

Bitangent Users Guide

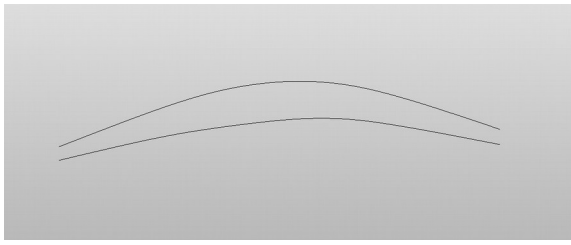
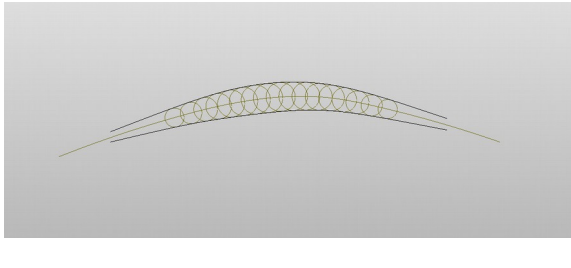
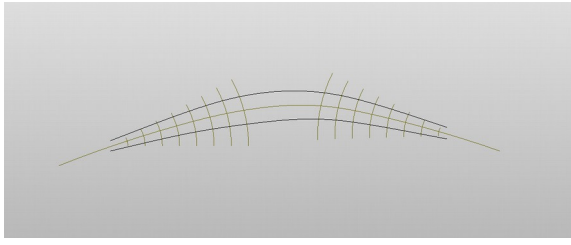
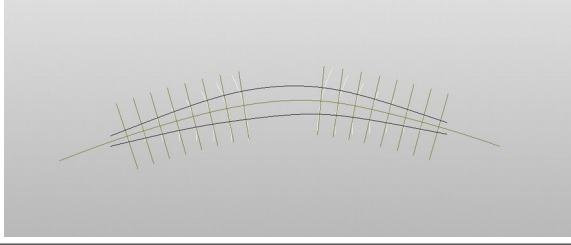
Table of Contents

Bitangent Users Guide.....	12
Overview.....	3
Compile Notes.....	3
Usage.....	4
Parameters.....	5
Input Files.....	8
Input files 'in_a.xyz', 'in_b.xyz'.....	8
Input file Param.txt.....	8
Output File.....	10
Output Calculation Data.....	12

Bitangent Users Guide

Overview

The purpose of the Bitangent program is to analyze scans and provide data useful for airfoil inspection. The following shows graphical examples of the function of this program.

<i>Image</i>	<i>Description</i>
	<p>Input data to the Bitangent program.</p> <p>The input is typically scan data from a CMM.</p>
	<p>Bitangent circles and camber line.</p> <p>A series of circles are fit between the two sets of input data. The point of contact from each side of the circle is tangent to the input data.</p> <p>The camber line is calculated from the center of each of the circles and extrapolated by 30% on each end.</p>
	<p>Intersection radii.</p> <p>One or more circles are created from points located at either end of the data. For each input an intersection point is calculated between the camber line and the circle.</p>
	<p>Edge thickness.</p> <p>For each intersection radius found along the camber line the thickness of the input data is calculated.</p> <p>The distance is calculated perpendicular to the camber line at the point of intersection.</p>

Compile Notes

The program is provided as source code or as a pre-compiled executable. To compile enter the command 'make' from a command prompt. A makefile is included based on the GNU GCC C++ compiler.

