CS 3200

Assignment 2

Lianrui Geng

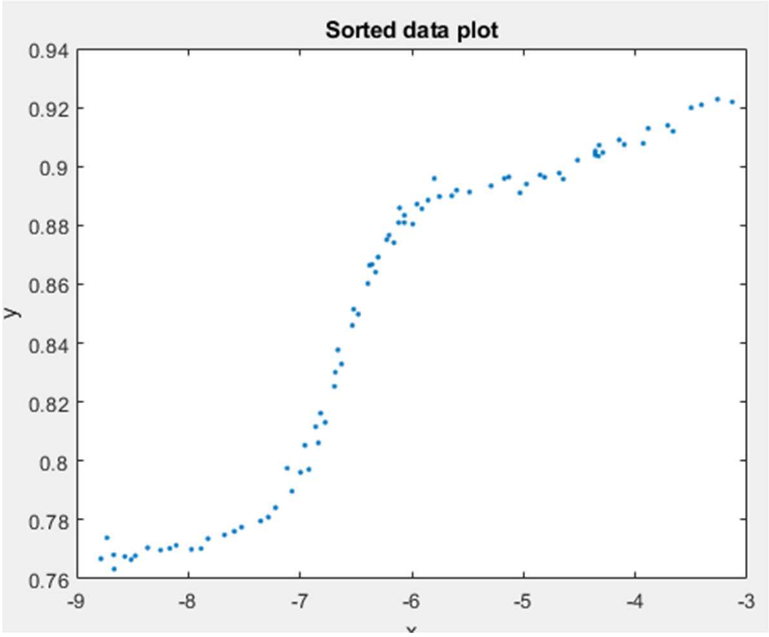
U1346008

Q1_a:

This code is reading data from a text file named "NIST_Filip_trim.txt". The text file contains two columns of data, which are being stored in the yy array. The code uses the fopen function to open the text file, and then uses the fgetl function inside a while loop to read each line of the text file. The sscanf function is used to extract the two values on each line of the text file and store them in the AAA array. The values are then stored in the yy array.

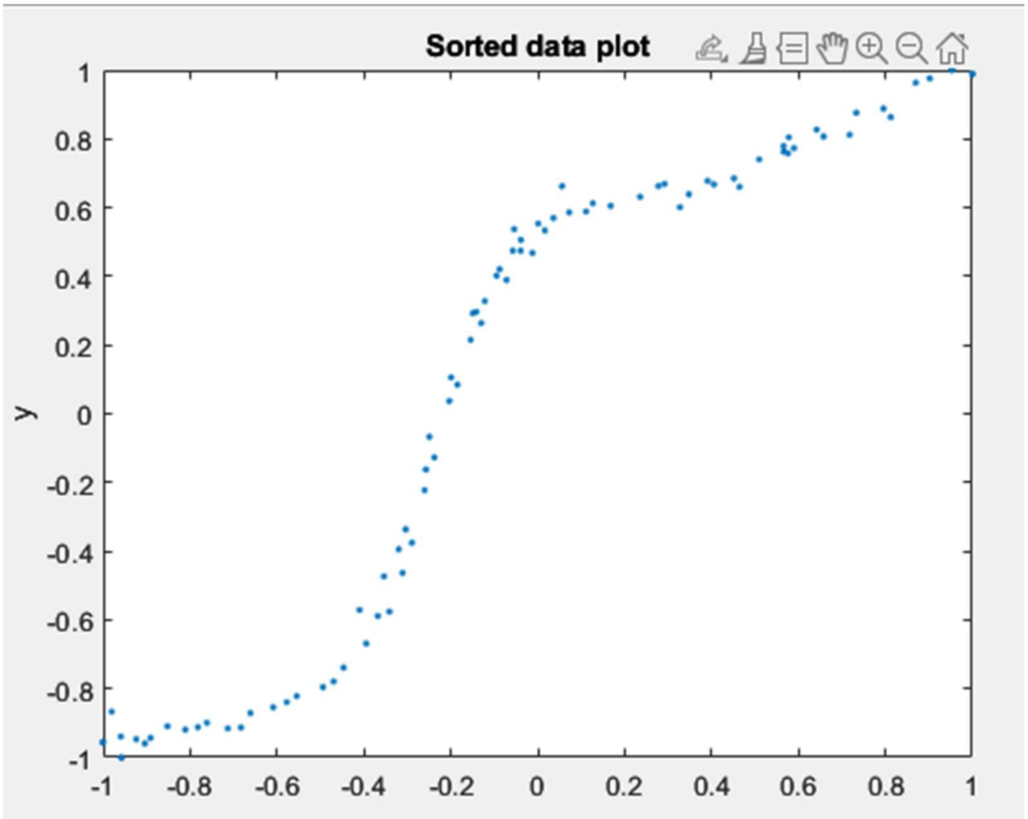
After all the lines of the text file have been read and stored, the code closes the file using the fclose function. The data is then sorted using the sortrows function, with the sorting being done based on the first column of the yy array. The sorted data is stored in the sorted_data array, with the x-values being stored in the x array and the y-values being stored in the y array.

