Extension of Simulation Software for Thermodynamic and Other Systems
Including Energy-Based Modeling

Forbes T. Brown
Professor of Mechanical Engineering, emeritus

Lehigh University
Bethlehem, PA
ftbmbh@aol.com or Ftb0@lehigh.edu

Key words: energy-based modeling, thermodynamics, thermodynamic properties, bond graphs, convection bond graphs

ABSTRACT

The author’s Simulation Package for Thermodynamic Systems, which is freely downloadable from the internet, was specially designed to accommodate convection bond graphs, although it also handles conventional and hybrid bond graphs. Several new backward-compatible features have been added to this simulation package to enhance the flexibility of its non-thermodynamic aspects, such as its treatment of electromechanical components. First, multiport compliances, particularly nonlinear ones, are treated when the user merely inputs expressions for their potential energy, using the symbolic math toolbox of MATLAB®, making otherwise more extensive and awkward pre-analysis unnecessary. Second, the user is prompted how to input nonlinearities and time-dependencies of the non-thermodynamic components of a model. Third, activated bonds are accommodated. Finally, the former need to avoid certain zero initial conditions in certain cases is eliminated. The result is a general-purpose simulation package that has some advantages over alternatives even when no thermodynamics is involved, although it retains its special role as the most practical way to simulate systems with flowing fluids that undergo temperature and volume changes.

1. BACKGROUND

The Achilles’ heel of conventional bond graphs is their virtual inability to deal effectively with systems that include flowing fluids with temperature and density changes with their essential consideration of entropy. This presents a major limitation, since bond graphs are purported to handle the general modeling and analysis of physical systems. Few proponents of bond graphs appear to be interested in thermodynamic systems, obscuring this fact. The transparency of convection bond graphs for thermodynamic systems, similar to the transparency of conventional bond graphs for non-thermodynamic systems, hopefully will encourage even those with little knowledge of thermodynamics to expand their horizon. The compatibility of the two graph forms is critical, as is the realization that convection graphs cannot be translated into simple conventional graphs because two intensive states are necessary to define the state of a thermodynamic substance, and the second law of thermodynamics must be incorporated.

The author is unaware of any other bond graph software that can simulate thermodynamic systems employing multiphase substances. There are many others that address non-thermodynamic systems; the author is uncertain whether these can handle the new features introduced herein.

The author is deeply engaged in a more significant development, as of this writing, in which hydraulic systems are addressed specifically. For those systems pressure needs to replace specific volume as a primary state variable in order to achieve accuracy. This development has also led to a restructuring of the primary simulation program, which shortly should lead to more robust and efficient coding for the multi-phase simulations as well. These results should be reported on the author’s website [1] later in 2012, and will not affect the results reported herein.

Complete information on convection bond graphs and the simulation package, including citations of ICBGM and other papers and updated basic software plus software for the examples described herein, is given on the author’s web site [1]. The graphs themselves were first introduced in 1991 [2], and the examples herein are taken from the author’s textbook [3]. The original Simulation Package was introduced in 2010 [4,5].

2. PROCEDURE; FIRST ENHANCEMENT

The standard procedure starts by drawing a bond graph of the system model of interest, including power half-arrows and a consistent set of causal strokes, numbering the elements and, separately, the
bonds. The author likes to circle the numbers of the elements to distinguish them from those of the bonds. The user enters the bond graph largely through the use of two matrices, $b$ and $el$; a graphical interface could be developed. (Anyone who may become interested in doing this is invited to contact the author.) The embedded computation of the thermodynamic properties of any of thirty-six

Table 1: Contents of the matrix $el$ (backward compatible)

