System Dynamics Modeling in Science and Engineering

Hans U. Fuchs
Department of Physics and Mathematics
Zurich University of Applied Sciences at Winterthur
8401 Winterthur, Switzerland
e-mail: hans.fuchs@zhwin.ch

Invited Talk at the System Dynamics Conference at the University of Puerto Rico
Resource Center for Science and Engineering, Mayaguez, December 8-10, 2006

Abstract: Modeling—both formal and computer based (numerical)—has a long and important history in physical science and engineering. Each field has developed its own metaphors and tools. Modeling is as diverse as the fields themselves.

An important feature of modeling in science and engineering is its formal character. Computer modeling in particular is assumed to involve many aspects a student can only learn after quite some time at college. These aspects include subject matter, analytical and numerical mathematics, and computer science. As a consequence, students learn modeling of complex systems and processes late.

The diversity and complexity of the traditional forms of modeling call for a unified and simplified methodology. This can be found in system dynamics modeling. System dynamics tools allow for an intuitive approach to the modeling of dynamical systems from any imaginable field and can be approached by even quite young learners.

In this paper, I shall argue that system dynamics modeling is not only simple but also powerful (simple ideas can be combined into models of complex systems and processes), useful (it makes the integration of modeling and experimenting a simple matter), and natural (the simple ideas behind SD models correspond to a basic form of human thought). It is a great educational tool and can be applied in the prototyping of complex real-life applications from science and engineering.

1 INTRODUCTION

System dynamics provides for a general and user friendly approach to the modeling of dynamical systems, irrespective of the field of application (Roberts et al., 1994; Fuchs, 2002a). Interestingly, the methods and tools developed in what is commonly called *system dynamics* are applied in only some of the many possible areas. Chief among these are social sciences (including economics and management) and environmental sciences. Also, system dynamics modeling has been made a part of some educational programs. There are (high) schools that make more or less extensive use of the practice of dynamical modeling in various subjects, ranging from biology to psychology. SD is conspicuously absent, however, from much of physics and engineering, particularly at college and in physics and engineering practice.

What is System Dynamics Modeling?

To understand why systems thinking, system dynamics, and system dynamics modeling are not generally used in science and engineering today we first need to define system dynamics.

System dynamics, as it was developed by Jay Forrester (1961), has its roots in control engineering, cybernetics, and general systems science—which, in turn, have their roots in early systems science in biology and physics. Donella Meadows wrote in 1991 (Meadows, 1991, p.1):

> *"System dynamics is a set of techniques for thinking and computer modeling that helps its practitioners begin to understand complex systems—systems such as the human body or the national economy or the earth's climate. Systems tools help us keep track of multiple interconnections; they help us see things whole."*

Here is an example of a model of a dynamical system that can introduce us to some aspects of system dynamics and the type of tools used in its practice. Imagine the economy of a country. In the most simple terms, the economy is understood as the processes of production and consumption of some types of goods. The goods themselves are accumulated in an inventory. This simple idea can be expressed pictorially by saying that there is a store of goods which is diminished by consumption and replenished by production (see the upper part of the diagram in Fig.1). The store may be represented as a rectangle (called a stock or reservoir) and the processes of production and consumption as pipelines or flows. By placing these graphical elements on the surface of the workspace of a system dynamics program (such as Stella, Vensim, Powersim, or Berkeley Madonna; see Sources for Tools at the end of the paper) and connecting them, we express a powerful idea—namely the *law of balance* of the inventory (the goods) of an economy. Put in standard language, we have formulated the idea that *the rate at which the inventory changes* must be given by the sum of the *production* and *consumption rates*. The software immediately converts this sentence into a formal expression:

```
Inventory(t) = Inventory(t-dt) + (Production - Consumption)*dt        (1)
INIT Inventory = Inventory(0)                                         (2)
```

This is the difference form (Euler form) of the *differential equation* that results from formulating a law of balance in instantaneous form, augmented by the initial value of the accumulating quantity. [It tells us that the inventory at time t is found from the inventory at a previous point in time, $t - dt$, by adding the net quantity accumulated as the result of the processes of production and consumption during the period dt. Here, dt stands for a finite interval of time.] In other words, by drawing a combination of a stock and one or several flows, the person creating a dynamical model writes a first order differential equation. The formulation of *initial value problems* (evolution equations) is at the fingertips of the learner or the practitioner of a field in which dynamical processes play a role.

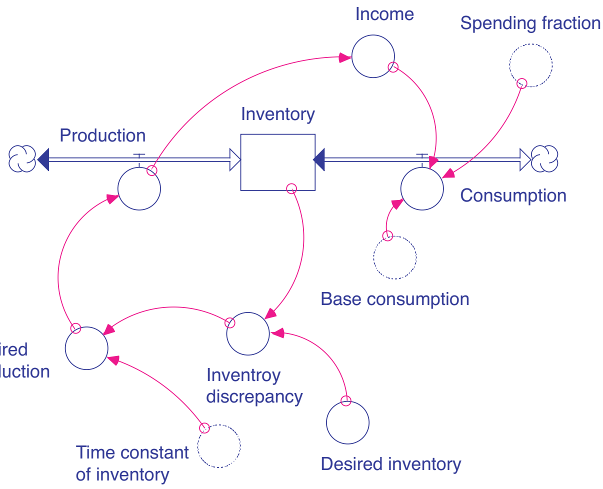


Figure 1: A model representing production, consumption, and accumulation of goods in an economy. Note the different elements. Rectangles (called stocks) represent stored quantities, pipelines (called flows) symbolize processes (flows or production rates). Combinations of stocks and flows represent laws of balance. The process quantities are determined by feedback relations expressed by circles (variables) and thin connectors. See Fuchs (2002a), p. 78.

The task that commonly takes more effort to complete concerns expressions for the process quantities (here the rates of production and consumption). Typically, proper models for processes involve the accumulating quantities which leads to feedback loops in the final model (the combination of storage elements with feedbacks is the hallmark of a model of a dynamical system; Fig.1). The formal expressions for the feedback relations normally are algebraic equations. In summary, the mathematical form of a model is a combination of first order differential equations and algebraic equations. Together, these constitute an initial value problem which is solved numerically by the software that supports the modeling process.

Let me stress what I see as an important aspect of system dynamics modeling. We use simple-to-use graphical software to express fundamental ideas concerning the operation of dynamical systems and processes. At the heart of these ideas is the search for laws of balance of easily visualized quantities that accumulate in a system as the result of processes. The laws of balance serve as the trunk to which the branches, i.e., the feedback relations defining the model, are added. Put still differently, we envision processes as resulting from the storage, flow, and production of some quantities that are metaphorically conceptualized as some kinds of stuff (Schmid, 1984, Fuchs, 1996, 1997b,c, 1998, 2002a, Herrmann, 1989-1999).

