Simulation Software for Thermodynamic Models, Part 1

Forbes T. Brown
Professor of Mechanical Engineering, emeritus
Lehigh University
Bethlehem, PA
ftbmhb@aol.com or Ftb0@lehigh.edu

Key words: thermodynamics, thermodynamic properties, bond graphs, convection bond graphs, software

ABSTRACT
The extensive software available for simulating systems modeled by standard bond graphs does not address most thermodynamic systems, because they are not modeled effectively by those graphs. The convection bond graphs which do address those systems have not previously been implemented by generic simulation software. The complexities inherent in these bond graphs make the development of such software quite challenging; heretofore differential equations for each model have been written on an ad hoc basis from the bond graphs. Free downloadable software is described herein that deduces, organizes and executes differential equations for most convection and hybrid bond graphs, thereby simulating the unsteady behavior of lumped models that represent hybrid thermodynamic, mechanical, electrical and hydraulic systems. The imbedded computation of the thermodynamic properties of any of thirty-six different single or multiphase substances is generally accurate and efficient, requiring virtually no iteration. The user enters the bond graph largely through the use of two matrices; a graphical interface could be developed. Examples given include a steam catapult and a rock drill driven by compressed wet air. The inclusion of fluid kinetic energy and overall conclusions are made in part 2 of the paper.

1. INTRODUCTION

Most attempts to treat thermodynamic systems with bond graphs have employed traditional bonds and bond-graph elements, with restricted success. Several highly developed commercial software packages are available that convert these bond graphs to give dynamic simulations, as an Internet search will reveal. This software and indeed the bond graphs themselves, however, are at best awkward when applied to thermodynamic systems, and give almost no access to the thermodynamic properties of substances. Extending bond graphs to thermodynamic systems on an equal footing to other system types is natural, given the energy basis of the graphs, and crucial to their wide acceptance for the modeling and analysis of dynamic physical systems.

Such an extension has been introduced by the author through the idea of non-conventional convection bonds and certain resulting bond-graph elements. These represent naturally and simply the facts that two intensive states are necessary to define the state of a thermodynamic substance and that the second law of thermodynamics applies. A summary in the form of the slides of a plenary lecture delivered at the 2007 ICBGM conference is available at the author's university website [1]; details are best available from the author's textbook [2] although the original idea was published much earlier [3]. To largely avoid the need for iteration in the treatment of thermodynamic properties, software was developed that deduced a large number of properties and gradients thereof for a very restricted list of substances. The coding is now revised for use with the software package being reported herein, and extended to 35 different substances in the vapor, saturated mixture and in most cases liquid phases, plus air as a mixture of nitrogen, oxygen, argon and water. The coding as well as descriptive information is freely downloadable.

At this point the user would still have to write his own differential equations, using the bond graph as a guide, as illustrated by the author in references cited above, amongst others. It was desired that a software package become available for convection bond graphs that, like the commercial programs available for conventional bond graphs, automatically converts the graph to differential equations which are then automatically executed numerically. This is a significant challenge, considering the relative complexity of the convection bond graph elements. The programs identified in the present paper represent a major advance in this direction. They are not the last word, however, because treating some non-linearities can be awkward and chemical reaction is not included. Further, there is no graphical interface; the user must code his or her model using matrices, which fortunately is easily done. For most purposes these are not major limitations. The new coding ("Simulation Package for Convection Bond Graphs") is also downloadable free from the same website as the fluid properties and the plenary lecture [1].

It is assumed here that the reader is familiar with conventional bond graphs. Note that the author truncates the traditional symbol for the transformer element, which is *TF*, to just *T*; this symbol doubles as the modulus of the transformer, or the ratio of the output flow to the input flow. Similarly, the symbol for the gyrator, *GY*, is truncated to *G*, and this symbol also doubles for its modulus, which is the ratio of the effort on either side to the flow on the opposite side. Considerable use also is made of the irreversible two-port element *RS* with conventional bonds, which conserves energy but not entropy and therefore is useful in representing heat conduction and frictional conversion from mechanical energy to thermal energy. This element is not a creation of the author and appears in many bond graph papers, but may not be familiar to the reader. The software treats friction and heat transfer as distinct variations of this element, so some description is given.

