

Model Generation from Physical Principles: A Progress Report

Ringo Ling
Louis Steinberg
Department of Computer Science
Rutgers University
New Brunswick, NJ 08904

ling@cs.rutgers.edu, lou@cs.rutgers.edu

Abstract

This paper describes a framework and a system for generating mathematical models (i.e. sets of equations) for analyzing physical systems. The models are derived from physical principles, and include not only models based on algebraic and ordinary differential equations (i.e. “lumped” models), but also those based on partial differential equations (i.e. “distributed” models). We are motivated by the need for analysis models to be used in designing artifacts, and focus on the domain of thermal manufacturing. Our framework involves three sequential subtasks: identify regions of interest on the artifact, determine and identify the relevant physical processes, transform the set of individual processes into equations and carry out mathematical simplification. We take the view that understanding the task of model generation is fundamental to our future research on approximate modeling in design.

Introduction

This paper describes a framework and a system for generating mathematical models for physical systems from physical principles, including not only models based on

algebraic and ordinary differential equations (i.e. “lumped” models), but also those based on partial differential equations (i.e. “distributed” models).

Our motivation for research in modeling arises from the desire to automate the process of design, in particular to automate the kind of design which involves repeatedly modifying a design and analyzing it to see if its behavior is satisfactory. In this process, it is important to base the analysis on a model of the artifact’s behavior that is as simple as possible (to allow fast analysis), but at the same time gives accurate enough results. The problem is that the appropriate model to use can change as design proceeds. E.g. a model of heat flow in a flat plate may assume the plate is much thinner than it is wide. If a modification of the design makes the plate thicker, the model may no longer apply. Thus, an automatic design system must be able to recognize when the current model no longer applies and to generate a new one as needed.

Colleagues at Rutgers have developed a system [Viswanath and Jaluria, 1990] which has a library of models and uses domain specific rules to choose among them. However, when we analyzed the rules and tried to extend the system to handle a broader class of situations, we began to face the questions, where do these models and their assumptions come from, and how do the heuristics for choosing them arise? In order to answer these questions, we have turned our attention to the question

What are the steps and knowledge involved in *generating* a mathematical model from a physical description of a problem (as opposed to selecting one from a library)?

We view the issue of model generation as fundamental to our understanding of the

*This work has been supported by NSF under Grant Number DMC-8610507, by the Rutgers Center for Computer Aids to Industrial Productivity, and by DARPA under Grant N00014-81-K-0394 and by DARPA and NASA under DARPA-funded NASA grant NAG2-645. The opinions expressed in this paper are those of the authors, and do not reflect any policies, either expressed or implied, of any granting agency.

use of approximate models in design.

The domain we have focused on is the modeling of heat transfer in thermal manufacturing processes, such as casting and heat treatment.

We have found that the process of model generation, at least in this domain,¹ can be divided into three sequential sub-tasks:

1. Identify a region of interest on the artifact,
2. Determine and identify the relevant physical processes for the region
3. Transform the set of individual processes into an equation based on the law of conservation of energy, and carry out mathematical simplification.

We have implemented a preliminary version of the system to generate mathematical models for rectangular shape objects. The paper will focus on the tasks and various kinds of knowledge used in deriving a mathematical model from a physical description of a problem. The knowledge used in the three tasks are:

- The relation between the query to be answered and the appropriate regions of interests
- Domain specific physical processes and laws
- Geometry
- A set of mathematical transformations

It should be noted that our focus is on producing a model expressed as a set of equations, and that for now we are not looking at the problems of validating the model's accuracy² or of using the model's equations to actually do analysis, e.g. by writing code to solve them numerically.

In the following sections we will first give an overview of the domain of heat transfer, then show more of how our system actually carries out tasks 1 - 3, largely by following a solution trace of an example problem, and discuss the current limits of our system and our ideas about further research.

¹We speculate that our framework applies to other domains as well, but have not yet investigated this in any detail.

