Unstructured Grids

Sometimes it is impossible to organize data as any type of structured grid. Imagine trying to model the reaction of an automobile to an impact with an immovable object. No amount of warping of a 2D matrix or a 3D matrix would be sufficient to make an accurate model of the hood, the windshield, the roof, the doors, or the seats. Researchers performing these types of calculations instead use *unstructured* grids.

In an unstructured grid, every data value contains information not only on its data locations, but also on its *nearest neighbors*. The figure below is identical to figure 10.1 except that we have removed the index values and have added arbitrary index numbers to every data value.

	X		
Y	**1** 0.0350 (0.5,0.5)	**4** 0.4911 (1.0,0.7)	**7** 0.5744 (1.5,0.5)
	2 0.0714 (0.7,1.0)	**5** 0.2422 (1.0,1.0)	**8** 0.3305 (1.3,1.0)
	3 0.3853 (0.5,1.5)	**6** 0.9207 (1.0,1.2)	**9** 0.8485 (1.5,1.5)

In figure 10.3 , we turn this 2D matrix dataset back into a column dataset, but with two new columns. The first column just lists the arbitrary index numbers we have added for every data value.

Index	X	Y	Speed	Neighbors
1	0.5	0.5	0.0350	2,4
2	0.7	1.0	0.0714	1,3,5
3	0.5	1.5	0.3853	2,6
4	1.0	0.7	0.4911	1,5,7
5	1.0	1.0	0.2422	2,4,6,8
6	1.0	1.2	0.9207	3,5,9
7	1.5	0.5	0.5744	4,8
8	1.3	1.0	0.3305	5,7,9
9	1.5	1.5	0.8485	6,8

The last column is the interesting one. It lists the index values of the nearest neighbors for every data value. This is, in fact, an example of an *unstructured grid*, where every data value carries information not only on data locations, but also on its nearest neighbors.

Unstructured Grids: *Every data value contains data locations and nearest neighbor information.*

In the warped grid example (fig. 10.1), the location of the data value in the (`row, column`) array *implicitly* tells you what the data value's nearest neighbors are. This is the reason we stored the warped grid data as a 2D matrix dataset in the first place, even though it took up as much disk space as the corresponding column dataset.

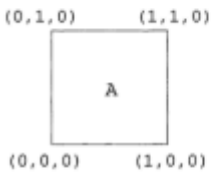
But in unstructured grids, the list of nearest neighbors is *explicit*. There is no need to use the matrix to define the nearest neighbor. Because of this, unstructured grid data are *always* stored in column form as opposed to matrix.

Node-Based Data vs. Cell-Based Data

The unstructured grid example shown in figure 10.3 is of a *node-based* dataset: one in which the *data values* are defined at the *data locations* (known here as *nodes*). This might seem a little silly. Where else would the data values be defined? Aren't data values always defined at the data locations? Not always.

In *cell-based* datasets, the data values are defined not at the data locations, but in a *cell*: some sort of object, usually a *polygon*. A polygon here means a closed surface defined by a series of nodes, or data locations. An example is shown below.

Figure 10.4
Polygonal cell example



In this example, the cell labeled `A` is a polygonal cell with four nodes (sometimes known as *vertices*). The (`X,Y,Z`) values of each node are given in the figure. The data value is defined at the center of the cell, right where the `A` is.