Since the convection bond and resulting bond graph elements are not so commonly known as conventional bond graphs, a review is presented next.

2. CONVECTION BOND GRAPHS FOR UNSTEADY MODELS

A convection bond, shown in part (a) of Fig. 1, describes the flow of a substance through a port. The flow variable, labeled below or to the right of the bond, is the mass flow rate, $\dot{m}$. The half-arrow, which is to the right in both of the examples shown, defines the direction of flow for which $\dot{m}$ is positive. Unlike a simple bond, which has a single effort variable labeled above or to the left of the bond, there are two independent effort variables, which are intensive thermodynamic properties. The need for two indepen-pendent efforts is represented by the use of a dashed line parallel to a solid line.

The causal stroke, a short line drawn perpendicular to the main bond lines, indicates from which end of the bond the effort causal variable, which is the pressure *P*, and $\dot{m}$ are causal for computational purposes. The case on the left in the figure describes *P* as causal from the left and $\dot{m}$ as causal from the right. The case on the right is the opposite. The enthalpy h is the power effort variable, in that the product $h\dot{m}$ is the power conveyed along the bond. Thus the most prominently cited sufficient pair of effort variables is *P* and *h*. Nevertheless, other intensive variables also are efforts, most importantly the temperature, θ, and specific volume, *v*, which are used as state variables, and the entropy, *s*. The causal stroke does not describe the causalities of *h*, θ, *v* or *s*, which are simply defined by the direction of the flow.

A lumped fluid volume with thermodynamic energy storage, two convection ports, a conventional heat conduction port with integral thermal resistance, and a conventional volume change port, is indicated with the symbol *CS*, as shown in part (b) of Fig. 1. The *C* indicates the energy storage, and the *S* the potential for irreversibility due to the mixing of an inflow having a different enthalpy than the volume itself, and any thermal conductance that may be present. A convection bond with outflow has the enthalpy of the substance in the volume as its power effort variable, which is also a causal output, whereas the enthalpy of a convection bond with inflow depends causally on whatever is connected to the other end of the bond. The volume change bond drawn below the element has as its flow $\dot{V}$, which is the rate of change of the volume of the element, *V*. Its effort variable is the pressure in the volume, and the product of the two is the mechanical power flowing out from the element. The bond drawn on top of the element (it could also be placed below)

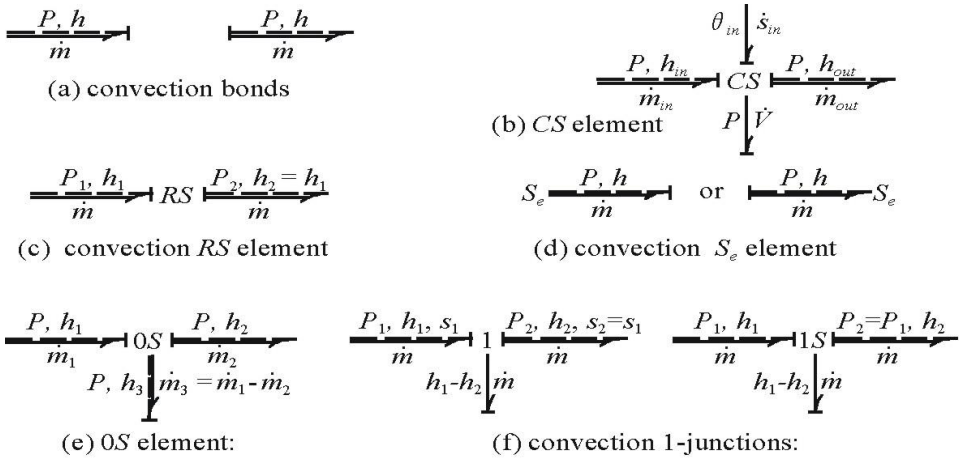

Fig. 1 Convection Bonds and Bond Graph Elements

represents a heat conduction port, with applied temperature θ_{in} and entropy flux $\dot{s}_{in}$, the product of which is the heat in, $\dot{Q} = H(\theta_{in} - \theta)$, where H is the thermal conductance. The volume change and heat conduction bonds are distinguished by their oppositely directed power half-arrows and causal strokes.

