ARMY RESEARCH LABORATORY

Natural Computing: Analysis of Tables for Computer Representation

Som Karamchetty

ARL-TR-2041 February 2000

Army Research Laboratory

Adelphi, MD 20783-1197

ARL-TR-2041 February 2000

Natural Computing: Analysis of Tables for Computer Representation

Som Karamchetty
Information Science and Technology Directorate

Abstract

Some fundamental objects in practical documents have not been implemented in software so that they can be used easily for calculating. One such object is the table—despite the mistaken view that databases are adequate representations of tables. A survey of practical tables found in a variety of real-world documents reveals that many of their useful features are not captured in software. This report proposes data structures and computer representation for table objects. Through the adoption of such structures and representations, practical table objects can be developed for use by domain specialists. Such tables embedded in electronic documents can be used in interactive applications to retrieve data, but most importantly, they can be used as functional representations for copying and pasting into procedures and programs. Use of these table objects, together with other natural computing objects (such as equations, graphs, and procedures), will permit electronic documents like handbooks, textbooks, journals, and bulletins to be used seamlessly for calculations by both domain specialists and naive users. Such developments will reduce the lag between the availability of information and its use in calculations, and encourage the further development of knowledge. Software development for computation will change and its costs will be contained.

Contents

Figures

Tables

1. Introduction

In earlier work (Karamchetty, 1997 and 2000), I have described an approach to software for computation that is based on the way people actually compute, using available information in a variety of formats. This report builds on that discussion by examining one aspect of Natural Computing: the use of tables. Before the invention of computers, most knowledge was captured in the form of books and other paper documents. Books can be further classified as textbooks, reference books, handbooks, and journals, based on the temporal nature of the information. More transient documents are flyers, brochures, and receipts. Information was printed on paper for storage, retrieval, and communication. The paper-based information was read by the end user. By reading the information from one or more documents and by combining it with one's own intuition, invention, and discovery, one generated and printed new information in the form of another paper document.

In the paper-based universe, when people dealt with technical matter, domain knowledge was captured in the form of text containing equations, tables, graphs, and pictures. Without pictures, descriptions of scenes were elaborate; as we recall, "a picture is worth a thousand words." In technical subjects, the pictures could be sketches, schematics, drawings, paintings, or photographs. Sketches stood for descriptions of parts and components in terms of shapes and sizes. Schematics and other diagrams showed the mutual relationships of components in a system and the state of the system and its temporal nature.

Both tables and graphs captured relationships among sets of variables. Graphs additionally provided a highly visual insight into the mutual dependency of the variables. Domain specialists read the text and concurrently used the included tables, graphs, and charts. They used note pads to make temporary notes and calculations. Simple calculations were done mentally. More complicated calculations required aids, such as log tables and slide rules. Domain specialists captured new ideas and information in the form of more equations, tables, graphs, and pictures; appended them to text; and communicated the new documents to others in the field. Documents were subject to three principal types of use: (1) reading and comprehension; (2) interactive calculations using the tables, equations, graphs, and pictures along with the text; and (3) development and recording of new functions (tables, equations, graphs, and pictures). Capturing these essential natural forms and processes in a computer software system is the goal of Natural Computing.

2. A Textbook Example of Calculation Features

Figure 1 shows a sample page from an engineering textbook describing mechanical springs. The page consists of a sketch of a mechanical spring, text, and equations. Figure 2 shows another sample page with a graph, more equations, and more text. By reading the explanations on these pages, an engineer can understand the domain of mechanical springs. By studying the graph on the page, the engineer can understand the trends. At any time, the engineer can obtain values given by the graphs—this activity is usually called reading a value from a graph. While using these pages, the engineer starts with values of D and d, proceeds to calculate the value of the variable C from equation 8-1 in figure 1, and reads the value of the Wahl correction factor K from the graph of spring index versus stress correction factor. This value of K and an input value of F (force) are next substituted into equation 8-4 (fig. 2) and the value of stress (τ) is calculated. The engineer may next proceed to the sample page shown in figure 3 and read the values of A and m for a given material. These values are substituted into equation 8-10 (fig. 3) for calculating the ultimate strength in tension of the spring material.

This description shows how domain specialists present information in textbooks for use by readers in performing calculations. Thus, textbooks are used both to explain the subject and to provide information in the form of text, sketches, equations, graphs, and tables for ready use in calculations.

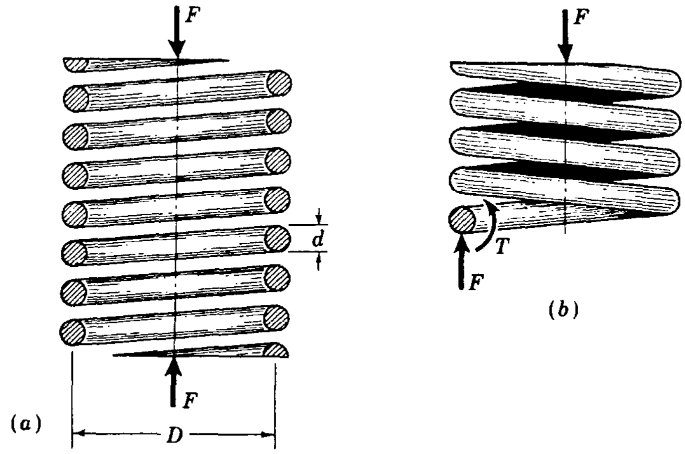


FIGURE 8-1
(*a*) Axially loaded helical spring; (*b*) free-body diagram showing that the wire is subjected to a direct shear and a torsional shear.

the hose in a straight line perpendicular to the plane of the coil. As each turn of hose is pulled off the coil, the hose twists or turns about its own axis. The flexing of a helical spring creates a torsion in the wire in a similar manner.

Using superposition, the maximum stress in the wire may be computed using the equation

$$\tau_{\max} = \pm\frac{Tr}{J} + \frac{F}{A} \qquad (a)$$

where the term Tr/J is the torsion formula of Chap. 2. Replacing the terms by $T = FD/2$, $r = d/2$, $J = \pi d^4/32$, and $A = \pi d^2/4$ gives

$$\tau = \frac{8FD}{\pi d^3} + \frac{4F}{\pi d^2} \qquad (b)$$

In this equation the subscript indicating maximum shear stress has been omitted as unnecessary. The positive signs of Eq. (*a*) have been retained, and hence Eq. (*b*) gives the shear stress at the inside fiber of the spring.

Now define *spring index*

$$C = \frac{D}{d} \qquad (8\text{-}1)$$

as a measure of coil curvature. With this relation, Eq. (*b*) can be arranged to give

$$\tau = \frac{8FD}{\pi d^3}\left(1 + \frac{0.5}{C}\right) \qquad (c)$$

Or designating

$$K_s = 1 + \frac{0.5}{C} \qquad (8\text{-}2)$$

Source: Joseph E. Shigley (1977). *Mechanical Engineering Design*, 3rd ed., McGraw-Hill Book Co., New York, NY.

Figure 1. A sample page containing text, sketch, and equations.

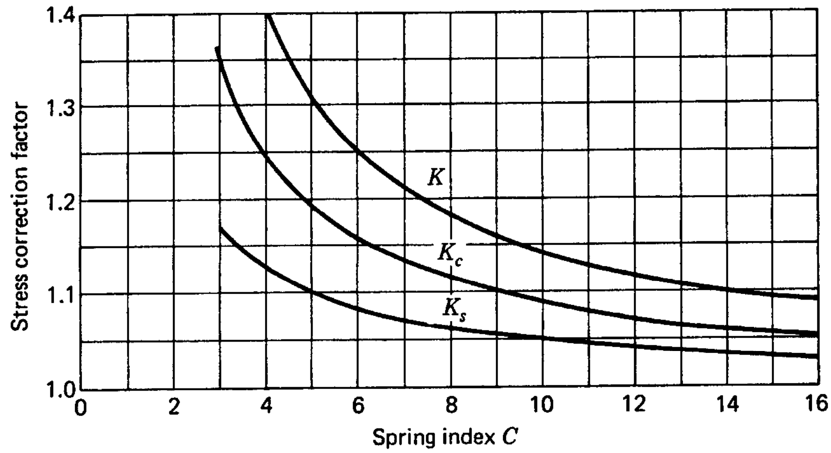


FIGURE 8-2
Values of the stress correction factors for round helical extension or compression springs.

then

$$\tau = K_s \frac{8FD}{\pi d^3} \qquad (8\text{-}3)$$

where K_s is called a *shear-stress multiplication factor*. This factor can be obtained from Fig. 8-2 for the usual values of C. For most springs, C will range from about 6 to 12. Equation (8-3) is quite general and applies for both static and dynamic loads. It gives the maximum shear stress in the wire, and this stress occurs at the inner fiber of the spring.

Many writers present the stress equation as

$$\tau = K \frac{8FD}{\pi d^3} \qquad (8\text{-}4)$$

where K is called the *Wahl correction factor*.* This factor includes the direct shear, together with another effect due to curvature. As shown in Fig. 8-3, curvature of the wire increases the stress on the inside of the spring but decreases it only slightly on the outside. The value of K may be obtained from the equation

$$K = \frac{4C - 1}{4C - 4} + \frac{0.615}{C} \qquad (8\text{-}5)$$

or from Fig. 8-2.

By defining $K = K_c K_s$, where K_c is the effect of curvature alone, we have

$$K_c = \frac{K}{K_s} \qquad (8\text{-}6)$$

Source: Joseph E. Shigley (1977). *Mechanical Engineering Design*, 3rd ed., McGraw-Hill Book Co., New York, NY.

Figure 2. A sample page containing text, graph, and equations.

strengths for various wire sizes and materials.* But the availability of the scientific electronic calculator now makes such a tabulation unnecessary. The reason for this is that a log-log plot of the tensile strengths versus wire diameters is a straight line. The equation of this line can be written in terms of the ordinary logarithms of the strengths and wire diameters. This equation can then be solved to give

$$S_{ut} = \frac{A}{d^m} \qquad (8\text{-}10)$$

where A is a constant related to a strength intercept, and m is the slope of the line on the log-log plot. Of course such an equation is only valid for a limited range of wire sizes. Table 8-2 gives values of m and the constant A for both English and SI units for the materials listed in Table 8-1.

Although the torsional yield strength is needed to design springs, surprisingly, very little information on this property is available. Using an approximate relationship between yield strength and ultimate strength in tension,

$$S_y = 0.75 S_{ut} \qquad (8\text{-}11)$$

and then applying the distortion-energy theory gives

$$S_{sy} = 0.577 S_y \qquad (8\text{-}12)$$

and provides us with a means of estimating the torsional yield strength S_{sy}. But this method should not be used if experimental data are available; if used, a generous factor of safety should be employed, especially for extension springs, because of the uncertainty involved.

Variations in the wire diameter and in the coil diameter of the spring have an effect on the stress as well as on the spring scale. Large tolerances will result in

* See, for example, the second edition of this book: Joseph E. Shigley, "Mechanical Engineering Design," 2d ed., p. 362, McGraw-Hill Book Company, New York, 1972.

Table 8-2 CONSTANTS FOR USE IN EQ. (8-10) TO ESTIMATE THE TENSILE STRENGTH OF SELECTED SPRING STEELS

Material	Size range, in	Size range, mm	Exponent, m	Constant, A	
				kpsi	MPa
Music wire[a]	0.004–0.250	0.10–6.5	0.146	196	2170
Oil-tempered wire[b]	0.020–0.500	0.50–12	0.186	149	1880
Hard-drawn wire[c]	0.028–0.500	0.70–12	0.192	136	1750
Chrome vanadium[d]	0.032–0.437	0.80–12	0.167	169	2000
Chrome silicon[e]	0.063–0.375	1.6–10	0.112	202	2000

[a] Surface is smooth, free from defects, and with a bright lustrous finish.
[b] Has a slight heat-treating scale which must be removed before plating.
[c] Surface is smooth and bright, with no visible marks.
[d] Aircraft-quality tempered wire; can also be obtained annealed.
[e] Tempered to Rockwell C49 but may also be obtained untempered.

Source: Joseph E. Shigley (1977). *Mechanical Engineering Design*, 3rd ed., McGraw-Hill Book Co., New York, NY.

Figure 3. A sample page containing text, table, and equations.

3. Usage of Tables

Tables are ubiquitous in books and paper documents of all types. Although tables have been in use for centuries, research into tables and their properties is nonexistent. Children as young as six years old understand tables when they see a table of items at their favorite ice-cream shop. Little League baseball players understand the scores of their favorite teams when the scores are presented in tables (see fig. 4) (*The Washington Post*, 1996). In general, people use tables quite intuitively and very little instruction is ever provided on tables. Although the structure of tables can become very complex, as I will show in this report, mature adults learn the structure, properties, and relationships of items in a table by trial and error, supplemented by intuition.

BASEBALL

AMERICAN LEAGUE STANDINGS

EAST	W	L	Pct	GB	L10	Strk	Home	Away
New York	27	19	.587	—	5-5	L1	18-8	9-11
Baltimore	27	20	.574	½	7-3	W1	18-11	9-9
Toronto	21	28	.429	7½	3-7	L3	10-13	11-15
Boston	19	28	.404	8½	6-4	L1	13-12	6-16
Detroit	12	38	.240	17	0-10	L11	6-17	6-21
CENTRAL	**W**	**L**	**Pct**	**GB**	**L10**	**Strk**	**Home**	**Away**
Cleveland	33	14	.702	—	8-2	W4	16-5	17-9
Chicago	29	18	.617	4	9-1	W8	16-5	13-13
Milwaukee	22	25	.468	11	5-5	L4	10-10	12-15
Minnesota	22	26	.458	11½	4-6	W3	12-13	10-13
Kansas City	23	28	.451	12	6-4	L2	10-15	13-13
WEST	**W**	**L**	**Pct**	**GB**	**L10**	**Strk**	**Home**	**Away**
Texas	30	19	.612	—	4-6	W2	18-7	12-12
Seattle	26	22	.542	3½	6-4	W1	15-12	11-10
California	23	25	.479	6½	3-7	W1	15-9	8-16
Oakland	22	26	.458	7½	3-7	L1	10-12	12-14

SUNDAY'S RESULTS

- Baltimore 6 .... Oakland 1
- Minnesota 9 .... Toronto 3
- Cleveland 5 .... Detroit 0
- Chicago 12 .... Milwaukee 1
- Texas 6 .... Kansas City 4
- at California 12 .... Boston 2
- at Seattle 4 .... New York 3

MONDAY'S GAMES

- Boston (Wakefield 2-5) at Oakland (Wojciechowski 5-0) 4:05
- Chicago (Fernandez 5-2) at Toronto (Janzen 2-0) 7:35
- Detroit (Gohr 2-6) at Kansas City (Belcher 5-2) 8:05
- Cleveland (McDowell 5-2) at Texas (Pavlik 6-1) 8:35
- New York (Pettitte 6-3) at California (Abbott 1-7) 10:05

TUESDAY'S GAMES

- Baltimore at Seattle 10:05
- Chicago at Toronto 7:35
- Minnesota at Milwaukee 8:05
- Cleveland at Texas 8:35
- New York at California 10:05
- Boston at Oakland 10:05

SATURDAY'S RESULTS

- Oakland 6 .... Baltimore 3
- Minnesota 6 .... Toronto 4 (10)
- Cleveland 7 .... Detroit 6
- Chicago 9 .... Milwaukee 7
- Texas 2 .... Kansas City 1
- Boston 10 .... California 3
- New York 5 .... Seattle 4

NATIONAL LEAGUE STANDINGS

EAST	W	L	Pct	GB	L10	Strk	Home	Away
Atlanta	32	17	.653	—	7-3	W1	20-9	12-8
Montreal	29	21	.580	3½	2-8	L3	16-7	13-14
Philadelphia	24	24	.500	7½	4-6	W1	8-11	16-13
Florida	25	26	.490	8	5-5	W1	16-11	9-15
New York	20	28	.417	11½	5-5	W1	10-11	10-17
CENTRAL	**W**	**L**	**Pct**	**GB**	**L10**	**Strk**	**Home**	**Away**
Houston	25	26	.490	—	5-5	W3	12-14	13-12
St. Louis	22	27	.449	2	6-4	L1	9-12	13-15
Cincinnati	19	25	.432	2½	3-7	L1	10-13	9-12
Chicago	21	29	.420	3½	3-7	L4	15-12	6-17
Pittsburgh	19	30	.388	5	3-7	L1	8-16	11-14
WEST	**W**	**L**	**Pct**	**GB**	**L10**	**Strk**	**Home**	**Away**
San Diego	31	19	.620	—	6-4	L1	17-11	14-8
Los Angeles	27	24	.529	4½	7-3	W3	16-9	11-15
San Francisco	25	23	.521	5	5-5	L1	11-13	14-10
Colorado	23	23	.500	6	8-2	W2	15-10	8-13

SUNDAY'S RESULTS

- Los Angeles 4 .... Montreal 3
- Florida 8 .... St. Louis 2 (7, rain)
- Atlanta 6 .... at Pittsburgh 3 (13)
- New York 1 .... San Diego 0
- Houston 7 .... Chicago 2
- Philadelphia 10 .... at San Francisco 1
- Cincinnati .... at Colorado (ppd. rain)

MONDAY'S GAMES

- Houston (Kile 5-3) at Pittsburgh (Ruebel 0-0) 1:05
- Colorado (Ritz 4-4) at St. Louis (Stottlemyre 4-2) 1:15
- Atlanta (Glavine 5-3) at Chicago (Telemaco 2-0) 4:05
- Cincinnati (Smiley 4-4) at Florida (Leiter 6-4) 4:35
- San Diego (Hamilton 7-3) at Montreal (Fassero 3-4) 7:35

TUESDAY'S GAMES

- Cincinnati at Florida 7:05
- Houston at Pittsburgh 7:05
- San Diego at Montreal 7:35
- Los Angeles at Philadelphia 7:35
- San Francisco at New York 7:40
- Atlanta at Chicago 8:05
- Colorado at St. Louis 8:05

SATURDAY'S RESULTS

- San Diego 7 .... New York 2
- San Francisco 3 .... Philadelphia 2
- Pittsburgh 6 .... Atlanta 2
- St. Louis 5 .... Florida 0
- Los Angeles 5 .... Montreal 3
- Houston 5 .... Chicago 2
- Colorado 7 .... Cincinnati 5

Source: *The Washington Post* (1996). Monday, May 27, p C6.

Figure 4. Status of baseball games on a given day shown by a number of tables.

An example of a simple table is a relationship between two columns of quantities, as shown in table 1. (The relationship can also be between two rows, but I use the column terminology for ease of discussion in this report.)

The basic property of a table is that the item at the top of each column is a descriptor and not a value, while the rest of the items in a column are values. The two columns can also be considered two lists, usually with a correspondence relationship between items in each row. This table represents relationships in a discrete domain; that is, the relationships are meaningful only at the values given in the table, and extensions (interpolation and extrapolation) are neither possible nor meaningful.

Table 2 gives another example of a simple table with two columns; once again, the first item in each column represents a descriptor, while the rest of the items are values, and there is a correspondence relationship between items in each column. Additionally, in such a table, one may be able to extend the relationship to other sets of values between those given in the table. Such a domain is continuous. For example, the value of **Cost** corresponding to an **Area** value of 2.5 can be calculated through interpolation. Several types of interpolations are available. (I do not discuss those various methods in this report.) Similarly, the value of **Cost** corresponding to an **Area** of 6.7 can be calculated by an extrapolation procedure. The important classification is that some tables represent continuous functional relationships and others represent discrete (point-wise) relationships. Database systems focus on the latter representation (a detailed discussion is presented in sect. 14).

