Jurnal Teknologi

Full paper

2D-Automatic Composition of Nodal Values from Numerical Model Using Script Programming

Rudi Heriansyah[a*], U. U. Sheikh[b], S. A. R. Abu-Bakar[b]

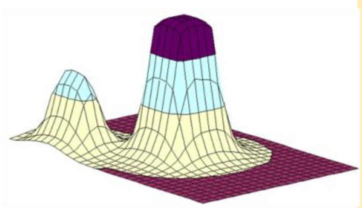
[a]*College of Engineering at Al-Lith, Umm Al-Qura University, Makkah, Saudi Arabia*
[b]*Computer Vision, Video & Image Processing Lab., Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia*

*Corresponding author: rudihe@gmail.com

Article history

Received :5 October 2012
Received in revised form :
29 November 2012
Accepted :15 April 2013

Graphical abstract



Abstract

This paper proposes an algorithm for automatic composition of nodal values obtained from the commercial software of finite element (FEM) and finite difference (FDM) modeling. The composed nodal values will be in two-dimensional form as this is the format used for uniform shaped model (square or rectangular). Since numerical software usually have facilities to save the data in a spreadsheet format, the proposed algorithm is implemented in this environment by using spreadsheet script programming.

Keywords: Finite element; numerical modeling; spreadsheet programming

Abstrak

Paper ini mencadangkan satu algoritma untuk penyusunan automatik nilai-nilai nod yang diperolehi daripada perisian komersial elemen hinggaan dan elemen batas. Nilai-nilai nod akan disusun dalam bentuk dua dimensi kerana ini biasanya format untuk model yang berbentuk teratur (segi empat atau bujursangkar). Kerana perisian numerik berkenaan biasanya mempunyai kemudahan untuk menyimpan data dalam format lembar-berlajur, algorithma yang dicadangkan diimplementasikan dalam persekitaran ini menggunakan pengaturcaraan lembar-berlajur.

Kata kunci: Elemen terhinggaan; pemodelan numeric; pemograman lembar-berlajur

1.0 INTRODUCTION

There are many commercial software to perform numerical modeling based on finite element (FEM) and finite difference (FDM) methods. It is often a requirement to the designer, that the values of the individual nodes in the numerical model are known. Usually, these software provide two methods to achieve this; firstly, by clicking directly onto the nodes of interest and secondly, by saving or exporting the whole nodal values to an external file. The former way is appropriate for models with small number of nodes, but as the number of nodes increases, it is no longer an efficient or effective way. Through the latter method, all nodal values are obtained, however the values are one-dimensional, and in some cases, only certain nodal values are required for presentation.

Commerical numerical modelling softwares based on FEM [1 - 5] and FDM [6] have the ability to save and export node values to external data files for further analysis. This is important to obtain individual node values in a certain area of interest and to make statistical interpretation on these data in another application, because such capabilities are usually not found in these commercial softwares.

In COSMOSWorks [5], the extraction of node values is possible by two ways: node-wise and area- or surface-wise. The first way is by *clicking* directly (while *Probe* window is open as in Figure 1-left) on the nodes of interest and save them in a *comma separated values* (CSV) or *plain text* (TXT) file format. The second way is to open the *List Selected* window (Figure 1-right), and then to *click* on the surface of interest, press *Update*, and then *Save* all data in the desired format (CSV or TXT). *Plain text* files can usually be opened by any word processor software such as *Notepad* in Microsoft® Windows. CSV files arranges data in a comma-delimited format will save the data in the way of which is supported by most spreadsheet and statistical softwares available today.