The three state variables for the *CS* element used in the software are V, m and θ. The causal strokes shown describe integral causality; $\dot{V}$ is integrated to give V, $\dot{m} = \dot{m}_{in} - \dot{m}_{out}$ is integrated to give m, from which $v = V/m$, and $\dot{\theta}$ is integrated to give θ, for which the first law of thermodynamics (as developed on p. 881 of [2]) gives

$$\frac{\partial\theta}{\partial t} = \frac{1}{m\,\partial u/\partial\theta}\left[\dot{Q} + (h_{in} - h)\dot{m}_{in} - (h_{out} - h)\dot{m}_{out} + \left(P + \frac{\partial u}{\partial v}\right)(v\dot{m} - \dot{V})\right]$$

Here, h_{out} is different from h if and only if there is any gravity separation of liquid and vapor phases. Once v and θ are known, all other intensive (effort) variables of interest are computed by the Thermodynamic Properties Package, described below.

When there is identically no flow on any of the bonds, the bond itself can be removed. For example, a fluid chamber of fixed size has no volume change bond.

Fluid flow through a restriction is represented by a convection *RS* element, as shown in part (c) of Fig. 1, which is not to be confused with the *RS* element used with conventional bonds. This is a generalization of that element, in which the *R* suggests resistance and the *S* suggests irreversibility. No heat conduction or fluid leakage is allowed, and therefore the two mass flows are equal, as are the two energy flows and therefore the two enthalpies; it is the two pressures that are different. The software assumes the admittance causality shown. The computation of the flow assumes ideal flow in either direction of a vapor, a liquid or a mixture thereof through an equivalent orifice of known area, accounting for any separation between phases that may result from the detailed geometry and gravity. Note that no difference is assumed between the stagnation and actual enthalpies, since the kinetic energy of the fluid is ignored.

A source or sink of fluid with specified intensive properties is designated by the S_e element, shown in part (d) of Fig. 1, which is a direct generalization of the conventional element with the same symbol. The temperature and specific volume must be specified, except in the case of a known sink for which the pressure is allowed as a sufficient substitute.

Flow-summing junctions of three convection bonds that have the same pressure are given by a generalization of the 0-junction designated as the 0*S*-junction, as shown in part (e) of Fig. 1. The *S* indicates the potential for irreversibility that results when two flows with different temperatures or enthalpies are merging, such as when $\dot{m}_1 > 0$ and $\dot{m}_2 < 0$. The causal strokes indicate the causal source of the common pressure. Six different combinations of the signs of the three flows are possible. The special case of diverging flows is reversible, but the software still recognizes it as an 0*S* junction.

Effort-summing or 1-junctions have two convection bonds and one conventional bond, as shown in part (f) of Fig. 1, allowing those portions of a system modeled by convection bonds to be attached to those portions modeled by conventional bonds. The flow, $\dot{m}$, is common, and the enthalpies sum. Two versions of this junction are used. The basic 1-junction, shown on the left, is reversible, which implies that the entropies of the two convection bonds are equal. For the irreversible 1*S*-junction shown on the right, on the other hand, the pressures of these two bonds are the same, instead. The causal stokes shown are assumed by the software.

These six elements are sufficient for models that do not include kinetic energy. Other convection elements described by the author elsewhere (*H*, *HS*, *HRS* and the convection *C*) are needed only for non-dynamic models or have largely theoretical or pedagogical value.

Two additional elements are introduced in Part 2 of this paper to represent the effects of fluid inertance and kinetic energy.

3. MODELING EXAMPLES AND MATRIX CODING

Critical details are now explained through the use of three examples; further examples are given in Part 2. The first two examples are the same as or similar to those given in the author's book [2], allowing the reader to compare the use of the new software with the more difficult method of writing differential equations based on inspection of the bond graph.

The first example, shown in Fig. 2, involves a simple fixed-field DC motor with a gear-driven simple mass-spring-dashpot load, which is offered to illustrate how the software applies to non-thermodynamic systems.

For purposes of the software, the bond graph elements are numbered and the numbers are circled. The elements are specified by the matrix *el*. Specifically, the i^{th} row of the matrix *el* describes the i^{th} element with three numbers. As given in Table 1, the first number designates the type of element. The second number is 0 in most cases but confers special meaning for six of the element types. The third number (or substitute symbol with value specified later) specifies a parameter value for the element, when such exists.