In popular vocabulary, tables are also called "tables of data." People say "the data is stored in a table." Such a statement implies that a table is *merely* a storage device. Ordinarily, a table (for example, either table 1 or table 2) stores data, and at the request of a user, the data are retrieved and presented. In such a system, storage is the primary purpose of tables. Database systems have been developed that essentially "store data" and return values upon a user's request. Note the common expression, "data

Table 1. Example table showing two columns of data that hold a discrete relationship.

Office	Person
President	John Johnson
Vice President	Mary Gorton
Treasurer	Catherine Weslock
Secretary	Timothy Wesbury

Table 2. Example table showing two columns of data that hold a continuous relationship.

Area	Cost
1	3
2	6
3	11
4	18
5	27
6	38

is *retrieved* from a database." In this sense (storage), databases and tables are used with synonymous meanings and for identical purposes.

However, it is very important to distinguish between tables that represent data in continuous and discrete domains. Table 2 can be replaced by a functional relationship of the form **Cost** = 2 + (**Area**)^2.

Thus, an equation (a functional relation) and a table become mutually substitutable. This means that a table can be used where an equation can be used. In this sense, a table is more than a data storage device. This calls for a representation for tables beyond databases.

Another example of tables that are much more than storage devices is the set of tables of logarithms. Before the advent of slide rules and calculators, tables of logarithms (see sample in table 3) were widely used for calculations in science and engineering fields. The logarithmic tables contained functional representations. Tables of values of reciprocals, sine, cosine, tangent, sinh, cosinh, tanh, and so forth are other examples of tables of functional relationships used directly to facilitate computations. As mathematics (specifically, algebra) progressed, people preferred to use equations rather than relying on tables of functions. However, this example shows that tables and equations can both represent the same functional relationship between variables in a given problem. Where equations are difficult to fit, tables of data may be the only recourse or else the best means, particularly when dealing with experimental data and empirical observations (such as baseball results).

The history of steam tables shows the evolution of tables of data to equations for representing complex relationships. For two centuries, thermal engineers dealt with steam tables obtained through laborious experimental work. Steam has a thermodynamic state that is defined as a number of properties (pressure, temperature, specific volume, internal energy, enthalpy, and entropy). Given any two of these properties, the state is completely obtainable. Historically, steam tables (fig. 5) have been the main means of representing these relationships between thermal states of steam (Baumeister and Marks, 1964). With the advent of digital computing, the complex functional relationships between these properties have been captured in equations. Since these equations are extremely complex and do not reveal the relationships in the data, the steam tables and charts are very widely used.

These arguments show that tables can be used in place of equations to represent functional relationships. Tables are more than storage devices. Tables are useful in doing calculations. For example, the ideal gas equation of state can be represented by a sample table, as shown in table 4.

Table 3. Sample table of logarithms.

1-18 MATHEMATICAL DATA AND CONVERSION TABLES

LOGARITHMS

If $x = \log_e n$, then $e^x = n$
$\log ab = \log a + \log b$
$\log a/b = \log a - \log b$
$= -\log b/a$
$\log 1/n = -\log n$

$\log n^c = c \log n$
$\log \sqrt[c]{n} = (1/c) \log n$
$\log_e x = 2.3026 \log_{10} x$
$\log_{10} x = 0.4343 \log_e x$
$e = 2.718282$

$\log 1 = 0$
$\pi = 3.141593$
$\log_{10} \pi = .497150$

LOGARITHMS TO BASE 10

Number	0	1	2	3	4	5	6	7	8	9	Proportional Parts								
											1	2	3	4	5	6	7	8	9
1.0	0000	0043	0086	0128	0170	0212	0253	0294	0334	0374	4	8	12	17	21	25	29	33	37
1.1	0414	0453	0492	0531	0569	0607	0645	0682	0719	0755	4	8	11	15	19	23	26	30	34
1.2	0792	0828	0864	0899	0934	0969	1004	1038	1072	1106	3	7	10	14	17	21	24	28	31
1.3	1139	1173	1206	1239	1271	1303	1335	1367	1399	1430	3	6	10	13	16	19	23	26	29
1.4	1461	1492	1523	1553	1584	1614	1644	1673	1703	1732	3	6	9	12	15	18	21	24	27
1.5	1761	1790	1818	1847	1875	1903	1931	1959	1987	2014	3	6	8	11	14	17	20	22	25
1.6	2041	2068	2095	2122	2148	2175	2201	2227	2253	2279	3	5	8	11	13	16	18	21	24
1.7	2304	2330	2355	2380	2405	2430	2455	2480	2504	2529	2	5	7	10	12	15	17	20	22
1.8	2553	2577	2601	2625	2648	2672	2695	2718	2742	2765	2	5	7	9	12	14	16	19	21
1.9	2788	2810	2833	2856	2878	2900	2923	2945	2967	2989	2	4	7	9	11	13	16	18	20
2.0	3010	3032	3054	3075	3096	3118	3139	3160	3181	3201	2	4	6	8	11	13	15	17	19
2.1	3222	3243	3263	3284	3304	3324	3345	3365	3385	3404	2	4	6	8	10	12	14	16	18
2.2	3424	3444	3464	3483	3502	3522	3541	3560	3579	3598	2	4	6	8	10	12	14	15	17
2.3	3617	3636	3655	3674	3692	3711	3729	3747	3766	3784	2	4	6	7	9	11	13	15	17
2.4	3802	3820	3838	3856	3874	3892	3909	3927	3945	3962	2	4	5	7	9	11	12	14	16
2.5	3979	3997	4014	4031	4048	4065	4082	4099	4116	4133	2	3	5	7	9	10	12	14	15
2.6	4150	4166	4183	4200	4216	4232	4249	4265	4281	4298	2	3	5	7	8	10	11	13	15
2.7	4314	4330	4346	4362	4378	4393	4409	4425	4440	4456	2	3	5	6	8	9	11	13	14
2.8	4472	4487	4502	4518	5433	4548	4564	4579	4594	4609	2	3	5	6	8	9	11	12	14
2.9	4624	4639	4654	4669	4683	4698	4713	4728	4742	4757	1	3	4	6	7	9	10	12	13
3.0	4771	4786	4800	4814	4829	4843	4857	4871	4886	4900	1	3	4	6	7	9	10	11	13
2.1	4914	4928	4942	4955	4969	4983	4997	5011	5024	5038	1	3	4	6	7	8	10	11	12
3.2	5051	5065	5079	5092	5105	5119	5132	5145	5159	5172	1	3	4	5	7	8	9	11	12
3.3	5185	5198	5211	5224	5237	5250	5263	5276	5289	5302	1	3	4	5	6	8	9	10	12
3.4	5315	5328	5340	5353	5366	5378	5391	5403	5416	5428	1	3	4	5	6	8	9	10	11
3.5	5441	5453	5465	5478	5490	5502	5514	5527	5539	5551	1	2	4	5	6	7	9	10	11
3.6	5563	5575	5587	5599	5611	5623	5635	5647	5658	5670	1	2	4	5	6	7	8	10	11
3.7	5682	5694	5705	5717	5729	5740	5752	5763	5775	5786	1	2	3	5	6	7	8	9	10
3.8	5798	5809	5821	5832	5843	5855	5866	5877	5888	5899	1	2	3	5	6	7	8	9	10
3.9	5911	5922	5933	5944	5955	5966	5977	5988	5999	6010	1	2	3	4	5	7	8	9	10
4.0	6021	6031	6042	6053	6064	6075	6085	6096	6107	6117	1	2	3	4	5	6	8	9	10
4.1	6128	6138	6149	6160	6170	6180	6191	6201	6212	6222	1	2	3	4	5	6	7	8	9
4.2	6232	6243	6253	6263	6274	6284	6294	6304	6314	6325	1	2	3	4	5	6	7	8	9
4.3	6335	6345	6355	6365	6375	6385	6395	6405	6415	6425	1	2	3	4	5	6	7	8	9
4.4	6435	6444	6454	6464	6474	6484	6493	6503	6513	6522	1	2	3	4	5	6	7	8	9
4.5	6532	6542	6551	6561	6571	6580	6590	6599	6609	6618	1	2	3	4	5	6	7	8	9
4.6	6628	6637	6646	6656	6665	6675	6684	6693	6702	6712	1	2	3	4	5	6	7	7	8
4.7	6721	6730	6739	6749	6758	6767	6776	6785	6794	6803	1	2	3	4	5	6	6	7	8
4.8	6812	6821	6830	6839	6848	6857	6866	6875	6884	6893	1	2	3	4	4	5	6	7	8
4.9	6902	6911	6920	6928	6937	6946	6955	6964	6972	6981	1	2	3	4	4	5	6	7	8
5.0	6990	6998	7007	7016	7024	7033	7042	7050	7059	7067	1	2	3	3	4	5	6	7	8
5.1	7076	7084	7093	7101	7110	7118	7126	7135	7143	7152	1	2	3	3	4	5	6	7	8
5.2	7160	7168	7177	7185	7193	7202	7210	7218	7226	7235	1	2	2	3	4	5	6	7	7
5.3	7243	7251	7259	7267	7275	7284	7292	7300	7308	7316	1	2	2	3	4	5	6	6	7
5.4	7324	7332	7340	7348	7356	7364	7372	7380	7388	7396	1	2	2	3	4	5	6	6	7

Source: Hans Gartmann, ed. (1970). *De Laval Engineering Handbook*, 3rd ed., McGraw-Hill Book Co., New York, NY, 1–18.

4-34 THERMAL PROPERTIES OF BODIES AND THERMODYNAMICS

Table 27. Properties of Saturated Steam

(From Keenan and Keyes, "Thermodynamic Properties of Steam")
(h_f and s_f are measured from 32F)

Abs press, psi	Temp, deg F	Specific volume		Enthalpy			Entropy			Internal energy
		Liquid	Vapor	Liquid	Evap	Vapor	Liquid	Evap	Vapor	Evap
1.0	101.74	0.01614	333.6	69.70	1036.3	1106.0	0.1326	1.8456	1.9782	974.6
1.2	107.92	0.01616	280.9	75.87	1032.7	1108.6	0.1435	1.8193	1.9628	970.3
1.4	113.26	0.01618	243.0	81.20	1029.6	1110.8	0.1528	1.7971	1.9498	966.7
1.6	117.99	0.01620	214.3	85.91	1026.9	1112.8	0.1610	1.7776	1.9386	963.5
1.8	122.23	0.01621	191.8	90.14	1024.5	1114.6	0.1683	1.7605	1.9288	960.6
2.0	126.08	0.01623	173.73	93.99	1022.2	1116.2	0.1749	1.7451	1.9200	957.9
2.2	129.62	0.01624	158.85	97.52	1020.2	1117.7	0.1809	1.7311	1.9120	955.5
2.4	132.89	0.01626	146.38	100.79	1018.3	1119.1	0.1864	1.7183	1.9047	953.3
2.6	135.94	0.01627	135.78	103.83	1016.5	1120.3	0.1916	1.7065	1.8981	951.2
2.8	138.79	0.01629	126.65	106.68	1014.8	1121.5	0.1963	1.6957	1.8920	949.2
3.0	141.48	0.01630	118.71	109.37	1013.2	1122.6	0.2008	1.6855	1.8863	947.3
4.0	152.97	0.01636	90.63	120.86	1006.4	1127.3	0.2198	1.6427	1.8625	939.3
5.0	162.24	0.01640	73.52	130.13	1001.0	1131.1	0.2347	1.6094	1.8441	933.0
6.0	170.06	0.01645	61.98	137.96	996.2	1134.2	0.2472	1.5820	1.8292	927.5
7.0	176.85	0.01649	53.64	144.76	992.1	1136.9	0.2581	1.5586	1.8167	922.7
8.0	182.86	0.01653	47.34	150.79	988.5	1139.3	0.2674	1.5383	1.8057	918.4
9.0	188.28	0.01656	42.40	156.22	985.2	1141.4	0.2759	1.5203	1.7962	914.6
10	193.21	0.01659	38.42	161.17	982.1	1143.3	0.2835	1.5041	1.7876	911.1
11	197.75	0.01662	35.14	165.73	979.3	1145.0	0.2903	1.4897	1.7800	907.8
12	201.96	0.01665	32.40	169.96	976.6	1146.6	0.2967	1.4763	1.7730	904.8
13	205.88	0.01667	30.06	173.91	974.2	1148.1	0.3027	1.4638	1.7665	901.9
14	209.56	0.01670	28.04	177.61	971.9	1149.5	0.3083	1.4522	1.7605	899.3
14.696	212.00	0.01672	26.80	180.07	970.3	1150.4	0.3120	1.4446	1.7566	897.5
15	213.03	0.01672	26.29	181.11	969.7	1150.8	0.3135	1.4415	1.7549	896.7
16	216.32	0.01674	24.75	184.42	967.6	1152.0	0.3184	1.4313	1.7497	894.3
17	219.44	0.01677	23.39	187.56	965.5	1153.1	0.3231	1.4218	1.7449	892.0
18	222.41	0.01679	22.17	190.56	963.6	1154.2	0.3275	1.4128	1.7403	889.9
19	225.24	0.01681	21.08	193.42	961.9	1155.3	0.3317	1.4043	1.7360	887.8
20	227.96	0.01683	20.089	196.16	960.1	1156.3	0.3356	1.3962	1.7319	885.8
21	230.57	0.01685	19.192	198.79	958.4	1157.2	0.3395	1.3885	1.7280	883.9
22	233.07	0.01687	18.375	201.33	956.8	1158.1	0.3431	1.3811	1.7242	882.0
23	235.49	0.01689	17.627	203.78	955.2	1159.0	0.3466	1.3740	1.7206	880.2
24	237.82	0.01691	16.938	206.14	953.7	1159.8	0.3500	1.3672	1.7172	878.5
25	240.07	0.01692	16.303	208.42	952.1	1160.6	0.3533	1.3606	1.7139	876.8
26	242.25	0.01694	15.715	210.62	950.7	1161.3	0.3564	1.3544	1.7108	875.2
27	244.36	0.01696	15.170	212.75	949.3	1162.0	0.3594	1.3484	1.7078	873.6
28	246.41	0.01698	14.663	214.83	947.9	1162.7	0.3623	1.3425	1.7048	872.1
29	248.40	0.01699	14.189	216.86	946.5	1163.4	0.3652	1.3368	1.7020	870.5
30	250.33	0.01701	13.746	218.82	945.3	1164.1	0.3680	1.3313	1.6993	869.1
31	252.22	0.01702	13.330	220.73	944.0	1164.7	0.3707	1.3260	1.6967	867.7
32	254.05	0.01704	12.940	222.59	942.8	1165.4	0.3733	1.3209	1.6941	866.3
33	255.84	0.01705	12.572	224.41	941.6	1166.0	0.3758	1.3159	1.6917	864.9
34	257.08	0.01707	12.226	226.18	940.3	1166.5	0.3783	1.3110	1.6893	863.5
35	259.28	0.01708	11.898	227.91	939.2	1167.1	0.3807	1.3063	1.6870	862.3
36	260.95	0.01709	11.588	229.60	938.0	1167.6	0.3831	1.3017	1.6848	861.0
37	262.57	0.01711	11.294	231.26	936.9	1168.2	0.3854	1.2972	1.6826	859.8
38	264.16	0.01712	11.015	232.89	935.8	1168.7	0.3876	1.2929	1.6805	858.5
39	265.72	0.01714	10.750	234.48	934.7	1169.2	0.3898	1.2886	1.6784	857.2

Source: T. Baumeister, and L. S. Marks, eds. (1964). *Mechanical Engineers' Handbook*, 6th ed., McGraw-Hill Book Co., New York, NY.

Figure 5. A sample table showing properties of saturated steam.

Table 4. Tabular representation of temperature-volume-pressure relationship for an ideal gas.

Table 1. Specific volume of air (ft^3).

		Pressure psia				
		14.7	20	30	50	60
Temperature deg R	529.7	13.35010157	9.812325	6.54155	4.906162	3.92493
	540.0	13.60969388	10.00313	6.66875	5.001563	4.00125
	550.0	13.86172525	10.18837	6.792245	5.094184	4.075347
	560.0	14.11375661	10.37361	6.915741	5.186806	4.149444
	570.0	14.36578798	10.55885	7.039236	5.279427	4.223542
	580.0	14.61781935	10.7441	7.162731	5.372049	4.297639
	590.0	14.86985072	10.92934	7.286227	5.46467	4.371736
	600.0	15.12188209	11.11458	7.409722	5.557292	4.445833

4. Ends and Means

The fact that tables can be used to represent functional relationships has important implications for software developers. Currently, most software methods generate tables that can be used as records and not as dynamic relationships. Contrast this with tables in books—a scientist can readily use tables from books (in paper) either to read or to be used as a relation in his or her computing. Regrettably, since computer tools are not available that treat tables as functional relationships, their reuse is limited and circuitous.

Coad and Yourdon (1991) caution software developers that if the application of a software engineering method produces a monument of paper, then something is wrong—in the method, in the application of the method, or perhaps both. They lament, "if we lose sight of people and begin producing charts, diagrams, and piles of paper as *ends* [italics added] unto themselves, we fail to effectively communicate."

By developing structures and methods to represent tables as *means*, as well as *ends*, we will be able to facilitate seamless reuse of tables in computer systems. In such a case, tables will become *means* for communication among people, understanding by people, and further generation of knowledge by people. That is, tables can be "copied and pasted" in developing new computer programs.

5. Treatment of Tables in Text Formatting Tools

Early word processing software tools could not handle tables. Subsequent word processors recognized the need for tables to be embedded in text documents. Initially, tables were provided as simple rectangular arrays. Recent developments cater for some basic variations in tables in text processing tools both for electronic (World Wide Web) and paper (word processing) documents. However, all these are display devices only.

Morris (1996, p 79) describes how tables are formatted and added as a new feature of HTML (HyperText Markup Language, the primary language of the World Wide Web) to make HTML a true publishing medium. As is common with HTML documents, tags are used to define a table and its components. A table is divided into rows and cells. Techniques are defined to format text in table cells. However, these formatting rules are merely to represent tables for display and visualization. Creation and editing of tables are permitted, but no other data manipulation is possible at present. That is, no calculations are made with tables. Lemay (1996) also discusses formatting HTML tables for use on the World Wide Web. The emphasis is on creating tables at the transmitter's terminal (or server) and displaying them at the receivers' (client) terminals. The Web browser Netscape features table heading cells and data cells. Lemay also suggests using lists, images, and preformatted text as alternatives to tables.

Microsoft Word 95 accommodated in its "Autoformat" 38 different table formats. It also allows for columns to be split (see table 5). However, the purpose of these operations is again to display a table in a text document in a natural fashion from a human interface point of view. But no methods exist to support the use of tables as functional relationships.

In the last few years, however, tables in word processors have become capable of performing simple calculations. For example, the totals of several items in a column (or a row) can be computed automatically as the items in the table are entered. This is somewhat similar to the capabilities in a spreadsheet. However, these features in tables are minor compared with the capabilities of tables in Natural Computing as described in this report.

Natural Computing also provides for use of spreadsheets with programming interfaces. Additionally, in Natural Computing (Karamchetty, 2000), spreadsheet strips are independent objects that behave like spreadsheets while being small in size, with positive implications on memory requirements and performance.

In this report, my focus is on the use of tables from the perspective of calculations. Therefore, the emphasis is on the representation of a table for calculation rather than for its presentation as part of a document.

Table 5. A table in a word processor showing an element spanning two columns.

