.datums[
24]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[10],trusspart.datums[
25]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[25],trusspart.datums[
26]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[26],trusspart.datums[
27]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[27],trusspart.datums[
28]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[15],trusspart.datums[
29]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[29],trusspart.datums[
30]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[30],trusspart.datums[
31]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[31],trusspart.datums[
32]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[20],trusspart.datums[
33]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[33],trusspart.datums[
34]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[34],trusspart.datums[
35]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[35],trusspart.datums[
36]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[3],trusspart.datums[8
]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[8],trusspart.datums[1
3]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[13],trusspart.datums[
18]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[18],trusspart.datums[
3]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[4],trusspart.datums[3
7]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[37],trusspart.datums[
20]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[19],trusspart.datums[
37]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[37],trusspart.datums[
5]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[9],trusspart.datums[3
8]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[38],trusspart.datums[
15]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[14],trusspart.datums[
38]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[38],trusspart.datums[
10]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[4],trusspart.datums[3
9]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[39],trusspart.datums[
10]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[9],trusspart.datums[3
9]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[39],trusspart.datums[
5]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[19],trusspart.datums[
40]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[40],trusspart.datums[
15]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[14],trusspart.datums[
40]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[40],trusspart.datums[
20]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[5],trusspart.datums[4
1]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[41],trusspart.datums[
33]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[20],trusspart.datums[
41]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[41],trusspart.datums[
21]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[10],trusspart.datums[
42]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[42],trusspart.datums[
29]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[15],trusspart.datums[
42]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[42],trusspart.datums[
25]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[5],trusspart.datums[4
3]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[43],trusspart.datums[
25]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[10],trusspart.datums[
43]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[43],trusspart.datums[
21]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[20],trusspart.datums[
44]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[44],trusspart.datums[
29]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[15],trusspart.datums[
44]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[44],trusspart.datums[
33]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[21],trusspart.datums[
45]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[45],trusspart.datums[
34]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[33],trusspart.datums[
45]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[45],trusspart.datums[
22]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[25],trusspart.datums[
46]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[46],trusspart.datums[
30]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[29],trusspart.datums[
46]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[46],trusspart.datums[
26]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[21],trusspart.datums[
47]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[47],trusspart.datums[
26]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[25],trusspart.datums[
47]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[47],trusspart.datums[
22]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[33],trusspart.datums[
48]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[48],trusspart.datums[
30]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[29],trusspart.datums[
48]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[48],trusspart.datums[
34]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[22],trusspart.datums[
49]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[49],trusspart.datums[
35]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[34],trusspart.datums[
49]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[49],trusspart.datums[
23]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[26],trusspart.datums[
50]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[50],trusspart.datums[
31]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[30],trusspart.datums[
50]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[50],trusspart.datums[
27]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[22],trusspart.datums[
51]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[51],trusspart.datums[
27]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[26],trusspart.datums[
51]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[51],trusspart.datums[
23]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[34],trusspart.datums[
52]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[52],trusspart.datums[
31]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[30],trusspart.datums[
52]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[52],trusspart.datums[
35]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[24],trusspart.datums[
53]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[28],trusspart.datums[
54]), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[32],trusspart.datums[
56]), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[36],trusspart.datums[
55]), ))

    ##trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=((trusspart.datums[CC_id], (0.0, 0.0,
towerh1)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh1), (0.0, 0.0,
towerh2)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh2), (0.0, 0.0,
towerh3)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh3), (0.0, 0.0,
towerh4)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh4), (0.0, 0.0,
towerh5)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh5), (0.0, 0.0,
towerh6)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh6), (0.0, 0.0,
towerh7)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh7), (0.0, 0.0,
towerh8)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0.0, 0.0, towerh8), (0.0, 0.0,
towerh9)), ))




    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((-nacelle_overhang*c_overhang,-
nacelle_overhang*s_overhang,towerh9+1.75), (-
nacelle_overhang*c_overhang,-
nacelle_overhang*s_overhang,towerh9+1.75+.05)), ))
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((hub_overhang*c_overhang,hub_overhang*
s_overhang,towerh9+2.40),
(hub_overhang*c_overhang,hub_overhang*s_overhang,towerh9+2.
40+0.05)), ))
```

```
    trusspart.WirePolyLine(mergeWire=OFF,
meshable=ON,points=(((0,0,18.15), (0,0,18.16)), ))

