TEACHING BUILDING SIMULATION WITH THE HELP OF GENERAL PURPOSE DYNAMICAL MODELING SOFTWARE

Hans U. Fuchs, Martin Simon
Department of Physics, Technikum Winterthur
8401 Winterthur, Switzerland

ABSTRACT

Learning how to model and simulate dynamical systems such as a building does not have to be restricted to specialists who have learned the physics, applied mathematics, and engineering necessary for the task. Modern software tools allow for a new approach to modeling which makes it accessible to a much wider group of scientists, eng neers, and architects. This paper presents an integrated modular method for training students in the methodology of modeling and simulation. Students can leave the process at different levels allowing them to either appreciate the content of models of dynamical systems and intelligently use existing models; to be able to quickly build simple models themselves for testing ideas; or to become full-fledged modelers.

INTRODUCTION

Simulation in the field of dynamical systems, be it in energy engineering or building science, is commonly left to specialists who have learned the physics, applied mathematics, and engineering necessary for the task. This leaves most of the students of architecture and of less specialized fields of engineering without training in this vital discipline, even though many of them apply the results of models of real world applications. Even for those who later turn to modeling and simulation, this methodology represents the end-result, rather than part of the process, of the education in physics and applied mathematics.

We have found that general purpose modeling software with graphical interfaces can serve as tools in the learning of modeling and simulation by students who may or may not become modelers themselves. For the latter (mostly students in architecture), the goal is to teach appreciation of the modeling process, which means joining in their minds simple laws of physics and the real behavior of buildings, and of the applicability of the results of this trade. For the former (mainly engineering students), early teaching of modeling and simulation supports the phase of learning about the physics of dynamical systems, and builds the abilities necessary for modeling of real applications in later phases of their training.

The software used at Technikum Winterthur falls in two categories, the second of which is reserved for later modelers. The categories are system dynamics modeling software (basically the program Stella, 1987), and block diagram software for dynamical systems known from engineering (basically the program Simulink, 1993). The interesting point is that neither product was written with building simulation or similar applications in mind. Still, they both support the modeling process in a unique way.

At the beginning of the education in the field of the physics of dynamical systems we employ Stella as the main engine. This general purpose system dynamics tool is particularly suited to the building of models of dynamical systems. Since system dynamics is a child of early cybernetics and control engineering theory, it is not surprising that this methodology has much in common with the basic structure of physical systems (see Figure 1 at the end of the paper). Therefore, modeling the system dynamics way can be used for teaching physics, rather than being the end-result of a lengthy education in an abstract field. This synergy is used for leading students towards the appreciation of modeling early on, by way of letting them express their knowledge of the laws of physics in terms of simple dynamical models.

Engineering students in the fields of control engineering and of energy and building sciences are then led to use Simulink, which is an industrial strength tool for building models of dynamical systems placed on top of the numerics engine Matlab. Simulink supports many mathematical features of dynamical systems, both linear and nonlinear, as they are known from control engineering. The graphical representation of the physics of a system is less straight forward than with Stella (Figure 2). However, the loss of simplicity is more than balanced by the strengths of the tool, especially if we wish to treat more demanding examples.

We have found that the same initial training made up of (1) systems thinking in the realm of physics, and (2) learning how to represent simple dynamical processes, can lead to different levels of mastering the methods of modeling and simulation. Those becoming fully adept at the art of modeling initially go through the same sequence of education as do those who are content with just understanding the meaning of a model of a dynamical system and applying existing models. There exists a natural sequence leading from a system dynamics description of physical processes, through using system dynamics tools for modeling of simple systems, to modeling and simulating dynamical systems as they are found in real-life applications in building simulation. The analogy between the laws of physics and the description of dynamical systems known from control engineering leeds to a profound synergy in the educational process.

The paper gives an overview of the integrated approach to systems thinking and modeling of dynamical systems as applied to a technical system (building). Even though we should not overemphasize the importance of the software used, some of the modern tools can make all the difference in our decision whether or not to teach modeling at all (Section 2). The inherent power of the tools could not be fully employed, hōwever, without a particular way of physical systems thinking (Section3). Section 4, finally, describes some examples and areas of application, and it discusses the question of whether or not the tools can be used in real life professional applications as well.