Monday	Tuesday	Wednesday	Thursday	
			AM	PM
—	—	—	—	—
—	—	—	—	—
—	—	—	—	—

6. Informal Survey and Analysis of Practical Tables

In this section, I provide a number of examples of tables from various textbooks, handbooks, brochures, and newspapers to demonstrate the richness and complexity of data, information, and functional relationship representations in practical tables. These examples allow me to identify and describe characteristics and properties of tables.

A table represents characteristics and values in a domain. A two-dimensional rectangular array is traditionally used to represent a table. But since a table contains a set of characteristics and their values, the representation should recognize this distinction by separating the table header, which contains the characteristics, from the data body, which contains the values. A table can be identified by a table caption, which gives it a unique identity in the document. In documents such as books, we find lists of tables, which bring together all the table captions to one location (generally with a table of contents (TOC) giving the page numbers of table locations). Thus, a simple table will consist of a table caption, a table header, and a table data body. The table caption is a string. The table header is a row (list) containing strings. The table data body is a two-dimensional matrix representing values of a given type (a number, a string, or a graphic symbol). Figure 6 is an example of a simple table with its parts identified.

Compare this simple table with the sample table in figure 7 (*Mechanical Engineers' Handbook*, Baumeister and Marks, 1964). In the sample, three types of values are used in the table data body: a string, a real number, and a range of real numbers. Such a combination of data types does not allow the use of an array in computer representation (since arrays represent data of the same type). Of course, one possible representation is an array of pointers, with each pointer pointing to a given data type (string, real number, and range).

Figure 6. Example of a simple table with its parts identified.

Caption → Table 7-1. Thermal properties.*

Header →

Body →

Pressure (psia)	Temperature (deg F)	Volume (ft^3)
100	212	5.4
150	239	4.3
200	320	3.6
250	360	3.1
300	410	2.7

Footnote → *Applicable to real gases.

Figure 7 shows that table header elements are usually complex data forms. In this sample, the second column label includes the descriptive name (absolute roughness) of the item, its symbol (ε), and its units (feet). From a textual visibility standpoint, representation as a string will suffice. However, a different and more elegant representation is needed to allow these three parts to be identified and accessed uniquely.

Figure 8 shows a sample table with data types that are numbers (both integers and a floating point), ranges of numbers (with range indicated by "–" as well as "to"), strings, and an algebraic expression (an equation). This table also has another feature: several footnotes (Baumeister and Marks, 1964).

Figure 7. A sample table with a variety of data types.

Table 4. Absolute Roughness Classification of Pipe Surfaces for Selection of Friction Factor *f* in Fig. 18

Commercial Pipe Surfaces (New)	Absolute Roughness ε, Ft
Glass, drawn brass, copper, lead	Smooth
Wrought iron, steel	0.00015
Asphalted cast iron	0.0004
Galvanized iron	0.0005
Cast iron	0.00085
Wood stave	0.0006–0.003
Concrete	0.001–0.01
Riveted steel	0.003–0.03

Source: T. Baumeister and L. S. Marks, eds. (1964). *Mechanical Engineers' Handbook*, 6th ed., McGraw-Hill Book Co., New York, NY.

Figure 8. A sample table showing distinct data types.

3–78 MECHANICS OF LIQUIDS

Table 5. Loss of Head in Flow of Water through Elbows, Meters, Etc.

(Also expressed in equivalent lengths of straight pipe to cause equal loss)

Nature of resistance	Loss of head as a decimal or multiple of $V^2/2g$ (safe avg values)	Equivalent length of straight pipe expressed in pipe diameters
Square-edged entry. Upstream end of pipe flush with inside face of reservoir wall	0.50	20
Entry like Borda's mouthpiece (see Orifices)	1.00	40
Rounded entry or very large radius bends	0–0.05	Zero
* 90 deg curves, smooth, same inside diam as pipe:		
Center-line radius = diam of pipe	0.50	20
Center-line radius = 2 to 8 diam	0.25	10
90 deg elbows, common screw end, short turn (experiments on ⅜ to 6 in. ells)	0.75	30
Tees, common screw end, full size branch (experiments on 1 to 4 in. tees)	1.50	60
Square elbow (intersection of two-cylinders)	1.25	50
Obtuse-angled elbows, deflection in pipe = a deg (less than 90 deg)	$1.25 \times (a/90)^2$	$50 \times (a/90)^2$
Water meters†		
Disk or wobble type	3.4 to 10	135–400
Rotary (disk of star or cog-wheel shape as piston)	10	400
Reciprocating piston (like a piston pump)	15	600
Turbine wheel type (double flow, balanced)	5–7.5	200–300

For pipe orifices in diaphragms see p. 16–19.
* For complete summary and details, see Pigott, *Trans. ASME*, 1950 and 1956.
† Different makes of meters and different sizes of the same make vary considerably.

Source: T. Baumeister and L. S. Marks, eds. (1964). *Mechanical Engineers' Handbook*, 6th ed., McGraw-Hill Book Co., New York, NY.

The sample table in figure 9 reveals another highly interesting use of tables in practical applications. Data elements in this table contain algebraic expressions that are the equivalent of several case and switch statements, as shown by pseudo-code in figure 10. It is excitingly obvious that such tables represent considerable information very elegantly.

Figure 9. Sample table showing algebraic expressions.

Table 7. Values of the Hydraulic Radius *m* for Various Cross Sections
(m = cross-sectional area ÷ wetted perimeter)

Form of cross section	Value of m
Circle, flowing full, also half full	$d/4$
Square, no top (depth = width = d)	$d/3$
Half square (width = $2d$, depth = d)	$d/2$
Trapezoidal channels (bottom width = b; depth = d):	
Half regular hexagon, side slopes 60 deg	$d/2$
Channel with 45 deg side slopes	$(bd + d^2)/(b + 2.83d)$
Channel with side slopes 1½ hor to 1 vert	$(bd + 1.5d^2)/(b + 3.61d)$
Channel with side slopes 2 hor to 1 vert	$(bd + 2d^2)/(b + 4.47d)$
Wide, shallow stream [1]	d (approx)

[1] Hence the term "hydraulic mean depth," sometimes used for m.

Source: T. Baumeister and L. S. Marks, eds. (1964). *Mechanical Engineers' Handbook*, 6th ed., McGraw-Hill Book Co., New York, NY.

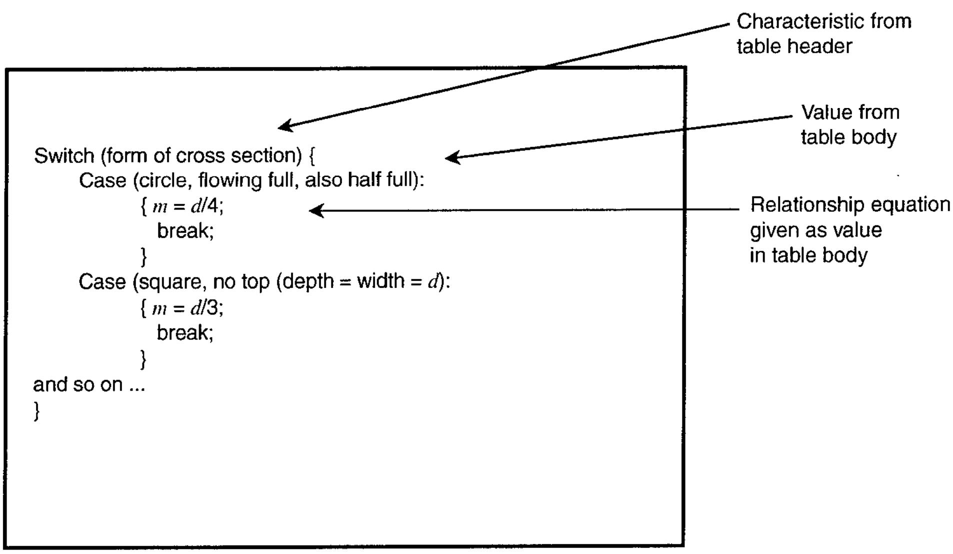


Figure 10. Example of complex case and switch statements required to replace a table such as that in figure 9.

In the sample table in figure 11, the fourth column represents a temperature range. However, this range is not an output or value range as in the previous sample (see fig. 7). It is a *conditional range*. This is a clever way of representing a number of "if statements" in a column. The equation given in column 3 is applicable only if the "Temp range, deg R," is within the range given at any element. This interesting feature leads to the representation of a table element as a data output value, or as a conditional range for checking. In other words, the table takes multiple data values. For example, if the gas is oxygen and if the temperature range is between 540 and 5000, then the equation is as given in column 3. This is indeed a very intriguing way of representing calculations through tables.

Figure 11. A sample table showing a conditional range value in an element.

Table 17. Specific Heats of Gases at 1 Atm

Gas	Symbol	Equation for C_p in Btu per mol	Temp range, deg R	Source
Oxygen	O_2	$11.515 - \left(\frac{172}{\sqrt{T}}\right) + \left(\frac{1530}{T}\right)$	540–5000	*a*
		$11.515 - \left(\frac{172}{\sqrt{T}}\right) + \left(\frac{1530}{T}\right) + \left(\frac{0.05(T - 4000)}{1000}\right)$	5000–9000	*a*
Nitrogen	N_2	$9.47 - \left(\frac{3.47 \times 10^3}{T}\right) + \left(\frac{1.16 \times 10^6}{T^2}\right)$	540–5000	*a*
Carbon monoxide	CO	$9.46 - \left(\frac{3.29 \times 10^3}{T}\right) + \left(\frac{1.07 \times 10^6}{T^2}\right)$	540–5000	*a*
Hydrogen	H_2	$5.76 + \left(\frac{0.578T}{1000}\right) + \left(\frac{20}{\sqrt{T}}\right)$	540–4000	*a*
		$5.76 + \left(\frac{0.578T}{1000}\right) + \left(\frac{20}{\sqrt{T}}\right) - \left(\frac{0.33(T - 4000)}{1000}\right)$	4000–9000	*a*
Water	H_2O	$19.86 - \left(\frac{597}{\sqrt{T}}\right) + \left(\frac{7500}{T}\right)$	540–5000	*a**
Carbon dioxide	CO_2	$16.2 - \left(\frac{6.53 \times 10^3}{T}\right) + \left(\frac{1.41 \times 10^6}{T^2}\right)$	540–6300	*a*
Methane	CH_4	$4.22 + 8.211 \times 10^{-3}T$	492–1800	*b*
		$27.0 - \frac{14,400}{T}$	1800–5940	*b*
Ethylene	C_2H_4	$6.0 + 8.33 \times 10^{-3}T$	720–1400	*c*
Ethane	C_2H_6	$6.6 + 13.33 \times 10^{-3}T$	720–1440	*c*
Ethyl alcohol	C_2H_6O	$4.5 + 21.1 \times 10^{-3}T$	680–1120	*c*
Methyl alcohol	CH_4O	$2.0 + 16.67 \times 10^{-3}T$	680–1100	*c*
Benzene	C_6H_6	$6.5 + 28.9 \times 10^{-3}T$	520–1120	*c*
Octane	C_8H_{18}	$14.4 + 53.3 \times 10^{-3}T$	720–1440	*c*
Dodecane	$C_{12}H_{26}$	$19.6 + 80.0 \times 10^{-3}T$	720–1440	*c*

[a] Sweigert and Beardsley, Empirical Specific Heat Equations Based upon Spectroscopic Data, *Ga. School Tech., State Eng. Expt. Sta. Bull.* 2, 1938.
[b] Schwarz, Die Spezifischen Wärmen der Gase als Hilfswerte zur Berechnung von Gleichgewichten, *Arch. Eisenhüttenw.*, **9**, 1936, p. 389.
[c] Parks and Huffman, *ACS, Mon.* 60, 1932.
* Approximate. An equation based on the most recent data is given by Keyes in *J. Chem. Phys.*, **15**, Aug. 1947, p. 602.

Source: T. Baumeister and L. S. Marks, eds. (1964). *Mechanical Engineers' Handbook*, 6th ed., McGraw-Hill Book Co., New York, NY.

In my presentation of table samples so far, the table header and the table data body are rectangular; i.e., a number of rows and columns form a regular rectangular array. Now consider the table in figure 12. Note that both columns 7 and 8 in the table header develop branches. First, the element "Specific heat per lb . . ." branches into c_p and c_v. Next, the element "Heat capacity per cu ft . . ." develops the same branches, c_p and c_v. The table header begins with 9 columns and, with these branches, ends with 11 columns (the same number as the table data body). At the junction of the table header and body, the number of columns in both the table header and body is the same. The table in figure 12 is an example of columns branching into two or more columns.

The most significant effect of the branched columns is that a simple rectangular array representation for tables is not adequate. We must be able to produce table header and table data body representations that will allow for any number of columns developing branches. To accommodate such irregular shapes (nonrectangular) for a table header and a table data body, we need to adopt a different data structure. I introduce the concept of a cell cage to accommodate this requirement (see sect. 7).

As a side point, figure 12 also has blanks in the table data body represented by several dots ("..."). Yet another feature of tables—footnotes—can also be observed in this example. The first column of the table data body has an entry (Steam) (about midway down) with a symbol *c* referring to a table footnote. Footnote symbols require another addition to the

Figure 12. Example table where elements develop branches as they progress along columns.

Table 23. Properties of Gases

Gas	Chemical symbol	Approx molecular weight[a]	Weight in lb of 1 cu ft at standard atmos pressure and 68 F[b]	Density relative to air	Gas constant, R, in ft-lb per lb, deg R	Specific heat per lb at room temperatures, Btu per lb, deg F		Heat capacity per cu ft at standard atmos pressure and 68 F, Btu per cu ft, deg F		$k = c_p/c_v$
						c_p	c_v	c_p	c_v	
Helium	He	4.0	0.01039	0.138	386.3	1.25	0.754	0.0130	0.0078	1.66
Argon	A	40.0	0.1037	1.377	38.70	0.124	0.0743	0.0129	0.0077	1.67
Air	.....	29.0	0.07528	1.000	53.30	0.241	0.1725	0.0181	0.0130	1.40
Oxygen	O_2	32.0	0.08305	1.103	48.31	0.217	0.1549	0.0180	0.0129	1.40
Nitrogen	N_2	28.0	0.07274	0.966	55.16	0.247	0.1761	0.0179	0.0128	1.40
Hydrogen	H_2	2.0	0.005234	0.0695	766.8	3.42	2.435	0.0179	0.0127	1.40
Nitric oxide	NO	30.0	0.07788	1.034	51.52	0.231	0.1648	0.0180	0.0128	1.40
Carbon monoxide	CO	28.0	0.07269	0.965	55.19	0.243	0.1721	0.0177	0.0125	1.41
Hydrochloric acid	HCl	36.5	0.09460	1.256	42.41	0.191	0.1365	0.0181	0.0129	1.40
Steam[c]	H_2O	18	....	0.623	85.81	0.46	0.36	......	......	1.28
Carbon dioxide	CO_2	44.0	0.1142	1.516	35.13	0.205	0.1599	0.0234	0.0183	1.28
Nitrous oxide	N_2O	44.0	0.1143	1.518	35.12	0.221	0.1759	0.0253	0.0201	1.26
Sulphur dioxide	SO_2	64.0	0.1663	2.208	24.13	0.154	0.1230	0.0256	0.0204	1.25
Ammonia	NH_3	17.0	0.04420	0.587	90.77	0.523	0.4064	0.0231	0.0179	1.29
Acetylene	C_2H_2	26.0	0.06754	0.897	59.40	0.350	0.2737	0.0236	0.0185	1.28
Methyl chloride	CH_3Cl	50.5	0.1309	1.738	30.62	0.24	0.2006	0.0314	0.0263	1.20
Methane	CH_4	16.0	0.04163	0.553	96.37	0.593	0.4692	0.0247	0.0195	1.26
Ethylene	C_2H_4	28.0	0.07280	0.967	55.11	0.40	0.3292	0.0291	0.0240	1.22

For more accurate values of specific heats, see pp. 4–9, 4–10.
[a] For accurate values of atomic weights, see p. 6–4.
[b] For values at 60 F, multiply by 1.0154.
[c] Very rough values applying to low pressures and temperatures only. See steam tables, pp. 4–34 to 4–41, for more exact values.

Source: T. Baumeister and L. S. Marks, eds. (1964). *Mechanical Engineers' Handbook*, 6th ed., McGraw-Hill Book Co., New York, NY.

item/value representation. The cell must be a composite, having both the value "Steam" and a footnote symbol. The table will then carry a footnote linked to that symbol. Figure 13 also shows a sample table where a number of columns develop multiple branches.

Figure 13. A number of columns develop variable numbers of branches or multiples.

4–34 THERMAL PROPERTIES OF BODIES AND THERMODYNAMICS

Table 27. Properties of Saturated Steam

(From Keenan and Keyes, "Thermodynamic Properties of Steam")

(h_f and s_f are measured from 32F)