Finally, the code creates a plot of the sorted data using the plot function, with the x-values being plotted on the x-axis and the y-values being plotted on the y-axis. The plot is labeled with x-axis and y-axis labels, and a title.



Q1_b:

This code reads data from a text file named "NIST_Filip_trim.txt". It uses the fopen and fgetl functions to read the data line by line, store it in an array named yy, and sort it based on the first column. The x-values and y-values are then stored in separate arrays named x and y. The values in these arrays are then normalized between -1 and 1 using the min and max functions. Finally, a plot of the normalized and sorted data is created and labeled with x-axis and y-axis labels, and a title.

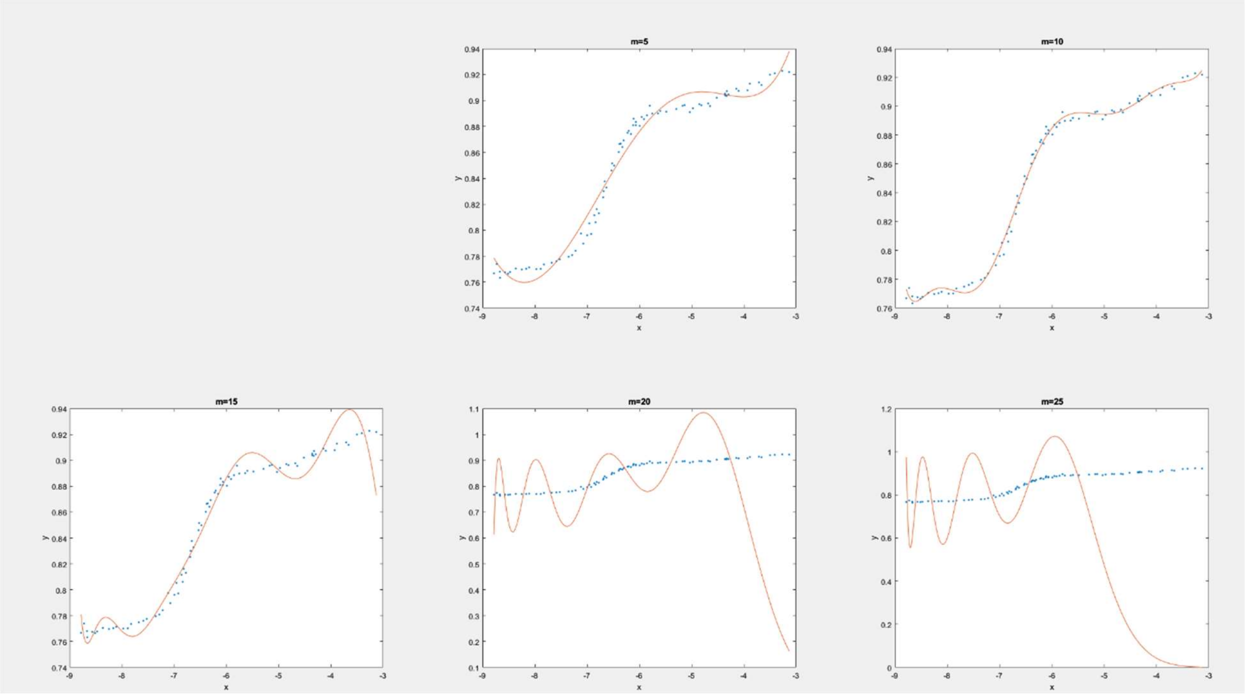


Q1_c:

This code reads data from a text file named "NIST_Filip_trim.txt". It uses the fopen and fgetl functions to read the data line by line and store it in an array named yy. The data is then sorted based on the first column and stored in separate arrays named x and y.