Bitangent Users Guide

Compile Example:

```
C:\MinGW\MyProjects\bitangent>make clean
rm -rf *.o bitangent.exe
```

```
C:\MinGW\MyProjects\bitangent>make
g++ -c -Wall -c main.cpp
g++ -c -Wall -c vector2.cpp
g++ -c -Wall -c vector3.cpp
g++ -c -Wall -c mat4.cpp
g++ -c -Wall -c refsys.cpp
g++ -c -Wall -c bitangent.cpp
g++ -c -Wall -c cmdline.cpp
g++ -c -Wall -c param.cpp
g++ main.o vector2.o vector3.o mat4.o refsys.o bitangent.o cmdline.o param.o -o bitangent
```

Command Line Options

The Bitangent program has the following command line options:

Command line options:

```
-debug_filter Output filter data to console
-extrapolation_length MCL extrapolation length
-extrapolation_sample MCL extrapolation data sample length
-filter_cutoff Cutoff length for filter
-filter_max_diameter Enable filter for max diameter search
-h Display argument help and exit
-inter_r1 Intersection circle 1 [X Y Z R]
-inter_r2 Intersection circle 2 [X Y Z R]
-inter_rn Intersection circle negative [X Y Z R]
-inter_rp Intersection circle positive [X Y Z R]
-order Spline Order
-param Name of parameter file
-path File path. Add quotes around path with spaces
-plane Projection plane [X Y Z I J K]
-quiet Suppress output
-u_end Ending position of bitangent circles
-u_incr Increment between bitangent circles
-u_start Starting position of bitangent circles
-v Show version and exit
```

The Bitangent program can be run without parameters. The input files 'in_a.xyz' and 'in_b.xyz' must exist in the same directory as the program. Default values are used for calculations.

```
C:\MinGW\MyProjects\bitangent
INFO: Input file path: C:\MinGW\MyProjects\bitangent
INFO: Loading data files in_a.xyz, in_b.xyz ... Done
INFO: Data size 'a': 1000
INFO: Data size 'b': 1000
INFO: Finding bitangent points ... Done
INFO: Finding maximum diameter ... Done
INFO: Writing output file ... Done
INFO: Processing complete
```

Bitangent Users Guide

The Bitangent program can be run with command line parameters to override the values used for calculations:

```
C:\MinGW\MyProjects\bitangent -path "g:\data to send" -inter_r1 0 0 0 15 -inter_r2 100 0 0 25 -order 3
-plane 0 0 10 0 0 1 -u_incr 0.01
INFO: Input file path: g:\data to send
INFO: Loading data files ... Done
INFO: Data size 'a': 1000
INFO: Data size 'b': 1000
INFO: Finding bitangent points ... Done
INFO: Finding maximum diameter ... Done
INFO: Writing output file ... Done
INFO: Processing complete
```

The Bitangent program can be run with a parameter file containing all input data and settings:

```
C:\MinGW\MyProjects\bitangent -path "g:\data to send" -param param.txt
INFO: Input file path: g:\data to send
INFO: Reading parameter data ... Done
INFO: Data size 'a': 1000
INFO: Data size 'b': 1000
INFO: Finding bitangent points ... Done
INFO: Finding maximum diameter ... Done
INFO: Writing output file ... Done
INFO: Processing complete
```

Parameters

The following describes the parameters used by the Bitangent program. The names of the parameter in the command line and those used in the parameter file are identical.

[u_start, u_incr, u_end]

These values are used to define the section of the input data to process. The input data is converted to a spline where the first point is represented by $u=0$ and the last is represented by $u=1$. The increment value, u_incr , is the relative spacing between samples of the data.

Example:

```
u_start = 0.1
u_incr = 0.2
u_end = 0.9
```

Result of Parameters:

The input data will be analyzed starting 10% ($u_start = 0.1$) of the total length. A measurement will be performed at increments of 20% ($u_incr = 0.2$) until 90% ($u_end = 0.9$) of the data is processed.

Bitangent Users Guide

[order]

The order parameter is passed to the spline function and controls the smoothness of the curve. A low order (value of two) will be equivalent to a polyline. As the order increases the smoothness of the spline increases. The spline approximates (not interpolate) the input data if the order value is set higher than two.

[plane]

The plane parameter describes the projection plane that the data will be calculated on. The projection of the point data is done perpendicular to the point normal.