    ############## Wire set creation

f=[[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[
N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],
[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1]
,[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1
],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N
1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[
N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],
[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1]
,[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],[N1],]
    f[0][0]=tuple(map(sum,zip(Mmid[0][0],Mmid[0][1])))
    f[1][0]=tuple(map(sum,zip(Mmid[1][0],Mmid[1][1])))
    f[2][0]=tuple(map(sum,zip(Mmid[2][0],Mmid[2][1])))
    f[3][0]=tuple(map(sum,zip(Mmid[3][0],Mmid[3][1])))
    f[4][0]=tuple(map(sum,zip(Mmid[4][0],Mmid[4][1])))
    f[5][0]=tuple(map(sum,zip(Mmid[5][0],Mmid[5][1])))
    f[6][0]=tuple(map(sum,zip(Mmid[6][0],Mmid[6][1])))
    f[7][0]=tuple(map(sum,zip(Mmid[7][0],Mmid[7][1])))
    f[8][0]=tuple(map(sum,zip(Mmid[8][0],Mmid[8][1])))
    f[9][0]=tuple(map(sum,zip(Mmid[9][0],Mmid[9][1])))
    f[10][0]=tuple(map(sum,zip(Mmid[10][0],Mmid[10][1])))
    f[11][0]=tuple(map(sum,zip(Mmid[11][0],Mmid[11][1])))
    f[12][0]=tuple(map(sum,zip(Mmid[12][0],Mmid[12][1])))
    f[13][0]=tuple(map(sum,zip(Mmid[13][0],Mmid[13][1])))
    f[14][0]=tuple(map(sum,zip(Mmid[14][0],Mmid[14][1])))
    f[15][0]=tuple(map(sum,zip(Mmid[15][0],Mmid[15][1])))
    f[16][0]=tuple(map(sum,zip(Mmid[16][0],Mmid[16][1])))
    f[17][0]=tuple(map(sum,zip(Mmid[17][0],Mmid[17][1])))
    f[18][0]=tuple(map(sum,zip(Mmid[18][0],Mmid[18][1])))
    f[19][0]=tuple(map(sum,zip(Mmid[19][0],Mmid[19][1])))
    f[20][0]=tuple(map(sum,zip(Mmid[20][0],Mmid[20][1])))
    f[21][0]=tuple(map(sum,zip(Mmid[21][0],Mmid[21][1])))
    f[22][0]=tuple(map(sum,zip(Mmid[22][0],Mmid[22][1])))
    f[23][0]=tuple(map(sum,zip(Mmid[23][0],Mmid[23][1])))
    f[24][0]=tuple(map(sum,zip(Mmid[24][0],Mmid[24][1])))
    f[25][0]=tuple(map(sum,zip(Mmid[25][0],Mmid[25][1])))
    f[26][0]=tuple(map(sum,zip(Mmid[26][0],Mmid[26][1])))
    f[27][0]=tuple(map(sum,zip(Mmid[27][0],Mmid[27][1])))
    f[28][0]=tuple(map(sum,zip(Mmid[28][0],Mmid[28][1])))
    f[29][0]=tuple(map(sum,zip(Mmid[29][0],Mmid[29][1])))
    f[30][0]=tuple(map(sum,zip(Mmid[30][0],Mmid[30][1])))
    f[31][0]=tuple(map(sum,zip(Mmid[31][0],Mmid[31][1])))
    f[32][0]=tuple(map(sum,zip(Mmid[32][0],Mmid[32][1])))
```