Abs press, psi	Temp, deg F	Specific volume		Enthalpy			Entropy			Internal energy
		Liquid	Vapor	Liquid	Evap	Vapor	Liquid	Evap	Vapor	Evap
1.0	101.74	0.01614	333.6	69.70	1036.3	1106.0	0.1326	1.8456	1.9782	974.6
1.2	107.92	0.01616	280.9	75.87	1032.7	1108.6	0.1435	1.8193	1.9628	970.3
1.4	113.26	0.01618	243.0	81.20	1029.6	1110.8	0.1528	1.7971	1.9498	966.7
1.6	117.99	0.01620	214.3	85.91	1026.9	1112.8	0.1610	1.7776	1.9386	963.5
1.8	122.23	0.01621	191.8	90.14	1024.5	1114.6	0.1683	1.7605	1.9288	960.6
2.0	126.08	0.01623	173.73	93.99	1022.2	1116.2	0.1749	1.7451	1.9200	957.9
2.2	129.62	0.01624	158.85	97.52	1020.2	1117.7	0.1809	1.7311	1.9120	955.5
2.4	132.89	0.01626	146.38	100.79	1018.3	1119.1	0.1864	1.7183	1.9047	953.3
2.6	135.94	0.01627	135.78	103.83	1016.5	1120.3	0.1916	1.7065	1.8981	951.2
2.8	138.79	0.01629	126.65	106.68	1014.8	1121.5	0.1963	1.6957	1.8920	949.2
3.0	141.48	0.01630	118.71	109.37	1013.2	1122.6	0.2008	1.6855	1.8863	947.3
4.0	152.97	0.01636	90.63	120.86	1006.4	1127.3	0.2198	1.6427	1.8625	939.3
5.0	162.24	0.01640	73.52	130.13	1001.0	1131.1	0.2347	1.6094	1.8441	933.0
6.0	170.06	0.01645	61.98	137.96	996.2	1134.2	0.2472	1.5820	1.8292	927.5
7.0	176.85	0.01649	53.64	144.76	992.1	1136.9	0.2581	1.5586	1.8167	922.7
8.0	182.86	0.01653	47.34	150.79	988.5	1139.3	0.2674	1.5383	1.8057	918.4
9.0	188.28	0.01656	42.40	156.22	985.2	1141.4	0.2759	1.5203	1.7962	914.6
10	193.21	0.01659	38.42	161.17	982.1	1143.3	0.2835	1.5041	1.7876	911.1
11	197.75	0.01662	35.14	165.73	979.3	1145.0	0.2903	1.4897	1.7800	907.8
12	201.96	0.01665	32.40	169.96	976.6	1146.6	0.2967	1.4763	1.7730	904.8
13	205.88	0.01667	30.06	173.91	974.2	1148.1	0.3027	1.4638	1.7665	901.9
14	209.56	0.01670	28.04	177.61	971.9	1149.5	0.3083	1.4522	1.7605	899.3
14.696	212.00	0.01672	26.80	180.07	970.3	1150.4	0.3120	1.4446	1.7566	897.5
15	213.03	0.01672	26.29	181.11	969.7	1150.8	0.3135	1.4415	1.7549	896.7
16	216.32	0.01674	24.75	184.42	967.6	1152.0	0.3184	1.4313	1.7497	894.3
17	219.44	0.01677	23.39	187.56	965.5	1153.1	0.3231	1.4218	1.7449	892.0
18	222.41	0.01679	22.17	190.56	963.6	1154.2	0.3275	1.4128	1.7403	889.9
19	225.24	0.01681	21.08	193.42	961.9	1155.3	0.3317	1.4043	1.7360	887.8
20	227.96	0.01683	20.089	196.16	960.1	1156.3	0.3356	1.3962	1.7319	885.8
21	230.57	0.01685	19.192	198.79	958.4	1157.2	0.3395	1.3885	1.7280	883.9
22	233.07	0.01687	18.375	201.33	956.8	1158.1	0.3431	1.3811	1.7242	882.0
23	235.49	0.01689	17.627	203.78	955.2	1159.0	0.3466	1.3740	1.7206	880.2
24	237.82	0.01691	16.938	206.14	953.7	1159.8	0.3500	1.3672	1.7172	878.5
25	240.07	0.01692	16.303	208.42	952.1	1160.6	0.3533	1.3606	1.7139	876.8
26	242.25	0.01694	15.715	210.62	950.7	1161.3	0.3564	1.3544	1.7108	875.2
27	244.36	0.01696	15.170	212.75	949.3	1162.0	0.3594	1.3484	1.7078	873.6
28	246.41	0.01698	14.663	214.83	947.9	1162.7	0.3623	1.3425	1.7048	872.1
29	248.40	0.01699	14.189	216.86	946.5	1163.4	0.3652	1.3368	1.7020	870.5
30	250.33	0.01701	13.746	218.82	945.3	1164.1	0.3680	1.3313	1.6993	869.1
31	252.22	0.01702	13.330	220.73	944.0	1164.7	0.3707	1.3260	1.6967	867.7
32	254.05	0.01704	12.940	222.59	942.8	1165.4	0.3733	1.3209	1.6941	866.3
33	255.84	0.01705	12.572	224.41	941.6	1166.0	0.3758	1.3159	1.6917	864.9
34	257.08	0.01707	12.226	226.18	940.3	1166.5	0.3783	1.3110	1.6893	863.5
35	259.28	0.01708	11.898	227.91	939.2	1167.1	0.3807	1.3063	1.6870	862.3
36	260.95	0.01709	11.588	229.60	938.0	1167.6	0.3831	1.3017	1.6848	861.0
37	262.57	0.01711	11.294	231.26	936.9	1168.2	0.3854	1.2972	1.6826	859.8
38	264.16	0.01712	11.015	232.89	935.8	1168.7	0.3876	1.2929	1.6805	858.5
39	265.72	0.01714	10.750	234.48	934.7	1169.2	0.3898	1.2886	1.6784	857.2

Source: T. Baumeister and L. S. Marks, eds. (1964). *Mechanical Engineers' Handbook*, 6th ed., McGraw-Hill Book Co., New York, NY.

The sample table in figure 14, which shows properties of superheated steam, is an example of a table with both a row header and a column header. In the column header, one column (Temperature of steam, deg F) develops nine branches. Each row in the row header (20, 40, 60, and so on) develops three branches. A number of elements are blank (indicated by a series of dots), indicating no meaningful values, or indicating that table values have bounds.

Figure 14. A sample table with both a column header and a row header.

Table 28. Superheated Steam Tables

(Abstracted from Keenan and Keyes, "Thermodynamic Properties of Steam")
(v = specific volume, cu ft per lb; h = enthalpy, Btu per lb; s = entropy)

Pressure, psia (Saturation temp, deg F)		Temperature of steam, deg F								
		340	380	420	460	500	550	600	650	700
20	v	23.60	24.82	26.04	27.25	28.46	29.97	31.47	32.97	34.47
(227.96)	h	1210.8	1229.7	1248.7	1267.6	1286.6	1310.5	1334.4	1358.6	1382.9
	s	1.8053	1.8285	1.8505	1.8716	1.8918	1.9160	1.9392	1.9671	1.9829
40	v	11.684	12.315	12.938	13.555	14.168	14.930	15.688	16.444	17.198
(267.25)	h	1207.0	1226.7	1246.2	1265.5	1284.8	1309.0	1333.1	1357.4	1381.9
	s	1.7252	1.7493	1.7719	1.7934	1.8140	1.8385	1.8619	1.8843	1.9058
60	v	7.708	8.143	8.569	8.988	9.403	9.917	10.427	10.935	11.441
(292.71)	h	1203.0	1223.6	1243.6	1263.4	1283.0	1307.4	1331.8	1356.3	1380.9
	s	1.6766	1.7135	1.7250	1.7470	1.7678	1.7927	1.8162	1.8388	1.8605
80	v	5.718	6.055	6.383	6.704	7.020	7.410	7.797	8.180	8.562
(312.03)	h	1198.8	1220.3	1240.9	1261.1	1281.1	1305.8	1330.5	1355.1	1379.9
	s	1.6407	1.6669	1.6909	1.7134	1.7346	1.7598	1.7836	1.8063	1.8281
100	v	4.521	4.801	5.071	5.333	5.589	5.906	6.218	6.527	6.835
(327.81)	h	1194.3	1216.8	1238.1	1258.8	1279.1	1304.2	1329.1	1354.0	1378.9
	s	1.6117	1.6391	1.6639	1.6869	1.7085	1.7340	1.7581	1.7810	1.8029
120	v	......	3.964	4.195	4.418	4.636	4.902	5.165	5.426	5.683
(341.25)	h	......	1213.2	1235.3	1256.5	1277.2	1302.6	1327.7	1352.8	1377.8
	s	......	1.6156	1.6413	1.6649	1.6869	1.7127	1.7370	1.7601	1.7822
140	v	......	3.365	3.569	3.764	3.954	4.186	4.413	4.638	4.861
(353.02)	h	......	1209.4	1232.3	1254.1	1275.2	1300.9	1326.4	1351.6	1376.8
	s	......	1.5950	1.6217	1.6458	1.6683	1.6945	1.7190	1.7423	1.7645
160	v	......	2.914	3.098	3.273	3.443	3.648	3.849	4.048	4.244
(363.53)	h	......	1205.5	1229.3	1251.6	1273.1	1299.3	1325.0	1350.4	1375.7
	s	......	1.5766	1.6042	1.6291	1.6519	1.6785	1.7033	1.7268	1.7491
180	v	......	2.563	2.732	2.891	3.044	3.230	3.411	3.588	3.764
(373.06)	h	......	1201.4	1226.1	1249.1	1271.0	1297.6	1323.5	1349.2	1374.7
	s	......	1.5596	1.5884	1.6139	1.6373	1.6642	1.6894	1.7130	1.7355
200	v	......	......	2.438	2.585	2.726	2.895	3.060	3.221	3.380
(381.79)	h	......	......	1222.9	1246.5	1268.9	1295.8	1322.1	1348.0	1373.6
	s	......	......	1.5738	1.6001	1.6240	1.6513	1.6767	1.7006	1.7232
220	v	......	......	2.198	2.335	2.465	2.621	2.772	2.920	3.066
(389.86)	h	......	......	1219.5	1243.8	1266.7	1294.1	1320.7	1346.8	1372.6
	s	......	......	1.5603	1.5874	1.6117	1.6395	1.6652	1.6892	1.7120
260	v	......	......	1.8257	1.9483	2.063	2.199	2.330	2.457	2.582
(404.42)	h	......	......	1212.4	1238.3	1262.3	1290.5	1317.7	1344.3	1370.4
	s	......	......	1.5354	1.5642	1.5897	1.6184	1.6447	1.6692	1.6922
300	v	......	......	1.5513	1.6638	1.7675	1.8891	2.005	2.118	2.227
(417.33)	h	......	......	1204.8	1232.5	1257.6	1286.8	1314.7	1341.8	1368.3
	s	......	......	1.5126	1.5434	1.5701	1.5998	1.6268	1.6517	1.6751
350	v	......	......	......	1.3984	1.4923	1.6010	1.7036	1.8021	1.8980
(431.72)	h	......	......	......	1224.8	1251.5	1282.1	1310.9	1338.5	1365.5
	s	......	......	......	1.5197	1.5481	1.5792	1.6070	1.6325	1.6563
400	v	......	......	......	1.1978	1.2851	1.3843	1.4770	1.5654	1.6508
(444.59)	h	......	......	......	1216.5	1245.1	1277.2	1306.9	1335.2	1362.7
	s	......	......	......	1.4977	1.5281	1.5607	1.5894	1.6155	1.6398

Source: T. Baumeister and L. S. Marks, eds. (1964). *Mechanical Engineers' Handbook*, 6th ed., McGraw-Hill Book Co., New York, NY.

This table has another extremely interesting feature. Note the characteristic value in the column header, "Pressure, psia (Saturation temp, deg F)." In the corresponding row values, note two numbers, one without and the other with parentheses. I refer to this set as a value and a parenthesis (or bracket) value. The pressure values are represented by the values in the open, and the saturation temperature values are placed in parentheses. From a computer representation point of view, this feature should be considered when representations for cells are developed (see sect. 8).

The sample table in figure 15 presents interesting examples of the use of ranges. As the footnote to the table indicates, when a range is given in an element, the first and the second numbers correspond, and linear interpolation is permitted.

Figure 15. A sample table showing ranges with value correspondence.

Table 1. Emissivity of Surfaces

Surface	Temp.* deg F	Emissivity*
Metals and Their Oxides		
Aluminum:		
Highly polished	440–1070	0.039–0.057
Polished	73	0.040
Rough plate	78	0.055–0.07
Oxidized at 1110 F	390–1110	0.11–0.19
Oxide	530–1520	0.63–0.26
Alloy 75ST	75	0.10
75ST, repeated heating	450–900	0.22–0.16
Brass:		
Highly polished	497–710	0.03–0.04
Rolled plate, natural	72	0.06
Rolled, coarse-emeried	72	0.20
Oxidized at 1110 F	390–1110	0.61–0.59
Chromium	100–1000	0.08–0.26
Copper:		
Electrolytic, polished	176	0.02
Comm'l plate, polished	66	0.030
Heated at 1110 F	390–1110	0.57–0.57
Thick oxide coating	77	0.78
Cuprous oxide	1470–2010	0.66–0.54
Molten copper	1970–2330	0.16–0.13
Dow metal, cleaned, heated	450–750	0.24–0.20
Gold, highly polished	440–1160	0.02–0.40
Iron and steel:		
Pure Fe, polished	350–1800	0.05–0.37
Wrought iron, polished	100–480	0.28
Smooth sheet iron	1650–1900	0.55–0.60
Rusted plate	67	0.69
Smooth oxidized iron	260–980	0.78–0.82
Strongly oxidized	100–480	0.95
Molten iron and steel	2730–3220	0.40–0.45
Lead:		
99.96%, unoxidized	260–440	0.06–0.08
Gray oxidized	75	0.28
Oxidized at 390 F	390	0.63
Mercury, pure clean	32–212	0.09–0.12
Molybdenum filament	1340–4700	0.10–0.29
Monel metal, K5700		
Washed, abrasive soap	75	0.17
Repeated heating	450–1610	0.46–0.65
Nickel and alloys:		
Electrolytic, polished	74	0.05
Electroplated, not polished	68	0.11
Wire	368–1844	0.10–0.19
Plate, oxid. at 1110 F	390–1110	0.37–0.48
Nickel oxide	1200–2290	0.59–0.80
Copper-nickel, polished	212	0.06
Nickel-silver, polished	212	0.14
Nickelin, gray oxide	70	0.26
Nichrome wire, bright	120–1830	0.65–0.79
Nichrome wire, oxid	120–930	0.95–0.98
ACI-HW (60Ni, 12Cr); firm black ox. coat	520–1045	0.89–0.82
Platinum, polished plate	440–2960	0.05–0.17
Silver, pure polished	440–1160	0.02–0.03
Stainless steels:		
Type 316, cleaned	75	0.28
316, repeated heating	450–1600	0.57–0.66
304, 42 hr at 980 F	420–980	0.62–0.73
310, furnace service	420–980	0.90–0.97
Allegheny #4, polished	212	0.13
Tantalum filament	2420–5430	0.194–0.33
Thorium oxide	530–1520	0.58–0.21
Tin, bright	76	0.04–0.06
Tungsten, aged filament	80–6000	0.03–0.35
Zinc, 99.1%, comm'l, polished	440–620	0.05
Galv., iron, bright	82	0.23
Galv., gray oxid	75	0.28
Refractories, Building Materials, Paints, Misc.		
Alumina, 50μ grain size	1850–2850	0.39–0.28
Alumina-silica, cont'g	1850–2850	
0.4% Fe_2O_3	....	0.61–0.43
1.7% Fe_2O_3	....	0.73–0.62
2.9% Fe_2O_3	....	0.78–0.68
Al paints (vary with am't lacquer body, age)	212	0.27–0.67
Asbestos	100–700	0.93–0.95
Candle soot: lampblack-water glass	70–700	0.95 ± 0.01
Carbon plate, heated	260–1160	0.81–0.79
Oil layers:		
Lube oil, 0.01" on pol. Ni	68	0.82
Linseed, 1–2 coats on Al	68	0.56–0.57
Rubber, soft gray reclaimed	76	0.86
Misc. I: shiny black lacquer, planed oak, white enamel, serpentine, gypsum, white enamel paint, roofing paper, lime plaster, black matte shellac	70	0.87–0.91
Misc. II: glazed porcelain, white paper, fused quartz, polished marble, rough red brick, smooth glass, hard glossy rubber, flat black lacquer, water, electrographite	70	0.92–0.96

* When two temperatures and two emissivities are given they correspond, first to first and second to second, and linear interpolation is permissible.

Source: T. Baumeister and L. S. Marks, eds. (1964). *Mechanical Engineers' Handbook*, 6th ed., McGraw-Hill Book Co., New York, NY.

Figure 16 shows a sample table for which dimensions or attributes are described by means of a sketch. This type of table is useful in generating the drawings of a part or machine component, as the table provides the flange sizes for a given load and nominal pipe size. A number of additional features are observable from the structure of this table. Notice the notes in the body of the table that redirect the calculations to a different part of the table.

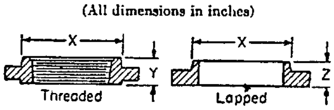
Figure 17 shows a sample table from a catalog of a plastics manufacturer. The size of the container is shown as $L \times W \times D$. But three separate columns are not used. In terms of cell representation, the column header should be represented as a composite cell containing the characteristic "size," units in parentheses, and dimensions L, W, D with separators ("×" marks). The corresponding data-body cells should accommodate the values for the three dimensions with the same separator marks "×."

The sample table in figure 18 uses pictures in the table data body to describe the attitude variable. If the attitude corresponds to the configuration shown in the picture, then the C_d value in the fourth column is the result. The interpretation of the rows is rich in meaning. The first row element develops seven multiples in the second column, but the seven multiple columns collapse into one element again in the third column. Again the fourth column has seven multiples corresponding to the seven items in the second column. The other rows also develop multiples and collapses.

Figure 19 shows the very popular use of graphical icons or symbols, as used in the *Consumer Reports* magazine. Each icon is explained in a footnote, end note, or other note (such as a key).

Table 33. Dimensions of American Standard Companion Flanges* (ASA)

(All dimensions in inches)



Nom pipe size	150 lb			300 lb			400 lb		
	X	Y	Z	X	Y	Z	X	Y	Z
1/2	1 3/16	5/8	5/8	1 1/2	7/8	7/8	For sizes below 4 in., use dimensions of the 600 lb flanges		
3/4	1 1/2	5/8	5/8	1 7/8	1	1			
1	1 15/16	11/16	11/16	2 1/8	1 1/16	1 1/16			
1 1/4	2 5/16	13/16	13/16	2 1/2	1 1/16	1 1/16	....	.....	.....
1 1/2	2 9/16	7/8	7/8	2 3/4	1 3/16	1 3/16	....	.....	.....
2	3 1/16	1	1	3 5/16	1 5/16	1 9/16	....	.....	.....
2 1/2	3 9/16	1 1/8	1 1/8	3 15/16	1 1/2	1 1/2	....	.....	.....
3	4 1/4	1 3/16	1 3/16	4 5/8	1 11/16	1 11/16	....	.....	.....
3 1/2	4 13/16	1 1/4	1 1/4	5 1/4	1 3/4	1 3/4	....	.....	.....
4	5 5/16	1 5/16	1 5/16	5 3/4	1 7/8	1 7/8	5 3/4	2	2
5	6 7/16	1 7/16	1 7/16	7	2	2	7	2 1/8	2 1/8
6	7 9/16	1 9/16	1 9/16	8 1/8	2 1/16	2 1/16	8 1/8	2 1/4	2 1/4
8	9 11/16	1 3/4	1 3/4	10 1/4	2 7/16	2 7/16	10 1/4	2 11/16	2 11/16
10	12	1 15/16	1 15/16	12 5/8	2 5/8	3 3/4	12 5/8	2 7/8	4
12	14 3/8	2 3/16	2 7/16	14 3/4	2 7/8	4	14 3/4	3 1/8	4 1/4
14 O.D.	15 3/4	2 1/4	3 1/8	16 3/4	3	4 3/8	16 3/4	3 5/16	4 5/8
16 O.D.	18	2 1/2	3 7/16	19	3 1/4	4 3/4	19	3 11/16	5
18 O.D.	19 7/8	2 11/16	3 13/16	21	3 1/2	5 1/8	21	3 7/8	5 3/8
20 O.D.	22	2 7/8	4 1/16	23 1/8	3 3/4	5 1/2	23 1/8	4	5 3/4
24 O.D.	26 1/8	3 1/4	4 3/8	27 5/8	4 3/16	6	27 5/8	4 1/2	6 1/4

Nom pipe size	600 lb			900 lb			1,500 lb		
	X	Y	Z	X	Y	Z	X	Y	Z
1/2	1 1/2	7/8	7/8	For sizes below 3 in., use dimensions of 1,500 lb flanges			1 1/2	1 1/4	1 1/4
3/4	1 7/8	1	1				1 3/4	1 3/8	1 3/8
1	2 1/8	1 1/16	1 1/16				2 1/16	1 5/8	1 5/8
1 1/4	2 1/2	1 1/8	1 1/8				2 1/2	1 5/8	1 5/8
1 1/2	2 3/4	1 1/4	1 1/4	....	...	....	2 3/4	1 3/4	1 3/4
2	3 5/16	1 7/16	1 7/16	....	...	....	4 1/8	2 1/4	2 1/4
2 1/2	3 15/16	1 5/8	1 5/8	....	...	....	4 7/8	2 1/2	2 1/2
3	4 5/8	1 13/16	1 13/16	5	2 1/8	2 1/8	5 1/4	2 7/8	2 7/8
3 1/2	5 1/4	1 15/16	1 15/16						
4	6	2 1/8	2 1/8	6 1/4	2 3/4	2 3/4	6 3/8	3 9/16	3 9/16
5	7 7/16	2 3/8	2 3/8	7 1/2	3 1/8	3 1/8	7 3/4	4 1/8	4 1/8
6	8 3/4	2 5/8	2 5/8	9 1/4	3 3/8	3 3/8	9	4 11/16	4 11/16
8	10 3/4	3	3	11 3/4	4	4 1/2	11 1/2	5 5/8	5 5/8
10	13 1/2	3 3/8	4 3/8	14 1/2	4 1/4	5	14 1/2	6 1/4	7
12	15 3/4	3 5/8	4 5/8	16 1/2	4 5/8	5 5/8	17 3/4	7 1/8	8 5/8
14 O.D.	17	3 11/16	5	17 3/4	5 1/8	6 1/8	19 1/2	.....	9 1/2
16 O.D.	19 1/2	4 3/16	5 1/2	20	5 1/4	6 1/2	21 3/4	.....	10 1/4
18 O.D.	21 1/2	4 5/8	6	22 1/4	6	7 1/2	23 1/2	.....	10 7/8
20 O.D.	24	5	6 1/4	24 1/2	6 1/2	8 1/4	25 1/4	.....	11 1/2
24 O.D.	28 1/4	5 1/2	7 1/4	29 1/2	8	10 1/2	30	.....	13

* Other dimensions are given in Tables 31 and 32. Finished bore on lapped flange to be such as method of attachment of pipe requires.