Next, a design matrix named X is created using the x values. The design matrix is used to fit polynomials of different degrees (specified by the m_values array) to the data. For each degree of the polynomial, a linear regression is performed using the \ operator to solve for the coefficients b. The fitted values are stored in the y_hat array.

Finally, the code creates a plot of the original data and the fitted polynomials for each degree of the polynomial. The plots are created using the plot function, and the x-axis and y-axis are labeled with xlabel and ylabel functions, respectively. The title of each plot is set using the title function.

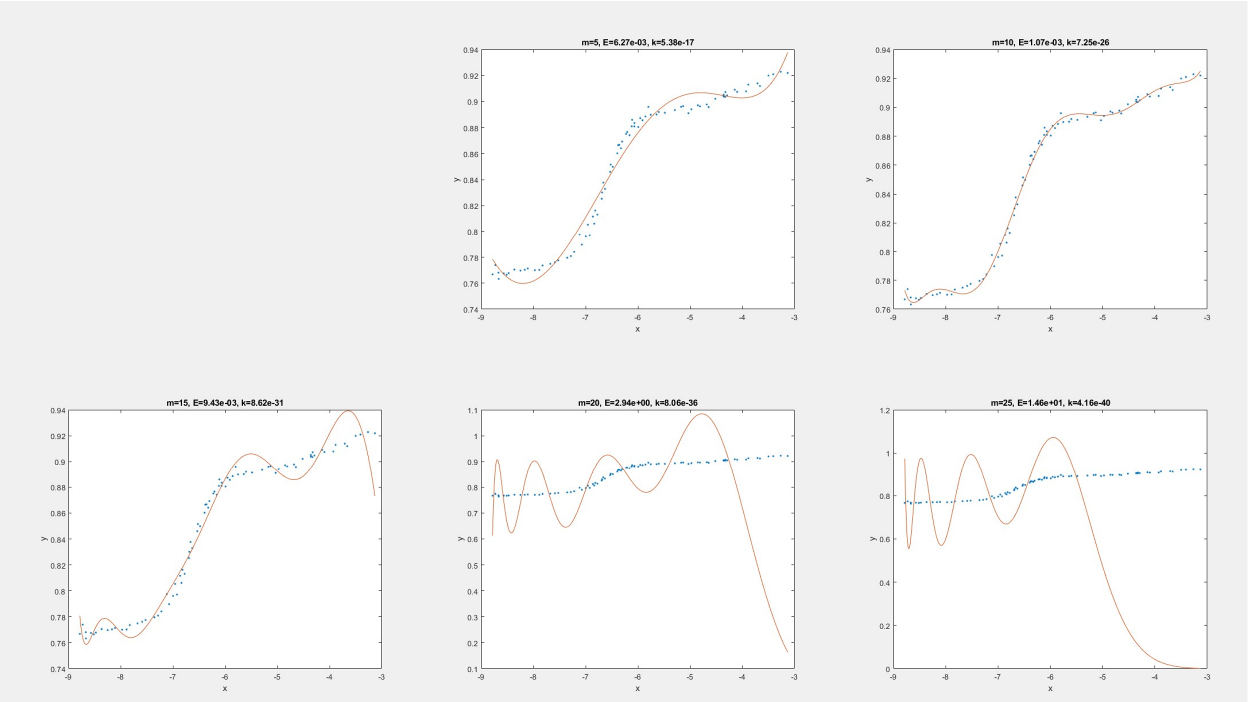


Q1_d:

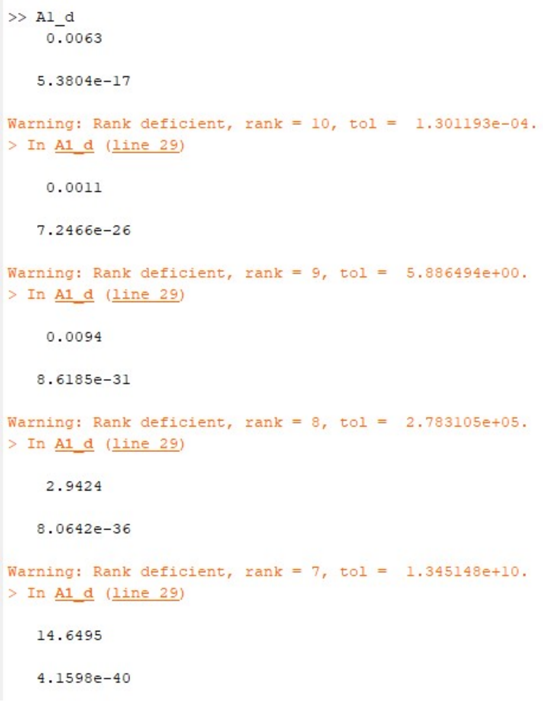
This code reads data from a text file named "NIST_Filip_trim.txt". It uses the fopen and fgetl functions to read the data line by line and store it in arrays named x and y.