²Once a model is produced its accuracy must be validated by comparing its predictions with known results, because there are no known methods for deciding what phenomena to include in the model to achieve the required accuracy, except for some rough heuristics.

In the last section we also discuss related research.

The domain: heat transfer in manufacturing

Our work on this problem has been driven by problems of designing thermal manufacturing processes, e.g. casting and heat treatment processes. In particular we have focused on issues of modeling heat flow in these processes.

In the domain of heat transfer, mathematical models are based on a fundamental physical law: the law of conservation of energy. This law says that the net heat flow into any bounded region plus the net heat generation within the region is equal to the net heat gain within the region. Because this law is so central, and because it refers to a "bounded region", models in this domain are written in terms of one or more "control volumes", specific regions of the artifact to which the conservation law is being applied.³

Other laws in this domain describe various kinds of heat transfer processes, such as conduction, convection, and radiation, giving conditions under which they happen and how the magnitude of the heat flow is related to properties of the physical system such as temperature, heat conductivity and area.

A typical model, then, is one or more equations describing the various heat flow processes in operation across the boundaries of the control volumes, describing the heat generation processes within the control volumes, and stating that energy is conserved.

Typical questions to be answered by a model include the temperature or amount of heat stored by a part of the physical system. It is often important to know not only an average or "lumped" temperature, say, but also how the temperature varies over space. E.g., in a casting process it can be important to keep the temperature gradient in the the casting below some value at every point in the casting, in order to prevent warping. This means that the kind of "lumped parameter" models used in most recent AI research on numerical modeling [Falkenhainer and Forbus, 1991] [Addanki *et al.*, 1991] are not sufficient. In these models, a parameter representing

³This is called the Eulerian approach of formulation

Input:
Dimensions and Physical Properties:
 - $\{(x, y, z) : x, y, z \text{ are dimensions}\}$,
 - $\{(k, \rho, Cp) : k \text{ is conductivity, } \rho \text{ is density, } Cp \text{ is specific-heat}\}$,
Initial and Boundary conditions:
 - $\{T(x, y, z, 0) : \text{initial temperature}\}$,
 - $\{(P^*)_i : P \text{ is boundary parameters at surface } i\}$,
 e.g. $(h, T_{env})_{y0}$, convection parameters at surface $y0$.
Query:
 - $\{(Qu, (D^*), E) : Qu \text{ is query, } D \text{ is spatial/temporal dependency, } E \text{ is allowable error}\}$
 e.g. $\{T, (x, y, z, t), 20\%\}$
Output:
 A mathematical model is a set of one or more equations.

Figure 1: Input/Output of the system

a physical quantity such as temperature is just a single number, representing the value of the quantity at a discrete point in space,⁴ and we can use ordinary differential equations. In contrast, in thermal modeling, such a parameter may represent a continuous distribution giving the temperature at each point in a region of space, leading to the need for *partial* differential equations.

The system

Input and Output

The input and output of the system is shown in figure 1. It requires three kinds of input. The first kind gives information about the dimensions of an object, and its physical properties. The second kind gives the initial conditions and the environment of the object, such as temperatures and convection coefficient parameters. For a rectangular parallelepiped, the system requires boundary parameters at six surfaces of the object. The final piece of input is the query which includes physical parameter, its spatial and/or temporal dependency requirement, and its accuracy requirement. The spatial dependency requirement indicates whether the user needs to know the spatial variation of the parameter or just a lumped value. For example, the query of

⁴Or representing an average value over some discrete component of the system.

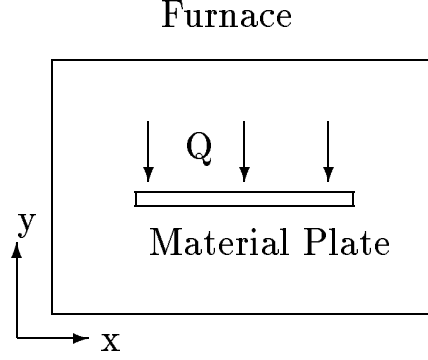


Figure 2: Heat Treatment

temperature distribution, $T(x, y, z, t)$ has the spatial dependency requirement along the directions of x, y, z in addition to its temporal dependency requirement, t .

