

SPROPS, PROFAIL, and OPTFAIL Manual

A1.1 Introduction

OPTFAIL is a computer code written in FORTRAN developed by Woodson (1994) to optimize circular frames laminated from uniaxial tape composites for improved energy absorption. It provides the design engineer with a user-friendly tool for the design of uniaxial tape layup composite frames.

The finite element analysis is based on a Vlasov-type curved-bar theory (Woodson et al., 1993) extended to include laminated composite wall construction, and a selective and progressive ply-by-ply material degradation model for graphite-epoxy laminates (Tsai, 1992). A genetic algorithm (Holland, 1975; Goldberg, 1989) is used to design the stacking sequence and cross-sectional dimensions, since the design variables are limited to discrete values and the objective function and constraint function are nonlinear, non-smooth functions of the design variables. The simplified analysis for progressive failure response and design are considered appropriate for preliminary design. More details related to the suitability of this model for graphite-epoxy uniaxial tape layup composite frames can be found in Woodson (1994).

Depending on the needs of the designer, there are three different versions of the code, in increasing degree of complexity of the analysis:

SPROPS: Section property calculation for thin-walled open cross-section laminated composite curved beams.

PROFAIL: In addition to the beam properties, this code predicts the load-deflection curve of the structure and calculates the displacements at the nodes and the resultant forces at the ends of the structure in global coordinates. In addition, it predicts the failure sequence and location of the main failure events in the structure.

OPTFAIL: This is the complete code. In addition to the above, it can optimize for the stacking sequence of the laminate and/or the geometry of the cross-section.

A1.1.1 Sample problem

The sample problem discussed through this appendix is an I-section tape layup composite frame of 6 ft in diameter that is clamped at the ends. The frame is made of graphite/epoxy material AS4/5208. The properties of this material are listed in Table A1.1. The cross-section of the frame is described in Figs. A1.1, and A1.2.

Table A1.1 Material properties of sample problem.

Stiffness Properties	value	Strength Properties	value
E_1 , msi	18.4	X , ksi	197.3
E_2 , msi	1.64	X' , ksi	160.0
ν_{12}	0.30	Y , ksi	5.50
G_{12} , msi	0.87	Y' , ksi	29.9
		S , ksi	14.0

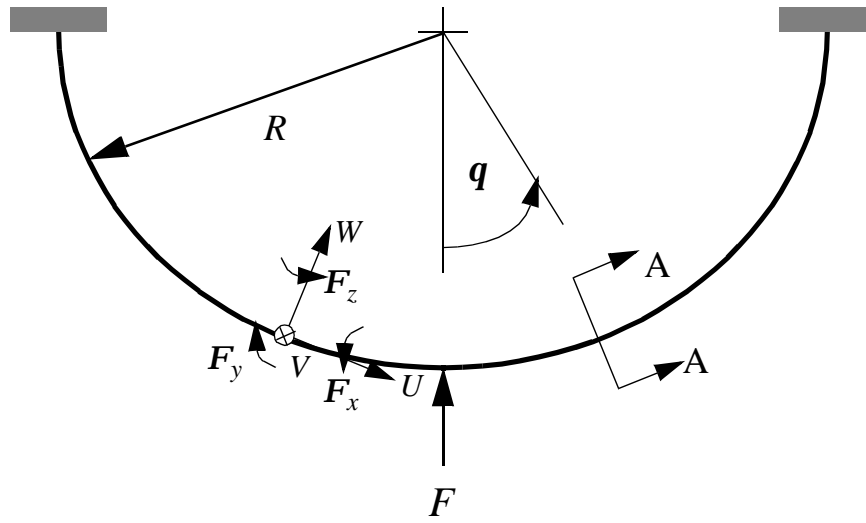


Fig. A1.1 Semicircular frame model.

