On the Rationale of the Transformity Method

Dennis Collins
Dept. of Mathematics, Box 9018
U. de Puerto Rico, Mayaguez
Mayaguez, Puerto Rico 00681

ABSTRACT

In terms of transformities, this paper presents some over-simplified or "inorganic" differential-equation models for ecological networks that nonetheless have some desirable mathematical properties. It is hoped that these models may provide a platform or springboard for creating more realistic or "organic" models based on transformities.

INTRODUCTION

From networks, the transformities of H.T. Odum or quality-equivalent mechanisms (QEM's) of M. Patterson can be calculated. These represent essential qualities or "trophic levels" of the system. Also, systems of differential or difference equations can be developed to describe the system. A kind of inverse problem is whether or not the system of differential equations can be written completely in terms of the transformities. For example, a "normalized" Lotka-Volterra predator-prey system with equilibrium point (*herbivores, carnivores*) = $(x, y) = (8, 1)$ might be written:

$$x' = x(1 - y)$$
$$y' = y(-8 + x).$$

Here the fact that eight herbivores units support one carnivore unit might roughly indicate a transformity of 8 to 1 for carnivores versus herbivores. The transformity "8" then represents the essential parameter of the differential equations. Also, the point (8, 1) is an equilibrium point of the system, since $x' = 0$ and $y' = 0$ at $x = 8, y = 1$. This paper presents an "inorganic" solution, that is, a solution that has only linear slope functions for the differential equations system, to the above problem. This model--which may be of primarily mathematical interest--roughly corresponds to a system of tanks which more-or-less gradually fill up to the required equilibrium values. A more realistic solution would require non-linear slope functions as the Lotka-Volterra model above that contains the risk of death ($(x, y) = (0, 0)$ *equilibrium*) as well as life. The model obtained in this paper for the predator-prey system would be the following:

$$x' = 8 - x$$
$$y' = 1/8\, x - y,$$

which also has equilibrium point (1, 8).

Then this paper will deal with a limited version of the emergy-transformity ecological models of H.T. Odum. The models here are digraphs (or directed graphs) with two types of nodes (or vertices): 1) Production nodes (here rectangles but in Odum's work, rectangles rounded in the direction of the output) combine various commodities or food to produce one or more output commodities or life forms. 2) Commodity-storage nodes (here circles but tank-shaped in Odum's work) receive branches from processes that produce the given commodity and branch out to production nodes or processes that use the commodity as input. The digraph is bipartite in the sense that branches from production nodes can only connect to storage nodes, and branches from commodity nodes can only connect to production nodes. This method agrees with Patterson's later work. Roughly, the storage nodes are capacitive and the production nodes are dissipative or resistive.

The discussion will cover three other types of model besides the digraph model (1.) discussed above: 2. Transformity models to calculate transformity; 3. Flow models to calculate steady-state flows; 4. Dynamic or differential equation models involving the storage nodes. There is a limited duality between the models 2. and 3. developed by Mikulecky and other previous writers. The amount stored in the tank nodes is not important in the steady-state theory, i.e., until the dynamic model 4.

The theory will be worked out for four digraph models (A, B, C, and D in Figure 1). The models will illustrate the "battle for uniqueness" within the approaches.

THE TRANSFORMITY MODEL

"Emergy" or "energy memory" measures energy required to produce something. Thus, emergy is never forgotten in the digraph flow process, unlike energy, which, although it is conserved, is mostly wasted in production processes, so that the useful energy flow is constantly decreasing from input to output of a production process. Only useful energy flow is accounted for in these models. Thus, all nodes satisfy "emergy in = emergy out," and storage nodes also satisfy (in this approximation) "energy in = energy out," but the production nodes satisfy "useful energy in $\geq$ useful energy out" due to dissipative losses of the second law of thermodynamics.

Every branch in the steady state process thus has an emergy flow and an energy flow assigned to it. In the diagrams (Figure 1, A to D), the emergy flow is put above the branch and the energy flow below the branch. The ratio of the emergy flow divided by the energy flow is called the transformity. It is assumed that the transformity of every branch from a given storage node is the same, since it reflects the flow of the same commodity. Thus, this identical transformity can be assigned to the storage node from which the commodity flows and also to the commodity itself. Knowing the transformities of the commodities plus the energy flow into each process allows the calculation of the emergy into each process or production node by summing "transformity x energy flow" (= emergy flow) over the branches entering a given production node.

The transformity model is simply the system of linear equations stating "emergy in = emergy out" for each production node in terms of the transformity of each commodity involved, i.e., each emergy term is represented by a product (called a constitutive relation) "emergy flow of given commodity = transformity energy flow." Thus, the number of equations is equal to the number of production nodes, and the number of variables is equal to the number of storage nodes. Transposing all terms to the left results in equations in terms of a commodity energy flow matrix J and a transformity column vector T:

$$J\,T = O$$

where the 0 is the column of zeros.