Source: T. Baumeister and L. S. Marks, eds. (1964). *Mechanical Engineers' Handbook*, 6th ed., McGraw-Hill Book Co., New York, NY.

Figure 16. A sample table connecting values to dimensions in a sketch.

Figure 17. A sample table showing high-density polyethylene tank data having three size values in one column.

Nalgene® Sturdy Rectangular Tanks With Covers **VERY POPULAR**

HIGH DENSITY POLYETHYLENE

Rectangular tanks have excellent chemical, abrasion, and impact resistance. They feature continuous operating service to 150°F. Large tanks need additional support at room temperature.

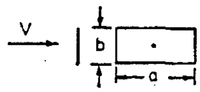
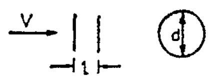
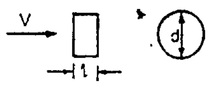
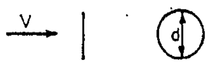
Part No. Tank With Cover	Capacity (Gal.)	Size (In.) L x W x D	Approximate Wall Thickness (In.)	Tank With Cover Price Each
19009LG	2	8 x 8 x 8	5/32	45.50
19013LG	6	14 x 10 x 10	5/32	50.20
19017LG	7	12 x 12 x 12	5/32	57.80
19020LG	11	18 x 12 x 12	5/32	78.00
19022LG	15	24 x 12 x 12	5/32	110.00
19010LG	30	24 x 24 x 12	5/32	173.00
19011LG	47	30 x 30 x 12	5/32	299.25
19024LG	6	18 x 4 x 18	5/32	93.10
19026LG	11	12 x 12 x 18	5/32	98.60
19028LG	15	18 x 12 x 18	5/32	106.40
19031LG	25	18 x 18 x 18	5/32	161.50
19033LG	22	24 x 12 x 18	5/32	159.90
19035LG	30	24 x 18 x 18	5/32	135.50
19037LG	60	36 x 20 x 20	5/32	227.50
19039LG	22	18 x 12 x 24	5/32	159.90
19041LG	30	24 x 12 x 24	5/32	169.80
19044LG	45	24 x 18 x 24	5/32	191.90
19046LG	90	36 x 24 x 24	3/16	325.50
19047LG	94	30 x 30 x 24	3/16	391.90
19048LG	12	24 x 4 x 30	1/8	156.90
19050LG	25	24 x 8 x 30	1/8	173.70
19051LG	117	30 x 30 x 30	3/16	423.90
19053LG	90	24 x 24 x 36	3/16	316.30
19055LG	140	30 x 30 x 36	3/16	457.45

Discount: Less 10% 2-4; 15% 5-7; 20% 8 or more.

Source: Consolidated Plastics Company Inc., 1996.

Figure 18. A sample table containing pictures and collapsing branched rows.

Table 4. Drag Coefficients

Object	Proportions	Attitude	C_D
Rectangular plate, sides a and b	a = 1 4 8 12.5 25 50 ∞		1.16 1.17 1.23 1.34 1.57 1.76 2.00
Two disks, spaced a distance l apart	$\frac{l}{d}$ = 1 1.5 2 3		0.93 0.78 1.04 1.52
Cylinder	$\frac{l}{d}$ = 1 2 4 7		0.91 0.85 0.87 0.99
Circular disk			1.11
Hemispherical cup, open back			0.41
Hemispherical cup, open front, parachute			1.35
Cone, closed base			α = 60°, 0.51 α = 30°, 0.34

Source: T. Baumeister and L. S. Marks, eds. (1964). *Mechanical Engineers' Handbook*, 6th ed., McGraw-Hill Book Co., New York, NY.

Figure 19. A sample table containing graphical icons.

Ratings Camcorders

& Recommendations

Shopping strategy

You can buy all the camcorder you need for $600. Use our Ratings to find the best picture quality and features.

Know the choices The basic compact tape formats are 8mm and VHS-C. At standard speed, 8mm models record for up to 2½ hours; VHS-C models, for up to 40 minutes. There are "high band" versions of both formats (Hi8 and S-VHS-C), with enhanced video resolution that's displayed at its best through a TV set with an S-video input. Compact camcorders weigh two to three pounds. Full-sized VHS models weigh at least five pounds.

Decide what to spend Basic models start at about $500; high-band models, at about $800.

Know where to shop Electronics stores have the widest selection. Discount stores usually have less selection but good prices. Department stores carry few camcorders.

Using the Ratings We tested 8mm, Hi8, and VHS-C camcorders. If you can't find a rated model, call the manufacturer; see page 64.

Overall Ratings *Listed in order of overall score*

Key no.	Brand and model	Format	Price	Overall score (0–100; P F G VG E)	Picture clarity: Hi8 SP	Picture clarity: SP	Picture clarity: EP	Picture clarity: LP	Picture color	Sound: Accuracy	Sound: Flutter
1	Hitachi VM-H620A	Hi8	$850		◉	◓	—	—	◉	◓	◉
2	Hitachi VM-E220A	8mm	600		—	◓	—	—	◓	○	◉
3	Sony CCD-TR84	8mm	780		—	◓	—	○	◓	○	◓
4	Canon ES600	8mm	675		—	◓	—	—	◓	○	◉
5	Samsung SCH985	Hi8	850		◓	◓	—	—	◓	○	◉
6	Panasonic PV-A206	VHS-C	600		—	◓	◓	—	◓	◒	○
7	Sharp VL-E34U	8mm	680		—	◓	—	—	◉	○	◉
8	Sony CCD-TR44	8mm	500		—	◓	—	◒	○	○	◉
9	Panasonic PV-D406	VHS-C	800		—	◓	○	—	○	◒	○
10	JVC GR-AX710	VHS-C	615		—	○	●	—	◓	○	◓
11	RCA CC617	VHS-C	550		—	○	●	—	◓	◒	◓

Source: *Consumer Reports,* 1996.

Figure 20 shows sample tables in which elements can contain relationships in terms of other table values and inputs for interactive computations. These representations were the forerunners of work sheets and spreadsheets. By developing suitable representations for the elements in these tables, we can realize dynamic tables. Section 5 presented a discussion of tables with cell values computed dynamically as in spreadsheets.

It is possible to continue this survey and discover many other table features in vogue in books. However, I believe I have cited a sufficient number and variety of examples so that we can discern the basic characteristics needed for computer representation of a variety of tables.

Figure 20. Sample tables where elements can contain values of computations.

Gas	v vol. %	m mol. wt.	vm	$\% \, G = \frac{vm}{M}$	C_p	$\% \, GC_p$	C_v	$\% \, GC_v$
C_2H_6	0.14	30.07	4.21	0.233	0.397	0.0926	0.325	0.0758
CH_4	0.85	16.04	13.63	0.752	0.593	0.4459	0.451	0.3392
N_2	0.01	28.02	0.28	0.015	0.244	0.0037	0.173	0.0026
Total . . .	1.00	$M = 18.12$		1.000	$C_p = 0.5422$		$C_v = 0.4176$	

Gas	V vol. %	Mc_p at 60°F	$V \times Mc_p$
C_2H_6	14	12.32	1.725
CH_4	85	8.46	7.191
N_2	1	6.95	0.069
	100		8.985

$$k = \frac{8.985}{8.985 - 1.99} = \frac{8.985}{6.995} = 1.284$$

Source: Hans Gartmann, ed. (1970). *De Laval Engineering Handbook*, 3rd ed., McGraw-Hill Book Co., New York, NY.

7. Anatomy and Morphology of Tables

The previous section's survey of practical tables found in books, newspapers, and brochures suggests a minimal structure for adopting such tables for computer applications. The anatomy of a minimal table structure is shown in figure 21. The structure consists of a table caption, a table column header (and/or row header), a table data body, footnotes, and table data in various memory locations. The column header and data body consist of cell cages. The cell cages contain pointers to cells. These cells are of different types (sect. 7.2). The cell cages can take any shape, rectangular or nonrectangular, the latter accommodating multiples and collapsing multiples. This representation separates the table structure from its data. Thus, it is possible to define generic table types and operations on them. Specific domain data can be connected to table cells by

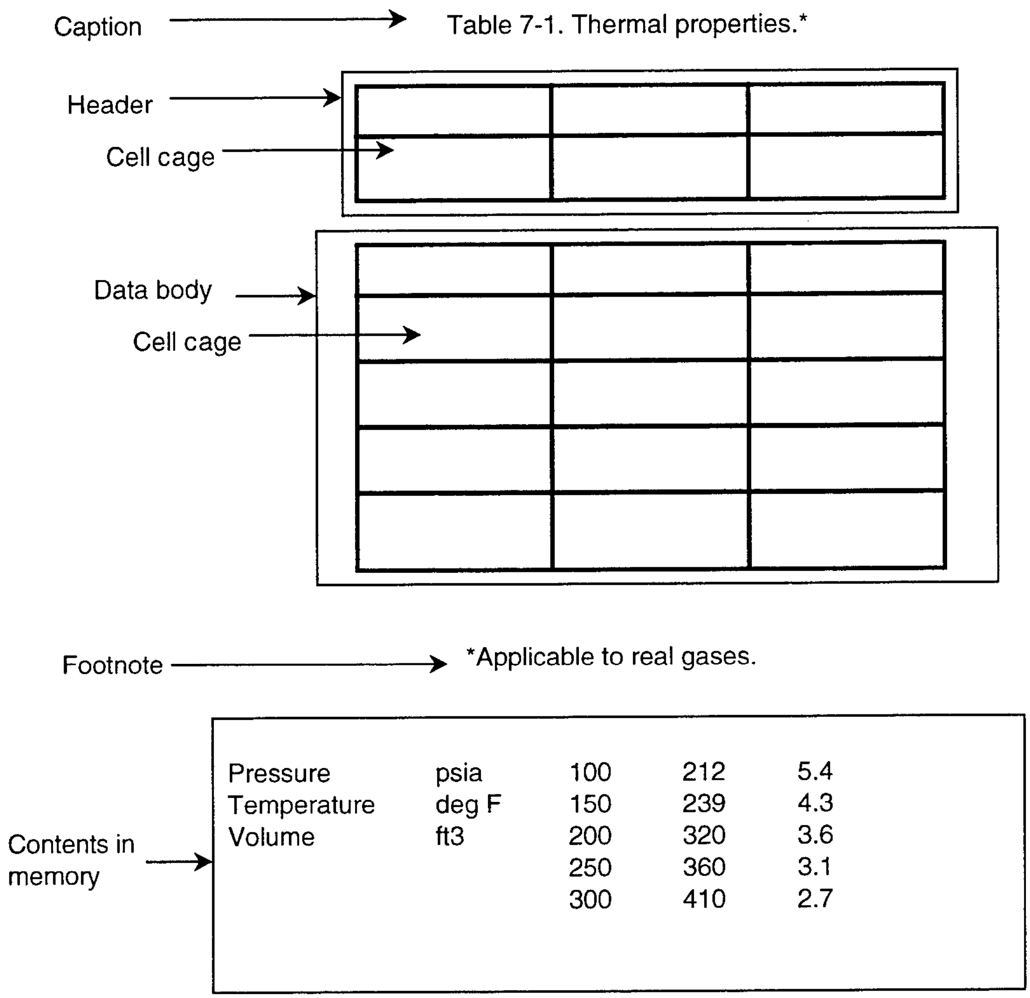


Figure 21. A generic structure for a simple table.

pointers. Since table data are not in an array, there are no restrictions on their type.

7.1 Cell Cages

The cell cage (see fig. 22) consists of a number of cells, each of which is identified by row and column positions. In the example, 18 cells are arranged in a rectangular fashion (3 columns and 6 rows). Each cell is identified by the position it occupies in the cell cage (0,0 to 5,2). The cells are themselves shown separately, and each cell can be of a specific type; in other words, they need not all be of the same type. By this arrangement, we can identify the relative position of the cells by the cell cage, while the cell type determines what can be stored in the cell. Pointers from a cell to the contents in memory will allow basic data types to be stored.

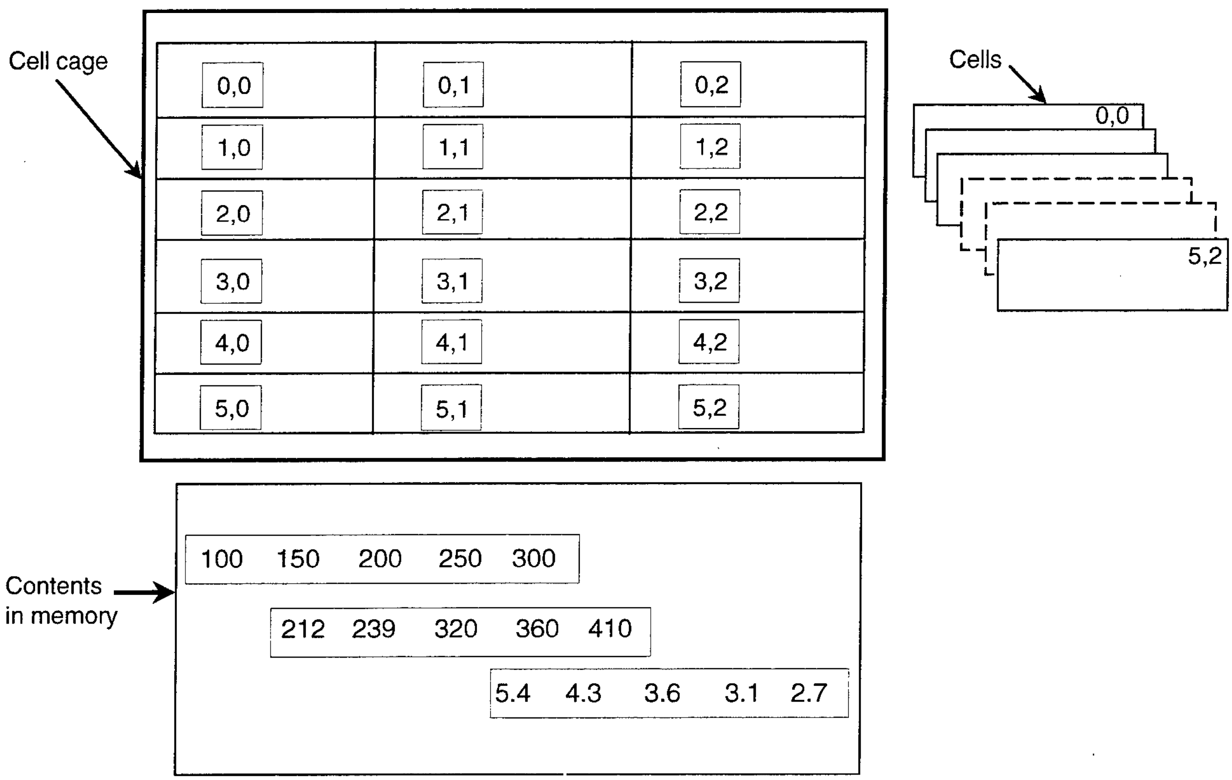


Figure 22. Cell cage and cells.

In a simple rectangular cell cage, adjacent cells bear certain relationships. If we take a cell i, j, its neighbors are north: $i - 1, j$; east: $i, j + 1$; south: $i + 1, j$; west: $i, j - 1$. For a cell cage with m rows and n columns, i will be between 0 and $m - 1$, and j will be between 0 and $n - 1$. Although the cell cage is rectangular, a single vector can be used to represent the cells if we arrange the cell cage position numbers in a sequence (a vector), as shown in figure 23. The position of a cell can be obtained from its sequence or serial number (N_S) by the following relationship:

$$i = N_S/n \text{ and}$$

$$j = N_S \% n.$$

The sequence number (in the vector) for a given cell i, j is obtained from the equation

$$N_S = in + j.$$

In terms of C++ programming language, a vector of void pointers corresponding to each N_S value can point to individual cells. Depending on the cell type, its contents can be processed. As an example, figure 23 shows integer data in a cell; the contents can be accessed via an integer pointer.

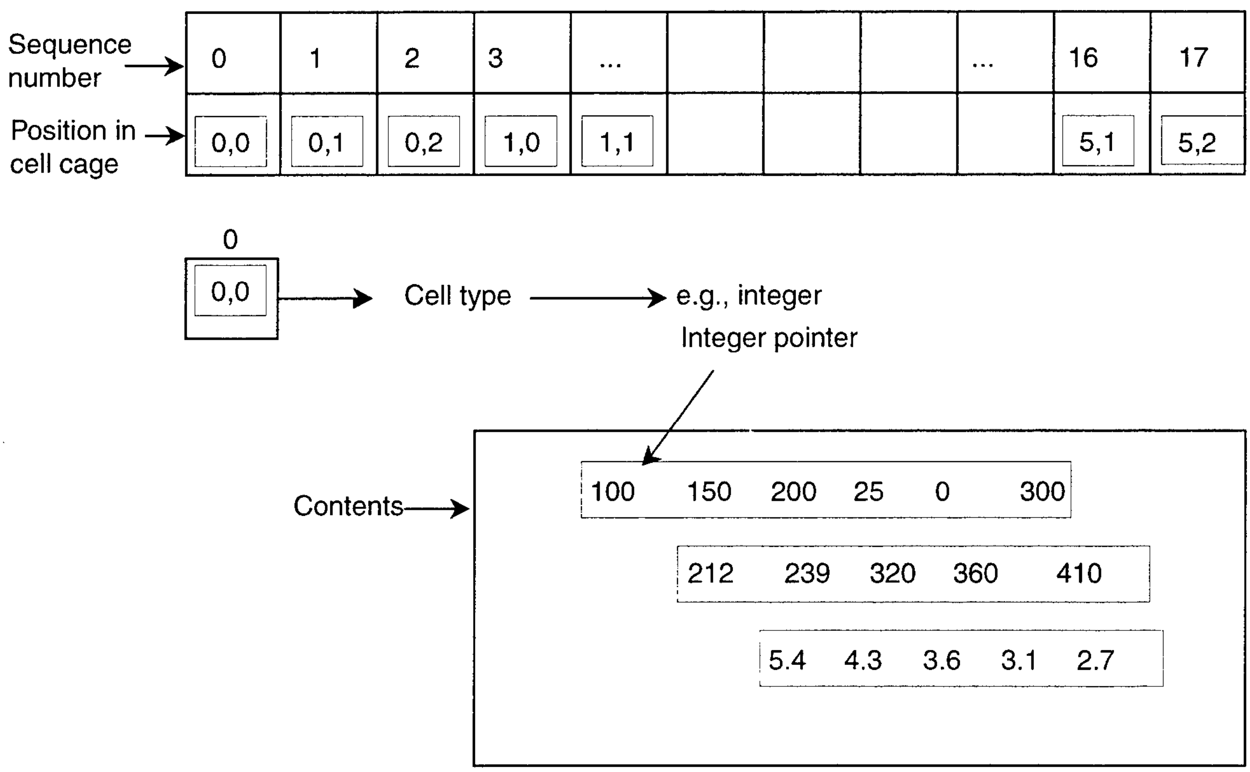


Figure 23. Cell-cage sequence numbers.