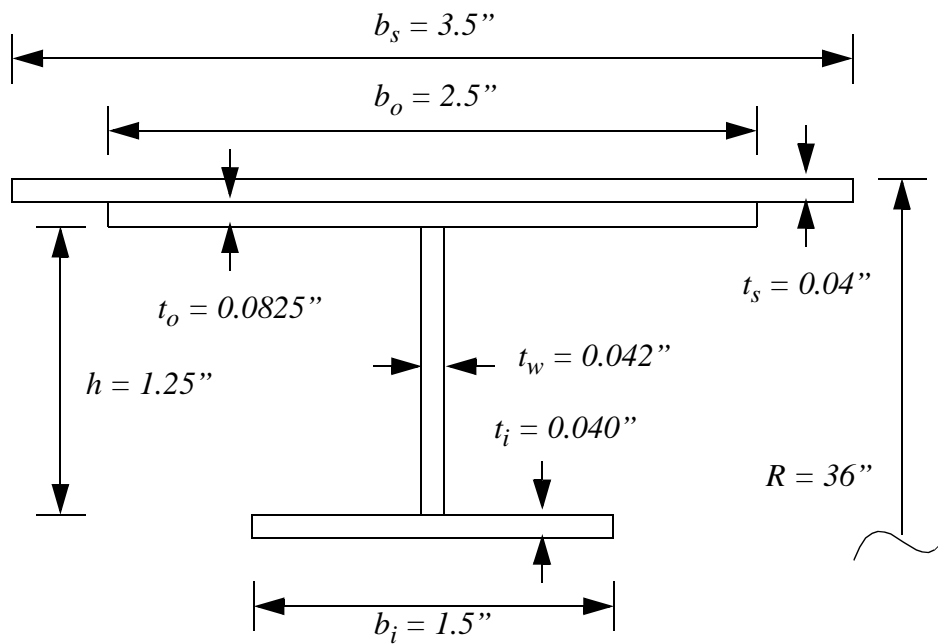


Fig. A1.2 Semicircular frame model cross-section A-A.

In Figs. A1.1, and A1.2, R is the radius of the frame in the outer flange, h is the height of the web, b_i , b_o , and b_s are the width of the inner flange, outer flange, and skin, respectively, and t_i , t_w , t_o , and t_s are the thicknesses of the inner flange, web, outer flange, and skin, respectively. The stacking sequence of the inner flange, the web, and the skin is $[45^\circ/-45^\circ/0^\circ/90^\circ]_S$, and the stacking sequence of the outer flange is $[45^\circ/-45^\circ/0^\circ/90^\circ/45^\circ/-45^\circ/0^\circ/90^\circ]_S$.

A1.2 SPROPS computer code

SPROPS performs section property calculation for thin-walled open cross-section laminated composite curved beams. The code needs the file INPUT.DAT to run, and produces the file OUTPUT.DAT.

A1.2.1 Input file

The input file for this code (INPUT.DAT) is presented in the next page. The input for this code includes: lamina material properties, lamina orientation, lamina arrangement in the laminate, and cross-sectional characteristics.

Material data

The input file must begin with the statement “*matl”. (Note that there is no space blank after the asterisk. The same applies to every other similar statement in the input file.) The next line must include all the following lamina parameters in order, separated by at least one space blank: E_1 = modulus of elasticity in the fiber direction, E_2 = modulus of elasticity in the transverse direction, ν_{12} = major Poisson’s ratio, G_{12} = modulus of rigidity, t = thickness of the lamina, ρ = density of the material.

Input file (INPUT.DAT) for SPROPS:

```

*matl
18.4e6 1.64e6 0.30 0.870e6 0.005000 0.003
18.4e6 1.64e6 0.30 0.870e6 0.005250 0.003
18.4e6 1.64e6 0.30 0.870e6 0.005156 0.003
18.4e6 1.64e6 0.30 0.870e6 0.005104 0.003
*endmatl
*lamina
1 45
1 -45
1 0
1 90
2 45
2 -45
2 0
2 90
3 45
3 -45
3 0
3 90
4 45
4 -45
4 0
4 90
*endlamina
*lamnte
8 1 2 3 4 4 3 2 1
8 5 6 7 8 8 7 6 5
16 9 10 11 12 9 10 11 12 12 11 10 9 12 11 10 9
24 13 14 15 16 13 14 15 16 16 15 14 13 16 15 14 13 14 15 16
16 15 14 13
*endlamnte
*bprop
8 7 36.0 0.0 0.7072
-0.75 1.3934
0 1.3934
0.75 1.3934
-1.75 0.0612
-1.25 0.0612
0 0.0612
1.25 0.0612
1.75 0.0612
2 1 2 0.0
2 2 3 0.0
1 6 2 0.0
3 4 5 0.0
3 7 8 0.0
4 5 6 0.0
4 6 7 0.0
*endbprop