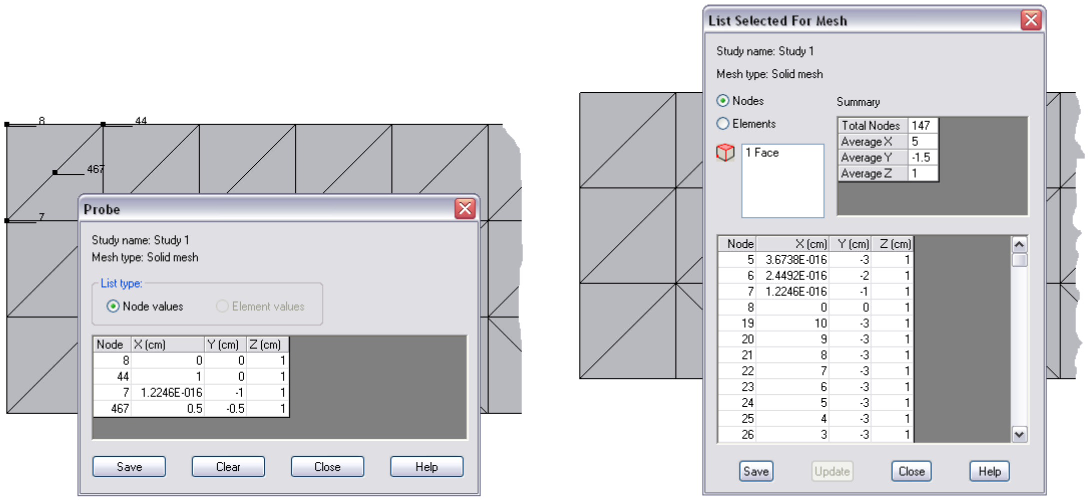


Figure 1 *Save* and export nodal values using *Probe* (left) and *List Selected* dialog box (right)

Like many other spreadsheet programs in the market, Microsoft® Excel [7] supports CSV format as well, in fact it was the first spreadsheet program to do so. Automation for the common tasks in Microsoft® Excel (and also in any Microsoft® Office family members) can be done using traditional *Macro* recording or for advance application, by using Visual Basic for Applications (VBA) scripting programming [8, 9]. VBA is basically similar to VB (Visual Basic) programming language, but the latter was designed for the development of stand alone applications. Both languages are visual programming languages which are event-driven and commonly known as RAD (rapid application development) environment. They were developed based on the legendary BASIC (Beginner's All-purpose Symbolic Instruction Code) language, which was famous in the past due to its simplicity and because it is easy to understand.

This paper demonstrates the use of VBA programming for composing exported data of FEM model (mainly COSMOSWorks) in two dimensional form and then to display them automatically from within application. The ways to implement this task is described step by step and the coding also given in this text.

■2.0 MESH AND NODES IN A NUMERICAL MODEL

In FDM applications, the generated meshes of the model commonly are in regular shapes (square or rectangular), hence nodal points which composes a mesh will always consist of regular node numbers (Figure 2). Although it is possible to mesh the model with curved or irregular boundaries, this is very difficult to implement [10, 11]. FEM then overcomes this limitation and is suitable for the application of deformed shapes, since it has the ability to handle truly arbitrary geometry. Some of its next most important features are the ability to deal with general boundary conditions and to include nonhomogeneous and anisotropic materials [12].

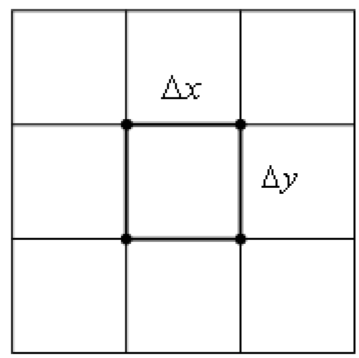


Figure 2 Typical finite difference mesh

There are three types of element (mesh) in FEM. The elements are one-dimensional (line segment), two-dimensional (triangle and rectangular/quadrilateral) and three-dimensional elements (brick, tetrahedron, and cylindrical) [13]. Figure 3 shows several of these element (meshes) types along with their nodals points [14].