Example:

```
-plane 0 0 10 0 0 1
```

Projection Sample Data:

- 1) 0.000, 0.000, 10.125, 0.000, 1.000, 0.000
- 2) 0.000, 0.000, 10.125, 0.000, 0.866,-0.500
- 3) 0.000, 0.000, 10.000, 0.000, 0.866,-0.500

1) Point will be projected from position (0.000, 0.000, 10.125) to (0.000, 0.000, 10.000). The perpendicular vector of the surface normal is (0.000, 0.000, 1.000)

2) Point will be projected from position (0.000, 0.000, 10.125) to (0.000, -0.072, 10.000) along the perpendicular vector of (0,0.5,0.866).

3) Point will not be changed. The position of the point is already (0.000, 0.000, 10.000)

The estimated projection position from example 2 is based on a 30 degree projection vector to Z ($\tan(30) \times 0.125 = 0.072169$).

The perpendicular vector of the point surface normal relies on the direction of the input data. It cannot be calculated from a single point.

[inter_r1, inter_r2]

These parameters allow intersection points to be found between the camber line and a user defined circle. Two circle locations are defined at each end of the data with multiple radius values. The default intersection point returned is the one closest to the center of the camber line data.

At each intersection of the camber line the distance (edge thickness) is calculated between the two sets of input data perpendicular to the camber line.

The command line parameter can only accept one radius input value. For multiple radius values use the parameter file method.

Bitangent Users Guide

[inter_rp, inter_rn]

These parameters allow intersection points to be found between the camber line and a user defined circle returning the intersection on either the positive or negative edge. The direction is relative to the camber line data (which is generally the same as direction as the 'a' data).

At each intersection of the camber line the distance (edge thickness) is calculated between the two sets of input data perpendicular to the camber line.

The command line parameter can only accept one radius input value. For multiple radius values use the parameter file method.

[param]

This option specifies the name of a parameter file. The parameter file contains values for all options along with the two sets of measurement point data.

[path]

The location of the input and output files. The default file location is the directory of the executable if the path is not specified.

[filter_max_diameter]

This option forces the maximum diameter to be found using the filtered edge thickness data.

[filter_cutoff]

The cutoff distance is the wavelength of the filter. At this distance any signal with has one full cycle will be reduced in amplitude by 50%. Refer to ISO 11562 for details.

[debug_filter]

Enabling this option will force additional information on the console window showing all parameters and intermediate values for the data filtering.

When using this option the output of the Bitangent program can be sent to a file using a standard redirection command.

Example:

```
bitangnet -filter_cutoff 10 -filter_max_diameter -debug_filter >outfile.txt
```

[extrapolation_sample]

Value representing the distance sampled to generated the extrapolated mean camber line. The minimum value is 3 (mm).

[extrapolation_length]

This value is the length of the extrapolated data generated from the mean camber line. This value can be zero if no extrapolation is required.

[quiet, v, h]

These options are general purpose and do not affect the output. The option 'quiet' will suppress

Bitangent Users Guide

output messages other than warnings or errors. Option 'v' will report the version and option 'h' will display the list of supported command line options.

Default parameter values

All values used by the Bitangent program are optional. The default values for the parameters are listed below:

```
u_start = 0.05
u_end = 0.95
u_incr = 0.05
spline_order = 2
projection plane vector = 0,0,1
projection plane position = <set to the first point of the 'a' data>
path = <current directory>
max_diameter_filter = off
filter_cutoff = 10
d_parameters.extrapolation_sample = 10;
d_parameters.extrapolation_length = 5;
```

It is not necessary to provide values for all parameters if the default values are satisfactory.

Input Files

There are two methods to provide input data to the Bitangent program. The first method is to create two files where each file contains one set of measurement data. The second method is to create a single file containing settings and two sets of measurement data.

Input files 'in_a.xyz', 'in_b.xyz'

These files are expected if the parameter file is not used. The measurement data files called 'in_a.xyz' and 'in_b.xyz' are standard XYZ files used by PC-DMIS.