```

Every line represents the input of a single material. Additional input lines can be written for any additional “material”. In the example, four “materials” are used. Although all the lamina material properties are the same in this data file, the thickness of the lamina will be different, so it is written as a different “material”.

The statement “*endmatl” terminates the material specifications.

A number will be assigned to every material included. This number will be used as reference in other areas of the input. The first material properties written will correspond to material number 1, the second to material number 2, and so on. For example, in the input file presented, the material with lamina thickness of 0.0052500 is material number 2, the one with 0.0051560 is material number 3. This number is going to be required in other sections of the input file, and to understand the output.

Lamina data

The lamina orientation input start with the statement “*lamina”.

Every lamina orientation needed for all the materials above must be written in the following format, one line per lamina (as before, every data information must be separated by at least one blank space): material id number, lamina orientation in degrees.

In the example, laminae with orientations of 0° , $\pm 45^\circ$, and 90° are used. Other problems will require only one or two lamina orientations.

This input section must finish with the statement “*endlamina”.

As in the previous section, every lamina included will be numbered according to the order in which it is written. For example, the lamina identified as “1 45.” in the input file is lamina number 1, the one identified as “2 0.” is lamina number 7, and the one identified as “4 90.” is lamina number 16. This information will also be needed for input in other parts of the code, and to understand the output.

Stacking sequence data

In this section, every laminate needed is constructed from the laminae from the previous section. The laminate input starts with the statement “*lamnte”.

Every laminate will be included in a single line following this format: number of laminae in the laminate, lamina id number (if more than one, separate every number with at least one blank space; if the number of laminae is 16, for example, 16 lamina id numbers should be listed).

In the input file presented there are four laminates. The first laminate has eight laminae; eight laminae id numbers follow. Notice that the fourth laminate has twenty-four laminae.

This part must finish with the statement “*endlmnte”.

The code will assign a number to the laminate in the same order as the laminate is written in the input file.

Beam cross-section data

The first line of this part of the input must be “*bprop”.

The next line must include the following information in order, separated by at least one blank space: number of joints, number of sections, radius of curvature, y-coordinate of the reference point, z-coordinate of the reference point.

The reference point defines the axis of the beam that is used in the finite element analysis.

The next lines will include the coordinates of the nodes (junctions). In the example from the input file there are 8 nodes. The input of each line must include the following,

separated at least by one blank space: y-coordinate of node (junction), z-coordinate of node (junction).

Every node is numbered by the code in the same order that they are written in the input file. In the sample problem, the node located at (0,0.0612) is node number 6. The next lines include information about the sections (branches). Every section is defined by two nodes. Every line must include: laminate id, initial node id, final node id, offset.

As in the nodes, every section is numbered by the program in the order in which it is written in the input file.

The input of this part must finish with the statement “*endbprop”.

If there are different cross-sections, a different set of nodes and sections for every cross-section are required. After entering the data of all the cross-sections, then the end statement “*endbprop” must be written.

A1.2.2 Output file

The output file of SPROPS is called “OUTPUT.DAT” and will allow the user to check the input file. The output file includes the following:

Material data for each material: moduli, Poisson’s ratio, thickness, and density

Lamina data: lamina id, material, ply angle

Stacking sequence data:

For every laminate, the following information is given: laminate id, ply number, lamina id, material id, ply angle, laminate thickness, extensional stiffness matrix [A], coupling stiffness matrix [B], bending stiffness matrix [D].