In summary, system dynamics modeling applies a graphical approach to building models of dynamical systems by combining the relations we perceive to hold in such systems. It makes use of very few structures which are projected onto virtually any type of dynamical system and its processes. As a result it applies a strong form of *analogical reasoning* (see Section 6).

Overview of the Paper

Why is standard system dynamics modeling hardly known in science and engineering? Since engineers, physicists, chemists and biologists all create dynamical models we can ask what the difference is between their practice and that of system dynamics. I shall take a look at this question in Section 2.

In the following sections I shall argue that the methodology of system dynamics modeling is perfectly suited to the subject of dynamical systems in the physical sciences and engineering. This holds for learning about dynamical systems as well as professional applications in these fields. I shall argue that system dynamics modeling is *simple* (simple tools make dynamical modeling accessible, Section 3), *powerful* (simple ideas can be combined into models of complex systems and processes, Section 4), *useful* (it makes the integration of modeling and experimenting a simple matter, Section 5), and *natural* (the simple ideas behind SD models correspond to a basic form of human thought, Section 6).

2 THE TRADITION OF MODELING IN PHYSICS AND ENGINEERING

Explicit formal mathematical modeling—with or without computers—has a long and important tradition in science and engineering (to the extent that they are based upon physical sciences). Therefore, it may not come as a surprise that system dynamics modeling as it has been practiced in the last 50 years is not found much in these fields. Physics and engineering have not exactly been waiting for a modeling methodology that at first seems to offer little new and actually appears to constitute a step backwards—away from the formal approaches developed in and so dear to the physical sciences.

Newtonian Mechanics

We may with some justification call Isaac Newton the first system dynamicist. All of us who have grown up in the tradition of physical sciences know how to formulate models of simple dynamical systems. Take the example of a damped oscillator, i.e., a body oscillating back and forth on a spring. We formulate Newton's Second Law:

$$ma = -Dx - \beta v \tag{3}$$

(Note that this includes more than just the basic Second Law. I have already introduced possible expressions for the forces acting upon the body.) We know that acceleration is the first derivative of speed, and speed is the first derivative of position. So we can write the model as a set of two first order differential equations which (including the initial values) take the form

$$\begin{aligned} \dot{x} &= v & \qquad x(0) &= x_0 \\ \dot{v} &= -\frac{D}{m}x - \frac{\beta}{m}v & \qquad v(0) &= v_0 \end{aligned} \tag{4}$$

We can now use a standard graphical system dynamics tool to *code* the equations of this model (see Fig.2). This only requires us to interpret a flow as a rate of change and the combination of a flow and a stock as an integrator. The expressions for the flows result from the model equations, i.e., the system of initial value equations in Eq.(4).

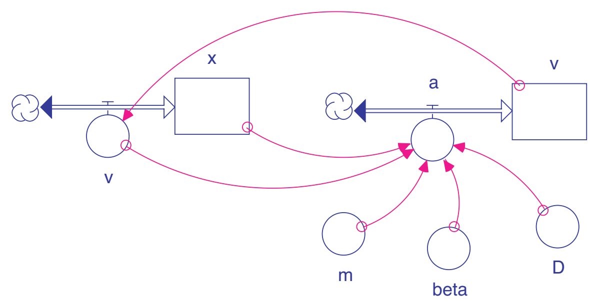


Figure 2: A system dynamics diagram of the model of a damped oscillator formulated in Eq.(4). There is no law of balance apparent in this formulation. (A proper system dynamics model is found in Fig.4.)

From the viewpoint of the tradition of physics, using a modern system dynamics tool does not add anything new. We may use such a tool to code the equations of a previously formulated model and let the software solve the equations numerically. Naturally, we can use any software that allows us to code and simulate initial value problems. Most likely, such tools (for example Matlab) are much more powerful from a purely mathematical standpoint than the simple system dynamics programs that have been developed since the 1980s. Therefore, anyone growing up in and following the tradition does not need Stella or Berkeley Madonna.

[Interestingly, when seen from the viewpoint of the practice of system dynamics modeling, the use of modern mathematical software such as Matlab misses two important aspects. First, there is no modeling involved, just coding and solving. Second, the model presented above does not seem to make use of a law of balance which—as described in Section 1—is a cornerstone of system dynamics modeling.

[Naturally, there is modeling. It is the formulation, by hand and in standard mathematical notation, of the initial value problem in Eq.(4) which precedes the coding and solving of the equations. Also, there is a law of balance behind the model; it is just not visible in the form that has evolved in the years following the introduction of Newtonian mechanics. We tend to forget that Newton expressed motion in terms of what we today call the law of balance of momentum.]

Dynamical modeling in physics can be quite sophisticated from a purely analytical viewpoint. However, many applications, particularly in education, do not go far beyond the example of the motion of a body presented here. There are only a few dynamical models treated in introductory courses. Much of the education is concerned with other aspects of the science. In some cases, physics does not even have a dynamical theory. Just think of thermodynamics which is still presented as a theory of the statics of heat. The development of a dynamical theory of heat had to await the last two or three decades of the 20th century (Fuchs, 1996). Clearly, we do not need system dynamics modeling in traditional physics, neither in teaching nor in research.

Continuum Physics and Finite Element Modeling

A general dynamical theory of macroscopic processes has evolved in continuum physics which was generalized and expanded during the second half of the 20th century (Truesdell and Toupin, 1960, Truesdell and Noll, 1965, Eringen, 1971-76, Truesdell, 1984, Müller, 1985, Jou et al., 1996). Here we see the conceptual elements of system dynamics modeling: Laws of balance and constitutive laws as expressions of the process quantities in the laws of balance (Fuchs, 1996). However, to do justice to a sophisticated continuum physics (or engineering) model, we need finite element tools. System dynamics programs can barely reflect the power (and the glory) of finite element modeling.

[If we take a less narrow view of the tradition, we see that system dynamics can serve as a stepping-stone into a complex world. In particular, if modeling rather than coding is asked for, we can use system dynamics tools to develop some simple distributed (pseudo-finite-element) versions of continuum models (Fuchs, 1997a, 2002a). I shall present a simple example below in Section 4.]