```
f[33][0]=tuple(map(sum,zip(Mmid[33][0],Mmid[33][1])))
f[34][0]=tuple(map(sum,zip(Mmid[34][0],Mmid[34][1])))
f[35][0]=tuple(map(sum,zip(Mmid[35][0],Mmid[35][1])))
f[36][0]=tuple(map(sum,zip(Mmid[36][0],Mmid[36][1])))
f[37][0]=tuple(map(sum,zip(Mmid[37][0],Mmid[37][1])))
f[38][0]=tuple(map(sum,zip(Mmid[38][0],Mmid[38][1])))
f[39][0]=tuple(map(sum,zip(Mmid[39][0],Mmid[39][1])))
f[40][0]=tuple(map(sum,zip(Mmid[40][0],Mmid[40][1])))
f[41][0]=tuple(map(sum,zip(Mmid[41][0],Mmid[41][1])))
f[42][0]=tuple(map(sum,zip(Mmid[42][0],Mmid[42][1])))
f[43][0]=tuple(map(sum,zip(Mmid[43][0],Mmid[43][1])))
f[44][0]=tuple(map(sum,zip(Mmid[44][0],Mmid[44][1])))
f[45][0]=tuple(map(sum,zip(Mmid[45][0],Mmid[45][1])))
f[46][0]=tuple(map(sum,zip(Mmid[46][0],Mmid[46][1])))
f[47][0]=tuple(map(sum,zip(Mmid[47][0],Mmid[47][1])))
f[48][0]=tuple(map(sum,zip(Mmid[48][0],Mmid[48][1])))
f[49][0]=tuple(map(sum,zip(Mmid[49][0],Mmid[49][1])))
f[50][0]=tuple(map(sum,zip(Mmid[50][0],Mmid[50][1])))
f[51][0]=tuple(map(sum,zip(Mmid[51][0],Mmid[51][1])))
f[52][0]=tuple(map(sum,zip(Mmid[52][0],Mmid[52][1])))
f[53][0]=tuple(map(sum,zip(Mmid[53][0],Mmid[53][1])))
f[54][0]=tuple(map(sum,zip(Mmid[54][0],Mmid[54][1])))
f[55][0]=tuple(map(sum,zip(Mmid[55][0],Mmid[55][1])))
f[56][0]=tuple(map(sum,zip(Mmid[56][0],Mmid[56][1])))
f[57][0]=tuple(map(sum,zip(Mmid[57][0],Mmid[57][1])))
f[58][0]=tuple(map(sum,zip(Mmid[58][0],Mmid[58][1])))
f[59][0]=tuple(map(sum,zip(Mmid[59][0],Mmid[59][1])))
f[60][0]=tuple(map(sum,zip(Mmid[60][0],Mmid[60][1])))
f[61][0]=tuple(map(sum,zip(Mmid[61][0],Mmid[61][1])))
f[62][0]=tuple(map(sum,zip(Mmid[62][0],Mmid[62][1])))
f[63][0]=tuple(map(sum,zip(Mmid[63][0],Mmid[63][1])))
f[64][0]=tuple(map(sum,zip(Mmid[64][0],Mmid[64][1])))
f[65][0]=tuple(map(sum,zip(Mmid[65][0],Mmid[65][1])))
f[66][0]=tuple(map(sum,zip(Mmid[66][0],Mmid[66][1])))
f[67][0]=tuple(map(sum,zip(Mmid[67][0],Mmid[67][1])))
f[68][0]=tuple(map(sum,zip(Mmid[68][0],Mmid[68][1])))
f[69][0]=tuple(map(sum,zip(Mmid[69][0],Mmid[69][1])))
f[70][0]=tuple(map(sum,zip(Mmid[70][0],Mmid[70][1])))
f[71][0]=tuple(map(sum,zip(Mmid[71][0],Mmid[71][1])))
f[72][0]=tuple(map(sum,zip(Mmid[72][0],Mmid[72][1])))
f[73][0]=tuple(map(sum,zip(Mmid[73][0],Mmid[73][1])))
f[74][0]=tuple(map(sum,zip(Mmid[74][0],Mmid[74][1])))
f[75][0]=tuple(map(sum,zip(Mmid[75][0],Mmid[75][1])))
f[76][0]=tuple(map(sum,zip(Mmid[76][0],Mmid[76][1])))
f[77][0]=tuple(map(sum,zip(Mmid[77][0],Mmid[77][1])))
f[78][0]=tuple(map(sum,zip(Mmid[78][0],Mmid[78][1])))
f[79][0]=tuple(map(sum,zip(Mmid[79][0],Mmid[79][1])))
f[80][0]=tuple(map(sum,zip(Mmid[80][0],Mmid[80][1])))
f[81][0]=tuple(map(sum,zip(Mmid[81][0],Mmid[81][1])))
```