Beam cross-section data:

For every cross-section, the following information is given:

number of joints, number of sections, radius of reference point, y-coordinate of reference point, z-coordinate of reference point

node id, y-coordinate, z-coordinate

section id, laminate id, initial node, final node, offset

Following the previous information, the code calculates the modulus-weighted section properties: EA , ES_y , ES_z , ES_w , E_h , EI_{yy} , EI_{yz} , EI_{wy} , E_{hc} , EI_{zz} , EI_{wz} , E_{hs} , EI_{ww} , E_{hq} , and GJ . There are three lists of these properties, according to the point used to calculate them. The points considered are the pole at reference (whose coordinates are given by the user in the input file), the pole at the centroid, and the pole at the shear center. The last two points are calculated by the code and their coordinates are listed in the output file.

A1.3 PROFAIL computer code

This program code will calculate the load-deflection curve and predicts the failure sequence, type (fiber or matrix) and locations. It creates three principal output files: OUTPUT.DAT (similar to the one produced by SPROPS), PLOT.F.OUT (load-deflection curve data points), and FAILURE.DAT (failure sequence and location data).

A1.3.1 Input file

The input file (INPUT.DAT) has many similarities with the one used with SPROPS, but additional information is needed. A sample input file for PROFAIL is presented at the end of this section.

Material data

First and second lines are the same as in the input file for SPROPS, but there is an additional line of material data that must be included for each material. This new data is to specify the strength characteristics of the material. It must include all the following properties in order, separated by at least one space blank: X = tensile strength in the fiber direction, X' = compressive strength in the fiber direction, Y = tensile strength in the transverse direction, Y' = compressive strength in the transverse direction, S = shear strength, F_{12}^* = interaction coefficient.

Lamina data, stacking sequence data, and beam cross-section data are the same as for SPROPS.

Node generation

This part must be added after the statement “*endbprop”, as shown in the input file. In this part, the user will specify the number of nodes along the arclength of the frame. To locate the load at the center of the frame, an odd number of nodes should be generated.

This part of the input must begin with the following two statements “*nodes” in the first line, and “*nodesgen” in the second.

The next line must include the following data separated by at least one space blank: radius of curvature at the reference point, number of nodes along the arclength.

This way to make the input will generate all the nodes by giving only the coordinates of the two end nodes. An additional node located along the line of symmetry of the frame (dummy node) must also be included. This information must be written in the following order, in three lines (one for each node):

initial end node x-coordinate, initial end node y-coordinate, initial end node z-coordinate

final end node x-coordinate, final end node y-coordinate, final end node z-coordinate

dummy node x-coordinate, dummy node y-coordinate, dummy node z-coordinate

To finish this part, write the following two statements: “*endgen” in the first line, and “*endnodes” in the following.

The nodes can also be written directly instead of being generated. For this, different instructions are needed.

Constraint input

This part is to specify the displacements and rotations in the nodes. In general, the displacements and rotations are specified for the initial node and the final node (the end nodes). The displacements for the dummy node must always be set to zero. There are seven degrees of freedom for each node: displacements and rotations in the three Cartesian coordinates (positive directions are shown in Figures 1 and 2), and warping of the cross-section.

The first line of this part is the statement “*con”.

The following lines should include the information shown for each node and each degree of freedom (in the example, 3 nodes * 7 dof's = 21 lines): node id, degree of freedom, displacement.

This must include the seven degrees of freedom of the initial end node (in this example, node 1), the final node (in this example, node 21), and the dummy node (in this example, node 22). The displacements of the dummy node are zero.

This part finishes with the statement “*endcon”.

Load input

This section of the input is necessary to specify the location and direction in which the load is applied. The load must be specified in a certain node, and in a certain degree of freedom.

This section begins with these two statements (each written in a different line): “*load”, and “case”.

The next line must include the following information in order, separated by at least one space blank: node id, degree of freedom, load magnitude.

The magnitude of the load must be different from zero. The load magnitude written in this part will not have any effect in the computation of the load-deflection curve. Nevertheless, this load magnitude will be of paramount importance when using the optimization code OPTFAIL because in that instance it represents the design load.