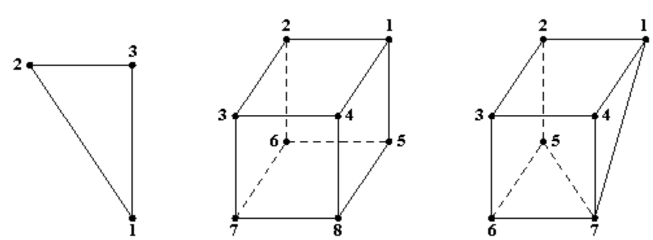


Figure 3 Finite element meshes (left to right): triangular, brick, and tetrahedron

In COSMOWorks, for *solid* meshes of bulky models, all elements are tetrahedrals with straight or curved edges. Triangular shapes with straight or curved edges and a uniform thickness is employed when generating *shell* mesh [5]. The size of generated elements depends on the size of the model and the initial predetermined size of element before generation along with its tolerance setting. Figure 4 shows a model with the size of 10 cm × 3 cm. The type of element is triangular. From the figure (second from the top) it is obvious that if the width and the height of the element are equal then the element will have regular shapes (in this case, two triangular elements will form one square shape). This paper will only concentrate on the 2-D composition nodal values from a model with these regular sized elements, since in many situations, these discrete nodal values are applicable.

3.0 IMPLEMENTATION

As mentioned previously, VBA programming language is employed for the purpose of composing the nodal values in 2-D form from a numerical model. In any Microsoft Office application, VBA Editor can be launched by accessing menu Tools>Macro>Visual Basic Editor, or by pressing the hot-key [Alt+F11]. Figure 5 shows this editor window. A good introduction on how to program in VBA environment can be referred in [8, 9, 15].

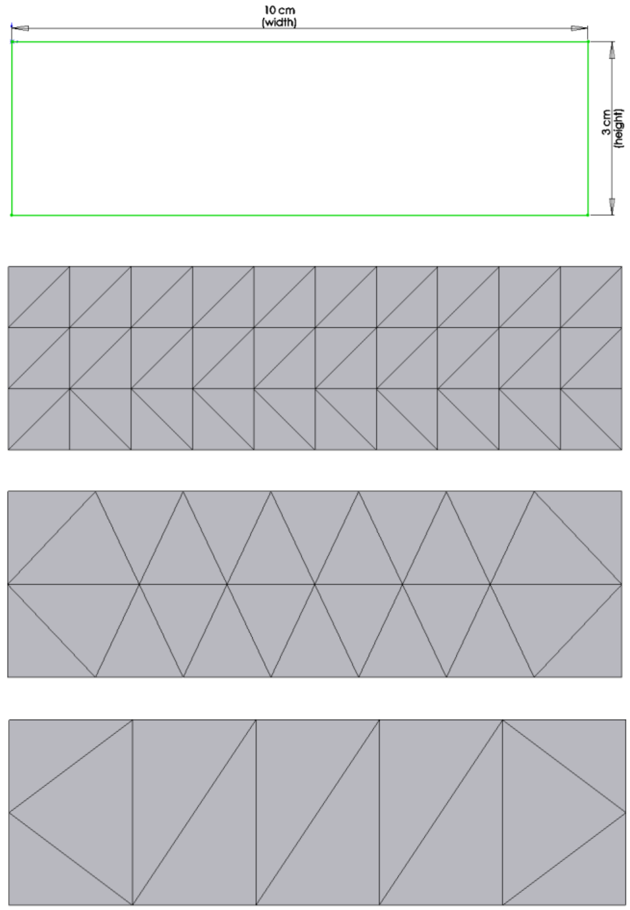


Figure 4 2-D model (10 cm × 3 cm) with generated elements (from top to bottom): model, 1 cm, 1.5 cm , 2 cm element size respectively