Input files are expected to contain XYZIJK comma separated values. Lines not containing six values are ignored. The 'XYZIJK METRIC' line of the file is ignored if present. The measurement units of the data is not considered as all results will be reported in the same units as the input data.

Example data file:

```
XYZIJK METRIC
0.00000000,0.00000000,0.00000000,0.09950315,-0.99503725,0.00000000
0.36493325,0.03649323,0.00000000,0.09950315,-0.99503725,0.00000000
0.59945265,0.05994492,0.00000000,0.09950210,-0.99503735,0.00000000
0.83087753,0.08308689,0.00000000,0.09950072,-0.99503749,0.00000000
1.06170242,0.10616849,0.00000000,0.09949889,-0.99503767,0.00000000
1.29192863,0.12918974,0.00000000,0.09949661,-0.99503790,0.00000000
1.52155747,0.15215067,0.00000000,0.09949387,-0.99503818,0.00000000
1.75059025,0.17505132,0.00000000,0.09949065,-0.99503850,0.00000000
1.97902828,0.19789169,0.00000000,0.09948696,-0.99503887,0.00000000
2.20687284,0.22067181,0.00000000,0.09948280,-0.99503928,0.00000000
2.43412524,0.24339172,0.00000000,0.09947818,-0.99503974,0.00000000
...
```


Bitangent Users Guide

Input file Param.txt

The param.txt file contains all settings and required data. Comment lines begin with a hash '#' and identifier sections start with a colon. The parameter options use the same names as the command line version.

Example param file:

```
# enable filter option for location of max diameter
:filter_max_diameter
:filter_cutoff
10

# starting position of data to process
:u_start
0.1

# ending position of data to process
:u_end
0.9

# increment of processed data
:u_incr
0.1

# projection plane
:plane
0,0,0,0,0,1

# spline order
: order
3

# extrapolation of MCL data
: extrapolation_sample
5
: extrapolation_length
10

# intersection radii
# x,y,z,r1,r2,r3,...
:inter_r1
0,1.5,0,5,10,15,20,25,30,35,40

# intersection radii
# x,y,z,r1,r2,r3,...
:inter_r2
100,4.5,0,5,10,15,20,25,30,35,40

# intersection radii positive side
# x,y,z,r1,r2,r3,...
:inter_rp
50,9,0,5,10,15,20,25,30,35,40

# intersection radii negative side
# x,y,z,r1,r2,r3,...
:inter_rn
50,9,0,5,10,15,20,25,30,35,40

# scan data a
:data a
0.00000000,0.00000000,0.00000000,0.09950315,-0.99503725,0.00000000
0.36493325,0.03649323,0.00000000,0.09950315,-0.99503725,0.00000000
```

Bitangent Users Guide

```
0.59945265,0.05994492,0.00000000,0.09950210,-0.99503735,0.00000000
0.83087753,0.08308689,0.00000000,0.09950072,-0.99503749,0.00000000
...
```

```
# scan data b
:data b
0.00000000,30.00000000,0.00000000,-0.19611541,0.98058082,0.00000000
0.24347757,30.04869543,0.00000000,-0.19611541,0.98058082,0.00000000
0.40010302,30.08002030,0.00000000,-0.19611408,0.98058109,0.00000000
0.55479794,30.11095882,0.00000000,-0.19611232,0.98058144,0.00000000
...
```

When using a parameter file additional command line parameter values are ignored. All required settings should be in the parameter file.

Output File

A single output file is created containing all of the calculated results and projected input data. Regardless of the input method (command line vs parameter file) the output format will be the same.