We also need an arrangement for a nonrectangular cell cage representing a table that has a branching column. To demonstrate this case, I modified the table of figure 21 and present it as figure 24.

In this case, it is immediately obvious that the table data body is still rectangular and can be represented by a rectangular cell cage. However, the column header should be represented by a nonrectangular cell cage, as shown in figure 25. Since we know that there are four columns at the junction of the column header and data body, the cell cage will have four columns, as shown in this figure. We choose to omit cell 0,2 in the first row of the cell cage. In general, we will follow a convention of missing the later cell number(s). We can also imagine that cell 0,1 will stand for (the missing) cell 0,2, if ever we look for it. For example, if we look for the northerly neighbor of cell 1,2 by the conventional (previously described) procedure, we will get the cell 0,2, and by this just-stated rule, we will convert it to 0,1. It may be worthwhile to note that the cell 0,1 will have one more neighbor now than it would ordinarily have; it has two southerly neighbors.

Figure 26 shows the vector representing the cell cage. It has seven sequence numbers connected to appropriate cell positions and memory locations.

While the representation and the theory for cell cages with expanding columns and collapsing multiple columns are complex, algebraic relationships can be developed, programmed once, and then be made available for use by downstream users. For brevity, I do not go into further detail in this report.

Caption ⟶ Table 7-1. Thermal properties.*

Header ⟶

Body ⟶

Pressure psia	Temperature		Volume ft^3
	deg F	deg C	
100	212	100	5.4
150	239	115	4.3
200	320	160	3.6
250	360	182	3.1
300	410	210	2.7

Footnote ⟶ *Applicable to real gases.

Figure 24. An example table showing a column with branches.

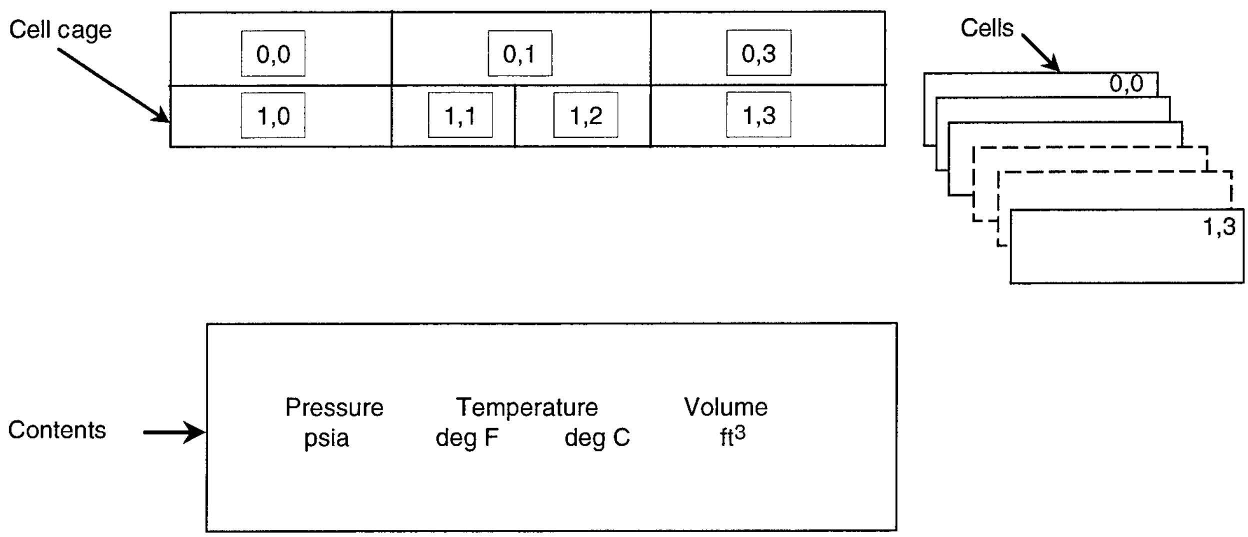


Figure 25. Representation of column header cell cage for table in figure 24.

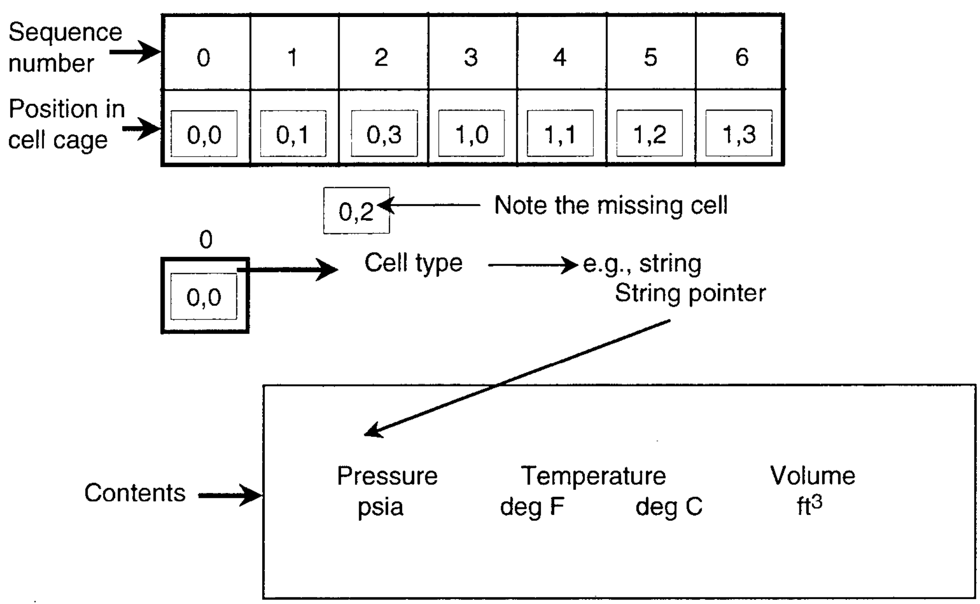


Figure 26. Cell-cage sequence numbers for a column header with branches.

7.2 Cells

As we have seen, there are different types of cells. The simplest cells are those containing basic data types available in the programming language, e.g., char, integer, real. By defining other data types, we can easily extend the cells to contain real numbers and strings. With object-oriented programming, it is only a little more complex to define cells of other types. Figure 27 shows a number of cell types and the data they contain.

7.3 Table Types

A spectrum of tables can be developed based on the parameters identified so far (see fig. 28). By using the representations described so far, software developers can generate several table types. First, a number of table types will be developed and made available as part of a Natural Computing tool box. In an iterative cycle (see fig. 29), developers will address almost all the requirements over several versions and years. Domain specialists will incorporate their knowledge into these tables and validate their correct and appropriate operation. Domain specialists will call for new versions to be developed by software developers if the existing table representations are inadequate in certain features. End users will use the tables provided to them by the domain specialists. Specialists will incorporate new domain knowledge, catering to the needs of end users.

Cell type		Data type	Example
Numeric	→	Number	2.34
String	→	String	Pressure
Blank	→	String or char	—
Parenthetic	→	Two of type	212 (100)
Graphic	→	Graphic or icon	
Multistring	→	Multiple strings	Pressure, psia
Units string	→	String(s) (from a list)	Btu per hr ft^2
Composite string	→	Combination	Composition (balance Al)

Figure 27. Example cell types with sample data.

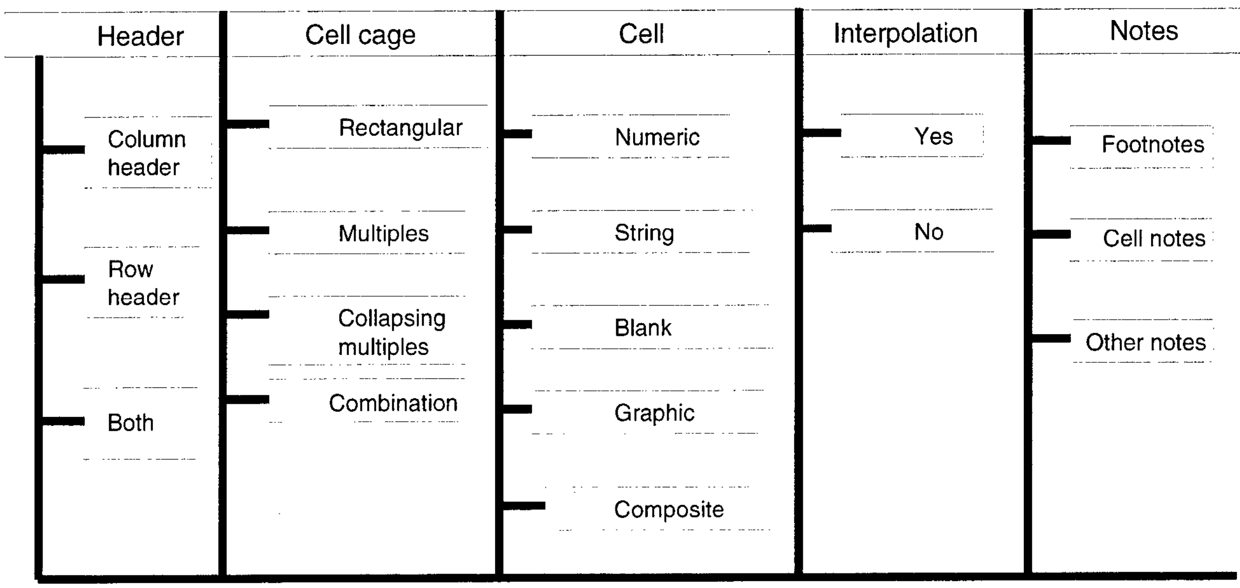


Figure 28. A spectrum of tables in profile representation that can be generated.

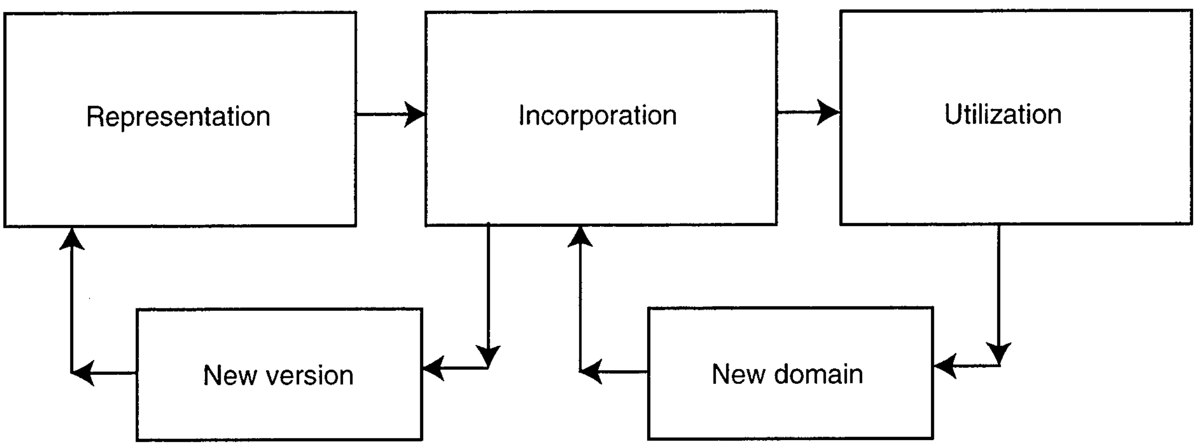


Figure 29. Iterative cycle of table development in future Natural Computing system.

Two types of system growth can be anticipated. One is the domain growth and the other is the Natural Computing tool growth. Initially, a set of Natural Computing tools will be developed and presented to domain specialists for use in their domains. A variety of domains can be incorporated. As more complex domain problems are incorporated, new Natural Computing tools will need to be developed every time a certain new table type is encountered for which a table representation is not yet available. As new table features (structures or behaviors) are encountered or invented, software developers will play a primary role in developing table objects with needed features. Once those features are implemented, the newer versions of Natural Computing software tools will become available to all domains. Eventually, most types of tables will be represented in Natural Computing and any domain can be captured.

8. Computer Representation of Tables

The foregoing discussion suggests that representing tables as objects with various attributes and methods for operating on their contents is promising. It is then obvious that object-oriented programming techniques and languages will be most suitable and perhaps essential in such a representation. A table class will be developed comprising a base class and a number of derived classes. As the need for new table features is recognized, more derived classes can be developed and added to the system. A container class should be chosen for a table class since it contains several components as identified in section 7. The component classes will include a caption, a header, a data body, and various notes classes (e.g., footnotes). The header and data body classes will consist of cell cages and arrays of cells. The cells themselves are represented in terms of a base class and a number of derived classes. The table hierarchy is shown in figure 30. (I reserve low-level details for a later report, as including them here would hinder the smooth presentation of the subject matter.)

Data are entered into a table at the cell level, the table being built up from the components by the various constructors in the table class. A number of operations must be defined to manipulate the contents of the table. Creation, deletion, modification, storage, recall from storage, and persistence operations are performed by conventional methods. Display, printing, and plotting of tables (as they appear in books and documents) (U.S. GPO, 1984) will be another set of operations. Perhaps the methods available up to this point in current and emerging state-of-the-art text processing systems will suffice for the display functionality of tables. Additionally, the table class will have operations that make the table serve its functional roles (for example, the relationship between input and output).

As stated in the introduction, the main role of a table is to serve as a functional relationship. That is, given an input value (or values), the table object should provide a result value. This functionality should be available both as an interactive feature and as part of a procedure (or program) building capability. In executing this function, the table class should behave responsibly by properly interpreting the qualifying information

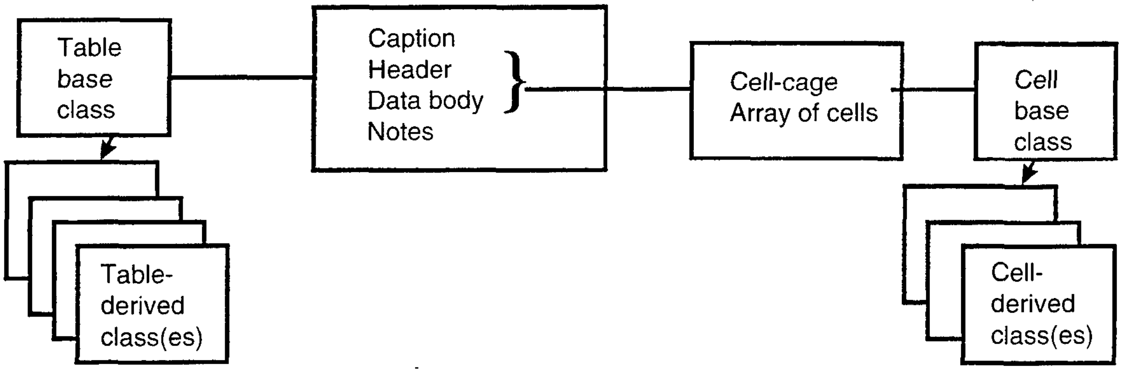


Figure 30. A hierarchy of table classes and component classes.

available in the header(s), footnotes, and other notes. Thus a table is not merely a lookup table; it both looks up information and *looks out* for footnoted interpretations based on conditions. I call this the *responsibility* of a table object during computations.

In programming, a table will be even more useful and helpful to a naive end user if it can adjust units automatically. Every quantity in a table should know that it carries units. In general, all calculation features in Natural Computing work with a set of four items. These four items are (1) a name for the characteristic, (2) a variable representing the characteristic (called notation), (3) the value of the characteristic, and (4) the units of the characteristic. The characteristic belongs to or is an attribute of a system under discussion or calculation. A variable is a symbol chosen to represent the physical quantity or characteristic for brevity in writing relationships, equations, or table header column cells. The characteristic has a definite value at a given state. Its value is given in some physical units. (In many documents, the relationships between the variables and their descriptions are often given in lists of notation or lists of symbols.)

A table header may contain a variable and its description and should be connected to the corresponding pair in the list of notation. The units of a quantity are given both in the notation and in the table and should be compatible but not necessarily identical. For example, a quantity such as velocity can be in feet per second and in miles per hour. When a table is queried, the units in the query (input) and the units for the quantity in the table should be compatible. Likewise, the units of a table quantity should be compatible with the units of the output quantity. The variable and units pair will also be connected when procedures are developed that use variables from tables. By treating units (and conversions) simultaneously with functional manipulation, we avoid the units-related errors that have caused much grief in programs of yesterday.

Other sets of operations exist that make tables very effective in programming: for example, mapping a table of one type into another. People usually copy tables of one type into another type. This can be done in software by the development of mapping operations that change table types and permit the development of supersets and subsets. These operations will enable abstracting part of a table data body and creating a subset table. In reverse, two tables can be combined to form a superset. For example, a table of **Year** versus **Income** and one of **Year** versus **Expenditure** can be combined and a superset table defined with three columns: **Year, Income,** and **Expenditure** (see fig. 31). We could further extend this table to yet another superset table by defining another column called **Profit**, which may be defined as the difference in the **Income** column and the **Expenditure** column. Thus, tables should be able to generate new tables by addition and/or deletion of columns and rows.

Mathematical operations on tables make them very valuable and highly useful. For example, figure 32 shows how two table values can be added to develop a result table. We should be able to achieve this result by simply writing a functional relationship, such as result table = table 1 + table 2. Other operations to be incorporated in tables are subtraction, scalar multiplication, division, columnar multiplication and division, and other complex operations.

Figure 31. Development of a superset table from two tables.

Year	Income ($M)
1950	120
1960	132
1970	130
1980	136
1990	145

Year	Expenditure ($M)
1950	110
1960	126
1970	115
1980	124
1990	137

Year	Income ($M)	Expenditure ($M)
1950	143	110
1960	157	126
1970	156	115
1980	166	124
1990	184	137

Income for North Region

Year	Income ($M)
1950	120
1960	132
1970	130
1980	136
1990	145

+

Income for South Region

Year	Income ($M)
1950	23
1960	25
1970	26
1980	30
1990	39

=

Income for North & South Regions

Year	Income ($M)
1950	143
1960	157
1970	156
1980	166
1990	184

Figure 32. Development of mathematical operations with tables.

9. Choice and Use of Tables

As shown in the foregoing section, tables come in a number of different types. Just as word processors contain document manipulation tools and graphics packages contain picture manipulation tools, a Natural Computing software tool box will contain several table tools for use by domain specialists. A domain specialist can select a table that best fits the needs of a particular application. Thus, an instance of a blank table is created (instance creation is the responsibility of the table object's constructor). In the creating/editing mode, all data and information are entered into a table. It is the domain specialist's responsibility to fill in the footnotes and other notes as applicable. The table object also develops a number of behavioral characteristics for the domain specialists' review. For example, the limit values (minimum, maximum, and singularities) of a characteristic are recorded, and the domain specialists check them for accuracy and applicability. The table object uses these limits to flag an error message when a user tries unallowable values; this is another feature imitated from human usage of tables. In real-world applications, domain specialists often provide such checks. Minimum and maximum values of a variable prevent extrapolation outside allowable bounds.