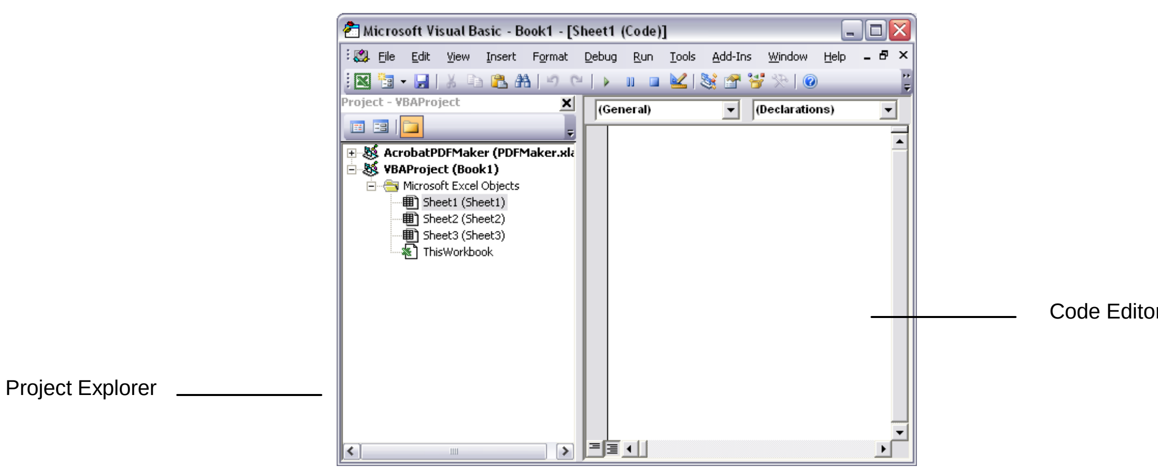


Figure 5 VBA editor window

3.1 Model and Data Saving

Figure 6 shows the model for this study along with its elements. The size of the model is 30 cm × 20 cm with 1 cm of thickness, and simulated spalling defects with uniform inner wall surface temperature of 100^{o}C. At the same time, convection and radiation occur at the outer surface wall with ambient temperature of 30^{o}C.

Size of element is 1 cm with the tolerance of 1×10^{-8} cm. Therefore for this model, the total number of elements are 5135 elements with 9820 nodes. The data is saved by using *List Selected* window (refer to Figure 1) from within COSMOSWorks. Figure 7 shows the saved data. The data is placed in five columns: *A, B, C, D* and *E*. All the files generated for any model will have the same format. But what we need is just the data in the first four columns, the *Z*-axis data (column E) is the same for every row as this is the surface where the interested nodal values are located. Another important data is the beginning (*bgnX*, *bgnY*) and the ending (*endX*, *endY*) coordinates. This can be obtained by using the *Probe* window, and by clicking on the top-left and bottom-right nodes on the model.

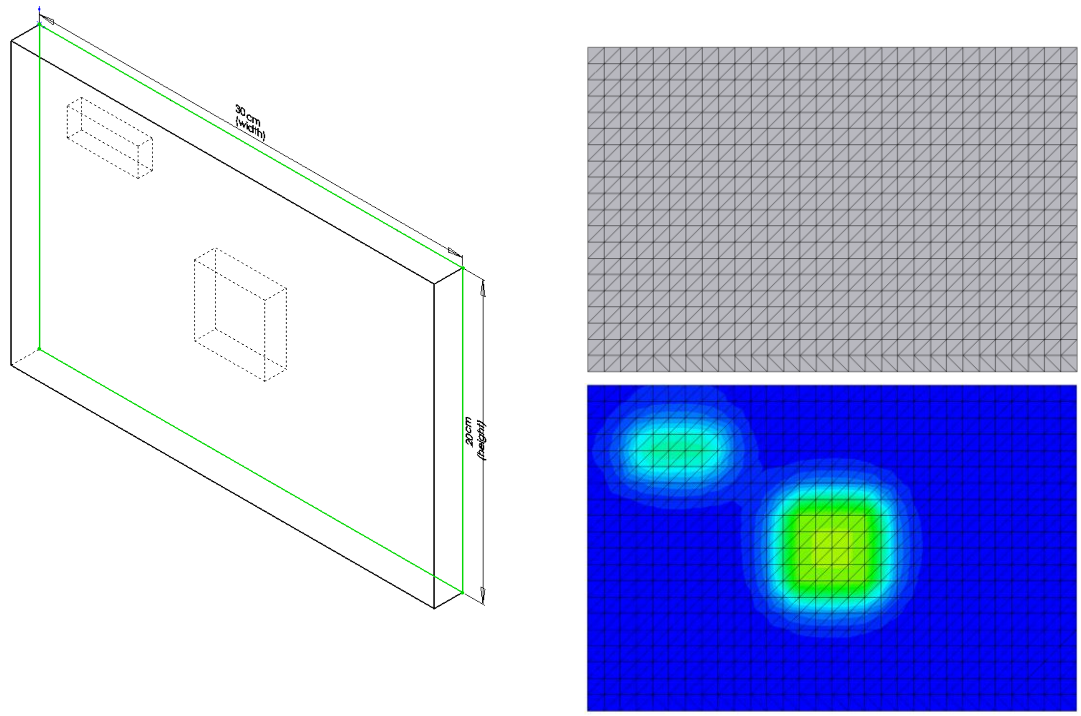
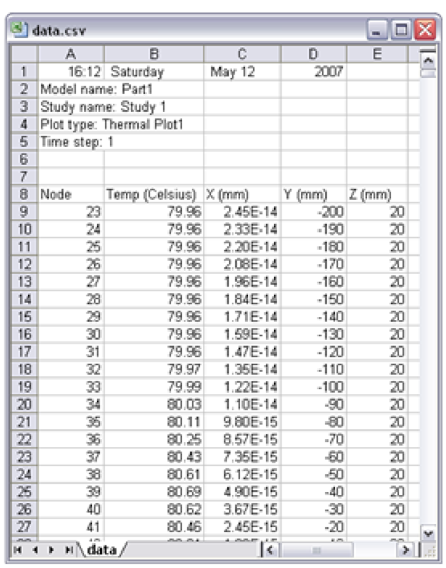