Close the input of this part by writing: “endcase”, and “*endload”.

Beam input

This section of the input is to locate beam elements and orient them properly with respect to the rest of the beam. Also in this part a cross-section will be assigned to every beam element of the model.

This part begins with the statement:

```
*beam
```

Then every beam element is specified by an initial node, a final node, a dummy node (the same for all the sections, node 22 in our example), and the cross-section id (in our example, the cross-section is the same for all). The data for every single beam element must be in the same line: initial node or arclength section, final node or arclength section, dummy node, cross-section id.

This section, and the input file for PROFAIL, finishes with the following statements:
“*endbeam”, “*progsol”, and “*end”.

Input file (INPUT.DAT) for PROFAIL:

```
*matl
18.4e6 1.64e6 0.30 0.870e6 0.005000 0.003
197.9e3 160.0e3 5.50e3 29.9e3 14.0e3 -0.50
18.4e6 1.64e6 0.30 0.870e6 0.005250 0.003
197.9e3 160.0e3 5.50e3 29.9e3 14.0e3 -0.50
18.4e6 1.64e6 0.30 0.870e6 0.005156 0.003
197.9e3 160.0e3 5.50e3 29.9e3 14.0e3 -0.50
18.4e6 1.64e6 0.30 0.870e6 0.005104 0.003
197.9e3 160.0e3 5.50e3 29.9e3 14.0e3 -0.50
*endmatl
*lamina
1 45
1 -45
1 0
1 90
2 45
2 -45
2 0
2 90
3 45
3 -45
3 0
3 90
4 45
4 -45
4 0
4 90
*endlamina
*lamnte
8 1 2 3 4 4 3 2 1
8 5 6 7 8 8 7 6 5
16 9 10 11 12 9 10 11 12 12 11 10 9 12 11 10 9
24 13 14 15 16 13 14 15 16 16 15 14 13 16 15 14 13 13 14 15 16
16 15 14 13
*endlamnte
*bprop
8 7 36.0 0.0 0.7072
-0.75 1.3934
0 1.3934
0.75 1.3934
-1.75 0.0612
-1.25 0.0612
0 0.0612
1.25 0.0612
```

```

1.75 0.0612
2 1 2 0.0
2 2 3 0.0
1 6 2 0.0
3 4 5 0.0
3 7 8 0.0
4 5 6 0.0
4 6 7 0.0
*endbprop
*nodes
*nodegen
36.0 21
0.000000e+00 0.000000e+00 0.000000e+00
7.200000e+01 0.000000e+00 0.000000e+00
3.600000e+01 0.000000e+00 -1.585622e+00
*endgen
*endnodes
*con
1 1 0.0
1 2 0.0
1 3 0.0
1 4 0.0
1 5 0.0
1 6 0.0
1 7 0.0
21 1 0.0
21 2 0.0
21 3 0.0
21 4 0.0
21 5 0.0
21 6 0.0
21 7 0.0
22 1 0.0
22 2 0.0
22 3 0.0
22 4 0.0
22 5 0.0
22 6 0.0
22 7 0.0
*endcon
*load
case
11 3 1000
endcase
*endload
*beam
1 2 22 1
2 3 22 1
3 4 22 1
4 5 22 1
5 6 22 1
6 7 22 1
7 8 22 1
8 9 22 1

```

```
9 10 22 1
10 11 22 1
11 12 22 1
12 13 22 1
13 14 22 1
14 15 22 1
15 16 22 1
16 17 22 1
17 18 22 1
18 19 22 1
19 20 22 1
20 21 22 1
*endbeam
*progsol
*end
```

A1.3.2 Output file

PROFAIL is run in a PC computer. It is really composed of two separate executable files: PROFAIL.EXE and POST.EXE. Both should be run to get the output files. The input file must be named “INPUT.DAT”. The following instructions must be followed to run PROFAIL:

step 1: write the name of the program and hit the “Enter” key

PROFAIL [*ENTER key*]

step 2: answer questions about the location of the load

degree of freedom of the direction of the load [*ENTER key*]

node to analyze [*ENTER key*]

In the sample problem shown in the input file, the answer to these questions is 3 and 11, respectively.

step 3: write the name of the other program and hit the “Enter” key

POST [*ENTER key*]

step 4: answer the questions that the code is making

degree of freedom of the direction of the load [*ENTER key*]

node to show [*ENTER key*]

After running PROFAIL and POST, three output files are created:

OUTPUT.DAT: Output file very similar to the one created by program code SPROPS. The main difference is that this one does not include the calculation of the centroid or the shear center.