Example output file:

```
# Bitangent output file

# Mean camber line data
# Format is XYZIJKD
# Parameters u_start: 0.10000000 u_incr: 0.10000000 u_end: 0.90000000 spline order: 3
:camber line circles
18.54215937,7.12237079,0.00000000,0.00000000,0.00000000,1.00000000,5.53793841
27.56719479,9.54572435,0.00000000,0.00000000,0.00000000,1.00000000,6.89066528
34.96649992,11.17347027,0.00000000,0.00000000,0.00000000,1.00000000,7.89176466
42.08822790,12.35181074,0.00000000,0.00000000,0.00000000,1.00000000,8.42320194
49.29173093,13.12146227,0.00000000,0.00000000,0.00000000,1.00000000,8.45122132
56.63009513,13.33180602,0.00000000,0.00000000,0.00000000,1.00000000,8.08756141
64.02702266,12.81879627,0.00000000,0.00000000,0.00000000,1.00000000,7.47160650
71.55143688,11.56806194,0.00000000,0.00000000,0.00000000,1.00000000,6.65959710
80.30479003,9.58108958,0.00000000,0.00000000,0.00000000,1.00000000,5.58344578

# Mean camber line extrapolation
# Format is XYZ
# Parameters extrapolation_sample: 5.00000000 extrapolation_length: 5.00000000
:camber line points
-15.08842964,-3.11884437,0.00000000
-9.52935636,-1.26456221,0.00000000
-3.95104452,0.53101427,0.00000000
1.64588598,2.26768553,0.00000000
7.26081311,3.94525856,0.00000000
12.89311288,5.56354693,0.00000000
18.54215937,7.12237079,0.00000000
27.56719479,9.54572435,0.00000000
34.96649992,11.17347027,0.00000000
42.08822790,12.35181074,0.00000000
49.29173093,13.12146227,0.00000000
56.63009513,13.33180602,0.00000000
64.02702266,12.81879627,0.00000000
71.55143688,11.56806194,0.00000000
80.30479003,9.58108958,0.00000000
85.80323260,8.28069936,0.00000000
91.28313039,6.90425258,0.00000000
96.74343139,5.45201348,0.00000000
```