The bonds are numbered separately; these numbers are not circled. The connectivity of the bonds is described by the matrix b. Specifically, the jth row of this matrix describes the jth bond, also with three numbers. The first is the number of the element adjacent to the causal stroke, and the second is the number of the element at the other end of the bond. The third number is one of the following: 1, -1, 2 or -2. 1 or -1 specifies a simple bond (the only kind present in this example), and 2 or -2 specifies a convection bond. Positive values indicate that the power convention half-arrow is at the same end of the bond as the causal stroke; negative values designate the opposite.

The two matrices applied to the DC motor example are, as the reader can verify,

$$el = \begin{bmatrix} 7 & 0 & Se \\ 1 & 0 & 0 \\ 11 & 0 & G \\ 9 & 0 & T \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 15 & 0 & R1 \\ 12 & 1 & C \\ 15 & 0 & R2 \\ 13 & 1 & I \end{bmatrix}; \qquad b = \begin{bmatrix} 2 & 1 & 1 \\ 3 & 2 & 1 \\ 3 & 4 & -1 \\ 4 & 5 & -1 \\ 6 & 5 & 1 \\ 2 & 7 & -1 \\ 5 & 8 & -1 \\ 6 & 9 & -1 \\ 10 & 6 & 1 \end{bmatrix}.$$

The parameters for the rotational spring *C* and inertia *I* used to produce the sample result plotted in Fig. 3 are the constants 0.05 rad/n-m and 0.2 kg m^2, respectively; although power-law functions are allowed. Other fixed parameters are G = 1 v/N m and R_1 = 1 ohm for the motor, gear ratio T = 0.2, and load damping R_2 = 1 N m s. The applied voltage is a sinusoid with an amplitude of 5.0 v and a frequency of 4 cycles/s. The system starts at rest. The results agree exactly with those found on p. 169 in the textbook [2] using more primitive methods.

The second example, shown in Fig. 4, is a primitive steam catapult for launching an aircraft on a carrier deck. It comprises only an adiabatic cylinder filled with a fixed quantity of saturated water and water vapor, a frictionless piston and an aircraft modeled as a simple mass with a fixed thrust force *F* from its engine. The *CS* element for the volume therefore has no convection bonds or heat transfer bonds, only a volume change bond.

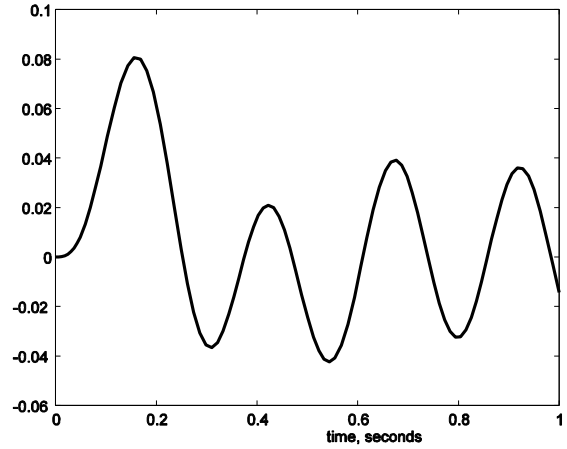

Fig. 3 Simulated Angular Position of Load of DC Motor System, rad

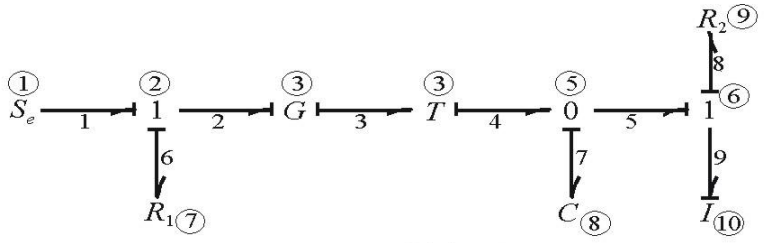

Fig. 2 Example of a System with a DC motor and Load

Table 1 Coding of the matrix *el*

Elements including convection bonds:

Element	1st column	2nd column	3rd column	comment
RS	3	see below*	orifice area, m^2	
CS	4	0	heat cond. coeff, W/K	
0*S*	18	0	0	includes convec. 0
1	1	0	0	also non-convection
1*S*	2	0	0	
S_e	14	spec. volume,m^3/kg	temperature, K	for sources or sinks
S_e	14	0	pressure, Pa	for sinks only
IRS	5	length of element	cross-sectional area	fluid inertance
AC	6	recip. of left-side area	recip. of right-side area	area change

Elements with simple bonds only:

Element	1st column	2nd column	3rd column	comment
0	0	0	0	
S_e	7	0	value of effort	
S_f	8	0	value of flow	
S_q	10	0	value of power	
T	9	0	transformer modulus	
G	11	0	gyrator modulus	
R	15	0	value of resistance	
C	12	exponent**	value of *C*	
I	13	exponent**	value of *I*	
RS	16	0	conduction modulus	for heat conduction
RS	17	exponent**	value of *R*	for friction

* The 2nd column for convection *RS* element gives numbers that allow for gravity separation of vapor or liquid in two-phase situations, as follows. ("Forward flow" is in the direction of the power half-arrows, and "reverse flow" is in the opposite direction.)

0	no gravity vapor or liquid separation
2	vapor separation for forward flow, no separation on reverse flow
3	liquid separation for forward flow, no separation on reverse flow
4	no separation for forward flow, vapor separation on reverse flow
5	no separation on forward flow, liquid separation on reverse flow
6	gravity vapor separation for both flow directions
7	liquid separation for forward flow, vapor separation on reverse flow
8	vapor separation on forward flow, liquid separation on reverse flow
9	liquid separation for both flow directions

** The constitutive relations for the *C*, *I* and the friction *RS* elements are in the form

Output variable = parameter · sign(input variable) · abs(input variable)^exponent,

in which the parameters are 1/*C*, 1/*I* and *R*, respectively, and the input variables are displacement, momentum and velocity, respectively.

The two matrices applied to the steam catapult are, as the reader can verify,

$$el = \begin{bmatrix} 4 & 0 & 0 \\ 9 & 0 & T \\ 1 & 0 & 0 \\ 13 & 1 & I \\ 7 & 0 & F \end{bmatrix}; \qquad b = \begin{bmatrix} 2 & 1 & 1 \\ 3 & 2 & 1 \\ 4 & 3 & 1 \\ 3 & 5 & 1 \end{bmatrix}.$$

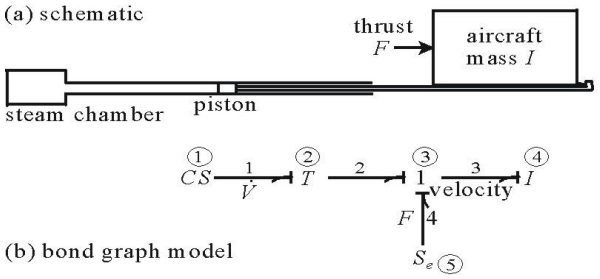

Fig. 4 Example of a Steam Catapult

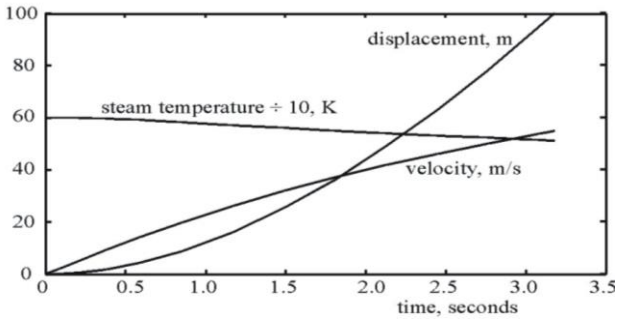

Fig. 5 Selected Results for a Steam Catapult

Here, T is the modulus of the transformer, which is the reciprocal of the area of the piston, I is the inertia (mass) of the aircraft (which is designated as constant or linear through the 1 in the second column) and F is the thrust force. In the results plotted in Fig. 5, the area is taken as 0.024 m^2, the mass as 16,000 kg, and the force as 10^5 N. (The model and results agree those in Problem 11.15 in the text, p. 926, and the solution manual.)