Control Engineering

System dynamics evolved out of systems science and control engineering, so we might expect a field of engineering close in philosophy and in practice to system dynamics and system dynamics modeling. Again, we will be disappointed. While there is a philosophical affinity between control engineering and system dynamics, the concrete applications and practice of the two fields seem totally disjointed. Applications of control engineering are narrowly focused using a highly developed formalism, whereas those of system dynamics are broad.

The focus of control engineering has brought about a type of computational tool that is quite different from what we have seen in system dynamics. In the last decade, a tool of choice (Simulink, based upon Matlab) has evolved that makes use of a graphical interface for building models of initial value problems, but the similarity stops there. The metaphors of system dynamics (as exemplified by Stella or Berkeley Madonna) are not the same as those employed in Simulink. Simulink uses the tools of signal processing, and modeling is guided by the metaphor of signals entering and leaving blocks that act upon these signals.

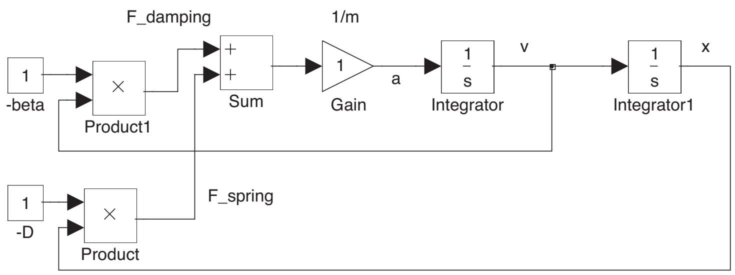


Figure 3: A simulink diagram of the model of a damped oscillator presented above in Eq.(4) and in Fig.2. There are no elements in Simulink that would let us directly express laws of balance.

[The graphical interface of Simulink supports the actual modeling process—rather than just the pure coding of pre-existing equations. However, it turns out to be very cumbersome to produce models of dynamical physical or chemical or biological processes if our thinking is guided by the structure of laws of balance and constitutive laws. My students typically start using Simulink for engineering applications only after having modeled physical processes with the help of system dynamics tools in the first year of their engineering degree courses.]

Chemical Engineering and Biochemistry

The three fields discussed (mechanics, continuum physics, and control engineering) are all based on the physical sciences and their models lead to initial value problems. We should expect modeling methodologies to be the same. As we have seen, this is not the case by a long shot. It might come as a relief that there is an established field in engineering that makes use of the basic metaphor that supports system dynamics modeling of physical processes.

In chemical engineering, students learn to apply the method of setting up laws of balance to create (steady-state or dynamical) models of relevant applications (Benitez, 2002; Richards, 2001). The models of chemical engineering are based upon transport processes and reactions and are guided by the results of continuum physics. Therefore, it is less surprising that the practice of system dynamics modeling finds a more sympathetic audience in this field of engineering. Indeed, Berkeley Madonna has turned out to be the tool of choice for some practitioners in the field (Ingham et al., 2007).

Aspects of engineering practice such as these have lead to the formulation of principles of a generalized engineering education that explicitly applies the same metaphors and methods as those found in system dynamics (The Foundation Coalition; see also Richards, 2001). We might say that the *physics of dynamical systems* has been made a part of basic engineering education. This is an example of the integrative power of the paradigm of dynamical systems.

Last but not least, let me add the example of system dynamics modeling developed by the creators of Berkeley Madonna. They apply the modeling methodology to examples from physiology, cell biology, biochemistry, genetics, and population dynamics (Macey and Oster, 2006). Their website provides a wealth of examples that help us understand dynamical processes in biology in a way we might not otherwise.

Multiport Modeling

In recent years, a modeling language and its associated tools have been developed that support the creation of dynamical models based on the paradigm of the physics of dynamical systems. To distinguish the approach and the tools from others, one speaks of multiport modeling. As an example, Modelica is a highly sophisticated tool that implements the methodology of multiport modeling. It lets the user create objects that exhibit well defined properties. By combining the objects, we can build models of complex dynamical engineering applications (Maurer, 2006).

Explicit Modeling in Physics Instruction

In the 1980s, David Hestenes pointed out that we teach models instead of modeling in our standard physics courses for scientists and engineers. [Incidentally, this may be a reason for our reluctance to apply system dynamics modeling in our introductory courses.] He pioneered a *modeling theory of physics instruction* to work against this trend (Hestenes, 1987, 1982; Halloun and Hestenes, 1987; Wells, Hestenes, Swackhamer, 1995; Hestenes, 2006). His work brought to our attention that modeling must be more than a professional activity that is practiced after one has learned a science. Rather, it should be made an explicit part of instruction.

The discussions in this section show one thing very clearly: There is a strong and diverse tradition of modeling in the physical sciences and engineering. If we wish to integrate approaches found in different fields and in particular, if we want to start students early on a path to modeling that can serve as a basis of later specialization, system dynamics may just be the way to go.

3 SYSTEM DYNAMICS MODELING IS SIMPLE

This section can be kept short. We have already seen in the introduction how simple it can be to create a model of a dynamical system. Here, I will produce the model of the damped oscillator in system dynamics form. At the same time we will recognize a structure of dynamical models that was not present in the simple economic model of Fig.1, namely the phenomenon of induction.

Let us look at motion as the result of what happens to the momentum of a body. This is Newton's original viewpoint. Momentum becomes the quantity accumulating in the body, and forces quantify the flow of momentum into and out of it. In other words, force is nothing but a momentum current (here denoted by the symbol I_p). Therefore, we express the law of balance of momentum involving the momentum of the body and two momentum transports (those due to the action of the spring and to damping; see the structure at the top in Fig.4).

We have to find expressions for the momentum flows in the model. The one for the damping is easy (here it is simply equal to the negative product of the speed of the body and a damping factor). The one representing the action of the spring leads to an interesting structure. Let us look at the relation describing the action of the spring in dynamical terms. We can say that the speed at which the end of the spring attached to the body moves determines how fast the force of the spring (the momentum current of the spring) changes. This is a typical law of induction: The particular physical situation determines the rate of change of a current. It happens quite frequently that nature gives us a relation not in the direct form we might wish for, but rather indirectly by specifying the rate of change of the quantity we are interested in. If this happens, we integrate the rate of change to obtain the quantity itself.

[Note that Stella or Berkeley Madonna do not include a separate element for an integrator. We have to use the same elements employed in the construction of a law of balance, i.e., a stock and a flow. However, in the case of a simple integration, we never have more than a single pipeline whereas a typical law of balance involves more than a single flow or production rate; compare the upper to the lower stock-flow combinations in Fig.4.]