GENERAL PURPOSE MODELING SOFTWARE FOR DYNAMICAL SYSTEMS

Advances in software and hardware development have made possible the creation of tools which we could only dream of as little as a few years ago. In the field of modeling and simulation of dynamical systems, this has led to the development of block diagram tools with user-friendly graphical interfaces. Even though it is true that the programs do not replace the fundamental knowledge and skill necessary for modeling, they facilitate the learning of this discipline, and they open new avenues to producing and employing models of dynamical systems in professional and industrial applications.

There is one class of modeling and simulation software for dynamical systems which has grown out of the methodology of system dynamics which was developed at the beginning of the sixties in order to apply the ideas of feedback and control strategies to very general applications such as industrial and urban dynamics (Forrester, 1961, 1968, 1969; Roberts et al. 1983). As far as software is concerned, the main application programs available today are Dynamo (1976), which is "industrial-strength" for system dynamics applications but does not have a simple graphical interface, Stella (1987), which is the first of the very easy to use tools, and Powersim (1993), which is very similar to Stella. Stella lets you build models of dynamical systems with only four graphical elements (Fig.1), and then supports the simulation step with the help of simple numerical methods and graphical displays. Mathematical, statistical, control, and general functions allow you to write sophisticated relations between the variables represented by the various blocks in a diagram representing the model of a dynamical system (see Table 1 for details).

Three of the four elements in a Stella diagram are fundamental building blocks. The element which looks like a rectangle represents a quantity which can be accumulated (stock), while the pipeline with valve stands for currents, fluxes, or rates of production (flow). The combination of stocks and flows normally represents a law of balance of quantities such as momentum, charge, energy, or entropy, if we restrict our view to applications in the field of physics. The third building block, the circle, is used for all other variables introduced in a model. The thin line connecting building blocks lets you transfer information from one element to another. The program allows you to build the model of a dynamical system with these four elements, with one sub-level for creating a two-stage hierarchical representation of a system.

The other category of modeling software is much more oriented toward technical and mathematical applications. Among these tools, Simulink (1993) and Extend (1988) stand out. Simulink, which builds upon the numerics engine Matlab, lets the user build models with the help of elements known from control engineering and signal processing (Figure 2; see Table 1 for a comparison with Stella). Extend, on the other hand, provides for general purpose programmable blocks which can be joined graphically to represent more complex models. The representation of a model in Simulink is much less straight forward in terms of the underlying physical structure than is a diagram produced in Stella; also, mathematical expressions are rather cumbersome to build. While we lose some simplicity in these respects, we gain much because of the basic flexibility of the tool.

In particular, the possibility of many layers of submodels, strong mathematical methods, and access to general programming, make Simulink a very respectable tool for both teaching and research and development. However, for teaching purposes only, the classical system dynamics approach and the related tools are to be preferred.

TEACHING THE MODELING PROCESS

The idea behind the training strategy presented here is that the fundamental knowledge needed by those who model dynamical systems can be appreciated by most students in the areas of architecture, engineering, or science. Put differently, this knowledge is instrumental for both the modeling process executed by professional modelers, and for learning the basic physics necessary for an appreciation of the functioning of a building. The basic ideas are known as systems physics (Burkhardt, 1987; Fuchs, 1996) and system dynamics (Forrester, 1968). Using both methodologies in an integrated approach can lead to great synergies in the learning of modeling and simulation (Simon and Fuchs, 1995). The following description of an approach to teaching building simulation applies to students in the department of mechanical engineering where we have gained most of our experience over a period of some five years. The modular nature of this approach allows it to be tailored to suit other students as well.

The most general approach to classical phenomena, namely continuum physics (Truesdell and Toupin, 1960; Truesdell and Noll, 1965), teaches us that there is a common underlying structure to the different branches of classical physics (Fuchs, 1996). Examples of applying this common description of otherwise distinct fields can be found in control engineering (White and Tauber, 1969) and in chemical engineering (Bird, Stewart, and Lightfoot, 1960). It is found that homogeneous systems (lumped systems) afford the simplest introduction to the fundamental laws underlying all physical processes, i.e. the laws of balance for physical quantities such as entropy, momentum, and energy. Learning how to apply the laws of balance is the first step in the sequence of education leading to modeling of dynamical systems, since these laws directly furnish the differential equations of the initial value problem. Combined with a system dynamics tool such as Stella, learning about accounting in dynamical systems is both easy and fun (see Figure 3). What often is considered to be a major stumbling block in the teaching of the physical sciences, now turns out to be quite simple for those who would never consider to become modelers; at the same time it provides the foundation for those who do.