<table>
<thead>
<tr>
<th>Element</th>
<th>1st column</th>
<th>2nd column</th>
<th>3rd column</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$RS$</td>
<td>3</td>
<td></td>
<td>see below*</td>
<td>orifice area, m$^2$</td>
</tr>
<tr>
<td>$CS$</td>
<td>4</td>
<td>0</td>
<td>heat cond. coeff, W/K</td>
<td></td>
</tr>
<tr>
<td>0S</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>includes conv. 0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>also non-convection</td>
</tr>
<tr>
<td>1S</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>$S_e$</td>
<td>14</td>
<td>spec. volume, m$^3$/kg</td>
<td>temperature, K</td>
<td>for sources or sinks</td>
</tr>
<tr>
<td>$S_f$</td>
<td>14</td>
<td>length of element</td>
<td>cross-sectional area</td>
<td>fluid inerance</td>
</tr>
<tr>
<td>$IRS$</td>
<td>5</td>
<td>recip. of upstream area</td>
<td>recip. downstream area</td>
<td>area change</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number</th>
<th>Exponent</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0 or 1**</td>
<td>value of effort</td>
</tr>
<tr>
<td>8</td>
<td>0 or 1**</td>
<td>value of flow</td>
</tr>
<tr>
<td>10</td>
<td>0 or 1**</td>
<td>value of power</td>
</tr>
<tr>
<td>9</td>
<td>0 or 1**</td>
<td>transformer modulus</td>
</tr>
<tr>
<td>11</td>
<td>0 or 1**</td>
<td>gyrator modulus</td>
</tr>
<tr>
<td>15</td>
<td>0 or 1**</td>
<td>value of resistance</td>
</tr>
<tr>
<td>12</td>
<td>exponent***</td>
<td>value of $C$</td>
</tr>
<tr>
<td>13</td>
<td>exponent***</td>
<td>value of $I$</td>
</tr>
<tr>
<td>16</td>
<td>0 or 1**</td>
<td>conduction modulus for heat conduction</td>
</tr>
<tr>
<td>17</td>
<td>exponent***</td>
<td>friction force for coulomb friction</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

* The 2nd column for convection $RS$ element gives numbers that allow for gravity separation of vapor or liquid in two-phase situations, as follows. (“Input” and “output” refer to the power half-arrows. The actual flow can be in either direction.)

0 no gravity vapor or liquid separation
2 vapor separation on input, none on output
3 liquid separation on input, none on output
4 no separation on input, vapor on output
5 gravity vapor separation on both
6 liquid separation on input, vapor on output
7 no separation on input, liquid on output
8 vapor separation on input, liquid on output
9 liquid separation on both input and output

** Use of the 1 precludes convec.m from asking for further details of a variable modulus.

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

Output variable = parameter · sign(input variable) · abs(input variable)$^{\text{exponent}}$.

In which the parameters are $1/C$, $1/I$ and $R$, respectively, and the input variables are displacement, momentum and flow (velocity), respectively.
different single or multiphase substances is generally accurate and efficient, requiring virtually no iteration. The matrix $b$ describes the interconnectivity of elements and bonds. It has three columns. The $j$th row describes the $j$th bond. Numbers in the first column identify the element adjacent to the causal stroke. The second column gives the number of the element at the other end of the bond. The third number is either 1, -1, 2 or -2. 1 or -1 specifies a simple bond, 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. For the new enhancement that accommodates activated bonds, the number 1 is replaced by 3 or 4, and the number -1 by -3 or -4. The magnitude 3 designates the propagation of effort and the interruption of flow; 4 designates the inverse.

3. VARIABLE VALUES OF $S_e$, $S_f$, $S_q$, $R$, $T$, $G$ AND HEAT CONDUCTION $RS$ ELEMENTS

In the standard procedure, as noted in Table 1, constant moduli of the simple-bonded elements $S_e$ for effort source, $S_f$ for flow source, $S_q$ for power source, $R$ for resistance, $T$ for transformer, $G$ for gyrator and $RS$ for conduction heat transfer are entered into the third element in the vector for that element within the matrix $el$. The new procedure also allows you to describe a parameter as variable, either as a function of time or a function of state, by entering 0 instead. When the program is run, this produces an error message along with a single line of instruction directing you how to input the desired behavior into the file $convec.m$. You may leave this coding in $convec.m$ without affecting the simulation of other models if you imbed it within an if loop that singles out the flag $FG$, which you would set at some distinguishing number in the master program for the particular system model.