Figure 6 The model (left), the front-view of its mesh (right-top), simulation result (right-bottom)



data.csv

	A	B	C	D	E
1	16:12	Saturday	May 12	2007	
2	Model name: Part1				
3	Study name: Study 1				
4	Plot type: Thermal Plot1				
5	Time step: 1				
6					
7					
8	Node	Temp (Celsius)	X (mm)	Y (mm)	Z (mm)
9	23	79.96	2.45E-14	-200	20
10	24	79.96	2.33E-14	-190	20
11	25	79.96	2.20E-14	-180	20
12	26	79.96	2.08E-14	-170	20
13	27	79.96	1.96E-14	-160	20
14	28	79.96	1.84E-14	-150	20
15	29	79.96	1.71E-14	-140	20
16	30	79.96	1.59E-14	-130	20
17	31	79.96	1.47E-14	-120	20
18	32	79.97	1.35E-14	-110	20
19	33	79.99	1.22E-14	-100	20
20	34	80.03	1.10E-14	-90	20
21	35	80.11	9.80E-15	-80	20
22	36	80.25	8.57E-15	-70	20
23	37	80.43	7.35E-15	-60	20
24	38	80.61	6.12E-15	-50	20
25	39	80.69	4.90E-15	-40	20
26	40	80.62	3.67E-15	-30	20
27	41	80.46	2.45E-15	-20	20

data

Figure 7 The CSV data imported by Microsoft® Excel

3.2 Coding and Result

In our approach, we employ one workbook and two sheets: the first one is to locate all required controls and the composed nodal values (named "Composer") and the second one is the sheet of the source data (named "Data"). It is also worth to note that in this paper, we are just interested on the nodes which have a separating distance of 1 cm from each other (in *X*-*Y* direction only), hence the elements of interest are just (30/1) × (20/1) = 600 elements or (30+1) × (20+1) = 651 nodes. The steps for the compositional task are as follows.