A BASIC program to solve for the transformities by an eigenvalue method was developed by the author and presented at a colloquium of H.T. Odum on July 2, 1997. A similar method is included in the 1998 paper of Patterson. A *Mathematica* version (Table 1) is used to approximate the transformities in the above programs (AT-DT). See Collins and Odum (2001) for explanation and applications to ecosystems. Further applications were reported in this volume (Odum and Colliins, 2002).

The matrices for the four examples are given in the programs AT, BT, CT, DT (Table 2) to solve for transformity.

If there is more than one product of the process node (by-product case), then additional equations could be added describing the partition of emergy coming out of the production node. This case occurs in example D in Figure 1.

THE FLOW MODEL

This paper is concerned with the inverse problem of calculating the steady-state flows, knowing the transformities. There is a problem because there are many more branches than nodes, so that there are more branch flows to calculate than there are node transformities to calculate them. As a consequence, the *m* transformity equations 1] must be "expanded" to cover the equation "emergy in = emergy out" as an equation over all branches. Of course, the flow and transformity variables have been interchanged in a kind of duality. Further, to specify the branch flows, two more sets of equations must be added: 2] commodity node equations stating "energy in = energy out" for each commodity node, and 3] "recipe" or proportion equations stating the proportion of energy required for each production node. These equations have the form in terms of a transformity matrix T_F and flow vector J_F:

$$T_F \cdot J_F = 0$$

If there are k branches, the above equations will uniquely solve for the energy flow in each branch. These equations are presented and solved in the programs AF, BF, CF, and DF (Table 3) by the same program. However, as in the Gauss elimination method or its unique row-reduced echelon form generalization, the flows not included in the original transformity equations can gradually be expressed in terms of the distinct process-to-dominant-commodity flows and eliminated from the flow equations (cf. pp. 134-145 in Mikulecky). Keeping track of the relations, all flows can be derived from the original transformity equations, written in terms of the reduced set of flow variables, and with the transformities determining the coefficient matrix (for the most part, except proportionality constants).

THE DYNAMIC MODEL

Now the equilibrium flows satisfy the J T = 0 of the Transformity Model section above, so that *if the flow equations are changed into differential equations for the rate of change of flow in the commodity nodes, then an equilibrium point of these equations will be the steady state flows.*

This creates--albeit artificially--a dynamic system which has as its equilibrium limit the steady-state flows. The creation of such systems is a stated goal of the network thermodynamics of Mikulecky (cf. pp. 14 and 78). Although the method described above is still what the network thermodynamics wishes to avoid, because it merely changes the equilibrium status into a system of differential equations, nonetheless it may give some indication of how to set up a correct dynamical system for the flows involved. The method differs from the electrical models in that the transformity typically goes up as one follows the emergy "current," whereas the voltage potential goes down. It still holds, however, that the sum of transformity changes around a closed loop is zero, similar to the Kirchhoff voltage law.

The method above appears to differ somewhat from Odum's version in that he regards emergy as being used up when a commodity follows a loop back to itself (cf. p. 89 in Odum's book), whereas here the "emergy in = emergy out" always holds, although the emergy may be diluted so as to approximate Odum's approach. For example, if energy is spent on water to purify it, and later the water is dumped into an untreated water storage, in this theory the water of the storage would increase its average emergy level somewhat (perhaps a small epsilon) above its previous value. Of interest is how other Odum symbols (such as the consumption symbol) should be related to emergy as treated here. Observe that since "emergy in = emergy out" in the steady-state model, emergy can cycle without building up. It may be that one can ignore the cycles and leave them out of the analysis as I believe Odum claims, but, as in the case of electrical systems where steady state current cycles driving appliances are a major interest, this procedure may also leave out the most important topics of analysis.

To test most dynamic differential equation properties, it is necessary to have an n x n system. Such a system can be obtained by gradually eliminating branch flows, as explained above. However, there is still some doubt as to which variables to include, and results can change somewhat, depending on these decisions.

Examples of systems of differential equations for the four systems A to D are given in Figure 2 (AE to DE). After dividing by the transformity (according to the "constituitive relation"), the coefficient in each equation of the variable differentiated is -1, which goes toward Odum's purpose of making the dynamics comparable regardless of energy scale.