For the list of elements above (only) and as noted in Table 1, the new instructions for the matrix $el$ tell you to set the second member of the vector for the particular element in question equal to one. This allows the parameter for that element to equal zero without causing problems, which may be useful under special conditions. This step is unnecessary if the parameter never equals zero.

3.1 Example of a ground-effect machine ($GEM.m$)

An example of the vertical motion of a ground-effect machine (or “Hovercraft”), taken from pages 376-378 (pages 316-319 in the first edition) of [3], is pictured in Fig. 1. There are two nonlinearities in the model: the resistance to airflow under the skirt, which is a function of the elevation $y$ of the craft as well as a nonlinear function of the pressure $P$, and the pressure-flow characteristic of the fan. The pressure-flow characteristic, also plotted in Fig. 1, is given in the book by a Thevenin equivalent combination of a fixed-effort (pressure) source and a nonlinear resistance. For present purposes it is simpler to replace this with a Norton equivalent of a fixed-flow source and a nonlinear resistance, and then since both this resistance and the resistance for the flow under the skirt would emanate from the same 0-junction, to combine these resistances into a single resistance. The resulting bond graph is also shown in the figure.

The flow is given as a function of the pressure $P$ and the elevation $y$ as follows:

$$Q = a_1 P + a_2 P^2 + a_3 P^4 + L \sqrt{2 \rho / \rho}$$

The first three terms, with constant coefficients, represent the fan characteristic; the final term, with the skirt length $L$, flow coefficient $c_d$ and air density $\rho$, represents the flow under the skirt.

![Fan Characteristic](image)

Fig. 1  Ground Effect Machine
There would be only a single energy-storage element, the inerante or mass, \( I = m \), were it not for the dependence of the flow resistance to the elevation \( y \). However, this dependence of a resistance on a displacement requires that \( y \) become the second state variable. It is generated in the bond graph by the compliance \( C \) bonded to the 1-junction by an activated bond. (The actual value given to this compliance is immaterial.) The matrices \( el \) and \( b \) become as follows:

\[
\begin{bmatrix}
7 & 0 & W \\
1 & 0 & 0 \\
13 & 1 & I \\
9 & 0 & T \\
0 & 0 & 0 \\
8 & 0 & Q_0 \\
15 & 0 & 0 \\
12 & 1 & C \\
\end{bmatrix}
\quad
\begin{bmatrix}
2 & 1 & -1 \\
3 & 2 & 1 \\
2 & 4 & 1 \\
4 & 5 & 1 \\
6 & 5 & -1 \\
5 & 7 & -1 \\
2 & 8 & -4 \\
\end{bmatrix}
\]

The master file, called \( GEM.m \), can be downloaded and run; it includes the values of the parameters. This calls the operational program \( convec.m \). The error message generated by the zero in the seventh row, third column of \( el \) prompts the user to insert coding for the fan characteristic into \( convec.m \), which the reader can find readily (note the flag setting \( FG=13 \)). This coding with its Newton-Raphson iteration is considerably more complicated than that usually required, since \( P \) is not a state variable and cannot be solved for explicitly. Note that the definition of a non-zero value of \( el(7,3) \) in the final line precludes any repeat of the error message. Since this parameter never equals zero it is unnecessary to set \( el(7,2) = 1 \), although there would be no harm in doing so.

The program generates the plot shown in Fig. 2 for the elevation \( y \) versus time, which agrees with that given in the book using a much more sophisticated approach.

4. VARIABLE VALUES OF \( C, I \) AND FRICTION \( R_S \) ELEMENTS; MORE ACTIVATED BONDS.

The new default modeling of the simple compliance and inerante elements allows for power-law relations between the effort and displacement and the momentum and flow, respectively. The user places the exponent of the relation as the second member of the vector for the element in the matrix \( el \); the coefficient is placed as the third member. The exponent is set equal to 1 when a linear relation is desired.