The second step of the process, which all the students in the department are treated to, involves an introductory physics lab where modeling and simulation of simple dynamical systems are taught. Examples such as cooling of a body of water, one-dimensional conduction of heat through a bar, and the heating and cooling of a building represented by a single-node model (such as in Figure 1), let all students get a feeling for applications in the fields of energy engineering and building science.

The third step is taken by only about half of the students in the department who specialize in fields such as energy engineering or chemical engineering. They learn about the applied mathematics behind simulation and are introduced to using Simulink for more advanced examples of dynamical systems. Some of these students take a course on renewable energy which, in a fourth step, deepens the knowledge of how to apply the modeling process. Finally, a few students go all the way in specializing in modeling and simulation in building science as part of their project work and thesis work.

EXAMPLES AND AREAS OF APPLICATION

Examples of dynamical models used in teaching building simulation cover the entire range from small and simple to large and complex, suiting different needs for different students. The simple one-node model of a building has been mentioned before. One example, which still is rather simple, demonstrates the difference between treating the building envelope as a single node versus dividing it into a small number of finite elements. While most students will be content with applying an existing model to learn about the significance of the effect of subdividing a complex body, others will become adept at creating models of this level of complexity by themselves. This example can lead all the way to the rather demanding subject of system identification, for which there are specialized tools to be found in Matlab. Another example teaches the concept of thermal optimization of a system by looking at the entropy produced as a result of the thermal processes; specifically, a simple model of a (solar) storage wall in a building can demonstrate that there is an optimal value of the thickness of the wall (all other parameters left unchanged) for which the irreversibility becomes minimal.

Let us present in more detail an example of model construction and model deployment using Stella as the modeling tool. Even though it is extremely simple, the one-node representation of a building can be used to investigate the relative importance of changing the heat capacity of the building versus changing its insulation (Figure 1). If we imagine the walls to be built of two layers of wood or brick with a layer of mineral wool as insulating material between them, changing the thickness of the wood or brick layers will mostly change the heat capacity while increasing the thickness of the insulating layer will lead to a change of the heat transfer value only.

The model is built as follows. We should first decide upon the number of elements (nodes) the system will be divided into. Here we settle with one node. Then we have to make up our mind as to which basic physical quantity stored in the building will be employed as the stock introduced into the model. We will take energy as the extensive quantity to be accounted for. Therefore we have to place a stock (small rectangle in Figure 1) onto a new document page in Stella. Next we count the different ways energy may be added to or withdrawn from the system. If we allow for solar radiation, heat losses, and supply of heating energy from a furnace, we have to connect three flows (i.e. energy currents; thick arrows in Figure 1) to the stock already drawn. It helps if the elements introduced are given names representing the physical significance of the quantities involved. For example, it would not make much sense to call the stock "building" since we do not count buildings but energy; the name for the stock therefore should reflect the name of the abstract physical quantity used in the law of balance.

With the stock and the three flows the law of balance of energy for the one-node building has been formulated; no equations have to be written up to this point. What is left to be done is the specification of the initial value of the energy of the building (which is entered into the icon of the stock) and the three energy currents. The latter requires more work than the previous steps. For each case we have to find appropriate constitutive laws. In the modeling process, we encourage students to work their way backwards through the problem. Since the energy currents have been introduced but have been left unspecified, the proper question to ask with regard to the heat loss is the following: which factors does the energy current due to heat loss from the building to the environment depend upon? The answer probably will be this: it depends upon both the temperature of the building and the ambient temperature, and the heat transfer coefficient of the building envelope multiplied by the surface area of the building. Therefore we are led to introduce three new variables into the model which are represented by the third type of element used in Stella, namely the simple converter (circles in Figure 1). We shall label them with appropriate names. Then the three variables are connected with the flow called "energy loss" with the help of thin wires. This makes the values of the variables accessible to the energy flux. We simply enter the relation

A*(temperature - ambient_temperature)

into the flow which finishes the formulation of the law of heat loss for the building.