Ordinarily it takes an entire matrix R to determine a system of linear differential equations, say $dz/dt = Rz$. Here the matrix R is determined by only the one vector of transformities T, plus perhaps some other constants involved in reducing the flow system to an n x n system. In this sense the vector T is like the infinitesimal generator of a continuous group representation, which can depend on only as many parameters as the group (say 3), even though the matrix has many more components (say 3 x 3 = 9). The paper by Terrell gives conditions under which a matrix can be considered to be generated by the one vector of coefficients of its characteristic polynomial. The systems of differential equations can be tested for different properties of control theory, such as complete controllability. The property of complete controllability depends on whether or not the vectors $\{b, Rb, R^2, R^3b, \ldots, R^{n-1}b\}$ are linearly independent, where b is the input vector. Complete controllability apparently implies the storage node flows can be driven to arbitrary amounts. The first example is completely controllable using the input

$$30 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and matrix } \begin{bmatrix} -2 & 0 \\ 2 & -3.75 \end{bmatrix} \text{ or } \begin{bmatrix} -1 & 0 \\ 2/3.75 & -1 \end{bmatrix}$$

The dynamic systems set up in AE to DE, which may be written

$$dz/dt = f = Az + b$$

can be easily solved numerically by programs such as *Mathematica* and various properties calculated, such as complete controllability.

STABILITY

The stability of linear systems of differential equations of the above type is discussed in Example A8, p. 27 (Higashi and Burns, 1991). Such a system is asymptotically stable if the real parts of the eigenvalues of the system matrix are all negative.

CONJECTURE: Any non-trivial system derived as above is asymptotically stable.

The reason is that the differential equations describe the energy flows z_i, which are always decreasing by the second law of thermodynamics. The eigenvalues are calculated for a couple of the cases (BE real parts -1.38, -1.38, -.239, and CE real parts -1.39, -1.39, -1, -1, -1, -.206), and seen to have negative real parts. An example output run of the dynamical system ("inverse method") for case B is given in Figure 3.

SENSITIVITY

The systems of differential equations can be rewritten in terms of the transformities (Figure 4). Now the transformities play the role of parameters, and it is possible to study the sensitivity of the steady-state flow vector (called x* in Higashi and Burns, pp. 27-28) as a function of these parameters. The equation expressing the sensitivity is given as

$$\frac{\partial x^*}{\partial p_k} = -A^{-1} \frac{\partial f}{\partial p_k}$$

In the notation above, x is replaced by z, and p by T.

Here the vector f represents the vector of differential equations.

For example in case BE, the sensitivity of steady-state value z* with respect to T_1 is calculated as follows:

$$\frac{\partial x^*}{\partial T} = A^{-1}\frac{\partial f}{\partial T} = -\begin{bmatrix} -1 & 5 & 0 \\ 1/20 & -1 & 1/5 \\ 1/4 & 0 & -1 \end{bmatrix}^{-1} \begin{bmatrix} 100(-1/T_1{}^2) + \frac{T_2 z_2}{2}(-1/T_1{}^2) \\ \frac{z_1}{2T_2} \\ \frac{z_1}{2T_3} \end{bmatrix}\Bigg|_{(z^*,T)}$$

$$= -\begin{bmatrix} -2 & -10 & -2 \\ -1/5 & -2 & -2/5 \\ -1/2 & -5/2 & -3/2 \end{bmatrix}\begin{bmatrix} -2 \\ 1/10 \\ 1/2 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ 0 \end{bmatrix}. \text{ Here } z^* = \begin{bmatrix} 20 \\ 2 \\ 5 \end{bmatrix} \text{ and } T = \begin{bmatrix} 10 \\ 100 \\ 20 \end{bmatrix}$$

DISCUSSION

There are several versions of how to compute transformity at present, including that discussed in Prof. Odum's book and that (the QEM) of Prof. Patterson. It is important to understand how these methods differ, since the success of transformity in applications will probably depend on whether or not different researchers can agree on values obtained. Fortunately the method is robust to the extent that many different approaches may give approximately the same result.

For example, Patterson's method seems to differ from Odum's in that Patterson does not require a "commodity unification node" to combine commodities produced by different processes, and therefore with different transformities. The consequence is a certain approximation to the results obtained in this paper, as illustrated by the two versions of Example C in Figure 1.

Another example of a difference between Odum's method and Patterson's is in the treatment of by-products. Prof. Odum's method, although consistent, has the mathematical problem that the emergy function is discontinuous when emergy disappears after completing a feedback loop. One way of handling this problem (Patterson's criticism) is to eliminate the feedback loops and work only with feedforward paths. If feedback loops include only the emergy of the source, they can be omitted in Odum's procedure to avoid double counting. The emergy of a closed loop is regarded as constant. However, where feedback energy has an independent source, some feedback of emergy is allowed without instability, and increases the transformity as expected. Also (part of Odum's theory), some increase of emergy at a node due to co-products (which may be termed "synergy") is allowed without instability, although this increase would violate the "emergy in = emergy out" rule of this study. Mathematically, the matrices $a = m^t{*}m$, which appear in the calculation of transformities, represent a wider class beyond "singular M-matrices" that have a positive eigenvector (Berman and Plemmons, 1994). A mathematical study up to 3 x 3 matrices is available as an appendix to this paper but is not included here. A condition ($1 - su - pv - qt - svq - upt > 0$, where u, v and s, t and p, q represent the feedback proportions from the three processes to the other two processes) for stability is given in this appendix. Some "synergy" (say $u = 1$ and $v = 1$, so that $u + v = 2 > 1$) is allowed provided some of the other feedback coefficients are small.