A design matrix named X is created using the x values, and linear regression is performed using the \ operator to fit polynomials of different degrees to the data. The fitted values are stored in an array named y_hat, and the residual sum of squares (RSS) is calculated and stored in the E variable. The condition number of the design matrix is also calculated and stored in the k variable.

Finally, the code creates a plot of the original data and the fitted polynomials for each degree of the polynomial. The plots are labeled with x-axis and y-axis labels, a title for each plot, and the degree of the polynomial, RSS, and condition number.



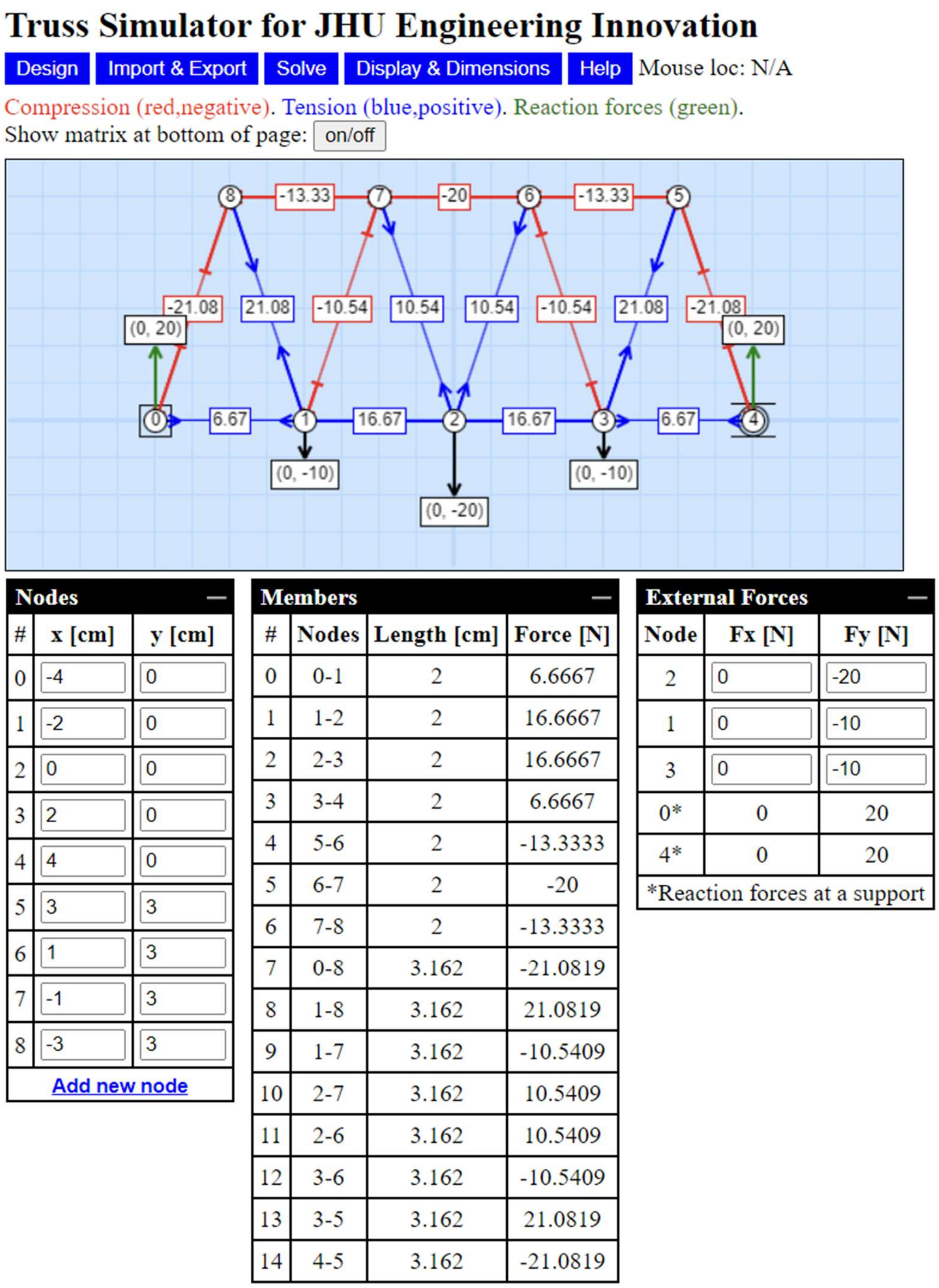
These are the E and k in this part:



```
>> A1_d
    0.0063

   5.3804e-17

Warning: Rank deficient, rank = 10, tol =  1.301193e-04.
> In A1_d (line 29)

    0.0011

   7.2466e-26

Warning: Rank deficient, rank = 9, tol =  5.886494e+00.
> In A1_d (line 29)

    0.0094

   8.6185e-31

Warning: Rank deficient, rank = 8, tol =  2.783105e+05.
> In A1_d (line 29)

    2.9424

   8.0642e-36

Warning: Rank deficient, rank = 7, tol =  1.345148e+10.
> In A1_d (line 29)

   14.6495

   4.1598e-40
```