```
    f[82][0]=tuple(map(sum,zip(Mmid[82][0],Mmid[82][1])))
    f[83][0]=tuple(map(sum,zip(Mmid[83][0],Mmid[83][1])))
    f[84][0]=tuple(map(sum,zip(Mmid[84][0],Mmid[84][1])))
    f[85][0]=tuple(map(sum,zip(Mmid[85][0],Mmid[85][1])))
    f[86][0]=tuple(map(sum,zip(Mmid[86][0],Mmid[86][1])))
    f[87][0]=tuple(map(sum,zip(Mmid[87][0],Mmid[87][1])))
    f[88][0]=tuple(map(sum,zip(Mmid[88][0],Mmid[88][1])))
    f[89][0]=tuple(map(sum,zip(Mmid[89][0],Mmid[89][1])))
    f[90][0]=tuple(map(sum,zip(Mmid[90][0],Mmid[90][1])))
    f[91][0]=tuple(map(sum,zip(Mmid[91][0],Mmid[91][1])))
    f[92][0]=tuple(map(sum,zip(Mmid[92][0],Mmid[92][1])))
    f[93][0]=tuple(map(sum,zip(Mmid[93][0],Mmid[93][1])))
    f[94][0]=tuple(map(sum,zip(Mmid[94][0],Mmid[94][1])))
    f[95][0]=tuple(map(sum,zip(Mmid[95][0],Mmid[95][1])))
    f[96][0]=tuple(map(sum,zip(Mmid[96][0],Mmid[96][1])))
    f[97][0]=tuple(map(sum,zip(Mmid[97][0],Mmid[97][1])))
    f[98][0]=tuple(map(sum,zip(Mmid[98][0],Mmid[98][1])))
    f[99][0]=tuple(map(sum,zip(Mmid[99][0],Mmid[99][1])))

f[100][0]=tuple(map(sum,zip(Mmid[100][0],Mmid[100][1])))

f[101][0]=tuple(map(sum,zip(Mmid[101][0],Mmid[101][1])))

f[102][0]=tuple(map(sum,zip(Mmid[102][0],Mmid[102][1])))

f[103][0]=tuple(map(sum,zip(Mmid[103][0],Mmid[103][1])))


    ############## Sets

P=[2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,3,3,
3,3,3,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,4,4,4,4]
    length=len(P)
    len2=length+1


trusspart.Set(edges=trusspart.edges.getSequenceFromMask(('[
#ffffffff:3 #ff ]', ), ), name='All_set')
    for i in range(1,len2):
        trusspart.Set(edges=trusspart.edges.findAt(((f[i-
1][0]),),), name='Wire-'+str(i)+'-Set-1')
        wire_region=trusspart.edges.findAt(((f[i-1][0]),),)
        globals()['wire%s' % i] = wire_region
    trusspart.Set(edges=trusspart.edges.findAt([CC]),
name='Concrete_help_wire')
```

```
    ############## Sections
    waterdensity=1000

    radius1=0.4
    radius2=0.6
    radius3=0.6
    radius4=0.6

    thickness1=0.02
    thickness2=0.05
    thickness3=0.035
    thickness4=0.04

    model.PipeProfile(name='Dim1', r=radius1, t=thickness1)
    model.PipeProfile(name='Dim2', r=radius2, t=thickness2)
    model.PipeProfile(name='Dim3', r=radius3, t=thickness3)
    model.PipeProfile(name='Dim4', r=radius4, t=thickness4)


    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Braces', poissonRatio=0.3, profile='Dim1',
temperatureVar=LINEAR)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius1, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Braces_submerged', poissonRatio=0.3, profile='Dim1',
temperatureVar=LINEAR, useFluidInertia=ON)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius1, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Braces_half_submerged', poissonRatio=0.3,
profile='Dim1', submerged=HALF, temperatureVar=LINEAR,
useFluidInertia=ON)

    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Lower_leg', poissonRatio=0.3, profile='Dim2',
temperatureVar=LINEAR)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius2, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Lower_leg_submerged', poissonRatio=0.3,
profile='Dim2', temperatureVar=LINEAR, useFluidInertia=ON)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
```

```
dius=radius2, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Lower_leg_half_submerged', poissonRatio=0.3,
profile='Dim2', submerged=HALF, temperatureVar=LINEAR,
useFluidInertia=ON)

    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Upper_leg', poissonRatio=0.3, profile='Dim3',
temperatureVar=LINEAR)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius3, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Upper_leg_submerged', poissonRatio=0.3,
profile='Dim3', temperatureVar=LINEAR, useFluidInertia=ON)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius3, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Upper_leg_half_submerged', poissonRatio=0.3,
profile='Dim3', submerged=HALF, temperatureVar=LINEAR,
useFluidInertia=ON)