Bitangent Users Guide

```
102.18308739,3.92426084,0.00000000
107.60105411,2.32128795,0.00000000
112.99629146,0.64340254,0.00000000
```

```
# Maximum circle diameter
# Format is XYZIJKD
:maximum diameter
45.85600378,12.81522938,0.00000000,0.00000000,0.00000000,1.00000000,8.49253073
```

```
# Intersection radii 1
# First line is center point of circle. Format is XYZIJK
# Subsequent lines are intersections points, normal vector at intersection, and edge thickness
# Format is [R] XYZIJKT where 'R' is the nominal radius and 'T' is edge thickness normal to camber line
# Failed intersections are shown as [R] <intersection not found>
# Failed edge thickness are shown as [R] XYZIJK<edge thickness not found>
:intersection radii 1
0.00000000,1.50000000,0.00000000,0.00000000,0.00000000,1.00000000
[5.00000000] 4.70836004,3.18266159,0.00000000,-0.28626668,0.95814998,0.00000000,3.54856964
[10.00000000] 9.51012118,4.59153614,0.00000000,-0.27615015,0.96111451,0.00000000,4.23796689
[15.00000000] 14.32225726,5.95791160,0.00000000,-0.26600293,0.96397222,0.00000000,4.93312782
[20.00000000] 19.14527803,7.28431694,0.00000000,-0.25932847,0.96578918,0.00000000,5.64377111
[25.00000000] 23.97609034,8.58146046,0.00000000,-0.25932847,0.96578918,0.00000000,6.36856224
[30.00000000] 28.82266162,9.82190988,0.00000000,-0.21484904,0.97664727,0.00000000,7.09740478
[35.00000000] 33.71462785,10.89807554,0.00000000,-0.21484904,0.97664727,0.00000000,7.76298759
[40.00000000] 38.65543732,11.78383116,0.00000000,-0.16323779,0.98658676,0.00000000,8.23549023
```

```
# Intersection radii 2
# First line is center point of circle. Format is XYZIJK
# Subsequent lines are intersections points, normal vector at intersection, and edge thickness
# Format is [R] XYZIJKT where 'R' is the nominal radius and 'T' is edge thickness normal to camber line
# Failed intersections are shown as [R] <intersection not found>
# Failed edge thickness are shown as [R] XYZIJK<edge thickness not found>
:intersection radii 2
100.00000000,4.50000000,0.00000000,0.00000000,0.00000000,1.00000000
[5.00000000] 95.18997942,5.86517453,0.00000000,0.25702789,0.96640399,0.00000000,3.57276419
[10.00000000] 90.35411784,7.13760298,0.00000000,0.24361361,0.96987237,0.00000000,4.24129444
[15.00000000] 85.50295678,8.35171506,0.00000000,0.23015257,0.97315456,0.00000000,4.90288194
[20.00000000] 80.63580755,9.50280342,0.00000000,0.23015257,0.97315456,0.00000000,5.55154812
[25.00000000] 75.75889007,10.61298872,0.00000000,0.22136409,0.97519123,0.00000000,6.16943830
[30.00000000] 70.87211791,11.68098071,0.00000000,0.16397360,0.98646473,0.00000000,6.75007304
[35.00000000] 65.92724000,12.50293553,0.00000000,0.16397360,0.98646473,0.00000000,7.29124759
[40.00000000] 60.92101395,13.03421176,0.00000000,0.06918824,0.99760362,0.00000000,7.76230120
```

```
# Intersection radii positive
# First line is center point of circle. Format is XYZIJK
# Subsequent lines are intersections points, normal vector at intersection, and edge thickness
# Format is [R] XYZIJKT where 'R' is the nominal radius and 'T' is edge thickness normal to camber line
# Failed intersections are shown as [R] <intersection not found>
# Failed edge thickness are shown as [R] XYZIJK<edge thickness not found>
:intersection radii positive
50.00000000,9.00000000,0.00000000,0.00000000,0.00000000,1.00000000
[5.00000000] 52.57310485,13.28708937,0.00000000,-0.05041075,0.99872857,0.00000000,8.33455093
[10.00000000] 59.06357795,13.22510994,0.00000000,0.04380293,0.99904019,0.00000000,7.91425291
[15.00000000] 64.52439311,12.74726913,0.00000000,0.14234617,0.98981694,0.00000000,7.43300359
[20.00000000] 69.78862691,11.90004216,0.00000000,0.18507117,0.98272512,0.00000000,6.86926993
[25.00000000] 74.93255879,10.83507814,0.00000000,0.21186582,0.97729876,0.00000000,6.27138585
[30.00000000] 79.99285434,9.65472515,0.00000000,0.22974566,0.97325070,0.00000000,5.63571219
[35.00000000] 84.99488742,8.40178754,0.00000000,0.24385444,0.96981184,0.00000000,4.96838090
[40.00000000] 89.95374336,7.07687972,0.00000000,0.26194175,0.96508369,0.00000000,4.29090018
```

```
# Intersection radii negative
# First line is center point of circle. Format is XYZIJK
# Subsequent lines are intersections points, normal vector at intersection, and edge thickness
# Format is [R] XYZIJKT where 'R' is the nominal radius and 'T' is edge thickness normal to camber line
# Failed intersections are shown as [R] <intersection not found>
```