The model builder will realise that while the constitutive law for the energy current has been written down, the three variables involved in this law have not been specified yet. An important step in modeling the thermal processes of the building has to do with realising that the temperature of the building is a result of the energy stored and the heat capacity. While the energy stored in the building is represented by the stock, the heat capacity still has to be furnished. Again we use one of the simple variables (circle), label it with a proper name and connect its icon and that of the stock with the variable called "temperature". Next, we should introduce parameters such as thickness of different layers of the walls, specific heats of material used, densities, conductivities and heat transfer coefficients from which the values of "UA" and "heat capacity" are calculated. The procedure for connecting the various variables and parameters and for writing the correct relations is the same as that already described. Later, during the simulation step, we can change parameters as we like and perform parameter studies.

Naturally, we still have to specify the other two flows. Let us just mention the case of the flow called "supply of energy"; as you can see in the model diagram in Figure 1, it is determined in terms of "solar radiation". This particular parameter actually is a one dimensional vector of insolation values at fixed points in time. Such values can easily be introduced from spread sheets or other data files. The program interpolates the tabular data appropriately. Therefore, values of "solar radiation" are fed into "supply of energy" at the time steps for which the numerical integration of the equations is performed. With some conversion factors (which have not been introduced in the diagram in Figure 1) the rate of absorption of energy by the building is calculated according to an equation written for the flow under consideration.

Assume the model has been built. We now determine the variables for which we would like to obtain graphical or tabular representations (as a function of time) during simulation. We then proceed to specify such parameters as the time step and the method used for integration whereupon we can perform the first model runs which should help us with debugging the model. Ideally, after debugging has been successful, we would like to have some means of validating the model; however, depending on the setting of the exercise, this may be too much to ask for.

Model deployment consists of various simulation runs with parameters changed according to the purpose of our investigation. The model described here may be used to demonstrate the influence of changing the thickness of the insulating layer of the walls versus changing the thickness of the outer layers made of wood or brick. Even our simple model lets us conclude that increasing the insulation decreases the amount of energy needed for heating the building while increasing the amplitude of the temperature of the building; increasing the thickness of the brick layers, on the other hand, lets the temperature amplitude of the building decrease without considerably changing the amount of heating energy needed.

Project and thesis work provide the bulk of experience gained with the application of general purpose modeling software for applications in building science. Considering that modeling was not included previously in the education of the students at the department of mechanical engineering, we are more than pleased with the results of design studies performed by our students using such tools in recent years. They easily apply the proper programs for their particular purpose, be it simple or complex, and deliver results which otherwise might have been difficult to derive.

Application of this methodology and the general purpose software tools (especially in the case of Simulink) extends all the way to "industrial" or "professional" examples. The model and the user interface created by a student for a solar hot water system is being transferred to applications in building simulation by a local company (Meyer, 1994). What has been difficult and cumbersome for them to create with standard programming tools promises to be much simpler and more cost effective with the modern general purpose application programs for modeling and simulation of dynamical systems.

CONCLUSIONS

Possibly the most important conclusion that can be drawn from our experience is that with the help of general purpose modeling software and the system dynamics methodology the education in modeling can start earlier and can reach a wider audience than would otherwise be possible with classical programming tools. Thinking about and understanding of the physical processes taking place in a building is greatly enhanced by the graphical user interfaces of modern tools. Students who would normally not have been exposed to modeling except through results of simulations performed by others may now take at least the first few steps into the art and practice of building simulation; this will let them gain a better understanding of the work done by modelers.

The importance of the modeling tools mentioned in this paper does not stop with the educational process, however. We have found that industrial strength programs such as Matlab/Simulink can serve as the main engines for modeling and simulation in environments such as building sciences and energy engineering.

REFERENCES

Bird R.B., Stewart W.E., Lightfoot E.N. (1960): *Transport Phenomena*. Chemical Engineering Series, MacGraw-Hill, New York.

Burkhardt H. (1987): Systems physics: A uniform approach to the branches of classical physics. *Am.J.Phys.* **55**,344-350.

Dynamo (1976): *Dynamo II User's Manual*. Pugh A.L., Cambridge, Mass. MIT Press

Extend (1988): Imagine That Inc., San Jose, CA.