The output of the system is a model, a set of one or more equations.

Example: Heat treatment Process

The query to illustrate the model derivation process is taken from designing a heat treatment process. In the heat treatment process, a material plate is heated up in a furnace, and maintained within a certain temperature range for material processing. The plate is clamped and insulated at two ends on the x -axis, i.e. the heat flux, $Q_{x0} = 0$ and $Q_{xL} = 0$. A heat flux, Q_{yM} is applied on top of the plate, along the y -axis, while the remaining surfaces of the plate normal to the y and z -axis are exposed to convection processes in the furnace environment, specified by the environment temperature T_{env} and the convection coefficient, h . The furnace and the plate are shown in the x , and y dimensions in figure 2.

The query is the temperature distribution in the plate, $T(x, y, z, t)$, which has a spatial dependency in x, y, z .

Algorithm and trace of example

The algorithm is shown in figure 5. The following is a trace of major steps in the algorithm that derives a model for $T(x, y, z, t)$. The algorithm also requires various kinds of knowledge, a sample of which is shown in figure 3

1. System boundary identification:

Since the query $T(x, y, z, t)$ has a spatial dependency requirement in x, y, z .

Type F: Formulation
Differential Formulation:
 $\{CV_{differential}, \Sigma CV_{boundary}\}$
Type H: Heat Transfer:
Conservation of energy: $\{\dot{E}, \Sigma Q_i\}$
Energy: $\dot{E} = \rho * V * C * \frac{dT}{dt}$
Conduction: $Q_i = -k * A_i * \frac{dT}{dX_i}$
Type G: Geometry
 $V_{differential} = dx * dy * dz$
Type M: Mathematical Transformations:
Taylor's Series expansion:
 $Q_{x+dx} = Q_x + \frac{\partial Q_x}{\partial x} dx$

Figure 3: Sample of the knowledge

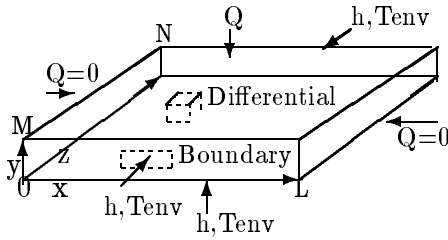


Figure 4: Distributed Formulation

The system will instantiate a differential control volume, $CV_{differential}$ within the plate, and a set of boundary control volumes, $CV_{boundary}$. The differential control volume is used to model interaction of energy processes within the plate,⁵ while the boundary control volumes are used to relate heat flux at the surfaces of the plate to the energy processes within the plate. Samples of the control volumes are shown in figure 4.

2. Instantiating relevant heat and energy processes:

For the $CV_{differential}$, we instantiate a set of energy processes $\{\dot{E}, Q_x, \dots, Q_{z+dz}\}$, where \dot{E} is energy storage, and Q_x, \dots, Q_{z+dz} are heat flux on the surfaces. The instantiation includes identifying the location and types of heat flux, i.e. conduction in this case. Similar steps are repeated for each $CV_{boundary}$.

3. Transformation and Simplification:

⁵Semantically, the model really contains an infinite number of infinitesimal control volumes, one for each point inside the plate, but syntactically we have only one volume, parameterized by its x, y, z location.

1. System boundary identification

Select and instantiate a set of control volumes CV based on knowledge type F.

2. Determining and Instantiation of processes

For each control volume CV , using the knowledge type H to:

- determine the type of processes for the CV , and
- instantiate each of them by their location.