When you wish to apply other than the power-law relation, you place a zero (for the "exponent") in both of these locations in the master file, which produces an error message and a single-line instruction as to how to code the relation in the file \( convec.m \). Since the subsequent coding continues to assume the power-law relation, the zero in the second position (the exponent) is retained, and the third parameter becomes the ratio \( \text{sign}(q)/e \) for the compliance, \( \text{sign}(f)/p \) for the inerance and the friction force for the friction element, all in the form of functions of the state vector \( x \) and the time \( t \). The presence of a non-zero value in this position then precludes the error message from reappearing.

4.1 Example of a DC motor and load

The second example is an elaboration on the model of a DC motor and spring-mass-dashpot load given in the papers and the web instructions for the basic simulation package \([1,4,5]\). Deadband is introduced into the compliance relation, representing backlash in the gears. Also, gear losses are incorporated using activated bonds, and the dashpot resistance is made nonlinear. The original model is given in the master file \( DCMotor.m \), which uses the flag \( FG=2 \); and the elaborated model is given in the master file \( DCMotors.m \), which uses the flag \( FG=18 \). The two respective bond graphs, shown in Fig. 3, differ only by the use of the activated bonds.

The coding for the matrix \( el \) in the master program \( DCMotors.m \) has zeros in the positions 1,2 and 1,3 for the effort source, positions 8,2 and 8,3 for the
activated transformer \( T_m \), positions 9,2 and 9,3 for the compliance, and positions 11,2 and 11,3 for the resistance \( R_2 \), all of which invoke error messages accompanied by instructions when the program is run. The bond activation is dictated in the matrix \( b \) by the -4 in the third position for the fourth bond, and by the -3 in the third position for the eighth bond:

\[
\begin{bmatrix}
7 & 0 & 0 \\
1 & 0 & 0 \\
15 & 0 & R I \\
11 & 0 & G \\
1 & 0 & 0 \\
9 & 0 & T d \\
0 & 0 & 0 \\
9 & 0 & T m \\
12 & 0 & 0 \\
1 & 0 & 0 \\
15 & 0 & 0 \\
13 & 1 & 1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
2 & 1 & 1 \\
2 & 3 & -1 \\
4 & 2 & 1 \\
4 & 5 & -1 \\
5 & 6 & -4 \\
6 & 7 & -4 \\
8 & 7 & -3 \\
5 & 8 & -3 \\
7 & 9 & -1 \\
10 & 7 & 1 \\
10 & 11 & -1 \\
12 & 10 & 1 \\
\end{bmatrix}
\]

The effort source is sinusoidal in time, and is entered into \textit{convex.m} in response to the instructions that appear with the error message as

\[
\text{el}(1,3) = 5\sin(8\pi t); \text{ el}(1,2) = 1;
\]

The setting \( \text{el}(1,2) = 1 \) accommodates the zero value of the function when \( t = 0 \) without causing the error message to reappear.

The desired compliance relation with its dead band is plotted in Fig. 4a, and the desired resistance relation for \( R_2 \) in Fig. 4b.

The error message for the compliance is as follows:

Code above in the form \( \text{el}(1,3)=f(x,t) \) the ratio \( \text{sign}(q)/e \) for the C element i, where \( e \) is effort and \( q \) is displacement

This is accomplished with the coding

\[
q_0=0.015; C=.05;
\]

if \( \text{abs}(x(1))<q_0; \text{el}(9,3)=1/0; \% \text{dead band for compliance}
\]

else

\[
\text{el}(9,3)=C/\text{abs}(x(1)) \cdot q_0; \% \text{effort proportional to } x(1) \cdot q0
\]

end

The coding in response to the error message for the resistance \( R_2 \) is

\[
\text{el}(11,3)=1/((\text{abs}(x(2)))^{1/3}+1e^{-9});
\]

This makes the damping torque become proportional to the 0.7 power of the velocity. The only practical effect of the 1e-9 is to prevent the resistance from becoming infinite. Since the resistance is never zero it is unnecessary to set \( \text{el}(11,2) = 1 \).