A table object will also have a set of input (or query) templates and output templates. Recall that a table is a functional relationship between quantities (characteristics) identified in the cells of the column header. Given one of the quantities, all others can be determined from the table. We can use this functional relationship in generating query templates. The user can choose the appropriate input template, type in a value for the independent variable, and submit it to the table object. The template also guides the user with the limits on the variable values. These guidance values in the templates protect the table from invalid or out-of-range queries. A table will return a result by means of the output templates. A table with input and output templates can be likened to a hardware component with its input and output sockets. These input and output templates are very useful in connecting different functional objects into a procedure or program.

10. Search for Data

Since the cells in a table can be connected by adjacency lists and pointers to neighbors, and pointers lead to data values or contents, searching a table for result data items is quite simple. A query consists of an independent variable/value pair and another dependent variable. If we wish to find the value corresponding to the dependent variable, we use the simple property of a table that two value cells bear the same neighborly or adjacency relationship as the corresponding two variable cells do. This relationship is depicted in figure 33.

Figure 33. Adjacency relationships are used to get table values.

Table 7-1. Thermal properties.*

Pressure psia	Temperature deg F	Volume ft^3
100	212	5.4
150	239	4.3
200	320	3.6
250	360	3.1
300	410	2.7

*Applicable to real gases.

Question:

Given pressure = 200 psia

Find volume = ? ft^3

Solution:

Adjacency relationships:

Pressure → second easterly neighbor → volume

Hence, second easterly neighborof 200 is → 3.6

11. Visibility of Table Data

An advantage of the use of tables in Natural Computing is that they provide the user a way to gain insight into the computation. Educators criticize current computational software systems as "black boxes," because the solution method is incomprehensible from the software. The user learns nothing from using the software. For that matter, even domain specialists do not understand what is in the code, once their domain information (actually, algorithms shorn of all information) is put into code by a software developer. With traditional media (paper, calculator, and pencil), students' learning improves proportionally with the number of problems solved. In contrast, with current computer software, a student's learning does not improve with the number of problems solved. Since tables in Natural Computing, as described here, reveal themselves and show relationships between variables, a student can realize opportunities available and watch out for pitfalls in the problem domain represented by each table.

12. Testing a Table in Isolation

Natural computing tables have the advantage that they can be tested in isolation. Testing is a key task in software development; capturing an application domain in software is equally critical. Isolating a table and testing it for a variety of inputs, together with the built-in justifications, limits, behaviors, and responsibilities prescribed for a table class, would go a long way toward eliminating bugs in software that used tables as described in this paper. Since the filters on a table will allow only preapproved types and ranges of values, isolated testing can come very near to guaranteeing both the software and the domain knowledge (the application).

13. Embedding Tables into Text

In textbooks and other paper documents, tables are embedded in text. A reader can read them and get a general idea of the information presented. Or a reader can interactively use the table by looking for values, with some input parameters. Any minimal computer representation should be able to duplicate that capability.

In Natural Computing, tables are embedded in text (as in fig. 34) and when a user wishes to use tables interactively, the tables can be activated by a computer command. Input and output templates (boxes) will appear, allowing the user to type in the input values and receive output values. Such an interactive facility is useful for studying a table and understanding trends in the domain.

While developing a procedure or a program, a user can copy and paste a table from the text. In this case, the table object is available to the procedure; this procedure can be saved separately from the original text object. In software as in paper-based computation, people should be able to develop procedures by connecting several tables, equations, and graphs to inputs and outputs.

22.2 H&R Block Basic Income Tax Course

PRACTICE 22.2. Prepare Form 1040X to correct Charles Albertson's 1985 return which was filed April 10, 1986. On May 10, 1988, he received another W-2 for 1985. The W-2 had been lost in the mail.

Federal Tax	Wages	FICA	State Tax	Employee
$300.00	$1,700.00	$119.85	$110.00	C. Albertson

While talking to Mr. Albertson, you learn that he paid all of the cost of maintenance of his home, which was the principal residence of his son, Johnnie (20), whose gross income was $600 for 1985. Assume that you have completed a support worksheet and that Mr. Albertson provided $3,000 of Johnnie's $3,600 total support.

In column A, on the Form 1040X, we have entered the information needed from Mr. Albertson's original Form 1040. The original return has not been examined and he has not been notified that it will be examined.

Notice that line 4 (deductions) of Mr. Albertson's Form 1040X is blank. That's because in 1985 the standard deduction, otherwise known as the zero bracket amount, was built into the tax table. Thus, taxpayers not itemizing deductions did not need to claim a standard deduction on the tax form.

The necessary portion of the 1985 Tax Table is shown below.

If line 37, 1040, line 19, 1040A, or line 7, 1040EZ (taxable income) is—		And you are—			
At least	But less than	Single	Married filing jointly	Married filing sepa-rately	Head of a house-hold
		Your tax is—			
14,000					
14,000	14,050	1,766	1,391	2,109	1,686
14,050	14,100	1,776	1,399	2,122	1,695
14,100	14,150	1,786	1,407	2,134	1,704
14,150	14,200	1,796	1,415	2,147	1,713
14,200	14,250	1,806	1,423	2,159	1,722
14,250	14,300	1,816	1,431	2,172	1,731
14,300	14,350	1,826	1,439	2,184	1,740
14,350	14,400	1,836	1,447	2,197	1,749
14,400	14,450	1,846	1,455	2,209	1,758
14,450	14,500	1,856	1,463	2,222	1,767
14,500	14,550	1,866	1,471	2,234	1,776
14,550	14,600	1,876	1,479	2,247	1,785
14,600	14,650	1,886	1,487	2,259	1,794
14,650	14,700	1,896	1,495	2,272	1,803
14,700	14,750	1,906	1,503	2,284	1,812
14,750	14,800	1,916	1,511	2,297	1,821
14,800	14,850	1,926	1,519	2,309	1,830
14,850	14,900	1,936	1,527	2,322	1,839
14,900	14,950	1,946	1,535	2,334	1,848
14,950	15,000	1,956	1,543	2,347	1,857

Source: H&R Block (1990). *Tax Preparation Guides.*

Figure 34. Example of a table embedded in text.

14. Tables and Databases

Although databases are commonly thought (albeit mistakenly) to represent tables, the foregoing description of tables should counter that impression. A brief comparison of database systems and real-world tables shows how much more complex actual tables are than the "tables" that database people claim to represent.

The vast literature on database systems might lead an unwary reader to conclude that the database community has already represented tables in a computer-usable format. However, after calling a table a relation (Date, 1995, p 79), Date then states, "a relation and a table are not really the same thing, although in practice it is frequently convenient to pretend that they are" (p 80). In a discussion of formalizing the concept of a table, Shaler and Mellor (1988) state that normalization rules can be viewed from two perspectives. The first focuses on the form of data in databases. The rules tell how to set up tables so that little redundancy is in the data; that is, the amount of data required to store a certain information content is minimized. The second perspective (the one most *natural* to us) looks at the normalization rules as statements about the repertoire of forms that we use in our model (the fact that we are using tables, for example) and at the meaning we imply whenever we use a form in a particular manner.

Date (1995) describes a database system as basically a computerized "record-keeping" system. Its overall purpose is to maintain information in a computer and to make that information available on demand to three classes of users: applications programmers, end users, and database administrators. In a slightly more precise definition, Date says, "A database consists of some collection of persistent data that is used by application systems of some given enterprise."

In defining a relational system, Date states that data in a relational system are perceived by the user as tables (and nothing but tables) (p 22). He goes on to state, "For most practical purposes, indeed, the terms relation and table can be taken to be synonymous." The relational model is a way of looking at data—that is, it is a prescription for a way of representing data (namely, by means of tables).

Continuing their discussion of regular tables, Shaler and Mellor explain the rules used in databases:

First Rule: One instance of an object has exactly one value for each attribute.

There is one and only one data element at each row-column intersection. This rule forbids the "repeating group" construct found in some databases (table 6) and true holes (as in table 7).

Second Rule: Attributes must contain no internal structure.

This is another expression of the requirement for fully factored attributes. The rule forbids tables with an internal structure such as table 8. But it allows a construct such as table 9.

Table 6. Example of a table with a forbidden repeating group.

Owner	Name of pet(s)	Address
Smith	Rover Rin Tin Tin Sarzak	100 Canine Court
Jones	Lassie	6 Dogwood Lane

Table 7. Example of a table with a forbidden true hole.

Owner	Model	Manufacturer	License No.
Brown	Sedan	Ford	16923A
Green	Van	Chevrolet	23004C
Jones	Truck		29-A-101

Table 8. Example of a table containing forbidden internal structure.

Name	Sex/breed
Lassie	F—Collie
Laddie	M—Collie
Fifi	F—Poodle

Table 9. Example of a table with an allowable structure to represent information in table 8.

Name	Sex	Breed
Lassie	F	Collie
Laddie	M	Collie
Fifi	F	Poodle

These examples illustrate that database relations do not allow the representation of the tables that scientists, engineers, and analysts use in a variety of domains, as described in section 6. This condition is due to the mathematical rigor of the database systems. Software developers are caught in a dilemma between mathematical rigor and the natural but highly flexible forms people use.

Study the baseball tables in figures 4 and 35. Young children grasp the nuances of these tables. These simple-looking tables represent many relationships. Lay readers can quickly compare and calculate desired outcomes. As new games are played each day, new tables of values can be calculated from old table values and predefined relationships (formulas).

The simplicity of the natural tables in representing a variety of relationships is yet to be matched by the best database systems. For example, I show in section 6 that ranges are represented, blanks are allowed with definite meanings, and data types are mixed with no problems. Notice how a batter and information on his field position are combined in the first column of the table in figure 35.

Figure 35. A table of baseball results representing a player and his position in same element.

NEW YORK	AB	R	H	BI	BB	SO	AVG
Boggs 3b	4	0	1	1	0	0	0.331
Girardi c	4	0	0	0	0	0	0.282
O'Neill rf	3	0	1	0	1	0	0.368
TMartinez 1b	4	0	0	0	0	0	0.246
Sierra dh	3	2	3	0	1	0	0.281
Duncan 3b	4	0	0	0	0	0	0.343
GeWilliams lf	3	0	0	1	0	0	0.333
RRivera cf	2	0	1	1	1	0	0.5
BeWilliams ph	1	0	0	0	0	1	0.278
DJeter ss	3	1	1	0	0	0	0.275
TOTALS	31	3	7	3	3	1	—

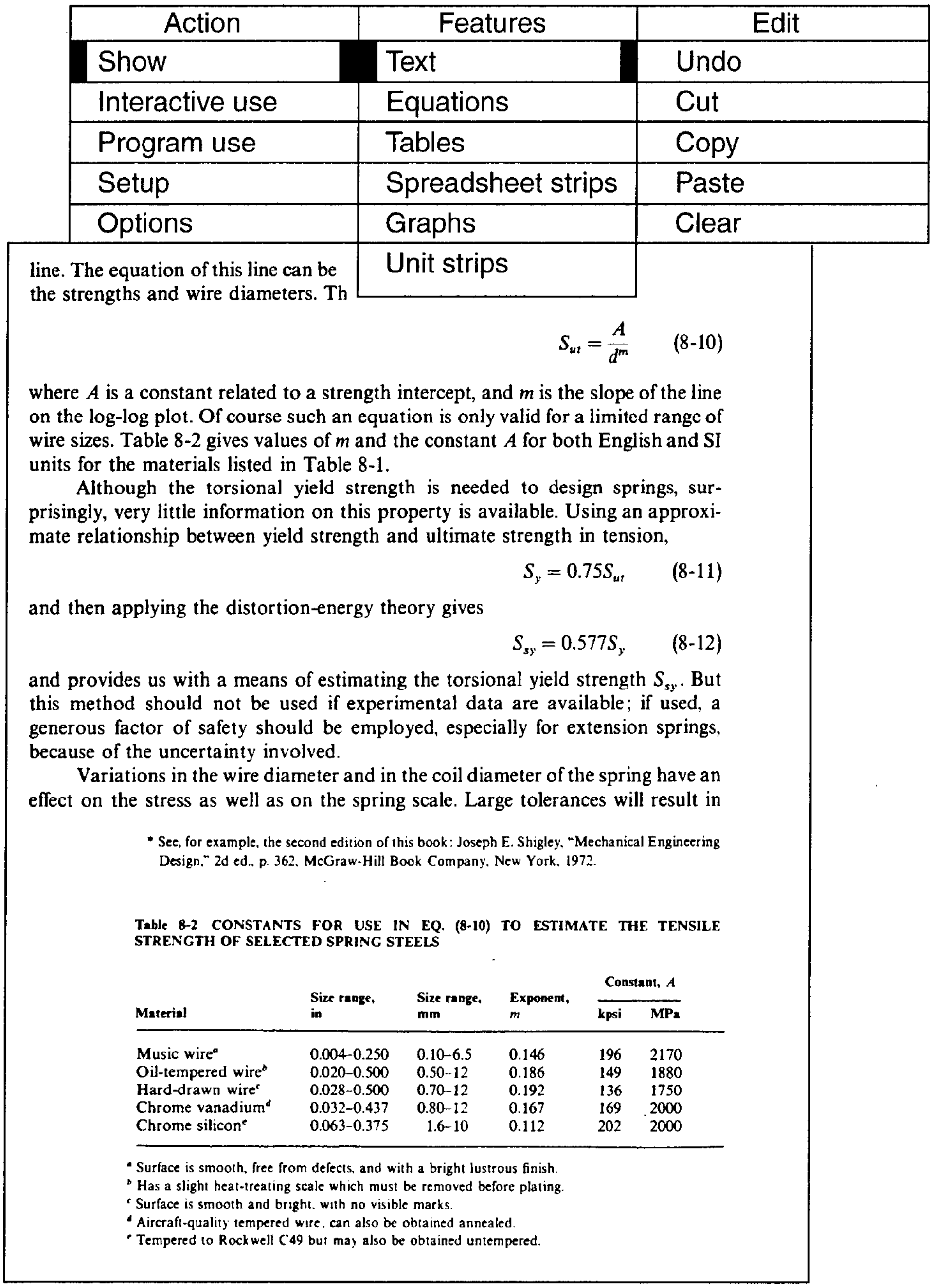
Why did I choose the baseball example? A six-year-old understands a baseball table. A child of four understands a table of menus and prices at a favorite restaurant! The knowledge of how to process tables is general. Humans gain such process knowledge independent of the domain. Initially, they apply the processing knowledge to simple (favorite) domains and later extend it to complex domains. A student may have problems with school homework but understands tables comparing automobiles in the April issue of *Consumer Reports*. These tables do not even use numbers in the cells. Circles filled with red and black colors are used (see fig. 19), and footnotes explain what the various colored circles mean. As people grow, they comprehend more complex table constructions. With help, they understand tables of increasing complexity. And they continually add more complex tables to their repertoire. That is the nature of our learning.

Although I argued in the foregoing that databases are not representations for tables, it is by no means suggested that tables (even Natural Computing tables) will replace databases. Natural Computing will provide interfaces to databases so that their attractive features can be used in addition to tables. Such a strategy allows tables to be used in their most natural form while large amounts of data can be handled efficiently through databases.

15. Example Usage of Natural Computing Tables

Based on the examples given in the survey in section 6 and the scheme for representing tables in computers given in section 8, one will be able to produce documents containing a variety of tables. As stated previously, such tables will manifest themselves in three basic forms: (1) tables embedded in text, (2) tables that can be used interactively, and (3) tables that can be built into procedures. Figure 36 shows a table embedded in text, along with the menus available to manipulate both the document and the tables. When the user wishes to use the table to obtain values, he or she activates the table (changing it to an interactive mode) as shown in figure 37. In this case, a highlighted table appears and presents input and output boxes (not shown in the figure in order to avoid clutter). When the user types in an input value, result values are outputted by the system. As stated in section 12, this interactive mode is used to test the table in isolation. An end user uses this mode to obtain values to comprehend trends or to use the result values in a series of calculations of a temporary nature. Finally, when a domain specialist wishes to incorporate a table into a procedure, a procedure is invoked and a table is connected to it, as shown in figure 38. Once a table is set into a procedure, it is embedded inside the procedure, and calculations are completed with the entire procedure (fig. 39).

Figure 36. A table embedded in text with Natural Computing menus.



Action	Features	Edit
Show	Text	Undo
Interactive use	Equations	Cut
Program use	Tables	Copy
Setup	Spreadsheet strips	Paste
Options	Graphs	Clear
	Unit strips	

line. The equation of this line can be
the strengths and wire diameters. Th

$$S_{ut} = \frac{A}{d^m} \qquad (8\text{-}10)$$

where A is a constant related to a strength intercept, and m is the slope of the line on the log-log plot. Of course such an equation is only valid for a limited range of wire sizes. Table 8-2 gives values of m and the constant A for both English and SI units for the materials listed in Table 8-1.

Although the torsional yield strength is needed to design springs, surprisingly, very little information on this property is available. Using an approximate relationship between yield strength and ultimate strength in tension,

$$S_y = 0.75 S_{ut} \qquad (8\text{-}11)$$

and then applying the distortion-energy theory gives

$$S_{sy} = 0.577 S_y \qquad (8\text{-}12)$$

and provides us with a means of estimating the torsional yield strength S_{sy}. But this method should not be used if experimental data are available; if used, a generous factor of safety should be employed, especially for extension springs, because of the uncertainty involved.

Variations in the wire diameter and in the coil diameter of the spring have an effect on the stress as well as on the spring scale. Large tolerances will result in

* See, for example, the second edition of this book: Joseph E. Shigley, "Mechanical Engineering Design," 2d ed., p. 362, McGraw-Hill Book Company, New York, 1972.

Table 8-2 CONSTANTS FOR USE IN EQ. (8-10) TO ESTIMATE THE TENSILE STRENGTH OF SELECTED SPRING STEELS

Material	Size range, in	Size range, mm	Exponent, m	Constant, A kpsi	Constant, A MPa
Music wire[a]	0.004–0.250	0.10–6.5	0.146	196	2170
Oil-tempered wire[b]	0.020–0.500	0.50–12	0.186	149	1880
Hard-drawn wire[c]	0.028–0.500	0.70–12	0.192	136	1750
Chrome vanadium[d]	0.032–0.437	0.80–12	0.167	169	2000
Chrome silicon[e]	0.063–0.375	1.6–10	0.112	202	2000

[a] Surface is smooth, free from defects, and with a bright lustrous finish.
[b] Has a slight heat-treating scale which must be removed before plating.
[c] Surface is smooth and bright, with no visible marks.
[d] Aircraft-quality tempered wire, can also be obtained annealed.
[e] Tempered to Rockwell C49 but may also be obtained untempered.