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

(1968): *Principles of Systems. Cambridge*. Mass. MIT Press.

(1969): *Urban Dynamics*. Cambridge, Mass. MIT Press.

Fuchs H.U. (1996): *The Dynamics of Heat*. Springer-Verlag, New York; to be published.

Meyer R. (1994): A dynamical model for a solar hot water system. Thesis, Technikum Winterthur (in German).

Powersim (1994): Powersim S.A., Bergen, Norway.

Roberts N. et al. (1983): *Introduction to Computer Simulation: The System Dynamics Approach*. Addison-Wesley, Reading, Mass.

Simon M., Fuchs H.U. (1995): System Dynamics in Engineering Education. Proceedings of the International System Dynamics Conference. Tokyo. In preparation.

Simulink (1993): The MathWorks, Inc., Natick, MA.

Stella (1987): High Performance Systems Inc., Hanover, NH.

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

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

White H.J., Tauber S. (1969): *Systems Analysis*. Saunders, Toronto.

Table 1: Comparison of Stella and Simulink for technical applications

	Stella	Simulink
Building blocks	4	virtually unlimited
Functions	mathematical, statistical, control statements	mathematical, statistical, control statements
Time-discrete systems	yes	yes
Expressions	very easy to build	rather cumbersome but very flexible representation of expressions
External data import	single column vectors of up to 2000 data points	matrices with basically unlimited number of data points
Data export	ASCII table	export to Matlab
Sub-model layers	2; strongly limited ability to handle hierarchical representation of model	many; good at hierarchical representation of model
Presentation of results	quick but limited graphs	simple and very limited graphs; sophisticated data presentation within Matlab (accessible through programming)
Block libraries	parts of other models can be copied and imported	easy to build and maintain block libraries
Sensitivity analysis	"quick and dirty"	has to be implemented by programming
User interface	easy to build but limited top layer above model for user to run a model ("flight simulator")	programmable interface in Matlab
Programming	no provision for external functions or commands	blocks can be written in either FORTRAN or C
Numerical methods	fixed-step Euler, Heun, and 4th order Runge Kutta methods	variable-step Runge Kutta methods; Gear method for stiff differential equations; recourse to all mathematical methods implemented in Matlab
Usefulness in real-life applications	useful for simple to medium size applications, including very simple one- or two-dimensional finite element models; if it is important to convince a customer of a basic dynamic feature of a building, this is the tool to use	very good for complex applications (including very simple one- or two-dimensional finite element models), including control problems; strong numerical background; suitable for research and development