The speed ratio of the gears is determined by the fixed ratio of teeth, and therefore the transformer modulus \( T_d \) is invariant. On the other hand, the transformer modulus \( T_m \) determines the torque loss due to friction, so as to give the specified mechanical efficiency \( \text{eta} \). The complicating factor is that the ratio of torques depends on whether the power is flowing from left to right or right to left. Therefore, the sign of the power is calculated on both the left and right sides, and \( T_m \) set accordingly. The coding is given below, and plots of the velocities of the output for the original linear model and the presently modified nonlinear model, in response to the original sinusoidal excitation, are given in Fig. 5.

Fig. 4 Characteristics of spring and resistance
Fig. 5 Velocities of the two motors

```matlab
eta=.8;
el(8,3)=el(6,3)/eta; % if power from left to right
Psigh=x(1)*x(2);
if Psigh<0; el(8,3)=eta*el(6,3); % reverse case
end
```

4.2 Example of a solenoid

The second example is a solenoid, taken from Problem 9.38 on page 677 of Ref. [3], second edition, and pictured in part (a) of Fig. 6.

![Solenoid Diagram](Image)

(a) cross section

![Solenoid Diagram](Image)

(b) model with two-port compliance

![Solenoid Diagram](Image)

(c) model with separate one-port compliances

Fig. 6 Solenoid

The modeling assumes that the only significant mmf drops in the magnetic path occur across the plastic sleeves, which are assumed to have the same magnetic permeability as free space. The flux lines are assumed to be radial. This gives the bond graph in part (b) of the figure, which represents the electromagnetic coupling as a two-port compliance. Before the new development described later below, however, only one-port compliances could be accommodated by the simulation package. In this first treatment of the system, therefore, the compliance is broken up into two compliances, as shown in the second bond graph, and the couplings between them are analyzed off-line. This treatment is more efficient computationally than the multiport-method presented subsequently, but requires more effort and acumen from the user and allows more opportunity for mistakes. The master program using separate compliances is `solenoid.m`; that for the combined compliance is `solenoids.m`. For the former, the coding for matrices `el` and `b` become

\[
\begin{bmatrix}
7 & 0 & e1 \\
1 & 0 & 0 \\
15 & 0 & R \\
11 & 0 & n \\
12 & 1 & 0 \\
12 & 0 & 0 \\
12 & 1 & Ck \\
13 & 1 & I \\
17 & 0 & 0 \\
12 & 1 & CT
\end{bmatrix} ; \quad
\begin{bmatrix}
2 & 1 & 1 \\
2 & 3 & -1 \\
4 & 2 & 1 \\
4 & 5 & -1 \\
7 & 6 & -1 \\
7 & 8 & -1 \\
9 & 7 & 1 \\
7 & 10 & -1 \\
10 & 11 & -1
\end{bmatrix} .
\]

while for the latter it becomes

\[
\begin{bmatrix}
7 & 0 & e1 \\
1 & 0 & 0 \\
15 & 0 & R \\
11 & 0 & n \\
19 & 0 & 2 \\
1 & 0 & 0 \\
12 & 1 & Ck \\
13 & 1 & I \\
17 & 0 & 0 \\
12 & 1 & CT
\end{bmatrix} ; \quad
\begin{bmatrix}
2 & 1 & 1 \\
2 & 3 & -1 \\
4 & 2 & 1 \\
4 & 5 & -1 \\
6 & 5 & -1 \\
6 & 7 & -1 \\
8 & 6 & 1 \\
6 & 9 & -1 \\
9 & 10 & -1
\end{bmatrix} .
\]
Note that the numbering for the bonds on the compliances must be in the same order as the numbering of the compliances themselves.

In either of the two approaches, one starts with the potential energy in the solenoid, which is the energy in the radial gap:

\[ \text{PotE} = \frac{1}{2} \phi^2; \quad C = \frac{1}{c_1(1/y+1/z)}; \quad c_1 = \frac{g}{\mu_0 \pi d} \]

The first state variable for both models is \( x(1) = \phi \), the magnetic flux, the second state variable \( x(2) = y \) is the mechanical displacement of the plunger, the third state variable \( x(3) \) is the displacement of the spring, the fourth state variable \( x(4) \) is the entropy of the thermal reservoir and the fifth is the momentum of the plunger. (The mechanical displacements of the spring and the plunger differ only by a constant.)