/

In general, the approach here is that energy flow recipes into a process must be prescribed as part of the definition of the process (as in a cookbook), and emergy flows of by-products must be estimated.

The question of what to do in the case of default of information is partly handled by Odum, and also may be considered part of Patterson's method. The author's view of the above approach may change on consideration of more complicated examples.

CONCLUSION

Transformities may have a value in describing the differential equations of a network as well as indicating rough "trophic levels," or levels of complexity. The problem of working out this relationship for non-linear systems such as the Lotka-Volterra system remains open; however this paper presents certain approaches which may be applied to the general problem. Roughly it is believed that the nonlinear systems can be studied by the process of Hopf bifurcation (Borrelli and Coleman, 1998). Although the models of this paper are stable, the stability of the linearization is not necessary for stable limit cycles to exist in supercritical Hopf bifurcations. This extends that fact that logistic growth defined by a nonlinear slope function $x' = x (a - bx))$ is limited by the saturation value $x = a/b$ even though the linearized model $x' = ax$ shows exponential or unstable growth. These considerations may be relevant to Hannon's unstable material flow models (Hannon, 1986), that is, if the correct nonlinear differential equations were obtained, the instability of the linear versions would not be a problem. Although the over-simplified models of this paper may be mainly of mathematical interest, unless the linear models are studied for their deficiencies, the nonlinear theory remains on an unstable footing.

Finally, the author would like to thank Prof. H.T. Odum for his many discussions and work on this paper. His awesome ability to point the way to solutions of ecological problems will be almost impossible to replace should he not recover from his present illness. Also, should this paper be included in the proceedings of the 2001 conference, it will be due to the tireless efforts of Prof. Mark Brown and Ms. Joan Breeze.

REFERENCES

Berman, A. and R.J. Plemmons. 1994. Non-negative Matrices in the Mathematical Sciences. Siam, Philadelphia, Pennsylvania.

Borelli, R.L. and C.S. Coleman. 1998. A Modeling Perspective. John Wiley, New York, pp. 490-494.

Collins, D. and H.T. Odum. 2001. Calculating transformities with an eigenvector method. pp. 265-280 in Emergy Synthesis, ed. by M.T. Brown. Center for Environmental Policy, Univ. of Florida, Gainesville.

Hannon, B. 1986. Ecosystem control theory. J. Theor. Biol. (1986) 121, 417-437.

Higashi, E.M. and T.P. Burns. 1991. pp. 26-27 in Theoretical Studies of Ecosystems: The Network Perspective. Cambridge U. Press, Cambridge, U.K.

Mikulecky, D.C. 1991. Network thermodynamics: A unifying approach to dynamic nonlinear living systems. pp. 71-99 in Theoretical Studies of Ecosystems: The Network Perspective, by E.M. Higashi and T.P. Burns. Cambridge U. Press, Cambridge, U.K.

Odum, H.T. 1996. Environmental Accounting. John Wiley, New York, 370 pp.

Odum, H.T. and D. Collins. 2002. More transformities for ecosystem energy webs with the eigenvalues method. Proc. of Emergy conference of 2001. Center for Environmental Policy, Univ. of Florida, Gainesville.

Patterson, M. 1998. Understanding energy quality in ecological and economic systems: a brief explanation of the QEM. pp. 257-274 in Advances in Energy Studies, Porto Venere, Italy, 26/30 May, 1998. Musis, Rome, Italy.

Terrell, W.J. 1999. Some fundamental control theory I: controllability, observability, and duality. American Mathematical Monthly 106(8):705-719.

LEGENDS FOR THE FIGURES

Figure 1. Four energy networks in which emergy flow is written above the pathways and energy flows below. The ratios are transformities by definition. A, pipeline topology; B, system with a split and feedback; C, web with commodity unification; D, web with co-product branch.

Figure 2. Differential equations for the four systems in Figure 1, where z_i is the flow from process i to storage i.

Figure 3. *Mathematica* output plots over time of the equations derived for system B of Figure 1. (a) z_1; (b) z_2; (c) z_3.

Figure 4. Differential equations for the four systems in Figure 1 in "normal form."