Step 1 *Store all one dimensional values* – The source data (in column A, B, C, D in sheet "Data") is stored into certain variables, the code in Figure 8 will retrieve all nodal values (2500 nodes) from the sheet "Data". datNum is number of nodal values and t, x, y are the variables to store the temperature and *X*-*Y* coordinates (node labels) respectively.

```
datNum = 1
With Worksheets("Data")
       While .Cells(8 + datNum, 2) <> ""
              t(datNum) = .Cells(8 + datNum, 2)
              x(datNum) = .Cells(8 + datNum, 3)
              y(datNum) = .Cells(8 + datNum, 4)

              datNum = datNum + 1
       Wend
End With
```

Figure 8 Retrieve all nodal values

Step 2 *Initialize the starting and ending nodal labels and also the element size* - Next is to initialize the values for bgnX, bgnY, endX, endY, and dis (distance between node, for this model dis is 1 cm to give a fine meshing results) to the variables. The relation between these parameters are shown in Figure 9. The values of bgnX, bgnY, endX, and endYare obtained by clicking the nodes in the model using *Probe* window (Figure 1) and the dis value is known when designing the model. These values are then stored in the worksheet Composer and retrieved using the code in Figure 10.

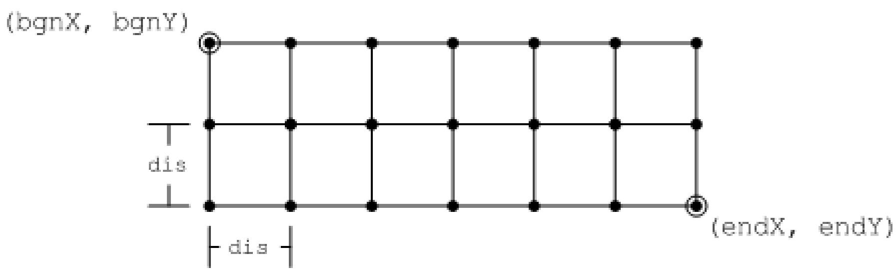


Figure 9 Initial parameter values situated in the model

```
With Worksheets("Composer")
       sheetName = .Range("K2")
       bgnX = .Range("N2")
       endX = .Range("N3")
       bgnY = .Range("Q2")
       endY = .Range("Q3")
       dis = .Range("S3")
End With
```

Figure 10 Initialization of the beginning and ending of nodes label

Step 3 *Generation of all node labels* – From the initial values obtained from the previous step, the labels for the whole nodes can be generated (see code in Figure 11). cntx will be the number of nodes in *X*-direction and cnty is the number of nodes in *Y*-direction.

```
If bgnX > endX Then
       disx = -dis
Else
       disx = dis
End If

cntx = 0
For i = bgnX To endX Step disx
       cntx = cntx + 1
       xc(cntx) = i
Next

cnty = 0
For i = bgnY To endY Step -dis
       cnty = cnty + 1
       yc(cnty) = i
Next
```

Figure 11 Generate all node labels in the model

Step 4 *Retrieve all nodal values and 1-D composition* - After the nodal labels for the whole nodes are generated, the next step is to retrieve the interested nodal values only (651 nodes) and then compose them in 1-D format, and sorted in ascending order. Code in Figure 12 facilitates this operation. Note that tol is the tolerance value (determined from within COSMOWorks for generation of the meshes), for the sample model this value is 1×10^{-8} cm. This tolerance value is needed because the actual nodal labels generated by COSMOSWorks meshing engine for certain nodes are in fraction numbers, but the nodal labels generated by our program are in integers for calculation simplification purpose. Thus, any difference less or equal to this tol will be considered as the same nodal values.

```
' Retrieve
cnt = 0
For i = 1 To datNum - 1
For m = 1 To cntx
       If (xc(m) >= x(i) - tol) And (xc(m) <= x(i) + tol) Then
               For n = 1 To cnty
                      If (yc(n) >= y(i) - tol) And (yc(n) <= y(i) + tol) Then
                     cnt = cnt + 1
                     tt(cnt) = t(i)
                     xx(cnt) = x(i)
                     yy(cnt) = y(i)
                       End If
               Next
       End If
       Next
Next

'Compose in 1-D format
With Worksheets("Compose")
       For i = 1 To cnt
              .Cells(5 + i, 2) = tt(i)
              .Cells(5 + i, 3) = xx(i)
              .Cells(5 + i, 4) = yy(i)
       Next
End With

'Sort - ascending
endRow = 5 + cnt
With Worksheets("Compose")
       .Range(.Cells(6, 2), .Cells(endRow, 4)).Select
End With

Selection.Sort Key1:=Range("C6"), Order1:=xlAscending, Key2:=Range("D6") _
       , Order2:=xlDescending
```