The magnetic permeability is given by \( \mu_0 \), and \( d \) and \( z \) are lengths defined in the drawing, and \( g \) is the thickness of the sleeve.

The method with separate one-port compliances requires you to compute the force \( F \) and the mmf \( M \) by hand:

\[ M = \frac{\phi}{C} \frac{\partial \text{PotE}}{\partial \phi}; \quad F = \frac{\phi}{C} \frac{\partial \text{PotE}}{\partial y} = \frac{c_1}{2} \phi^2. \]

It then requires that you insert these terms at the designated place in \( \text{convec.m} \), making sure the signs are correct, as the error message directs. The coding is

```plaintext
g=.001; mu0=4*pi*1e-7; d=.01; z=.015; k0=1000; c1=g/mu0/pi/d; el(5,3)=1/c1/(1/x(2)+1/z); % note this is set up in solenoid.m for the compliance to be linear in x(1) (but not in x(4)).

% The compliance relation uses its required nonlinear form:
el(6,3)=sign(x(2))/(c1*x(1)^2/2*x(2)^2);

The model also assumes a load very close to coulomb friction plus some stiction, represented by a friction \( RS \) element with special characteristics as plotted in Fig. 7.

Note that the actual velocities range over 0.5 m/s, considerably off the horizontal scale of the plot.

The coding for this friction for the case with a 2-port compliance is given below. The coding for the case with separate 1-port compliances is the same except for the indices.

```plaintext
vmag=abs(x(5)/el(8,3));
if vmag<.001272; el(9,3)=1000*vmag;
if vmag==0; el(9,3)=1e-6; end % to prevent error message from appearing
else; el(9,3)=abs(1+exp(-1200*vmag));
end
```

The resulting displacement of the plunger following a step application of a 12 volt source is plotted in Fig. 8.

5. MULTIPORT COMPLIANCES

The newly introduced multiport compliance element \( C_f \) has two or more bonds; even if you have more than one such compliance, you must consider them as a single multiport element, which does not in fact restrict the modeling. When you run the master program you get an error message within \( \text{convec.m} \).

You will find detailed instruction at the indicated line number, including examples, telling you how to proceed. In essence, all you do is enter an expression for the potential energy of the multiport compliance, as a function of the state variables. The program
takes over from there, using the Symbolic Math Toolbox of MATLAB® to find the respective efforts, first in symbolic form and then numerically. You do not have to worry about analytical or sign mistakes, or whether the element is linear or nonlinear. Any dependencies on remote displacements also are treated automatically.

All multiport compliances are lumped together in a single $C_f$ element of however many ports are needed. The automatic ordering of state variables, as described in the program `convec templates.m`, starts with those of the thermodynamic CS elements, as usual, then proceeds on to those of the $C_f$ element, followed by other simple-bonded $C$ elements, etc. You are well advised to label these on the bond graph for the system. In the coding for the potential energy, the scalar variables $x_1=x(1)$, $x_2=x(2)$, $x_3=x(3)$, etc., are used.

1-port compliances also may be included within the $C_f$ element if you wish to take advantage of the automatic differentiation feature for them also.

The coding for the two-port compliance, which is placed at the designated location in `convec.m`, is very simple:

```matlab
g=.001; mu0=4*pi*1e-7; d=.01; z=.015; k0=1000;  
x0=.005;  
c1=g/mu0/pi/d;  
PotE=c1/2*(1/x2+1/z)*x1^2;
```

Note that the state variables for this compliance must be $x(1)$ and $x(2)$. The results are identical to those of version 1, but execution is slower.

### 5.1 Example of an electromechanical balance

The electromechanical balance pictured in Fig. 9 is taken from Problem 9.31 (p. 646) of [3]. A solenoid pulls the balance one way, and a variable-plate capacitor pulls it the other way; both elements share a common electrical current. Balance is achieved when the frequency of the sinusoidal excitation is set at a particular value that depends on the various parameters.

In practice, one knows all the parameters with some accuracy except for the magnetic flux and its induced mechanical force. The instrument can be used to measure this property (the parameter $\alpha$). The capacitor comprises two large parallel flat plates separated by a small gap $x(4) = d + x$. The bond graph treats both two-port compliances together as a single four-port compliance. The $x$ in the figure is $x(5)$ in the bond graph; its value is zero when the spring is relaxed. The coil or solenoid is taken to be the same as in the example of the solenoid above.

For the device to operate properly, the electrical current needs to be virtually sinusoidal, but use of a current source would produce differential causality for one of the bonds on the compliance field. The desired result is accomplished instead by modeling the source as a sinusoidal voltage (effort) source of very large amplitude combined with a series resistance of very large magnitude. The same solenoid as in the examples above is employed, except that there are more turns on the coil and the friction in the sleeve is replaced by a linear dashpot, which is still modeled by the friction RS element, however, so heat generated by friction is still monitored.

The capacitance is virtually $C = \varepsilon C_A x(4)$, where $\varepsilon$ is the dielectric constant. Thus, the total potential energy of the solenoid and the capacitor is

$$PotE = \frac{c_1}{2} \left( \frac{1}{y} + \frac{1}{z} \right) \phi^2 + \frac{x + x_0}{2A\varepsilon} q^2.$$  

The system is excited with three sinusoidal excitations with frequencies of 90%, 100% and 110% of the theoretical balance frequency, with the balance beam started in balance. The resulting displacements of the beam from its balanced position are plotted in Fig. 10.

An enormous number of cycles are computed, so data compression is used and execution requires considerable time.
6. ELIMINATION OF REQUIREMENT FOR NON-ZERO INITIAL CONDITIONS

Under special circumstances, primarily compliances bonded to 1-junctions and inertances bonded to 0-junctions, it used to be necessary to avoid identically zero initial conditions. This limitation no longer applies as a result of a fundamental change in the coding which is otherwise transparent to the user.

7. CONCLUSIONS AND RECOMMENDATIONS

The Simulation Package for Thermodynamic Systems has been extended to make it attractive for simulating non-thermodynamic systems as well. In particular, incorporating nonlinearities and activated bonds in the non-thermodynamic components has been expedited, and a special feature in which compliances can be described by the energy stored therein rather than their sometimes more difficult constitutive relations has been added. An earlier complication regarding certain zero values of initial conditions has been overcome.

Achieving a desirable bridge to a graphical interface rather than the use of defining matrices, and more generally producing some conversion of convection bond graphs to conventional bond graphs, have thus far been elusive because of the inherent relative sophistication of the convection graphs. The former is certainly possible, however. The latter is doubtful.

The author believes that for bond graphs to achieve their potential they must accommodate thermodynamic systems with their compressible fluid flow, phase change and heat transfer, and convection bond graphs appear to be the only practical way to do this. The fundamental reason is the need to account for the added independent variable, entropy. Approaching the end of his professional career, he hopes others will take up the mantle, as much more needs to be done.

8. REFERENCES

1. Brown, F.T., 2008, website www.lehigh.edu/~inmem; click on faculty, emeritus, and F. Brown

9. BIOGRAPHY

Forbes T. Brown retired from active teaching at Lehigh University in 2004, but continued his work on bond graphs including completing the second edition of his textbook and writing papers for the ICBGM.

His advisor for his doctoral thesis at M.I.T. was Henry M. Paynter, the creator of bond graphs. It was the first university project in what became know as fluidics. Dr. Brown served on the faculty there from 1958 to 1970, when he moved to Lehigh University at the rank of Professor. He taught bond graphs at both schools, as well as design, control, system modeling, mechanics, dynamics, graphics, fluid mechanics, hydraulics and thermodynamics. His research focused on fluidics, fluid line dynamics, hydraulics, dynamic system modeling including basic energy-based methods, simulation and control, as well as on bond graphs.

At present he is deeply engaged in a adapting his simulation package to hydraulic systems. This will also likely lead to a new release of a more robust and efficient version of the primary program in his simulation package for multiphase thermodynamic systems, convec.m.