    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel', name='Top',
poissonRatio=0.3, profile='Dim4', temperatureVar=LINEAR)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius4, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Top_submerged', poissonRatio=0.3, profile='Dim4',
temperatureVar=LINEAR, useFluidInertia=ON)

model.BeamSection(consistentMassMatrix=False,crossSectionRa
dius=radius4, fluidMassDensity=waterdensity,
integration=DURING_ANALYSIS, material='Steel',
name='Top_half__submerged', poissonRatio=0.3,
profile='Dim4', submerged=HALF, temperatureVar=LINEAR,
useFluidInertia=ON)



    ################ Tower sections

    model.PipeProfile(name='Tower2', r=5.577/2, t=0.032)
    model.PipeProfile(name='Tower3', r=5.318/2, t=0.030)
    model.PipeProfile(name='Tower4', r=5.082/2, t=0.028)
    model.PipeProfile(name='Tower5', r=4.800/2, t=0.024)
```

```
    model.PipeProfile(name='Tower6', r=4.565/2, t=0.024)
    model.PipeProfile(name='Tower7', r=4.329/2, t=0.022)
    model.PipeProfile(name='Tower8', r=4.118/2, t=0.030)
    model.PipeProfile(name='Tower9', r=4.000/2, t=0.030)


    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_2', poissonRatio=0.3, profile='Tower2',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_3', poissonRatio=0.3, profile='Tower3',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_4', poissonRatio=0.3, profile='Tower4',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_5', poissonRatio=0.3, profile='Tower5',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_6', poissonRatio=0.3, profile='Tower6',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_7', poissonRatio=0.3, profile='Tower7',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_8', poissonRatio=0.3, profile='Tower8',
temperatureVar=LINEAR)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Tower_section_9', poissonRatio=0.3, profile='Tower9',
temperatureVar=LINEAR)


    towermid0=(0,0,(18.15+towerh1)/2)
    towermid1=(0,0,(towerh1+towerh2)/2)
    towermid2=(0,0,(towerh2+towerh3)/2)
    towermid3=(0,0,(towerh3+towerh4)/2)
    towermid4=(0,0,(towerh4+towerh5)/2)
    towermid5=(0,0,(towerh5+towerh6)/2)
    towermid6=(0,0,(towerh6+towerh7)/2)
    towermid7=(0,0,(towerh7+towerh8)/2)
    towermid8=(0,0,(towerh8+towerh9)/2)
```

```
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid1])),
sectionName='Tower_section_2',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid2])),
sectionName='Tower_section_3',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid3])),
sectionName='Tower_section_4',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid4])),
sectionName='Tower_section_5',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid5])),
sectionName='Tower_section_6',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid6])),
sectionName='Tower_section_7',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid7])),
sectionName='Tower_section_8',thicknessAssignment=FROM_SECT
ION)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([towermid8])),
sectionName='Tower_section_9',thicknessAssignment=FROM_SECT
ION)



trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid1])))
```

```
trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid2])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid3])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid4])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid5])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid6])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid7])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([towermid8])))


    ################ Dummy sections
    Concrete_center=[0,0,18.155]
    model.PipeProfile(name='Dummy', r=0.1, t=0.010)
    model.BeamSection(consistentMassMatrix=False,
integration=DURING_ANALYSIS, material='Steel',
name='Dummy_section', poissonRatio=0.3, profile='Dummy',
temperatureVar=LINEAR)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([nacelle_cg])),
sectionName='Dummy_section',thicknessAssignment=FROM_SECTIO
N)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
region=Region(edges=trusspart.edges.findAt([hub_cg])),
sectionName='Dummy_section',thicknessAssignment=FROM_SECTIO
N)
    trusspart.SectionAssignment(offset=0.0, offsetField='',
offsetType=MIDDLE_SURFACE,
```

```
region=Region(edges=trusspart.edges.findAt([Concrete_center
])),
sectionName='Dummy_section',thicknessAssignment=FROM_SECTIO
N)

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([nacelle_cg])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([hub_cg])))

trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(0.0, 1.0, 0.0),
region=Region(edges=trusspart.edges.findAt([Concrete_center
])))

    ################ Steps

    model.ImplicitDynamicsStep(initialInc=0.025,
maxNumInc=24001, name='Dynamic', noStop=OFF, nohaf=OFF,
previous='Initial', timeIncrementationMethod=FIXED,
timePeriod=600.0)