Bitangent Users Guide

```
# Failed edge thickness are shown as [R] XYZIJK<edge thickness not found>
:intersection radii negative
50.00000000,9.00000000,0.00000000,0.00000000,0.00000000,1.00000000
[5.00000000] 46.87544654,12.90348085,0.00000000,-0.08984860,0.99595543,0.00000000,8.48934704
[10.00000000] 40.49609032,12.11057854,0.00000000,-0.14980492,0.98871557,0.00000000,8.35209947
[15.00000000] 35.16351791,11.20880095,0.00000000,-0.17651147,0.98429858,0.00000000,7.92406501
[20.00000000] 30.03130367,10.11856211,0.00000000,-0.22643443,0.97402641,0.00000000,7.27621835
[25.00000000] 25.00026219,8.88541471,0.00000000,-0.24912642,0.96847097,0.00000000,6.52338861
[30.00000000] 20.03563302,7.53823456,0.00000000,-0.26824862,0.96334972,0.00000000,5.77823164
[35.00000000] 15.12026312,6.10103490,0.00000000,-0.28600321,0.95822866,0.00000000,5.04496481
[40.00000000] 10.23982544,4.62638450,0.00000000,-0.29959704,0.95406583,0.00000000,4.33569670
```

```
# Calculations using tangent to normal vector projected points
# The point normal vectors are not projected only the position of the points
# Projection plane XYZ: 0.00000000,0.00000000,0.00000000
# Projection plane IJK: 0.00000000,0.00000000,1.00000000
```

```
# Projected A data
# Format is XYZIJK
:data a
0.00000000,0.00000000,0.00000000,0.23224817,-0.97265656,0.00000000
2.52072000,0.60073200,0.00000000,0.23126869,-0.97288992,0.00000000
4.89888000,1.16452800,0.00000000,0.23005161,-0.97317843,0.00000000
7.13448000,1.69138800,0.00000000,0.22867289,-0.97350332,0.00000000
...
```

```
# Projected B data
# Format is XYZIJK
:data b
0.00000000,3.00000000,0.00000000,-0.36235581,0.93203984,0.00000000
2.52072000,3.97884000,0.00000000,-0.36149382,0.93237451,0.00000000
4.89888000,4.89936000,0.00000000,-0.36042231,0.93278924,0.00000000
7.13448000,5.76156000,0.00000000,-0.35920792,0.93325756,0.00000000
...
```

Output Calculation Data

The output file contains the following calculated values:

```
:camber line circles
18.54215937,7.12237079,0.00000000,0.00000000,0.00000000,1.00000000,5.53793841
...
```

This represents the position and diameter of a circle that is placed between the two sets of input data. The IJK values represent the projection vector of the circle. The contact points on the edge of the circle are tangent to the contact point from each side of the input data.

```
:camber line points
-15.08842964,-3.11884437,0.00000000
...
```

The camber line point data is similar to the camber line circle data except no diameters are shown and the data is extrapolated at each end by 30% of the camber line circle data length. The extrapolation follows the curvature of the camber line data based on the last 10% of the length.

```
:maximum diameter
45.85600378,12.81522938,0.00000000,0.00000000,0.00000000,1.00000000,8.49253073
```

Bitangent Users Guide

This is the location of the largest Bitangent circle found between the two sets of input data.

```
:intersection radii x
0.00000000,1.50000000,0.00000000,0.00000000,0.00000000,1.00000000
[5.00000000] 4.70836004,3.18266159,0.00000000,-0.28626668,0.95814998,0.00000000,3.54856964
...
```

When one or more radii values are provided an intersection point is calculated between the camber line and each radius. The first line of the data is the center position of the circle used and each subsequent line shows the intersection information in this form:

[Nominal Radius] XYZ IJK Edge Thickness

Where:

[Nominal Radius] is the radial value entered from the center position of the circle.

[XYZ] is the intersection point of the circle and the camber line

[IJK] is the normal vector of the camber line at the point of intersection

[Edge Thickness] is the distance between the input data at the position of the intersection point perpendicular to the camber line.

```
:data x
0.00000000,0.00000000,0.00000000,0.09950315,-0.99503725,0.00000000
...
```

This is the projected input data used for the calculations. Two sets of data are expected; 'data a' and 'data b'. If the input values are already projected then these values will be identical to that of the input. The projection plane information is shown in a comment line prior to the data for reference.

Example of projection plane data comment:

```
# Projection plane XYZ: 0.00000000,0.00000000,0.00000000
# Projection plane IJK: 0.00000000,0.00000000,1.00000000
```

The IJK value of the input data is not projected onto the plane. The intention is to move the point data to a common plane but keep the 3D normal vector. For example, if the input IJK value was 0.000, 0.866, 0.500 then the output IJK value would be the same.