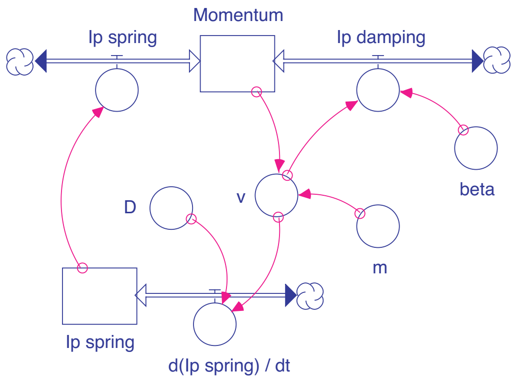


Figure 4: A system dynamics version of the model of the damped oscillator. It consists of the law of balance of momentum for the body, the law of mechanical induction describing the action of the spring, and an expression for the damping force. Forces are expressed as momentum currents.

Simulation of the model is made simple by the tools available for system dynamics modeling. Disregarding numerical problems, the simulation only involves the setting of some simulation parameters and the definition of an output medium. A simulation of the model in Fig.4 can be seen in Fig.5.

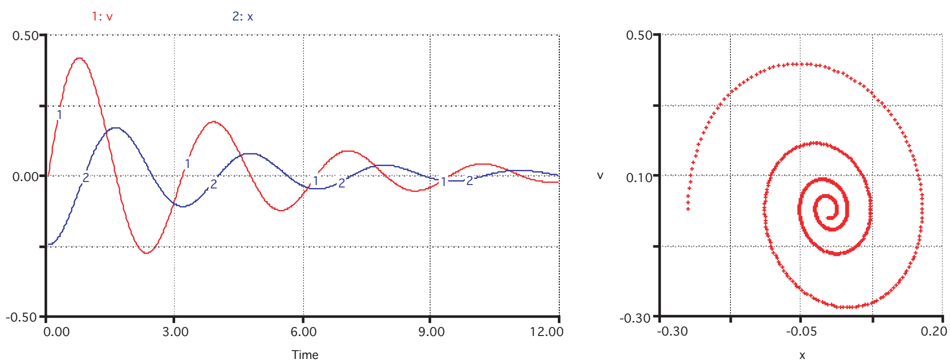


Figure 5: The simulation of the model in Fig.4 results in the relevant quantities as functions of time (left). If the speed of the body is represented as a function of its position, a phase plot (right) is obtained. The position *x* (not included in the model diagram of Fig.4) is calculated from –Ip_spring / D.

4 SYSTEM DYNAMICS MODELING IS POWERFUL

In this section, I would like to demonstrate how simple ideas lead to simple models that can serve as elements of more complex ones. In other words, models of complex systems are obtained by combining relatively simple building blocks. This is seen here in a sequence of models of hydraulic systems starting with tanks and pipes that culminates in a distributed model of the aorta of a mammal.

Consider two cylindrical tanks connected by a pipe at the bottom (Fig.6). The tanks contain oil which is allowed to flow through the pipe. We know how the dynamical process works: It results in a flow that is strong at first, weaker later on, and zero in the end when equilibrium of levels has been established. Creating a dynamical model of this system and its processes is quite intuitive and follows a sequence of thoughts and actions that can be repeated for other systems and processes. The thoughts and actions are guided by only a few considerations and question:

- Find the quantity (or quantities) that accumulate(s) in storage elements. Find the flows and/or production rates for each quantity and each element. Express the law(s) of balance.
- To create expressions for the relevant processes (for the flows), ask the following question (divide it into two steps): What quantities does the process directly depend upon…
- …and how?

The answer to the first part of the question is a tentative and unordered list of possible quantities. This list must be streamlined and ordered to be useful for the construction of a proper relation.

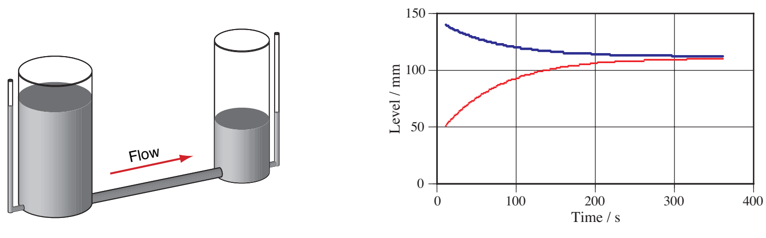
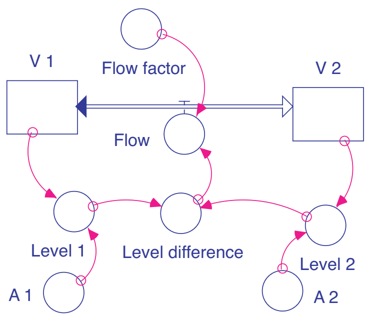


Figure 6: Two communicating tanks (containing a liquid such as oil) are joined at the bottoms by a pipe. The levels of the liquid behave as shown in the accompanying graph. We note that the levels change quickly at first and more slowly later on. Finally, they stop changing when they have become equal.

Taking these steps and answering these questions results in a creative and constructive process leading to a useful dynamical model like the one in Fig.7.



*V_1(t) = V_1(t - dt) + (- Flow) * dt*
*INIT V_1 = A_1*h1_init*

*V_2(t) = V_2(t - dt) + (Flow) * dt*
*INIT V_2 = Cross_section_2*h2_init*

Level_1 = V_1/A_1
Level_2 = V_2/A_2
Level_difference = Level_1-Level_2
*Flow = Flow_factor*Level_difference*

Figure 7: The model diagram shows a possible answer to the question of why the system behaves as observed (model equations are on the right). In particular, we need a model that explains why the levels equilibrate and why they change fastest at the beginning and more slowly later on. The quality of the model is judged on the basis of comparisons of simulations and experimental data.

Let us now change the system step by step and make it more complex. We first add an outflow to one of the two tanks in Fig.6. In the model, this is accomplished by adding a flow to the stock representing the volume of liquid in one of the tanks. The model diagram looks like the one in Fig.8.