The third example is a pneumatically driven commercial rock drill, which employs a steel hammer, three chambers with variable volumes, and four restrictions to air flow that depend on the position of the hammer. Due to the proprietary nature of this device, only a schematic drawing and the bond graph are shown in Fig. 6, and the parameter values have been changed. Chambers 1 and 4 provide the downward force on the hammer, and chamber 2 provides an upward force. There are also five fixed chambers and associated fixed flow restrictions. The air is assumed to contain 1% of water by mass, and is modeled as a mixture of nitrogen, oxygen, argon and the water. The impact of the hammer with the rock is modeled as plastic. Selected simulation results are given in Fig. 7. Note that its water component is a saturated mixture for a portion of each cycle.

The matrices *el* and *b* become, as the reader can verify,

$$el = \begin{bmatrix} 4 & 0 & 0 \\ 4 & 0 & 0 \\ 4 & 0 & 0 \\ 4 & 0 & 0 \\ 4 & 0 & 0 \\ 4 & 0 & 0 \\ 4 & 0 & 0 \\ 4 & 0 & 0 \\ 3 & 0 & AS \\ 3 & 0 & AS \\ 3 & 0 & AS \\ 3 & 0 & AS \\ 3 & 0 & AS \\ 3 & 0 & asa0 \\ 3 & 0 & aa10 \\ 3 & 0 & a1e \\ 3 & 0 & aa20 \\ 3 & 0 & a2e \\ 13 & 1 & ms \\ 18 & 0 & 0 \\ 14 & vs & Ths \\ 14 & ve & The \\ 9 & 0 & -1/Ap(1) \\ 9 & 0 & 1/Ap(2) \\ 9 & 0 & -1/Ap(3) \\ 1 & 0 & 0 \\ 7 & 0 & -ms*g \end{bmatrix}; \quad b = \begin{bmatrix} 9 & 21 & 2 \\ 9 & 5 & -2 \\ 10 & 5 & 2 \\ 10 & 6 & -2 \\ 11 & 6 & 2 \\ 11 & 7 & -2 \\ 12 & 7 & 2 \\ 12 & 8 & -2 \\ 13 & 8 & 2 \\ 13 & 3 & -2 \\ 14 & 3 & 2 \\ 14 & 4 & -2 \\ 20 & 4 & 2 \\ 15 & 20 & 2 \\ 15 & 1 & -2 \\ 16 & 1 & 2 \\ 16 & 22 & -2 \\ 17 & 20 & 2 \\ 17 & 2 & -2 \\ 18 & 2 & 2 \\ 18 & 22 & -2 \\ 23 & 1 & 1 \\ 26 & 23 & 1 \\ 25 & 4 & 1 \\ 26 & 25 & 1 \\ 24 & 2 & 1 \\ 26 & 24 & 1 \\ 19 & 26 & 1 \\ 26 & 27 & 1 \end{bmatrix}.$$

. Another example on the author's web site is the charging of a tank by a compressor, which illustrates amongst other things the use of convection 1 and 1*S* junctions.

4. THE THERMODYNAMIC PROPERTIES PACKAGE

Most sources of thermodynamic properties require either interpolation or iteration, or fail to provide gradients particularly useful for dynamic simulations, such the NIST [4]. The author has prepared a