Figure 12 Retrieve the nodal values of interest and compose them in 1-D

Step 5 *2-D Composition and chart display* – Code in Figure 13 is dedicated to compose the 2-D version of these nodal values and also to automatically display the nodal values as a surface type chart. The results are shown in Figure 14 and 15 respectively.

```
' Compose the nodal value in 2-D format
With Worksheets("Compose")

        For i = 1 To cntx
              For j = 1 To cnty
                     .Cells(6 + j, 6 + i) = .Cells((5 + j) + syc * (i - 1), 2)
              Next
        Next
End With

' Display the chart
With Charts("Chart")
       .SetSourceData Source:=Sheets("Compose").Range(Cells(7, 7), _
              Cells(6 + cnty, 6 + cntx))
       .ChartTitle.Characters.Text = "Data"
       .Activate
End With
```

Figure 13 Compose in 2-D format and display it as a chart

2-D Composer for Numerical Nodal Values (CVVIP, FEE UTM).xls

	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
6															
7	80.25	80.31	80.48	80.69	80.86	80.96	80.97	80.88	80.7	80.45	80.25	80.12	80.04	80.01	79
8	80.31	80.4	80.66	81.03	81.28	81.41	81.42	81.32	81.05	80.6	80.32	80.15	80.06	80.02	
9	80.46	80.65	81.35	82.27	82.97	83.16	83.14	82.89	82.17	81.3	80.55	80.24	80.11	80.06	80
10	80.62	80.95	82.21	84.69	85.92	86.1	86.09	85.8	84.6	82.19	80.9	80.37	80.2	80.15	80
11	80.69	81.08	82.72	85.65	87.1	87.34	87.31	87.08	85.65	82.6	81.09	80.5	80.39	80.38	80
12	80.61	80.88	82.13	84.67	85.9	86.04	85.94	85.72	84.51	82.34	81.05	80.72	80.82	81.03	81
13	80.43	80.62	81.43	82.19	82.87	83.05	83.18	83.08	82.21	81.51	80.96	81.17	81.81	82.77	83
14	80.25	80.32	80.58	80.88	81.08	81.29	81.29	81.15	81.01	80.83	81	81.92	84.54	87.22	87
15	80.11	80.13	80.23	80.33	80.41	80.46	80.47	80.47	80.47	80.59	81.11	82.78	87.17	92.5	92
16	80.03	80.04	80.07	80.1	80.13	80.16	80.17	80.19	80.27	80.52	81.22	83.05	87.87	92.99	93
17	79.99	79.99	80	80.02	80.03	80.04	80.06	80.09	80.2	80.48	81.21	83.2	87.92	92.98	93
18	79.97	79.97	79.98	79.98	79.99	79.99	80.01	80.05	80.16	80.42	81.16	83.06	88.13	93.02	93
19	79.96	79.96	79.96	79.97	79.97	79.98	79.99	80.02	80.12	80.36	80.99	82.7	87.17	92.48	92
20	79.96	79.96	79.96	79.96	79.96	79.97	79.98	80.01	80.08	80.26	80.65	81.87	84.27	87.23	87
21	79.96	79.96	79.96	79.96	79.96	79.96	79.97	79.99	80.04	80.15	80.41	80.92	81.84	82.65	83
22	79.96	79.96	79.96	79.96	79.96	79.96	79.97	79.98	80.01	80.07	80.19	80.4	80.7	81	81
23	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.97	79.98	80.01	80.07	80.15	80.25	80.36	80
24	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.97	79.98	80.01	80.04	80.08	80.11	80
25	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.97	79.98	79.99	80	80.01	80
26	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.97	79.97	79.98	79.98	79
27	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.96	79.97	79.97	79.97	79
28															