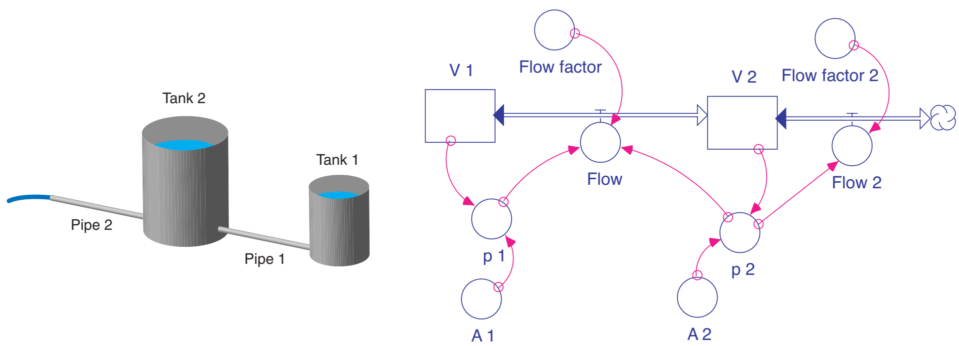


Figure 8: Two communicating tanks having an additional outflow. The model diagram is shown on the right. We can see how it evolves from the model of the communicating tanks in Fig.7. However, important details such as the particular flow laws used cannot be seen in the diagram.

There is a possible change to the model that cannot be seen in the typical graphical model diagrams. What if we change the liquid from oil to water? The diagram in Fig.8 remains the same, but the equations for the two flows have to be adapted to the new situation (in general, oil leads to laminar flow having a linear characteristic, and water flows are turbulent which may be represented by a square root characteristic).

Let us now change the system to a typical windkessel consisting of a pump, a short pipe with a valve leading into a storage tank, and a long pipe leading out of the tank (Fig.9; this is the standard first step in understanding blood flow in the systemic circuit from the left ventricle of the heart through the aorta and through the body of a mammal). The model diagram is nearly as simple as that of two communicating tanks (Fig.9). The model assumes that the pressure set up by the pump is given as a function of time. In other words, this particular quantity is assumed to be a driving function of the model and is not included as a part of the dynamical model. The boundary of the model is around the valve, tank, and long pipe, and it does not include the pump. The boundary of a system is an important element of any type of model. It allows us to focus on some parts of a (larger) system and express the function of the environment (of the rest of the world) with the help of given data. Naturally, this strongly limits the applicability of a given model. If the environment changes in response to changes in the modeled system, we should normally be concerned with including the environment explicitly in the model.

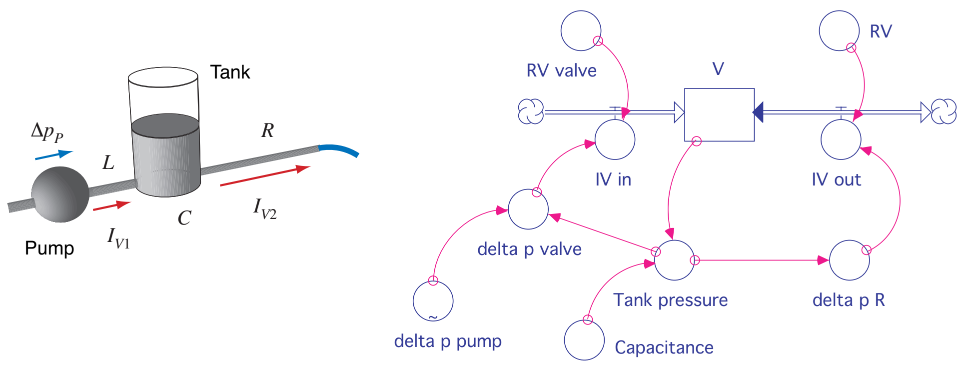


Figure 9: A system dynamics model diagram for a simple windkessel (left). The windkessel was designed for fire pumps, and it serves as a model of the interaction of the left ventricle of the heart and the aorta.

As a final step, let us apply the windkessel model to the operation of the systemic circuit of a mammal's blood flow system. Furthermore, we divide the aorta into a number of segments each of which is treated as a storage and flow element (Fig.10).

All it takes to create a distributed (pseudo-finite-element) model is to copy an already existing part several times and then to connect the models of each element properly.

System dynamics tools were not developed for the kind of work needed to model complex multi-dimensional spatially distributed engineering applications. Nevertheless, colleagues and

I have been successful in using the simplest tools for model prototyping in mechanical engineering and solar energy engineering. SD tools can be quite powerful after all.

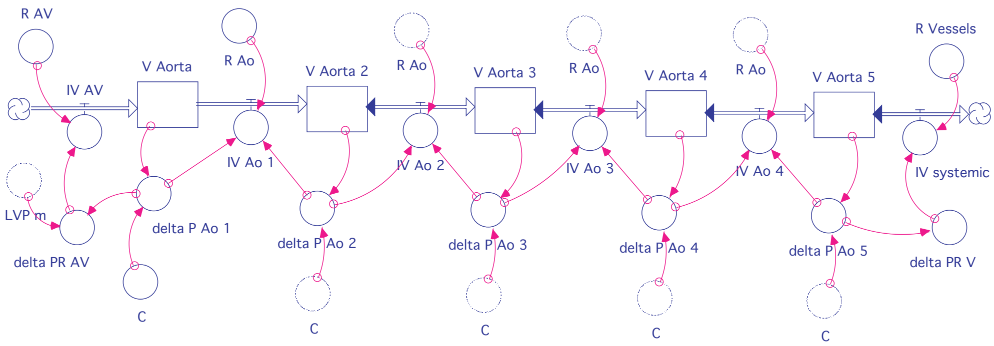


Figure 10: A multi-compartment (or pseudo-finite-element) model of the aorta of a mammal created on the basis of the windkessel model presented in Fig.9.

5 SYSTEM DYNAMICS MODELING IS USEFUL

As seen in the previous section, system dynamics modeling allows us to approach quite complex tasks. It is useful in still another important practical educational respect—that of integrating the lab with model creation, simulation, and model validation. In other words, the simplicity and power of SD modeling tools let us apply the scientific method quite easily and successfully (Fuchs, Ecoffey, and Schuetz, 2001-2005; Fuchs 2006).

Reconsider the development of the short series of models of two tanks containing either oil or water, with or without an additional outflow (Fig.6 - Fig.8). A first typical idea for a flow relation is that of a linear characteristic which works very well in the case of laminar flow. Indeed, the first model (Fig.7) was validated and parameters were adjusted for the case of oil flowing through the pipe connecting the two tanks (data had been taken previously and was imported into the model for comparison).

Finally, the model was applied to the case of two tanks with an additional outflow, this time with water as the liquid. A simulation showed qualitatively similar results as those found in the lab, but it was clear that the fit was not nearly as good as in the case of the flow of oil. This led to the investigation of water flowing from a single tank and the realization that the flow characteristic had to be changed. (This step involves the construction of a smaller model for the simple case of draining of a single tank.) Applying this new idea to the model of two communicating tanks having an additional outflow was clearly successful (Fig.11).