    ############## Properties

    S=[0]*len(P)
    member_sets=[0]*len(P)

    for i in xrange(len(P)):
        if P[i-1]==1:

            if M[i-1][0][2]>=0 and M[i-1][1][2]>=0:
                S[i-1]='Braces'
            elif M[i-1][0][2]<0 and M[i-1][1][2]<0:
                S[i-1]='Braces_submerged'
            else:
                S[i-1]='Braces_half_submerged'

        elif P[i-1]==2:
            if M[i-1][0][2]>=0 and M[i-1][1][2]>=0:
                S[i-1]='Lower_leg'
            elif M[i-1][0][2]<0 and M[i-1][1][2]<0:
                S[i-1]='Lower_leg_submerged'
            else:
                S[i-1]='Lower_leg_half_submerged'
```

```
        elif P[i-1]==3:
            if M[i-1][0][2]>=0 and M[i-1][1][2]>=0:
                S[i-1]='Upper_leg'
            elif M[i-1][0][2]<0 and M[i-1][1][2]<0:
                S[i-1]='Upper_leg_submerged'
            else:
                S[i-1]='Upper_leg_half_submerged'

        else:
            if M[i-1][0][2]>=0 and M[i-1][1][2]>=0:
                S[i-1]='Top'
            elif M[i-1][0][2]<0 and M[i-1][1][2]<0:
                S[i-1]='Top_submerged'
            else:
                S[i-1]='Top_half_submerged'

        member_sets[i-1]='Wire-'+str(i)+'-Set-1'
        member_sets[len(P)-1]='Wire-'+str(len(P))+'-Set-1'



    ############## Section assignment

    for i in xrange(len(P)):

trusspart.SectionAssignment(offset=0.0,offsetField='',
offsetType=MIDDLE_SURFACE,
region=trusspart.sets[member_sets[i]], sectionName=S[i],
thicknessAssignment=FROM_SECTION)


    ############## Mesh


trusspart.setElementType(elemTypes=(ElemType(elemCode=B31,
elemLibrary=STANDARD), ),
regions=(trusspart.edges.getSequenceFromMask(('[#ffffffff:3
#ff ]', ), ), ))

    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([CC]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([nacelle_cg]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([hub_cg]), number=5)
    for i in xrange(len(P)+1):
            trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt(f[i-1]), number=5)
```

```
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid1]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid2]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid3]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid4]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid5]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid6]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid7]), number=5)
    trusspart.seedEdgeByNumber(constraint=FINER,
edges=trusspart.edges.findAt([towermid8]), number=5)

    trusspart.generateMesh()


    ############## Assembly

    model.rootAssembly.DatumCsysByDefault(CARTESIAN)
    model.rootAssembly.Instance(dependent=ON, name='Part-1-
1',  part=trusspart)

    model.rootAssembly.regenerate()

    model.rootAssembly.Set(name='Rotor_nacelle',
vertices=model.rootAssembly.instances['Part-1-
1'].vertices.findAt([nacelle_cg],[hub_cg]))

model.rootAssembly.Set(name='Concrete_jacket_connection',
vertices=model.rootAssembly.instances['Part-1-
1'].vertices.findAt([N24],[N28],[N32],[N36],[CC]))


model.Coupling(controlPoint=Region(vertices=model.rootAssem
bly.instances['Part-1-1'].vertices.findAt([point_mass_1])),
couplingType=KINEMATIC, influenceRadius=WHOLE_SURFACE,
localCsys=None,
        name='Constraint-2',
surface=model.rootAssembly.sets['Concrete_jacket_connection
'], u1=ON, u2=ON, u3=ON, ur1=ON, ur2=ON, ur3=ON)


model.Coupling(controlPoint=Region(vertices=model.rootAssem
bly.instances['Part-1-1'].vertices.findAt([point_mass_3])),
```

```
couplingType=KINEMATIC, influenceRadius=WHOLE_SURFACE,
localCsys=None,
        name='Constraint-3',
surface=model.rootAssembly.sets['Rotor_nacelle'], u1=ON,
u2=ON, u3=ON, ur1=ON, ur2=ON, ur3=ON)


    ############## Point masses


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=concrete_Iw,
i22=concrete_Iw, i33=concrete_Ih, mass=concrete_mass, name=
        'Inertia-1',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([CC])))