Table 1. Template for Evaluating Transformities with *Mathematica* with Values of Energy Flows of Transformation Processes Entered in the Matrix m.

```
m = {{20000000.0, -100, 0, 0, 0, 0, 0, 0},
   {0, 100, -10, 0, 0, 0, 0, 0},
   {0, 0, 8, -1, 0, 0, 0, 0},
   {0, 0, 2, 0, -0.2, 0, 0, 0},
   {0, 0, 0, 0, 0.2, 0.1, -0.01, 0},
   {0, 0, 0, 1, 0, -0.1, 0, 0},
   {0, 0, 8, 0, 0, 0, 0, -.02}};

MatrixForm[m]
a = Transpose[m].m;
p = Eigenvectors[a];
err = m.Transpose[p];
    MatrixForm[err];
MatrixForm[Transpose[p]];
Eigenvalues[a]
u = Min[Abs[Take[Eigenvectors[a], -1]]];
t = (1/u) Take[Eigenvectors[a], -1];
MatrixForm[Transpose[t]]
```

Table 2. Transformities Calculated by the *Mathematica* Program in Table 1 for the Four Systems in Figure 1

System	Matrix J*	Error	Transformities Vector T
A	[30 -15 0	1181, 232, -2.8E-15	x1 = 1
	0 15 -8]		x2 = 2
			x3 = 3.75
B	[100 -20 1 0 0	10408, 194, 52, 1.95,	x1 = 1
	0 10 -2 5 0	-3.1E-15	x2 = 10
	0 10 0 -5 0		x3 = 100
	0 0 1 0-1]		x4 = 20
			x5 = 100
			x1 = 1
C	[100 -5 0 0 0 0 0	30059, 281, 177, 42,	x2 = 20
	100 0 -5 0 0 0 0	28, 1.36, -4 E-15	x3 = 20
	100 0 0 -10 0 0 5		x4 = 40
	0 5 0 0 -1 0 0		x5 = 100
	0 0 5 10 0 -9 0		x6 = 55.6
	0 0 0 0 1 9 -10]		x7= 60
Pattersons Method (omit "Commodity Unification" in Fig. 1)			
C	[100 -5 0 0 0	30059, 281, 177, 42,	x1 = 1
	100 0 -5 0 0	28, 1.36, -4 E-15	x2 = 15.8
	100 0 0 -10 5		x3 = 22.1
	0 5 0 0 -1		x4 = 39.8
	0 0 5 10 -9]		x5 = 57
D	[100 -20 0 0 0	21747, 3642, 956,	x1 = 1
	100 0 -50 0 0	504, -107 E-14	x2 = 5
	0 20 50 -5 -10		x3 = 2
	0 0 0 5 -30]		x4 = 30
			x5 = 5

* Energy flows under transformity columns: x1, x2, x3,etc

Table 3. Energy Flows Calculated by the *Mathematica* Program in Table 1 for the Four Systems in Figure 1

System	Matrix*	Error	Transformities Vector	Energy Flow Vector
A	[30 -15 0	1181, 232, -2.8 E-15	x1 = 1	30
	0 15 -8] = J		x2 = 2 = T	z1* = 15
			x3 = 3.75	z2* = 8
Alternate Method				
A	[1 -2 0 0	914, 19, 3.9, -4.4 E-16	x1 = 1	30
	0 2 -3.75 0 = T_F		x2 = 2	z1* = 15 = J_F
	0 0 3.75 -30]		x3 = 3.75	z2* = 8
			x4 = 30	1
B	[1 -10 100 0 0 0 0 0	20050, 10500, 500, 52,	x1 = 1	100
	0 0 0 10 0 0 20 -100	5.5, 1.6, 1.39, -1.33 E-13	x2 = 10	z1* = 20
	0 0 0 0 10 -20 0 0			1
	0 0 0 1 -1 0 0 0 = T_F			10 = J_F
	1 0 -100 0 0 0 0 0			10
	0 0 0 1 0 0 -2 0		x3 = 100	z2* = 2
	0 1 0 -1 -1 0 0 0]		x4 = 20	5
			x5 = 100	z3* = 5
C	[1 0 0 -20 0 0 0 0 0 0 0	24415, 7107,		100
	0 1 0 0 -20 0 0 0 0 0 0	4556, 1813, 376,		100
	0 0 1 0 0 -40 0 0 0 0 0	331, 67, 2.98,		99.8
	0 0 0 20 0 0 -100 0 0 0 0	1.0, 0.99, 0.0011		z1* = 5
	0 0 0 0 20 40 0 -55.6 0 0 0			z2* = 5
	0 0 0 0 0 0 100 55.5 -60 0 0			z3* = 9.97 = J_F
	0 0 0 0 0 0 1 1 -1 0 0 = T_F			z4* = +1
	0 0 0 0 0 0 0 0 1 -1 -1			z5* = 8.97
	0 0 0 0 1 1 0 -1 0 0 0			z6* = 9.98
	0 0 1 0 0 0 0 0 0 -20 0			5.0
	0 0 0 0 0 0 9 -1 0 0 0			5.0