3. Transformation and Simplification

Transform the processes into their mathematical representation using the knowledge type H, and simplification using the knowledge of types G and M, until the model explicitly contains the desired physical quantity.

Figure 5: Subtasks in Model Generation

The transformation maps the set $\{\dot{E}, Q_x, \dots, Q_{z+dz}\}$ from previous stage into a mathematical equation

$$Q_x - Q_{x+dx} + Q_y - Q_{y+dy} + Q_z - Q_{z+dz} = \dot{E}$$

Then further simplification by applying mathematical transformations, such as Taylor's series expansion, and then expanding the mathematical definitions of the terms from the domain knowledge, gives

$$-\frac{\partial Q_x}{\partial x} dx - \frac{\partial Q_y}{\partial y} dy - \frac{\partial Q_z}{\partial z} dz = \dot{E}$$

and then

$$\frac{\partial}{\partial x}(k \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(k \frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(k \frac{\partial T}{\partial z}) = \rho C \frac{\partial T}{\partial t}$$

The final model has one governing equation (PDE), which describes the temperature distribution within the plate, six boundary equations for the surface of the plate and a equation for the initial condition, figure 6.

Note that while this example resulted in a partial differential equation, our system is capable of producing ordinary differential equations or algebraic equations when these are appropriate.

Discussion

Having briefly described how our system works, we now turn to a discussion of its

Governing Equation:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = \frac{\rho C}{k} \frac{\partial T}{\partial t}$$

The Initial condition:

$$T(x, y, z, 0) = T_{init}$$

Boundary Conditions:

$$x0: 0 = k * A_{x0} * \frac{dT}{dx}$$

$$xL: 0 = k * A_{xL} * \frac{dT}{dx}$$

$$yM: Q_{yM} = k * A_{yM} * \frac{dT}{dy}$$

$$y0: k * A_{y0} * \frac{dT}{dy} = h_{y0} * A_{y0} * (T -$$

T_{env})

$$z0: k * A_{z0} * \frac{dT}{dz} = h_{z0} * A_{z0} * (T -$$

T_{env})

$$zN: k * A_{zN} * \frac{dT}{dz} = h_{zN} * A_{zN} * (T - T_{env})$$

Figure 6: Distributed Model

current status, limitations, and future directions. We also summarize the kinds of knowledge used in each subtask and discuss briefly how this work might transfer to other domains.

Status

The system described in this paper has been implemented in Common Lisp and CLOS. It was developed and validated based on discussions with three researchers in the field of thermal modeling, and examples and knowledge taken from three classical textbooks in the domain, [Incropera and DeWitt, 1985] [Jaluria and Torrance, 1986] [Arpaci, 1966]. The system runs quite quickly, taking roughly 2 seconds on a SparcStation 2 to do the example presented above.

Limitations and future directions

We now discuss the main limitations of our current system and the issues involved in extending it to deal with those limitations.

The biggest limitation of our system is that two major kinds of approximation have not yet been fully implemented. They are: approximation by order of magnitudes and approximation by looking for invariants. The order of magnitude approximation is based on two standard techniques used by expert engineers: *dimensional analysis* and *order of magnitude analysis*. This approximation can be carried out in two ways. One way is done within the last

task of model generation, after a mathematical representation has been generated. Dimensional analysis is used to transform the governing equation, i.e. the equation that represents the law of conservation of energy, into dimensionless form, and at the same time, to establish the order of magnitude scales for all the terms in the equation. Then order of magnitude reasoning is used to examine the terms and to eliminate those which are of order of magnitude less than the dominant ones in the equation.