Q2:

The code above is solving a linear system of equations using the backslash operator (). The matrix A represents the coefficients of the variables and b is the vector of constants. The solution x of the linear system is calculated by dividing b by A, which is done using the backslash operator. The solution x is then displayed using the disp function.



Truss Simulator for JHU Engineering Innovation

Mouse loc: N/A

Compression (red,negative). Tension (blue,positive). Reaction forces (green).

Show matrix at bottom of page: on/off

Nodes

#	x [cm]	y [cm]
0	-4	0
1	-2	0
2	0	0
3	2	0
4	4	0
5	3	3
6	1	3
7	-1	3
8	-3	3

Add new node

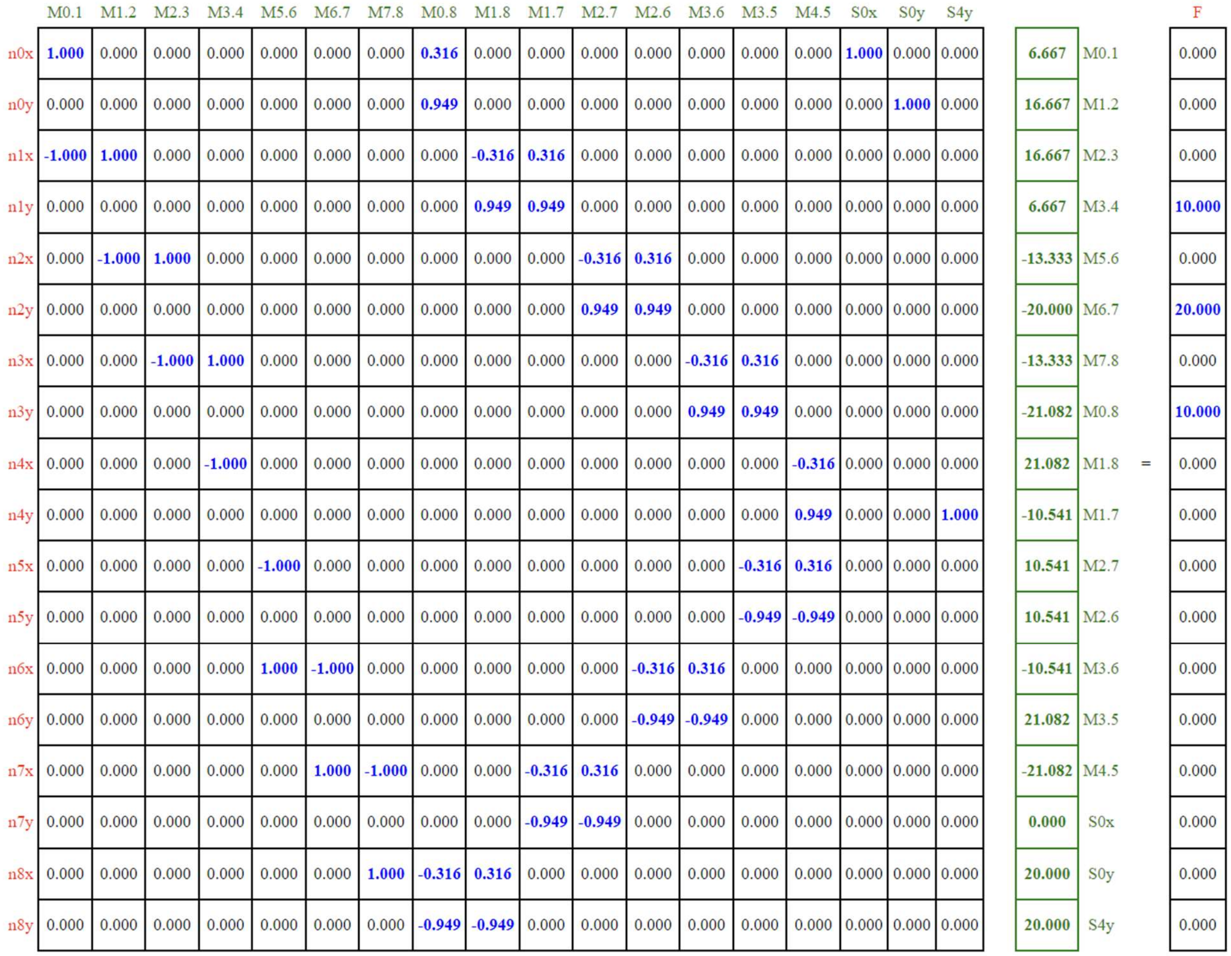
Members

#	Nodes	Length [cm]	Force [N]
0	0-1	2	6.6667
1	1-2	2	16.6667
2	2-3	2	16.6667
3	3-4	2	6.6667
4	5-6	2	-13.3333
5	6-7	2	-20
6	7-8	2	-13.3333
7	0-8	3.162	-21.0819
8	1-8	3.162	21.0819
9	1-7	3.162	-10.5409
10	2-7	3.162	10.5409
11	2-6	3.162	10.5409
12	3-6	3.162	-10.5409
13	3-5	3.162	21.0819
14	4-5	3.162	-21.0819

External Forces

Node	Fx [N]	Fy [N]
2	0	-20
1	0	-10
3	0	-10
0*	0	20
4*	0	20

*Reaction forces at a support



	M0.1	M1.2	M2.3	M3.4	M5.6	M6.7	M7.8	M0.8	M1.8	M1.7	M2.7	M2.6	M3.6	M3.5	M4.5	S0x	S0y	S4y				F
n0x	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.316	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	6.667	M0.1		0.000
n0y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.949	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000	16.667	M1.2		0.000
n1x	-1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.316	0.316	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	16.667	M2.3		0.000
n1y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.949	0.949	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	6.667	M3.4		10.000
n2x	0.000	-1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.316	0.316	0.000	0.000	0.000	0.000	0.000	0.000	-13.333	M5.6		0.000