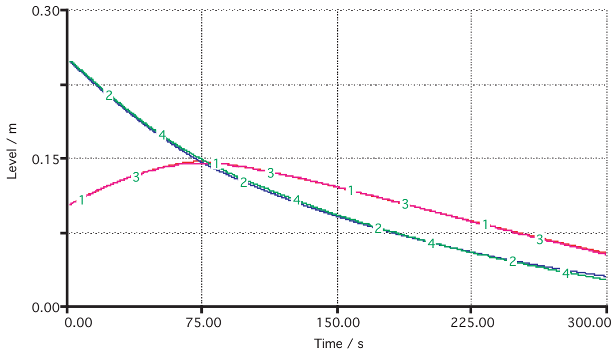


Figure 11: Simulation result compared to experimental data in the case of the model of two communicating tanks having an additional outflow (Fig.8). Note that the fit between simulation (curves 1 and 2) and experimental data (curves 3 and 4) is almost perfect. The model for the flow through the connecting pipe is Flow = IF ((P1-P2) > 0) THEN SQRT((P1-P2)/k1) ELSE -SQRT(-(P1-P2)/k1).

As a final example, let us discuss experiments with and models of a Peltier thermoelectric device used to cool water (Fig.12). The device serves as the wall separating two bodies of water in a fairly well insulated tank. When it is operated, one body of water becomes cooler while the temperature of the other one goes up.

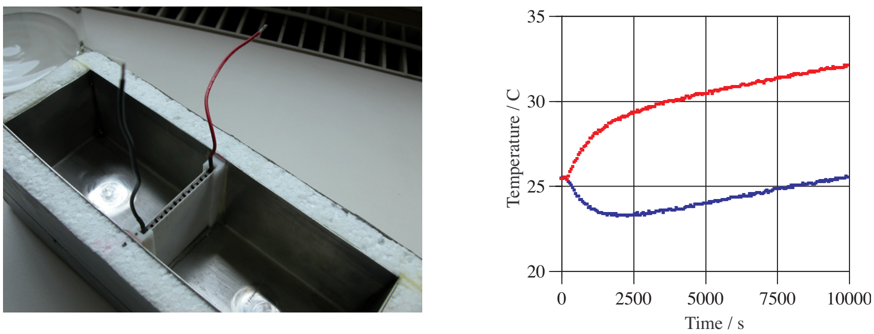
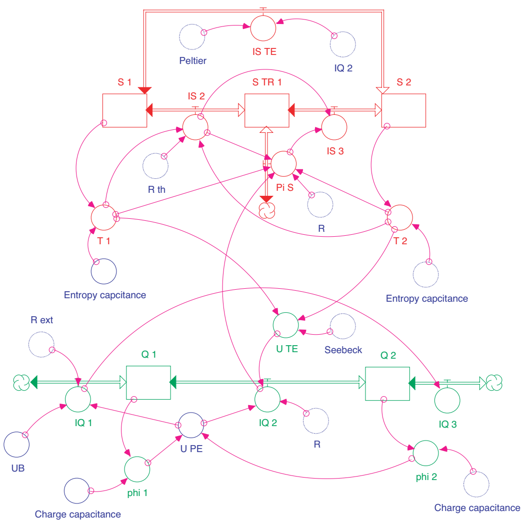


Figure 12: A Peltier thermoelectric device operated as a heat pump between two bodies of water. The device serves as the separating wall and as the heat pump. The two bodies of water are stirred with magnetic bars to insure homogenous conditions. Apart from the electric quantities, the temperatures of the two bodies of water are measured as functions of time (graph on the right). There is a lid on the container.

Getting a proper model of the entire system—Peltier device, water, and tank—required several rounds of going back and forth between modeling and experimenting. The system dynamics model of the device itself involves a thermal and an electric part (Fig.13) and is relatively easy to build once one understands how to model electric and thermal processes in analogy to the hydraulic examples presented before (see Section 6).



U_TE = Seebeck(T_2-T_1)*
*IS_TE = Peltier*IQ_2*
Pi_S = P_diss/T_2
P_diss =
-IS_2(T_2-T_1) + R*IQ_2^2*
Seebeck = Peltier

Figure 13: SD model of a Peltier device showing thermal and electric aspects, and their coupling. Note the entropy production rate (Pi_S) that models thermal and electric irreversibilities (Fuchs, 2002b).

It turns out that modeling the Peltier device is only one part of the entire job (see the expression for the balance of entropy of the entire setup, Fig.14), and taking the data shown in Fig.12 is only one of several different experiments necessary to properly specify the many parameters of the system. Among other things, thermal aspects of the system had to be investigated with the Peltier device in passive mode (just serving as the separating wall). In fact, one can think of half a dozen of experiments that could or should be performed that help to determine some of the parameters of the system before the complete setup is modeled. In summary, a moderately complex system can serve as an example of the cyclic nature of the scientific method as we go back and forth between the lab and the computer, always comparing the results of modeling with those found in experiments (Fuchs, 2006).

6 SYSTEM DYNAMICS MODELING IS NATURAL

There is another reason why we might want to consider system dynamics modeling to introduce learners to the art and practice of creating dynamical models. Unlike other methods and tools, the programs created for system dynamics modeling make use of a metaphor which resembles most closely the one we humans use to conceptualize phenomena and experiences ranging from fluids, electricity, heat, and motion, to biological-psychological ones such as love, anger, or pain. The list even includes social-philosophical examples such as justice.