Table 3 Energy Flows (continued)

System	Matrix*											Error	Transformities Vector	Energy Flow Vector
	1	-1	0	0	0	0	0	0	0	0	0			
	0	1	-1	0	0	0	0	0	0	0	0]			

System	Matrix T_F*										Error	Transformities	Energy Flow J_F
Patterson's method													
C	[1	0	0	-15.8	0	0	0	0	0	0	6862, 3795, 3662, 441,		100 -> 58.8
	0	1	0	0	-22.1	0	0	0	0	0	238, 91 3.0, 1.01, 1.00		100 -> 59.2
	0	0	1	0	0	-39.8	0	0	+57.6	0	0.0027		99.8 -> 59.6
	0	0	0	15.8	0	0	-57.6	0	0	0			z1* = 5 -> 3.68
	0	0	0	0	22.1	+39.8	0	-57.6	0	0			z2* = 5 -> 2.68
	0	0	0	0	0	0	1	1	-1	-1			z3* = 9.97 -> 5.82
	0	0	0	0	1	1	0	-1	0	0			z4* = +1 -> 1.0
	0	0	1	0	0	0	0	0	-20	0			z5* = 8.97 -> 5.05
	0	0	0	0	0	0	9	-1	0	0		Node eliminated	9.98
	1	-1	0	0	0	0	0	0	0	0			5.0 -> 2.98
	0	1	-1	0	0	0	0	0	0	0			5.0 -> 3.07
Recipe for splitting study													
D	[1	0	-5	0	0	0					957, 32, 5.8, 2.1,		z1* = 20
	0	1	0	-5	0	0					1.7, 6.7 E-15		z2* = 50
	0	0	5	2	-30	-5							z3* = 4 = J_F
	1	-1	0	0	0	0							z4 *= 10
	0	0	2.5	-1	0	0							z5* = 1
	0	0	0	0	2	-1]							z6* = 2

* Energy flows arranged under transformity columns: x1, x2, x3, x4, etc.