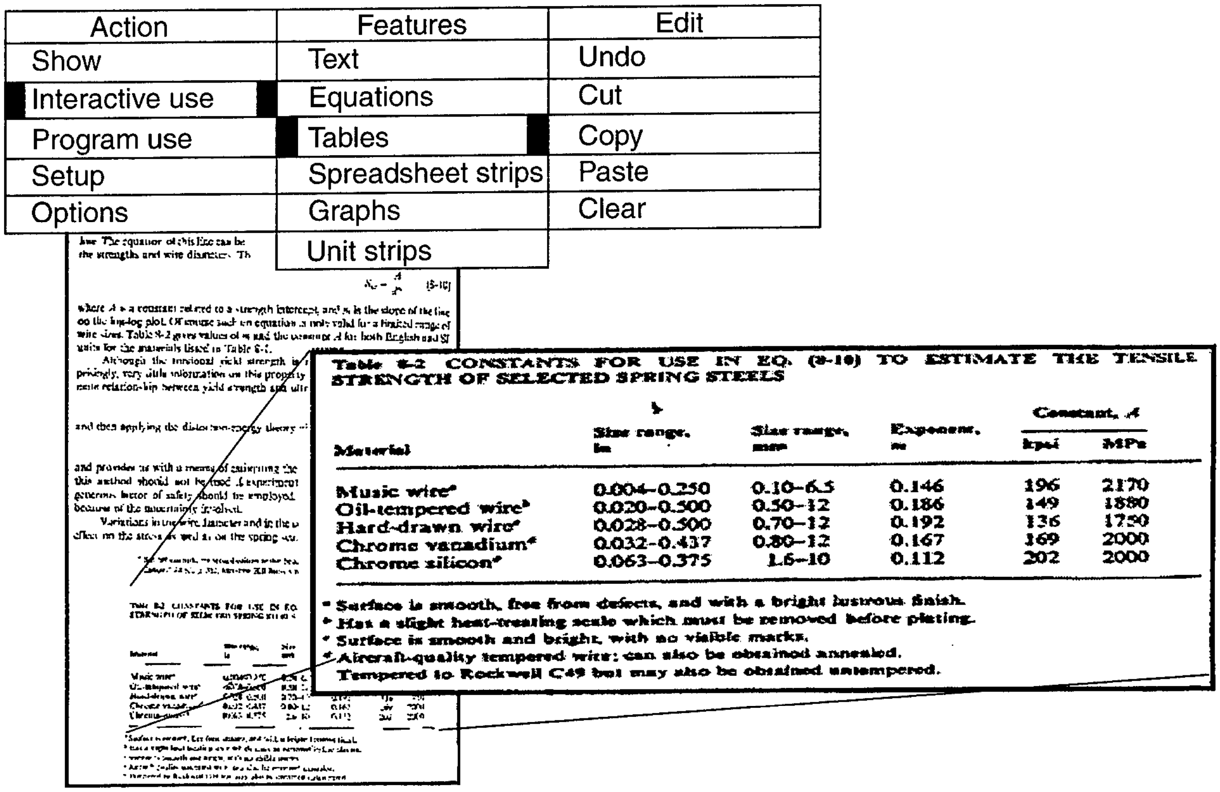
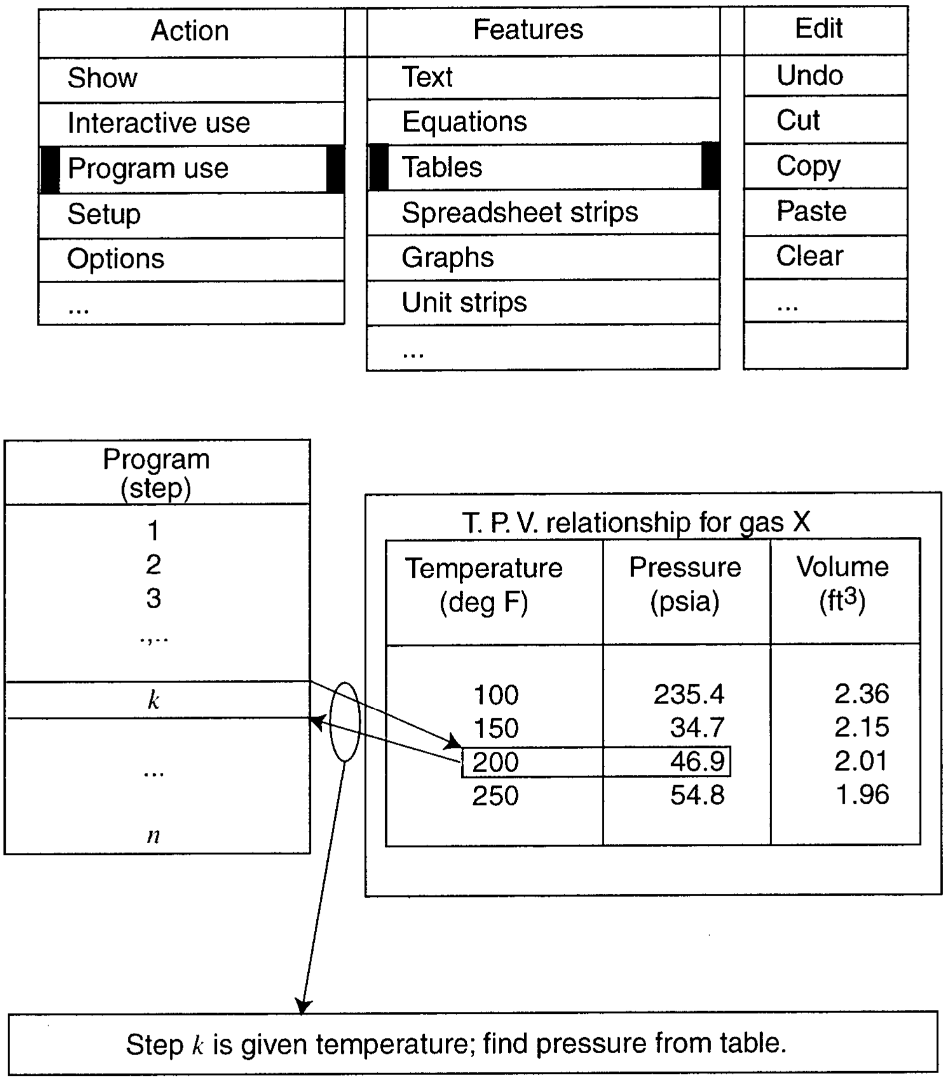


Figure 37. A table activated for interactive use.

Figure 38. A table being set into a procedure.



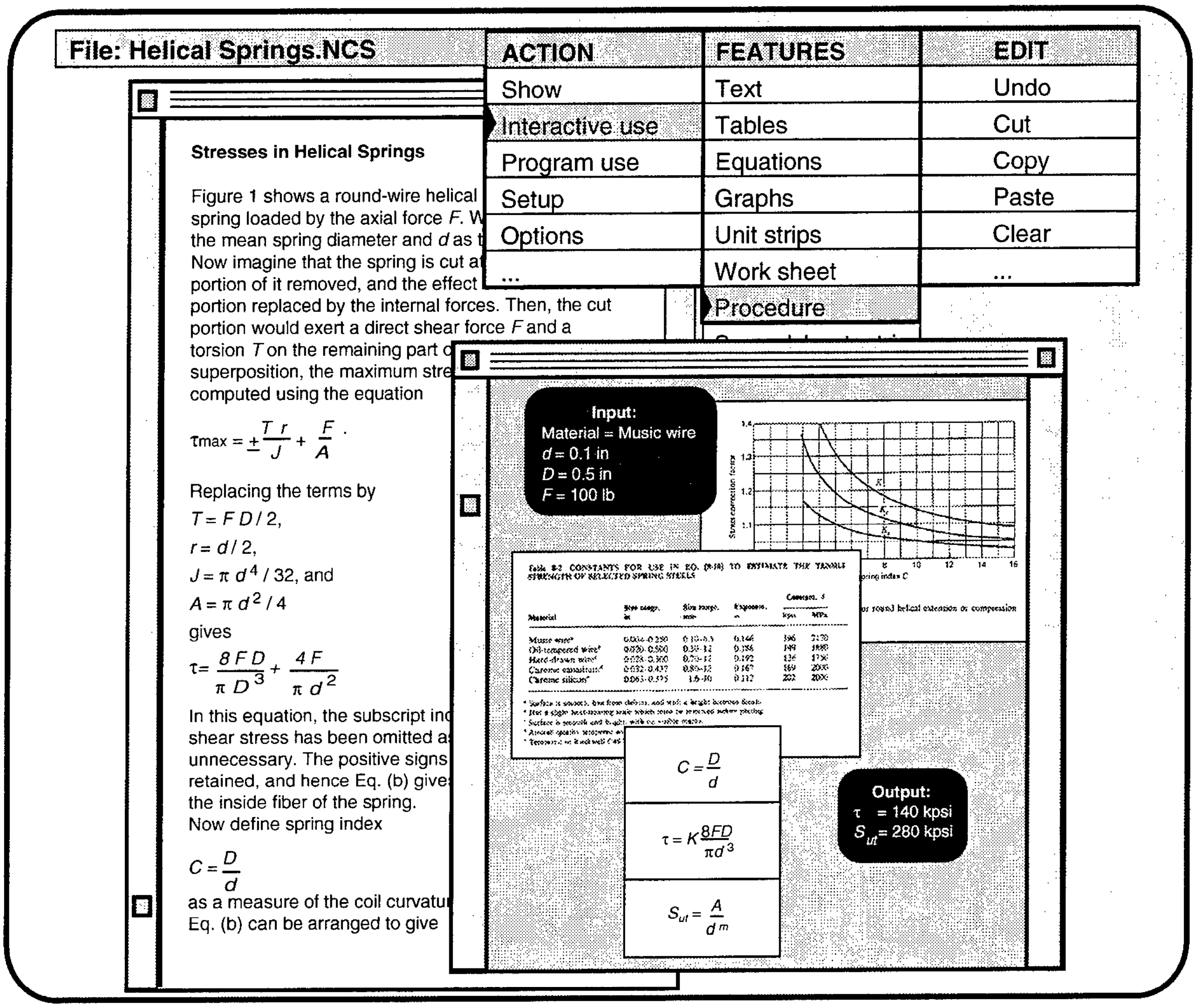


Figure 39. A Natural Computing screen showing a document with a procedure.

16. Conclusions

Tables are ubiquitous means of depicting functional relationships used freely in paper-based documents, such as textbooks, handbooks, journals, newspapers, and flyers. Although tables are complex representations, people use them with consummate ease. Although database systems have been misunderstood to represent real tables, they are at best representations of simple tables containing discrete domain data. Although database systems have been and will continue to be highly useful, they are not replacements for true tables. I have demonstrated in this paper the need for a richer and more appropriate computer representation for tables.

The development of a base table class and several derived classes is the appropriate method to represent and use tables. The structure of a table has been analyzed, and the components have been identified. The table anatomy presented in this report has the advantage that it allows a variety of tables to be represented and used. Various component classes are also described. Interpolation and extrapolation, where permissible, can be performed with the table methods briefly presented in this report. The display of information in natural-looking tables both allows the user to visualize opportunities in the data in a table and warns the user of pitfalls in the data.

If tables are represented and used as described in this report, computing will take a giant step toward making a reality the natural forms of data representation (such as tables). Once such forms of representation and processing are available, we can realize electronic handbooks, textbooks, documents, journals, and bulletins with the capability to interchange data and objects seamlessly. Knowledge from these various electronic library sources can be exchanged and combined, and computational procedures and systems can be developed that will be more powerful, economical, and expeditious than those created in the traditional software development cycle.

Acknowledgments

The author appreciates Ms. Laura Mann's efforts in obtaining permission to use copyrighted material from several publishers.

The author gratefully acknowledges permission to use copyrighted material granted by—

- McGraw-Hall Book Co., New York, NY,
- The Washington Post Company, Washington, DC,
- Consolidated Plastics Company, Inc.,
- Consumer Reports, New York, NY, and
- H&R Block Company.

References

Baumeister, T. and L. S. Marks, eds. (1964). *Mechanical Engineers' Handbook*, 6th ed., McGraw-Hill Book Co., New York, NY.

Coad, Peter, and Edward Yourdon (1991). *Object-Oriented Analysis*, 2nd ed., Yourdon Press Computing Series.

Cumpsty, N. A. (1989). *Compressor Aerodynamics*, Longman Scientific and Technical, Harlow, England.

Date, C. J. (1995). *An Introduction to Database Systems*, 6th ed., Addison-Wesley Publishing Company, Reading, MA.

U.S. Government Printing Office (GPO) (1984). *U.S. Government Printing Office Style Manual* (March).

Gartmann, Hans, ed. (1970). *De Laval Engineering Handbook*, 3rd ed., McGraw-Hill Book Co., New York, NY, 1–18.

H&R Block (1990). *Tax Preparation Guides*.

Karamchetty, Som D. (1997). Natural Computing, U.S. Patent No. 5,680,557 (October 21).

Karamchetty, Som D. (2000). *Natural Computing: Its Impact on Software Development*, U.S. Army Research Laboratory, ARL-TR-2040.

Keenan, Joseph H., Frederick G. Keyes, Philip G. Hill, and Joan G. Moore (1969). *Steam Tables: Thermodynamic Properties of Water Including Vapor, Liquid, and Solid Phases*, John Wiley and Sons, Inc., New York, NY.

Lemay, Laura (1996). *Teach Yourself Web Publishing With HTML 3.0 in a Week*, 2nd ed., Sams.Net Publishing, Indianapolis, IN, 194, 440.

Lester, C. B. (1994). *Hydraulics for Pipeliners, Vol. 1. Fundamentals*, 2nd ed., Gulf Publishing Co., Houston, TX.

Microsoft Word Version 7.0 (1995). Microsoft Corporation, Redmond, WA.

Morris, Mary E. S. (1996). *HTML for Fun and Profit*, Sunsoft Press.

Shaler, Sally, and Stephen J. Mellor (1988). *Object-Oriented Systems Analysis—Modeling the World in Data*, Yourdon Press Computing Series, Prentice Hall, Englewood Cliffs, NJ.

Shigley, Joseph E. (1977). *Mechanical Engineering Design*, 3rd ed., McGraw-Hill Book Co., New York, NY.

Van Wylen, Gordon J., and Sonntag, Richard E. (1965). *Fundamentals of Classical Thermodynamics*, John Wiley and Sons, New York, NY.

Streeter, V. L. (1971). *Fluid Mechanics*, 5th ed., McGraw-Hill Book Co., New York, NY.

The Washington Post (1996). Monday, May 27, p C6.

Wolfram, Stephen (1991). *Mathematica, A System for Doing Mathematics by Computer*, 2nd ed., Addison-Wesley Publishing Company, Reading, MA.

Distribution

Admnstr
Defns Techl Info Ctr
Attn DTIC-OCP
8725 John J Kingman Rd Ste 0944
FT Belvoir VA 22060-6218

Ofc of the Secy of Defns
Attn ODDRE (R&AT)
The Pentagon
Washington DC 20301-3080

Ofc of the Secy of Defns
Attn OUSD(A&T)/ODDR&E(R) R J Trew
3080 Defense Pentagon
Washington DC 20301-7100

AMCOM MRDEC
Attn AMSMI-RD W C McCorkle
Redstone Arsenal AL 35898-5240

CECOM
Attn PM GPS COL S Young
FT Monmouth NJ 07703

Dir for MANPRINT
Ofc of the Deputy Chief of Staff for Prsnnl
Attn J Hiller
The Pentagon Rm 2C733
Washington DC 20301-0300

TECOM
Attn AMSTE-CL
Aberdeen Proving Ground MD 21005-5057

US Army ARDEC
Attn AMSTA-AR-TD M Fisette
Bldg 1
Picatinny Arsenal NJ 07806-5000

US Army Info Sys Engrg Cmnd
Attn ASQB-OTD F Jenia
FT Huachuca AZ 85613-5300

US Army Natick RDEC
Acting Techl Dir
Attn SSCNC-T P Brandler
Natick MA 01760-5002

US Army Simulation, Train, & Instrmntn Cmnd
Attn J Stahl
12350 Research Parkway
Orlando FL 32826-3726

US Army Soldier & Biol Chem Cmnd
Dir of Rsrch & Techlgy Dirctrt
Attn SMCCR-RS I G Resnick
Aberdeen Proving Ground MD 21010-5423

US Army Tank-Automtv Cmnd Rsrch, Dev, & Engrg Ctr
Attn AMSTA-TR J Chapin
Warren MI 48397-5000

US Army Train & Doctrine Cmnd
Battle Lab Integration & Techl Dirctrt
Attn ATCD-B J A Klevecz
FT Monroe VA 23651-5850

US Military Academy
Mathematical Sci Ctr of Excellence
Attn MDN-A LTC M D Phillips
Dept of Mathematical Sci Thayer Hall
West Point NY 10996-1786

Nav Surface Warfare CtrA
Attn Code B07 J Pennella
17320 Dahlgren Rd Bldg 1470 Rm 1101
Dahlgren VA 22448-5100

DARPA
Attn S Welby
3701 N Fairfax Dr
Arlington VA 22203-1714

Palisades Inst for Rsrch Svc Inc
Attn E Carr
1745 Jefferson Davis Hwy Ste 500
Arlington VA 22202-3402

NASA Langley Rsrch Ctr
Vehicle Techlgy Ctr
Attn AMSRL-VT W Elber
Hampton VA 23681-0001

Distribution (cont'd)

US Army Rsrch Lab
Attn AMSRL-WM I May
Aberdeen Proving Ground MD 21005-5000

US Army Rsrch Lab
Attn AMSRL-RO-D C Chang
Attn AMSRL-RO-EN W Bach
PO Box 12211
Research Triangle Park NC 27709

US Army Rsrch Lab
Attn AMSRL-CI N Radhakrishnan
Aberdeen Proving Ground MD 21005-5067

US Army Rsrch Lab
Attn AMSRL-HR R L Keesee
Aberdeen Proving Ground MD 21005-5425

US Army Rsrch Lab
Attn AMSRL-VP R Bill
21000 Brookpark Rd
Cleveland OH 44135-3191

US Army Rsrch Lab
Attn AMSRL-SL J Wade
White Sands Missile Range NM 88002

US Army Rsrch Lab
Attn AMSRL-DD J Miller
Attn AMSRL-CI-AI-A Mail & Records Mgmt
Attn AMSRL-CI-AP Techl Pub (3 copies)
Attn AMSRL-CI-LL Techl Lib (3 copies)
Attn AMSRL-IS J D Gantt
Attn AMSRL-IS-CB L Tokarcik
Attn AMSRL-IS-CD P Jones
Attn AMSRL-IS-CI B Broome
Attn AMSRL-IS-CS G Racine
Attn AMSRL-IS-D COL M R Kindl
Attn AMSRL-IS-D P Emmerman
Attn AMSRL-IS-D R Slife
Attn AMSRL-IS-E D Brown
Attn AMSRL-IS-TA J Gowens
Attn AMSRL-SE J Pellegrino
Attn AMSRL-SE-EP S Karamchetty
(30 copies)
Attn AMSRL-ST C I Chang
Adelphi MD 20783-1197

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 2000	Final, 1995–1997

4. TITLE AND SUBTITLE Natural Computing: Analysis of Tables for Computer Representation

5. FUNDING NUMBERS
DA PR: N/A
PE: N/A

6. AUTHOR(S) Som Karamchetty

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Attn: AMSRL-IS-C email: skaramch@arl.mil
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-2041

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
ARL PR: N/A
AMS code: N/A

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
Some fundamental objects in practical documents have not been implemented in software so that they can be used easily for calculating. One such object is the table—despite the mistaken view that databases are adequate representations of tables. A survey of practical tables found in a variety of real-world documents reveals that many of their useful features are not captured in software. This report proposes data structures and computer representation for table objects. Through the adoption of such structures and representations, practical table objects can be developed for use by domain specialists. Such tables embedded in electronic documents can be used in interactive applications to retrieve data, but most importantly, they can be used as functional representations for copying and pasting into procedures and programs. Use of these table objects, together with other natural computing objects (such as equations, graphs, and procedures), will permit electronic documents like handbooks, textbooks, journals, and bulletins to be used seamlessly for calculations by both domain specialists and naive users. Such developments will reduce the lag between the availability of information and its use in calculations, and encourage the further development of knowledge. Software development for computation will change and its costs will be contained.

14. SUBJECT TERMS
Natural Computing, software engineering, object-oriented programming

15. NUMBER OF PAGES
64

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102