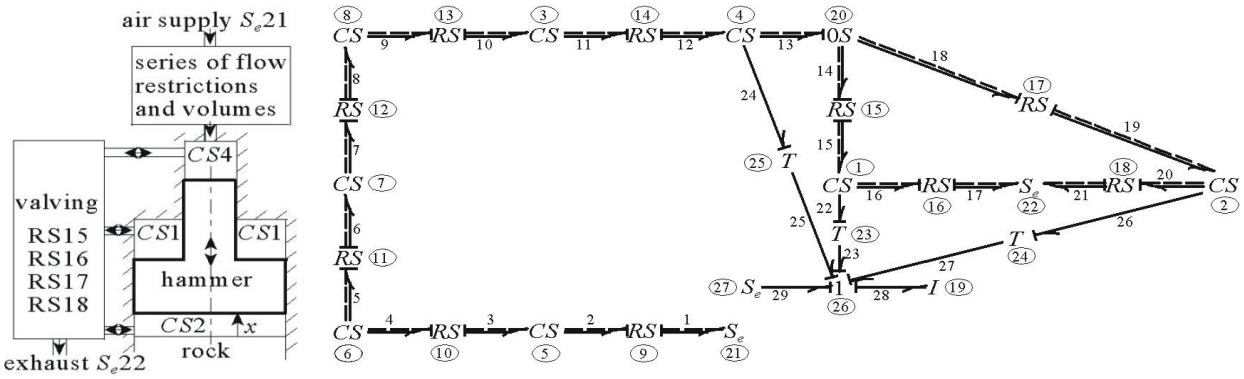

Fig. 6 Schematic and Bond Graph for a Rock Drill

MATLAB®-based package of programs ("Thermodynamic Properties Package"), freely downloadable from his web site [1], which efficiently and generally accurately computes 18 thermodynamic properties, and 16 more if needed, of 35 different common pure substances plus wet air as a mixture of pure substances. The vapor and saturated liquid/vapor mixture regions are included for all the substances. The compressed liquid region is included for most of them, but not the refrigerants and the metals. In these latter cases the simple bulk-modulus model is substituted. The files replace the much narrower ones listed in the author's book. More substances may be added at a later date, for example certain modern refrigerants.

To use the package, one first runs the data file for the substance of interest, for example "*datammonia.m*." Nothing further is needed if one subsequently runs the master file containing the matrices *el* and *b* to produce a simulation. If one simply wishes to make a particular evaluation of properties, the file "*therprop.m*" or "*airmix.m*" is run instead. For simulation, other files are called automatically as needed.

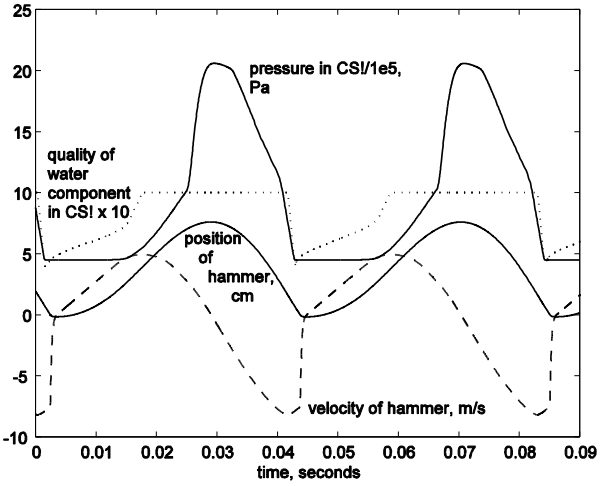

Fig. 7 Selected Results for the Rock Drill

5. THE SIMULATION FILES

Several example annotated simulation files and further explanations are given in the "Simulation Package for Convection Bond Graphs" that is freely downloadable from the author's website [1]. The master script MATLAB files, which include the matrices *el* and *b* and other defining information, are named *DCmotor.m, catapult.m* and *Rockdrill.m.* for the examples discussed herein. Such files can be adapted from the template file "*convectemplate.m*", which directs its various entries and describes the meanings of various symbols. The values of the parameters are specified and, unless modified in the generic function file "*convec.m*" that is called repeatedly, they are treated as constants. The number of *CS* elements is automatically computed as *p*, the number of *C* elements as *q*, and the number of *I* elements as *r*; the state vector comprises *p* fluid masses, *p* temperatures, *p* volumes, *q* displacements and *r* momenta, in that order, to give a total order of $3p+q+r$. (Further elements, described in Part 2, describe fluid inertia and kinetic energy.) The user also specifies the substance used for the thermodynamic portion of the system, if any, the initial conditions for the state vector and the duration of the simulation.