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=0, i22=0, i33=0, mass=1900,
name=
        'Pointmass-1',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([point_mass_1])))


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=0, i22=0, i33=0, mass=1400,
name=
        'Pointmass-2',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([point_mass_2])))


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=0, i22=0, i33=0, mass=1000,
name=
        'Pointmass-3',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([point_mass_3])))


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
```

```
        0.0, composite=0.0, i11=0, i22=0, i33=0,
mass=240000, name=
        'Pointmass-4',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([nacelle_cg])))


model.rootAssembly.engineeringFeatures.PointMassInertia(alp
ha=
        0.0, composite=0.0, i11=0, i22=0, i33=0,
mass=110000, name=
        'Pointmass-5',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([hub_cg])))


    ############## Load
    model.ConcentratedForce(amplitude='lc1-fz', cf1=1.0,
        createStepName='Dynamic', distributionType=UNIFORM,
field='', localCsys=None
        , name='lc1-fx',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([load_point])))
    model.ConcentratedForce(amplitude='lc1-fy', cf2=-1.0,
        createStepName='Dynamic', distributionType=UNIFORM,
field='', localCsys=None
        , name='lc1-fy',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([load_point])))


    model.Moment(amplitude='LC1-mx', cm1=1.0,
createStepName=
        'Dynamic', distributionType=UNIFORM, field='',
localCsys=None, name=
        'LC1-mx',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([load_point])))
    model.Moment(amplitude='LC1-my', cm2=1.0,
createStepName=
        'Dynamic', distributionType=UNIFORM, field='',
localCsys=None, name=
        'LC1-my',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([load_point])))
    model.Moment(amplitude='LC1-mz', cm3=1.0,
createStepName=
        'Dynamic', distributionType=UNIFORM, field='',
localCsys=None, name=
```

```
        'LC1-mz',
region=Region(vertices=model.rootAssembly.instances['Part-
1-1'].vertices.findAt([load_point])))
    model.Gravity(comp3=-9.81, createStepName='Dynamic',
        distributionType=UNIFORM, field='', name='Gravity')

    ############## BC

    model.rootAssembly.Set(name='BC_set',
vertices=model.rootAssembly.instances['Part-1-
1'].vertices.findAt([N1],[N6],[N11],[N16]))
    model.DisplacementBC(amplitude=UNSET,
createStepName='Dynamic',distributionType=UNIFORM,
fieldName='', fixed=OFF, localCsys=None, name='BC-1',
        region=model.rootAssembly.sets['BC_set'], u1=0.0,
u2=0.0, u3=0.0, ur1=0.0, ur2=0.0, ur3=0.0)


    ############## Beam orientations


    beamvectors=[0]*len(P)
    beamnormals=[0]*len(P)
    for i in xrange(len(P)):
            beamvectors[i]=[0]*3
            beamnormals[i]=[0]*3
    for i in xrange(len(P)+1):
        for j in xrange(1,4):
            beamvectors[i-1][j-1]=M[i-1][1][j-1]-M[i-
1][0][j-1]

    for i in xrange(len(P)):
        if beamvectors[i][0]==0:
            beamnormals[i][0]=1
            beamnormals[i][1]=0
            beamnormals[i][2]=0
        elif beamvectors[i][1]==0:
            beamnormals[i][0]=0
            beamnormals[i][1]=1
            beamnormals[i][2]=0
        else:
            beamnormals[i][1]=1
            beamnormals[i][2]=1
            beamnormals[i][0]=-
(beamvectors[i][1]*beamnormals[i][1]+beamvectors[i][2]*beam
normals[i][2])/(beamvectors[i][0])


    for i in xrange(len(P)):
```

```
trusspart.assignBeamSectionOrientation(method=N1_COSINES,
n1=(beamnormals[i]), region=trusspart.sets['Wire-
'+str(i+1)+'-Set-1'])

    ############## Outputs
    model.fieldOutputRequests['F-Output-
1'].setValues(variables=('S', 'U'),frequency=1)
    mdb.models['Model-
1'].HistoryOutputRequest(createStepName='Dynamic',
frequency=
        LAST_INCREMENT, name='H-Output-1',
variables=('MASS', ))

    mdb.models['Model-
1'].keywordBlock.synchVersions(storeNodesAndElements=False)
    keywordblocks=model.keywordBlock.sieBlocks
    stepblock='*Step, name=Step-1, nlgeom=NO'
    if count==1:
        model.keywordBlock.insert(595, """*AQUA\n-50, 0,
9.81, 1021\n*WAVE, TYPE=STOKES\n1.395, 5.705, 0, 1, 0""")


    if q>0:
        wiredel="Wire-"+str(q)
        del mdb.models['Model-1'].parts['Part-
1'].features[wiredel]