Composer / Data / Chart

Figure 14 2-D composed nodal values

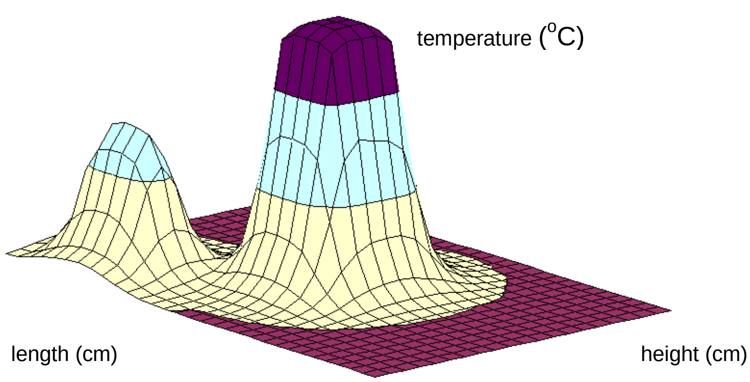


Figure 15 Derived chart from the data in Figure 14

4.0 CONCLUSION

This paper has demonstrated a way to compose nodal values from a numerical data set in an efficient and effective manner. The important parameters to make this task possible are bgnX, bgnY, endX, endY, tolerance (tol), distance between nodes (dis), and 1-D data in CSV format. Spreadsheet programming is suitable for this purpose, since the 1-D data format exported by numerical software is in spreadsheet format. Another advantage by using spreadsheet programming is that there is no need for special programming software for 2-D data composition. The proposed method will work well if the distance of nodes are regular, thus may not work well if distance is irregular such as in a deformed shape model.

Finally, the composed 2-D data can then be used by other applications, for instance to do further analysis using software other than Microsoft Excel.

References

[1] Moaveni, S. 2003. *Finite Element Analysis: Theory and Applications with ANSYS*. Upper Saddle River, NJ: Pearson Eduction.

[2] Stolarski, T. A., Y. Nakasone, and S. Yoshimoto. 2006. *Engineering Analysis with ANSYS Software*. Elsevier Butterworth-Heinemann.

[3] Liu, G. R. and S. S. Quek. 2003. *The Finite Element Method: Practical Course*. Elsevier Butterworth-Heinemann.

[4] FEA Modeling using ABAQUS. 2001. *Seminar on FEA Modelling using ABAQUS*. Singapore.

[5] *Introducing COSMOSWorks*. 2003. Structural Research and Analysis Corporation (SRAC).

[6] *Building your First Model in SINDA/G for FEMAP 9.0*. 2005. Network Analysis, Inc.

[7] Cashman, T. J. & J. S. Quasney. 2004. *Microsoft Office Excel 2003: Comprehensive Concepts and Techniques*. Boston, MA: Thomson/Course Technology.

[8] Kimmel, P. T., S. Bullen, J. Green, R. Bovey, R. Rosenberg, and B. Patterson. 2004. *Excel 2003 VBA Programmer's Reference*. Wrox.

[9] Birnbaum, D. 2005. *Microsoft Excel VBA Programming for the Absolute Beginner*. 2nd Ed. Course Technology.

[10] Ozisik, M. N. 1994. *Finite Difference Heat Transfer*. CRC Press, Inc.

[11] Croft, D. R. and D. G. Lilley. 1977. *Heat Transfer Calculations using Finite Difference Equations*. Applied Science Publishers, Ltd.

[12] Akin, J. E. 2005. *Finite Element Analysis with Error Estimatiors: An Introduction to the FEM and Adaptive Error Analysis for Engineering Students*. Elsevier Butterworth-Heinemann.

[13] ElShayeb, M. and Y. K. Beng. 2000. *Application of Finite Difference and Finite Element Methods for Thermal Problems*. Sabah: Universiti Malaysia Sabah.

[14] Bathe, J. K. 1996. *Finite Element Procedures*. Upper Saddle River, NJ: Prentice-Hall, Inc.

[15] Walkenbach, J. 2007. *Excel 2007 VBA Programming for Dummies*. Indianapolis, Indiana: Wiley Publishing, Inc.