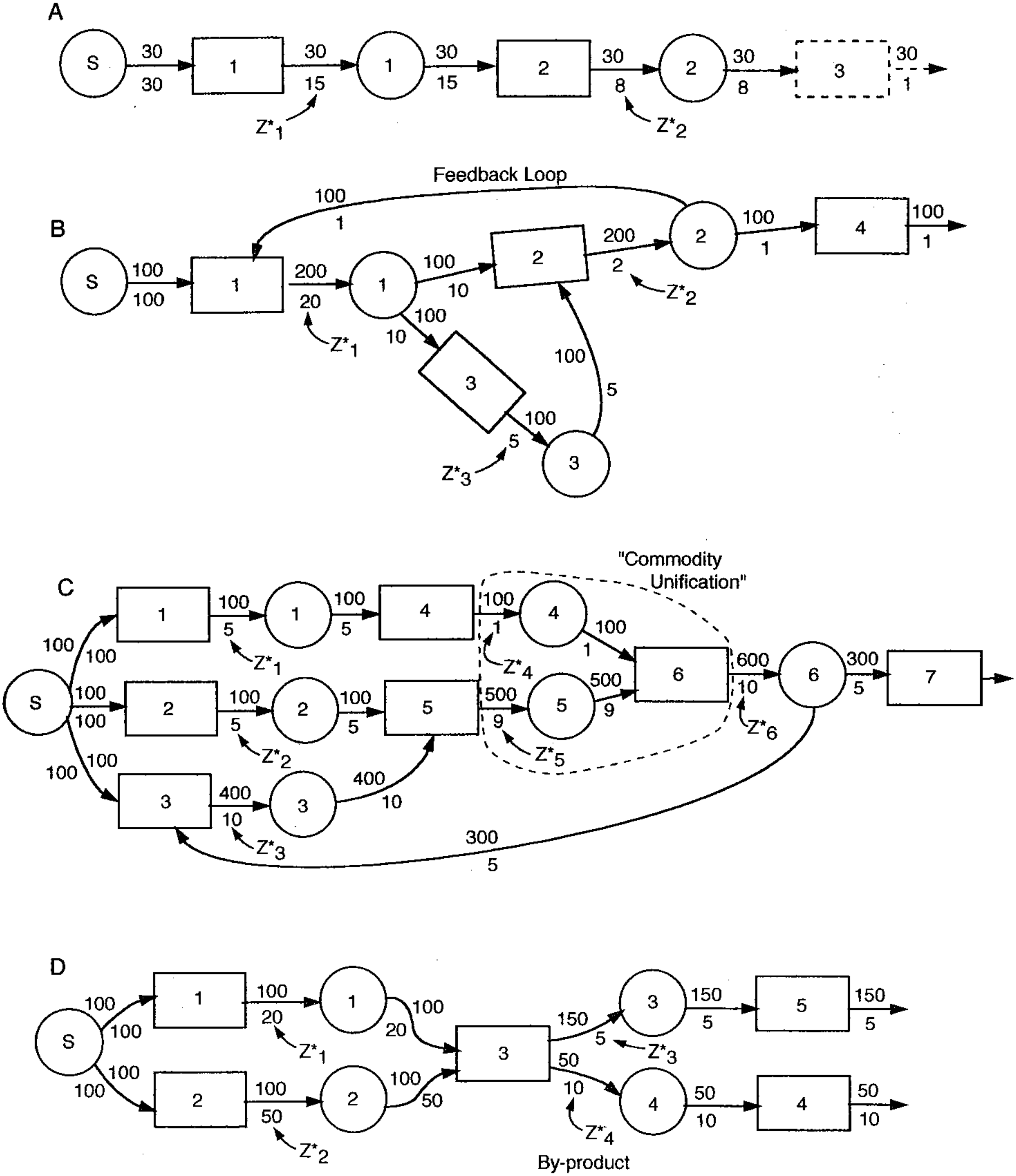

Figure 1. Four energy networks in which emergy flow is written above the pathways and energy flows below. The ratios are transformities by definition. A, pipeline topology; B, system with a split and feedback; C, web with commodity unification; D, web with co-product branch.

$$AE: \begin{bmatrix} 2\ z_1' \\ 3.75\, z_2' \end{bmatrix} = \begin{bmatrix} 30 - 2\, z_1 & \\ 2z_1 & -3.75\, z_2 \end{bmatrix}$$

$$BE: \begin{bmatrix} 10\ z_1' \\ 100\, z_2' \\ 20\, z_3' \end{bmatrix} = \begin{bmatrix} 100 - 10\, z_1 & +100(\frac{1}{2} z_2) & \\ 10(\frac{1}{2} z_1) & -100\, z_2 & +20\, z_3 \\ 10(\frac{1}{2} z_1) & & -20\, z_3 \end{bmatrix}$$

$CE:$

$$\begin{bmatrix} 20\, z_1' \\ 20\, z_2' \\ 40\, z_3' \\ 100\, z_4' \\ 55.5\, z_5' \\ 60\, z_6' \end{bmatrix} = \begin{bmatrix} 100 - 20\, z_1 & & & & & \\ 100 & -20\, z_2 & & & & \\ 100 & & -40\, z_3 & & & +60(\frac{1}{2} z_6) \\ & 20\, z_2 & & -100\, z_4 & & \\ & 20\, z_2 & +40 z_3 & & -55.5\, z_5 & \\ & & & 100\, z_4 & +55.5\, z_5 & -60\, z_6 \end{bmatrix}$$

$$DE: \begin{bmatrix} 5\, z_1' \\ 2\, z_2' \\ 30\, z_3' \\ 5\, z_4' \end{bmatrix} = \begin{bmatrix} 100 - 5\, z_1 & & & \\ 100 & -2\, z_2 & & \\ 5\, z_1 & +2 z_2 & -30\, z_3 & -5\, z_4 \\ & & 5(2 z_3) & -5\, z_4 \end{bmatrix}$$

Figure 2. Differential equations for the four systems in Figure 1, where z_i is the flow from process i to storage i.