    ############## Run job
    mdb.jobs[jobname].submit()
    mdb.jobs[jobname].waitForCompletion()
```

APPENDIX 7 – MATLAB SCRIPT FOR RAINFLOW COUNTING (REQUIRES MATLAB PLUGIN (12))

```
clear all
clc
li=[0,3,4,7,8,17,18,19,20,21,22,23,24,33,34,36,37,40,41,45,
46,47,48,53,56,57,60,61,62,63,64,69,72,73,76,77,78,79,80,85
,88,89,92,93,94,95,96];
len=length(li);
damage=zeros(len,56);
for j=1:len
    member=li(j);
    memberstr=num2str(member);
    loadcase='lc1';
    name=strcat(loadcase,'-',memberstr,'.txt');
    a=strcat(name);
    Input =dlmread(a, '\t',1,0);

    for i=1:56
        Var=Input(:,i);

        % S-N curve parameters:
        sigaf=35000000;    % endurance limit
        m=8;           % slope of the curve
        Nk=2e6;        % number of cycle for knee point


        % length in second of the time history
        To=12.5;

        tp=sig2ext(Var);      % turning points
        rf=rainflow(tp);       % rainflow
        CycleRate=rf(3,:);     % number of cycles
        siga=rf(1,:);          % cycle amplitudes

        % calculation of the damage

damage(j,i)=1/(sum((CycleRate/Nk).*((siga/sigaf).^m)));
    end
end
ref=damage;
for m=1:len
    for n=1:56
        ref(m,n)=damage(m,n)/damage(1,n);
    end
end
```

APPENDIX 8 – OPTIMISED GEOMETRY (FIGURE)

NTNU
Det skapende universitet

MASTERKONTRAKT

- uttak av masteroppgave

1. Studentens personalia

Etternavn, fornavn	Fødselsdato
Bakkom, Ole Edvard	**28. aug 1990**
E-post **oleedvba@stud.ntnu.no**	Telefon **94359243**

2. Studieopplysninger

Fakultet	
Fakultet for ingeniørvitenskap og teknologi	
Institutt **Institutt for bygg, anlegg og transport**	
Studieprogram **Bygg- og miljøteknikk**	Studieretning **Beregningsmekanikk**

3. Masteroppgave

Oppstartsdato **14. jan 2014**	Innleveringsfrist **10. jun 2014**
Oppgavens (foreløpige) tittel **Computer-aided optimization of an offshore jacket for a wind turbine with a simplified load model**	

Oppgavetekst/Problembeskrivelse

The task is to optimize the structural design of a jacket foundation for an offshore wind turbine, using ABAQUS finite-element software. The aerodynamic loads will be provided by a simplified load model that will be developed during the project. Both the optimization of the topology of the jacket (i.e., location of nodes) and the sizing of the members shall be studied.

This shall include the following subtasks:
- Literature study (wind turbine loads, structural optimization) [2 weeks]
- Implementation of an ABAQUS model of the jacket structure [5 weeks]
- Development of the rotor load model [2 weeks]
- Validation of structural response with a full wind turbine simulation [2 weeks]
- Implementation of gradient-based optimization for optimizing section heights [3 weeks]
- Optimization of the structure [2 weeks]
- Writing [4 weeks]

Hovedveileder ved institutt	Medveileder(e) ved institutt
Forsker Michael Muskulus	
Merknader **1 uke ekstra p.g.a påske.**	

4. Underskrift

Student: Jeg erklærer herved at jeg har satt meg inn i gjeldende bestemmelser for mastergradsstudiet og at jeg oppfyller kravene for adgang til å påbegynne oppgaven, herunder eventuelle praksiskrav.

Partene er gjort kjent med avtalens vilkår, samt kapitlene i studiehåndboken om generelle regler og aktuell studieplan for masterstudiet.

……………………………
Sted og dato

……………………………
Student

……………………………
Hovedveileder

Originalen lagres i NTNUs elektroniske arkiv. Kopi av avtalen sendes til instituttet og studenten.