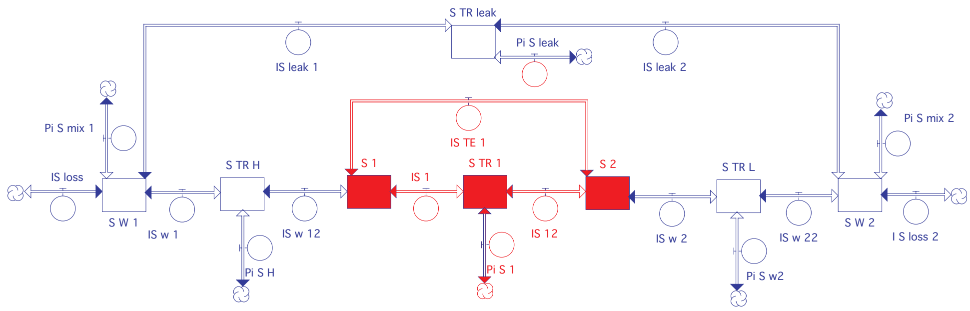
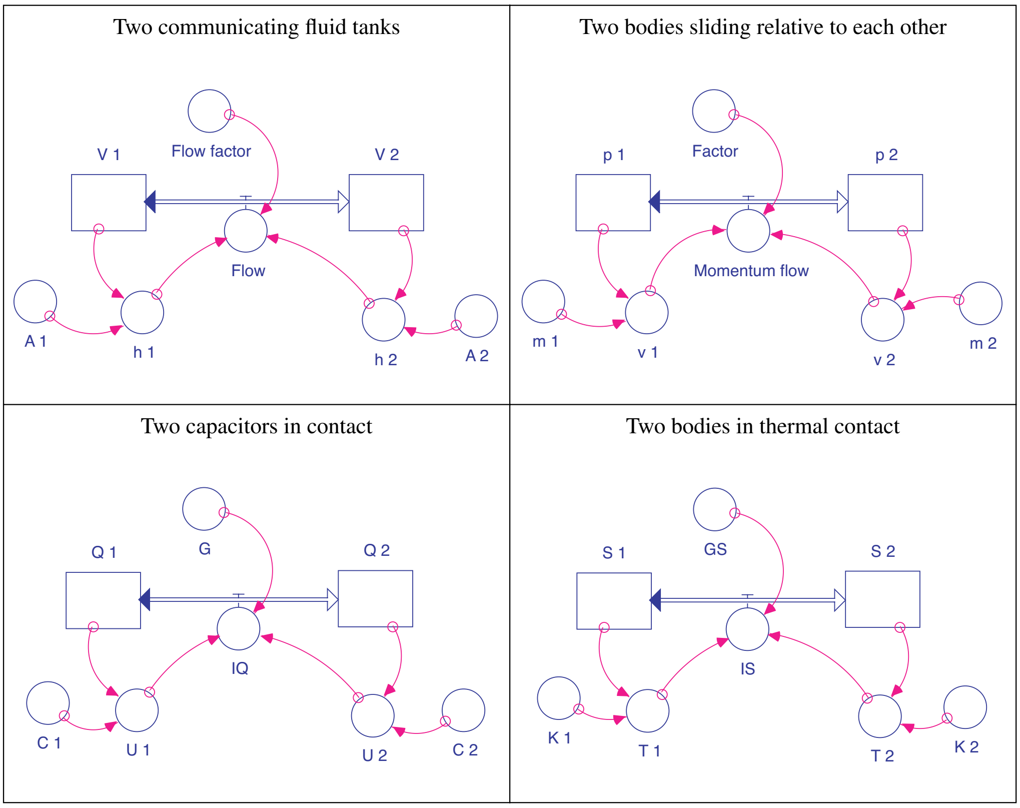


Figure 14: Expression for the balance of entropy of all the parts involved in the thermal system. The solid part at the center refers to the two sides (hot and cold) of the Peltier device (see Fig.13). The stocks on the left and on the right represent the entropy of the two bodies of water. There are entropy flows through the device into and out of the bodies of water, entropy loss to or gain from the environment, and a bypass from the warm to the cold body of water through the crack between Peltier device and styrofoam lid. The additional stocks serve as entropy nodes used to anchor the entropy production rates.

It appears that we humans construct a *gestalt* from these experiences that has, among others, the aspects of *quantity* (more heat, less pain, more justice), *intensity* or *quality* (higher intensities of heat or pain, lower quality of justice), and *force* or *power* (the power of water, of heat, or of pain or justice). These aspects, in turn, are conceptualized metaphorically (Lakoff and Johnson, 1980, Johnson, 1987) using object (or fluid substance) schemas for quantity, the orientational schema of up and down for intensity, and the gestalt of direct manipulation for force or power (see Fuchs, 2005, 2006, for a more detailed discussion of these ideas).

The important point for us here is this: If phenomena as diverse as those encountered in fluids, electricity, heat, and motion are all conceptualized using the same basic gestalt and aspects, these phenomena become similar. This allows us to apply analogical reasoning (Fuchs, 1996, 2002, 2006; Borer et al., 2005). We should expect to be able to apply strong analogies to our models of these phenomena. Indeed, aided by a typical system dynamics tool, we create models of hydraulic, electric, thermal, and mechanical processes that all look fundamentally alike (see the table below). Having a tool that was created for processes in general irrespective of their origin, what else should we expect? Note that system dynamics tools use—at least partly—the same metaphorical expressions for experiences as those I identified above. We have the elements of substance (storage of a substance-like quantity), and of interaction (conceptualized as the exchange of the substance). Other aspects (such as intensity) are less conspicuous, but they can be added to the graphical representation of a model in a fairly intuitive manner. [In an ideal world, we might construct a generilized modeling tool that implements the gestalt of physical processes outlined here. It would support thinking about processes in science and engineering along the lines discussed in this paper. A system dynamics program would be an element of this generalized tool.]



Graphically oriented system dynamics tools let us express our metaphors visually. This is an aspect whose power should not be underestimated. Being able to assemble the structure of a dynamical model graphically is quite different from having to write (first order differential and algebraic) equations on paper, and having to manipulate them. That is not to say that we can always do everything with a particular tool that can be done with another. Each tool has its advantages. Let us explore and use those offered by system dynamics.

7 SUMMARY

I hope I have been able to demonstrate that system dynamics modeling can play an important and constructive role in science and engineering, both in education and in real-life applications. If modeling is not an activity that follows *after learning* has taken place but rather is used as a method *for learning*, we should consider carefully what system dynamics has to offer.

In an important series of papers, David Hestenes has made it clear that modeling is a path to learning (Hestenes, 1987, 1982, 2006). In a very general sense it is even *the* path to learning in a science. Modern computer tools have made explicit modeling of physical, chemical, biological and engineering applications accessible to a much larger audience than would have been considered possible only a few years ago. Finally, the development of pedagogy has added a sense of how important activity based learning can be. There is hardly a simpler method of integrating the lab with model construction than that offered by system dynamics modeling tools.

I would like to finish by quoting again what Donella Meadows wrote in 1991 (Meadows, 1991, p.1):

> *"System dynamics is a set of techniques for thinking and computer modeling that helps its practitioners begin to understand complex systems—systems such as the human body or the national economy or the earth's climate. Systems tools help us keep track of multiple interconnections; they help us see things whole."*

Science and engineering have become truly complex and divided. It is time we help our students to see them whole again.