For the example of the DC motor, $p = 0$, $q = 1$ and $r = 1$, and the coding for the assumed initial condition vector is x0 = [.0001 .0001] (virtually zero values of displacement and velocity). For the example of the steam catapult (file *catapult.m*), $p = 1$, $q = 0$ and $r = 1$, and coding for the assumed initial condition vector is x0 = [V0/v0 600 V0 1e-8] where the initial volume is V0 = 0.335 m^2 and specific volume is v0 = 0.002 m^3 /kg. In the unusual instances of a compliance bonded to a 1-junction or an inertance bonded to a 0-junction, the intial conditions must not be identically 0. Finally, should there be a bond connecting a 1-junction to a 1*S*-junction or a *CS* element with positive inflow, the user makes an initial estimate of its temperature and specific volume, which need not be accurate. The default

integrator is the MATLAB ode45; one can substitute a different integrator, and also choose "options" that affect the speed and accuracy of the simulation. Finally, one can place instructions for plotting results at the end of the master program, or wait until the simulation is complete before issuing these commands.

If all the various parameters in the matrix *el* are intended to be the constants inserted into the master file, and certain other conditions are met, the relatively complex function file called by this file to carry out the simulation (*convec.m*) can be used without modification. This applies to the example of the steam catapult. Otherwise, appropriate adjustments are made, normally near the top of this file. A nonlinearity easily relatable to the state vector or to time can be satisfied by correcting the relevant element in the matrix *el.* For the DC motor (identified in the files by its flag number FG = 2), the sinusoidal dependence on time of the excitation voltage is specified. For the rock drill (FG=7), the areas of the four variable flow restrictions are made functions of the displacement of the hammer. Also, whenever the rate of penetration of the rock is positive, the contact force is set proportional to the displacement of the hammer into the rock plus a damping term. Whenever the rate of penetration is negative, the contact force is set equal to zero. For convenience, the constant gravity force on the hammer is added to the contact force. The file *convec.m* also contains special equations for other examples with other flag values discussed in Part 2 or otherwise discussed in the website.

The vectors *eb* and *fb*, respectively, give the efforts and flows on all of the bonds in the program *convec.m,* which is fairly well annotated. The first of its three major parts focuses on the convection portion of the model, calling either the program *therprop.m* or *airmix.m* (which in turn calls one of a set of vapor programs) to evaluate thermodynamic states and the program *convRS.m* to evaluate flows through restrictions. The directions of the flows through the restrictions are determined first, in so far as possible, the value 1 being flow in the direction of its power half-arrow and -1 being flow in the reverse direction. In the special case of convection bonds attached to 1-junctions, it is necessary for the user to specify these directions at the beginning of the program, as illustrated in Part 2. Much of the code in the first part deals with the problem of determining whether the directions of flow for the bond attached to 0*S*-junctions produce a diverging or a merging flow.

The program *convRS.m* is a radical revision of the approach documented in the book for calculating flows through orifices. Linearized relations are replaced by more accurate implementation of the conservation of entropy and use of the thermodynamic property package. Newton-Raphson iterations converge very rapidly, due to the use of analytical derivatives of the properties

The second major part of the program focuses on the conventional bond graph portion of the model. A matrix equation is established between the initially known and the unknown variables, which is solved by inversion. The vector of initially unknown variables is divided into sub-vectors, the first for bonds having only unknown effort, the second for bonds having only unknown flow and the third for bonds having both unknown effort and flow. If a *friction RS* element exists in the system, the final obligation of the user requires, usually at the beginning of the program, that the velocity variable attached to this element be related explicitly to input or other known variables, because of the nonlinearity of this element. (The piston-cylinder compressor described in Part 2 includes this requirement.)

The third major part of the program assembles the differential equations.

Should the user wish to plot a variable such as a pressure or a flow that is not readily related to state variables, a new state variable can be defined so its time derivative equals the desired variable. Then, this variable can be plotted by using the "diff" function of MATLAB. This feature is illustrated in several of the examples discussed in both parts of this paper.

Conclusions and recommendations are given at the end of Part 2.

6. REFERENCES

1. Brown, F.T.,2008, website www.lehigh.edu/~inmem; click on *faculty, emeritus*, and *F. Brown*
2. Brown, F.T., 2007, *Engineering System Dynamics, a Unified Graph-Centered Approach*, 2nd Ed., CRC Press, Taylor and Francis Group, FL
3. Brown, F.T., 1991, "Convection Bond Graphs," *Journal of the Franklin Institute*, v 328 n 5/6, pp.871-886
4. NIST Thermodynamic Properties of Refrigerants and Refrigerant Mixtures Database (REF_PROP), version 4.0, 1994, National Institute of Standards and Technology, Gaithersburg, MD