FAILURE.DAT: Output file that describes the failure sequence and location of the main failure events in the structure. When interpreting this file, the cross-section of the structure, the nodes, the sections, the junctions and the segments have to be taken into account. Whether the failure is a fiber failure or a matrix failure is also shown in this file.

PLOTF.OUT: File that includes the load and displacement points needed to plot the load-deflection curve of the response of the frame.

A1.4 OPTFAIL computer code

OPTFAIL is a computer code that will optimize a composite fuselage frame for laminae stacking sequence and/or cross-sectional dimensions of the frame for improved energy absorption.

A1.4.1 Input file

The input file (INPUT.DAT) is shown at the end of this section, and is almost the same as the one used in PROFAIL, with the following exceptions:

Laminate data

In this part, the line following “*lamnte” will include an additional piece of information: total number of laminae in the laminate, optimization option, lamina id number (if more than one, separate every number with at least one blank space; if the number of laminae is 16, for example, you must have 16 lamina id numbers in this part). The “optimization option” is either “0” or “1”: “0” means no stacking sequence optimization will be performed, “1” means stacking sequence optimization will be performed.

Beam cross-section data

In this part, the line following “*bprop” must include additional information: number of junctions (nodes), number of branches (sections), radius of curvature, y-coordinate of the reference point, z-coordinate of the reference point, percentage of branch (section) length, shift variable.

The “percentage of branch length” will be the maximum range of values that will be analyzed in every run of the optimization. It is related to how far the junctions (nodes) can move in every run. The smaller this number is, the most precise the optimization could be. For example, if the length of the branch that is going to be optimized is 1 in., and the percentage of branch length to be considered is 10 %, then the range that will be considered in the optimization has a size of 0.1 in.. Seven points will be evaluated along this range, from the beginning to the end.

The shift parameter changes the shape of the cross-section to optimize the energy absorbed. It can be either “-1”, “0”, or “1”:

-1: With this option, the program code will only consider branch lengths smaller than the original one (it will only shrink the branch length). For example, if the branch is 1 in. in length, and we use 10%, then branches of lengths ranging from 0.9 in. to 1 in. will be considered. Seven points will be considered in the analysis: the branch length corresponding

to the first one will be 1 in. (the original dimension), the second will be 0.983 in., the third 0.967 in., then 0.95 in., 0.933 in., 0.917 in., and the seventh point will be 0.9 in..

0: With this option, the program code will consider branch lengths larger and smaller than the original one. If the section is 1 in. in length, and we use 10 %, then sections of lengths ranging from 0.95 in. to 1.05 in. will be considered. The points considered will range from the first one (0.95 in.) to the seventh (1.05 in.); point number four corresponds to 1 in. (the original dimension).

1: With this option, the program code will only consider branch lengths larger than the original one (it will only expand the branch length). If the section is 1 in. in length, and we use 10 %, then sections of lengths ranging from 1.0 in. to 1.1 in. will be considered. The first point will be 1.0 in. (original dimension), the seventh 1.1 in..

The next lines of information must have the following format: y-coordinate of junction (node), z-coordinate of junction (node), y-coordinate direction of motion, z-coordinate direction of motion.

In the example shown, the y-coordinate of joint 1 will move in the opposite direction of joint 3, the z-coordinate of joint 1 will not be allowed to move (a "0" is used to describe this). Whenever the number written in this part is the same as the node number, that means that it is independent, free to move.

Load input

As in PROFIL, this section if the input is necessary to specify the location and direction in which the load is applied. The load must be specified in a certain node, and in a certain degree of freedom.