REFERENCES

Benitez J. (2002): *Principles and Modern Applications of Mass Transfer Operations*. Wiley, New York.

Borer, Frommenwiler, Fuchs, Knoll, Kopacsy, Maurer, Schuetz, Studer (2005): *Physik – ein systemdynamischer Zugang*, 2. Auflage, h.e.p. verlag, Bern.

Eringen A.C. (1971-1976): *Continuum Physics*, Vol.I–IV. Academic Press, New York.

Forrester J. (1961): *Industrial Dynamics*. MIT Press, Cambridge, MA.

Foundation Coalition: http://www.foundationcoalition.org/.

Fuchs H. U. (1996): *The Dynamics of Heat*. Sprinnger-Verlag, New York.

Fuchs H. U. (1997a): The Missing Link: Introductory Continuum Physics for Engineering Students. Paper presented at the First European Conference on Physics Teaching in Engineering Education (PTEE 1997) in Kopenhagen, Denmark.

Fuchs H. U. (1997b): The Continuum Physics Paradigm in physics instruction I. Images and models of continuous change. Zurich University of Applied Sciences at Winterthur. http://home.zhwin.ch/~fuh/LITERATURE/Literature.html

Fuchs H. U. (1997c): The Continuum Physics Paradigm in physics instruction II. System dynamics modeling of physical processes. Zurich University of Applied Sciences at Winterthur. http://home.zhwin.ch/~fuh/LITERATURE/Literature.html

Fuchs H. U. (1998): The Continuum Physics Paradigm in physics instruction III. Using the Second Law. Zurich University of Applied Sciences at Winterthur. http://home.zhwin.ch/~fuh/LITERATURE/Literature.html

Fuchs H. U. (2002a): *Modeling of Uniform Dynamical Systems*. With Computer Based Training Unit. Orell Füssli, Zurich.

Fuchs H. U. (2002b): A simple continuum model leading to the reciprocity relation for thermoelectric effects. Report, Zurich University of Applied Sciences. http://home.zhwin.ch/~fuh/LITERATURE/Literature.html

Fuchs H. U. (2005): Metaphern machen Physik—Welche Beziehung haben Wörter zur Welt? Kolloquiumsvortrag. Zurich University of Applied Sciences at Winterthur. http://home.zhwin.ch/~fuh/LITERATURE/Literature.html

Fuchs H. U. (2006): System Dynamics Modeling in Fluids, Electricity, Heat, and Motion. Procedding of the 2006 GIREP Conference on Modeling in Physics and Physics Education, University of Amsterdam.

Fuchs H. U., G. Ecoffey, E. Schuetz (2001-2006): Physics as a Systems Science. Materials for an introductory university physics course.

Halloun I. A. and Hestenes D. (1987): “Modeling instruction in mechanics,” Am.J.Phys. **55**(5),455-462.

Herrmann F. (1989-1999): Der Karlsruher Physik-Kurs. Sekundarstufe 1 (3 Bände), Sekundarstufe 2 (2 Bände), 2 Lehrerbände. Universität Karlsruhe. ISBN 3-761-42095-1 etc.

Hestenes D. (1987): “Toward a modeling theory of physics instruction,” Am.J.Phys. **55**(5),440-454.

Hestenes D. (1992): “Modeling games in the Newtonian World,” Am. J.Phys. **60**(8),732-748.

Hestenes D. (2006): Notes for a Modeling Theory of Science, Cognition and Instruction. Proceedings of the 2006 GIREP Conference on Modeling in Physics and Physics Education, University of Amsterdam.

Ingham J., Dunn I. J., Heinzle E., Prenosil J. E., Snape J. B. (2007): *Chemical Engineering Dynamics: An Introduction to Modelling and Computer Simulation*. Wiley, New York.

Johnson M. (1987): *The Body in the Mind. The Bodily Basis of Meaning, Imagination, and Reason.* University of Chicago Press, Chicago.

Jou D., Casas-Vazquez J., Lebon G. (1996): *Extended Irreversible Thermodynamics.* Springer-Verlag, Berlin.

Lakoff G. and Johnson M (1980): *Metaphors We Live By.* University of Chicago Press, Chicago (with a new Afterword, 2003).

Macey R., Oster G. (2006): Dynamic modeling of biological systems. http://mcb.berkeley.edu/courses/mcb137/

Maurer W. (2006): DyMoRail–oder wie simuliert man Eisenbahnzüge? http://home.zhwin.ch/~mau/Vortraege/TKoll_2005.pdf

Meadows, D. H. (1991): *System Dynamics meets the press*. In: The Global Citicen. Island Press, Washington.

Müller I. (1985): *Thermodynamics*. Pitman, Boston MA.

Richards D. E. (2001): *Basic Engineering Science: A Systems, Accounting, and Modeling Approach*. See http://www.rose-hulman.edu/~richards/textbooks/index.htm

Roberts N., W. Shaffer, M. Garet, R. Deal, D. Andersen (1994): *Introduction to Computer Simulation: A System Dynamics Modeling Approach*. Productivity Press, University Park IL.

Schmid G. B. (1984): “An up-to-date approach to Physics,” Am. J.Phys. **52**(9),794-799.

Truesdell C. A. (1984): *Rational Thermodynamics*. Springer-Verlag, New York, 2nd ed.

Truesdell C. A. and Toupin R. A. (1960): "The Classical Field Theories," in *Encyclopedia of Physics*, v. III/1, S.Flügge ed. (Springer-Verlag, Berlin, 1960).

Truesdell C. A. and Noll W. (1965): "The Non-Linear Field Theories of Mechanics," in *Encyclopedia of Physics*, v. III/3, S.Flügge ed. (Springer-Verlag, Berlin, 1965).

Wells M., Hestenes D., Swackhamer G. (1995): "A modeling method for high school physics instruction," Am.J.Phys. **63**(7),606-619.

SOURCES FOR TOOLS

Berkeley Madonna: Graphical (Flowchart) and text based model development. Strong tools for simulation and automatic curve fits. http://www.berkeleymadonna.com/

Matlab: A tool for doing mainly numerical mathematics, data analysis, graphical display. http://www.mathworks.com/

Modelica: A multi-port modeling engineering tool. http://www.modelica.org/

Powersim: http://www.powersim.com/

Simulink: A graphical editor for creating dynamical models for signal processing and control engineering. http://www.mathworks.com/

Stella: The first of the graphically oriented system dynamics tools. Still one of the easiest to use. http://www.isee-systems.com/

Vensim: Similar to Stella. There is a free (but limited) fully functional version of the program. http://www.vensim.com/