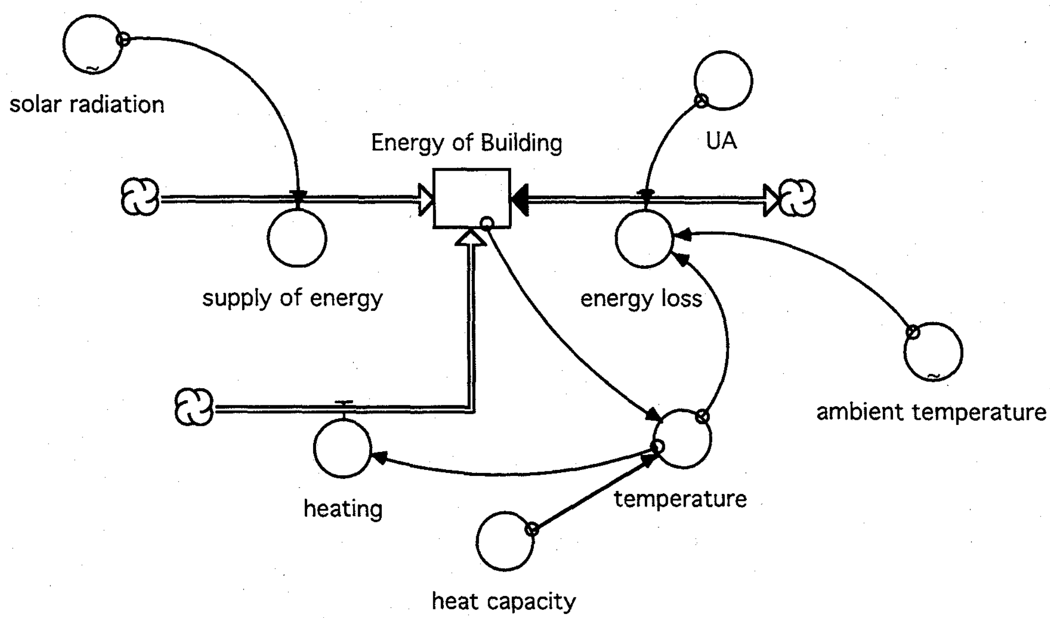


Figure 1: Primitive one-node representation of the energy balance of a building prepared in Stella. This software tool lets you build models of dynamical systems with only four elements, and then supports the simulation step with the help of simple numerical methods and graphical displays.

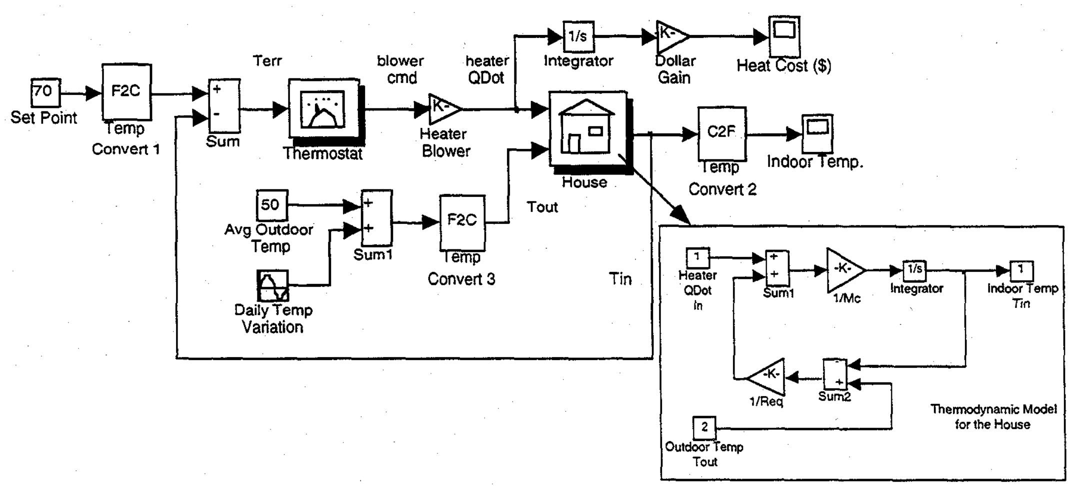
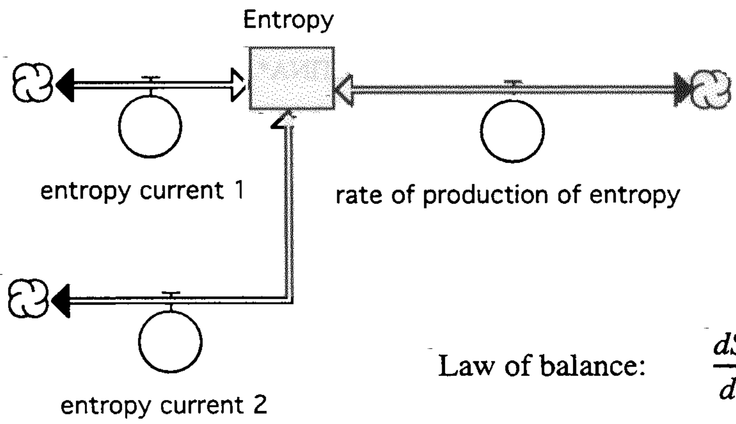


Figure 2: Model of the thermal control system of a building done with Simulink. The block called "House" contains a sub-model of the energy balance of the house, also represented as a one-node model. The example presented here is one supplied with the software package.



Law of balance: $$\frac{dS}{dt} = I_{s1} + I_{s2} + \Pi_s$$

Entropy(t) = Entropy(t - dt) + (entropy_current_1 + entropy_current_2 + rate_of_production_of_entropy) * dt

Figure 3: The Stella representation of the time dependent law of balance of entropy is assembled from a stock and some flows. The figure shows the block diagram, the formal differential equation (initial value problem), and the Euler representation of the law for later numerical solution.