This section begins with these two statements (each written in a different line):
"*load", and "case".

The next line must include the following information in order, separated by at least one space blank: node id, degree of freedom, load magnitude.

The load magnitude here represents the design load, the load that will serve as constraint in the optimization scheme. The optimization values obtained with the code will yield a load-deflection curve that will have this value as the maximum (if the design has no penalty).

Close the input of this part by writing: “endcase”, and “*endload”

The rest of the input file used to run the code OPTFAIL is the same as the input file to run PROFAIL.

An additional input file, called TINPUT, must be prepared to run OPTFAIL in the Unix System. This file must include three pieces of data, each one in a different line, in this order: dof, node for analysis, integer seed number. The integer seed number can be any number.

Input file (INPUT.DAT) for OPTFAIL:

```
*matl
18.4e6 1.64e6 0.30 0.870e6 0.005000 0.003
197.9e3 160.0e3 5.50e3 29.9e3 14.0e3 -0.50
18.4e6 1.64e6 0.30 0.870e6 0.005250 0.003
197.9e3 160.0e3 5.50e3 29.9e3 14.0e3 -0.50
18.4e6 1.64e6 0.30 0.870e6 0.005156 0.003
197.9e3 160.0e3 5.50e3 29.9e3 14.0e3 -0.50
18.4e6 1.64e6 0.30 0.870e6 0.005104 0.003
197.9e3 160.0e3 5.50e3 29.9e3 14.0e3 -0.50
*endmatl
*lamina
1 45
1 -45
1 0
1 90
2 45
2 -45
2 0
2 90
```

```

3 45
3 -45
3 0
3 90
4 45
4 -45
4 0
4 90
*endlamina
*lamnte
8 0 1 2 3 4 4 3 2 1
8 0 5 6 7 8 8 7 6 5
16 0 9 10 11 12 9 10 11 12 12 11 10 9 12 11 10 9
24 0 13 14 15 16 13 14 15 16 16 15 14 13 16 15 14 13 13 14 15 16
    16 15 14 13
*endlamnte
*bprop
8 7 36.0 0.0 0.7072 10 0
-0.75 1.3934 -3 0
0 1.3934 0 0
0.75 1.3934 -3 0
-1.75 0.0612 -6 0
-1.25 0.0612 0 0
0 0.0612 0 0
1.25 0.0612 0 0
1.75 0.0612 6 0
2 1 2 0.0
2 2 3 0.0
1 6 2 0.0
3 4 5 0.0
3 7 8 0.0
4 5 6 0.0
4 6 7 0.0
*endbprop
*nodes
*nodegen
36.0 21
0.000000e+00 0.000000e+00 0.000000e+00
7.200000e+01 0.000000e+00 0.000000e+00
3.600000e+01 0.000000e+00 -1.585622e+00
*endgen
*endnodes
*con
1 1 0.0
1 2 0.0
1 3 0.0
1 4 0.0
1 5 0.0
1 6 0.0
1 7 0.0
21 1 0.0
21 2 0.0
21 3 0.0
21 4 0.0

```

```

21  5  0.0
21  6  0.0
21  7  0.0
22  1  0.0
22  2  0.0
22  3  0.0
22  4  0.0
22  5  0.0
22  6  0.0
22  7  0.0
*endcon
*load
case
11  3  1000
endcase
*endload
*beam
1  2  22  1
2  3  22  1
3  4  22  1
4  5  22  1
5  6  22  1
6  7  22  1
7  8  22  1
8  9  22  1
9  10  22  1
10 11  22  1
11 12  22  1
12 13  22  1
13 14  22  1
14 15  22  1
15 16  22  1
16 17  22  1
17 18  22  1
18 19  22  1
19 20  22  1
20 21  22  1
*endbeam
*progsol
*end

```

A1.4.2 Output file

There are two output files created by OPTFAIL: OUTPUT.DAT and OPGEN.OUT. OUTPUT.DAT is like the one created by PROFAIL. OPGEN.OUT has all the design history on it, followed by information details about the optimum design selected, including laminae stacking sequence and cross-sectional dimensions.