n2y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.949	0.949	0.000	0.000	0.000	0.000	0.000	0.000	-20.000	M6.7		20.000
n3x	0.000	0.000	-1.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.316	0.316	0.000	0.000	0.000	0.000	-13.333	M7.8		0.000
n3y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.949	0.949	0.000	0.000	0.000	0.000	-21.082	M0.8		10.000
n4x	0.000	0.000	0.000	-1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.316	0.000	0.000	0.000	21.082	M1.8	=	0.000
n4y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.949	0.000	0.000	1.000	-10.541	M1.7		0.000
n5x	0.000	0.000	0.000	0.000	-1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.316	0.316	0.000	0.000	0.000	10.541	M2.7		0.000
n5y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.949	-0.949	0.000	0.000	0.000	10.541	M2.6		0.000
n6x	0.000	0.000	0.000	0.000	1.000	-1.000	0.000	0.000	0.000	0.000	0.000	-0.316	0.316	0.000	0.000	0.000	0.000	0.000	-10.541	M3.6		0.000
n6y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.949	-0.949	0.000	0.000	0.000	0.000	0.000	21.082	M3.5		0.000
n7x	0.000	0.000	0.000	0.000	0.000	1.000	-1.000	0.000	0.000	-0.316	0.316	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-21.082	M4.5		0.000
n7y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.949	-0.949	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	S0x		0.000
n8x	0.000	0.000	0.000	0.000	0.000	0.000	1.000	-0.316	0.316	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	20.000	S0y		0.000
n8y	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-0.949	-0.949	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	20.000	S4y		0.000

This is my code get:

```
>> A2
    6.6596
   16.6491
   16.6491
    6.6596
  -13.3193
  -19.9789
  -13.3193
  -21.0748
   21.0748
  -10.5374
  10.5374

  10.5374

 -10.5374

  21.0748

 -21.0748

  -0.0000

  20.0000

  20.0000
```

Which is the data shown on the website, means I got the correct data.