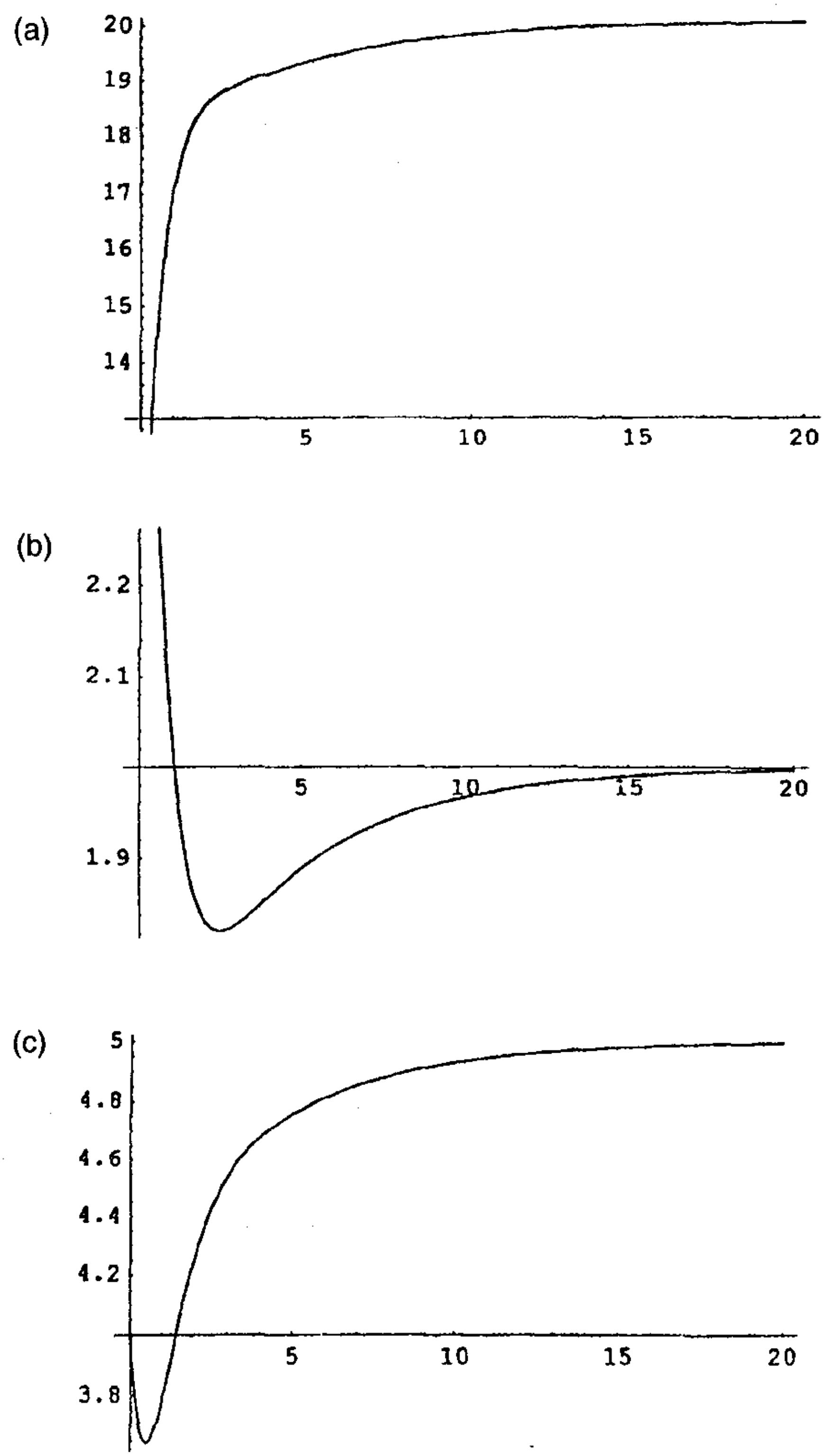

Figure 3. *Mathematica* output plots over time of the equations derived for system B of Figure 1. (a) z_1; (b) z_2; (c) z_3.

$$AE: \begin{bmatrix} z_1' \\ z_2' \end{bmatrix} = \begin{bmatrix} \frac{30}{T_1} - z_1 & \\ \frac{T_1}{T_2} z_1 & - z_2 \end{bmatrix}$$

$$BE: \begin{bmatrix} z_1' \\ z_2' \\ z_3' \end{bmatrix} = \begin{bmatrix} \frac{100}{T_1} - z_1 & + \frac{T_2}{T_1}(\frac{1}{2} z_2) & \\ \frac{T_1}{T_2}(\frac{1}{2} z_1) & - z_2 & + \frac{T_3}{T_2} z_3 \\ \frac{T_1}{T_3}(\frac{1}{2} z_1) & & - z_3 \end{bmatrix}$$

$$CE:$$

$$\begin{bmatrix} z_1' \\ z_2' \\ z_3' \\ z_4' \\ z_5' \\ z_6' \end{bmatrix} = \begin{bmatrix} \frac{100}{T_1} - z_1 & & & & & \\ \frac{100}{T_2} & - z_2 & & & & \\ \frac{100}{T_3} & & - z_3 & & & + \frac{T_6}{T_3}(\frac{1}{2} z_6) \\ & \frac{T_1}{T_4} z_2 & & - z_4 & & \\ & \frac{T_2}{T_5} z_2 & + \frac{T_3}{T_5} z_3 & & - z_5 & \\ & & & \frac{T_4}{T_6} z_4 & + \frac{T_5}{T_6} z_5 & - z_6 \end{bmatrix}$$

$$DE: \begin{bmatrix} z_1' \\ z_2' \\ z_3' \\ z_4' \end{bmatrix} = \begin{bmatrix} \frac{100}{T_1} - z_1 & & & \\ \frac{100}{T_2} & - z_2 & & \\ \frac{T_1}{T_3} z_1 & + \frac{T_2}{T_3} z_2 & - z_3 & - \frac{T_4}{T_3} z_4 \\ & & \frac{T_3}{T_4}(2 z_3) & - z_4 \end{bmatrix}$$

Figure 4. Differential equations for the four systems in Figure 1 in "normal form."