While this way of order of magnitude approximation can be easily done for a single component object where a single governing equation is involved, the technique can become complicated when multiple governing equations are involved in a multi-components object. We have observed that expert engineers tend to employ another way of carrying out order of magnitude approximation. Their approach is to use a set of rules for pruning out less significant energy processes during the second task of model generation, i.e. the task of identifying and instantiating relevant energy processes. These rules are based on a set of physical quantities known as “dimensionless numbers.”⁶ These rules can be inferred from the same kind of order of magnitude reasoning described above, and so can be viewed as a compiled form of it, but they have the advantage that they can be applied at the stage where the physical components and processes are still represented explicitly, rather than at the stage where all we have is equations. As we argue in [Ling and Steinberg, 1990], doing this reasoning on physical entities make it much simpler, especially for complex objects, but there is a cost in generality. We are implementing both ways of doing order magnitude approximation, to compare the tradeoffs and advantages of both of them.

The second kind of approximation is based on looking for invariants in material properties and parameter values. Many kinds of approximation can be attributed to invariants in parameter values. For example, a non-linear heat conduction equation can be turned into a linear one if its conductivity is constant, instead of varying with temperature. Two dimensional partial differential equation can become one-dimensional if its temperature is constant

⁶The specific set of quantities is domain specific, but such quantities are known in a number of domains.

along one dimension. Also a transient energy process can be eliminated if its temperature is constant with respect to time. Some of these invariants can be easily identified from input data, such as the values of conductivity. Other have to be estimated by using rough calculation and heuristics, such as in estimating time and spatial variation of temperatures. We are also implementing this kind of approximation.

Another major current limit of our system is that it can handle only artifacts with a single component, e.g. the flat plate in the example above, along with its environment. It thus cannot handle problems like casting, where there are multiple components (e.g. the mold, the solidified portion of the casting, and the still-molten portion of the casting). Handling multiple components will require more complex methods of choosing control volumes. E.g., for certain approximate models, some components are not modeled at all, i.e. are not inside any control volume. It will also require more complex ways of finding relevant heat flow processes. E.g., any place where two solid components are in contact can give rise to a conduction process. Extending our system to handle multi-component artifacts is a high priority for us.

The final major limitation of our system is that it can only handle simple rectangular shapes. Extending it to handle other regular shapes (e.g. spheres) is straight forward. However, handling more complex shapes raises the issue of approximating a complex shape by a simpler one. Shape approximation is not well understood in general, but may be more tractable if approached in this kind of restricted context. We plan to put off dealing with this issue until others mentioned above have been handled.

Kinds of knowledge used

The current system uses several kinds of knowledge in the various sub-tasks. Besides the obvious, knowledge of how to choose control volumes and the specific laws of heat flow and conservation of energy, the system uses knowledge of geometry (e.g. to derive the area of a surface) and of various mathematical operations, e.g. simplification and Taylor series expansions.

Transfer to other domains

The key things that make our approach work in this domain include the existence of a strong domain theory and a conservation law that ties the individual processes together. We believe that these features are present in other domains as well, such as fluid mechanics, but we have not investigated this issue in any detail yet. While it is clear that the knowledge for instantiating physical processes is domain specific, and the knowledge for mathematical transformation is domain independent, it is not clear how general the knowledge for choosing control volumes or for filtering out irrelevant processes is.

Our system does not assume that processes can be tied to components of the artifact, the way that, e.g., the work of Falkenhainer and Forbus do [Falkenhainer and Forbus, 1991], since in our domain processes are associated with *configurations* of components (e.g. contact between solids) rather than with single components. In a domain like theirs, it would be useful to do the kind of indexing of process to component that they do.

Related Work

Both Addanki *et al* [Addanki *et al.*, 1991] and Weld [Weld, 1990] focus on model selection and switching. The work of Addanki *et al* uses graphs to represent models of physical domains, and domain specific parameter change rules to select models which resolve the conflicts between predictions and observation. Weld's work uses the domain independent technique of inter-model comparative analysis to select appropriate models. In both cases, model equations are explicitly input to the system, and they both pointed out the need of model generation and abstraction [Addanki *et al.*, 1991], [Weld and Addanki, 1990]. Our current work is a first step in that direction.

The work of Falkenhainer and Forbus [Falkenhainer and Forbus, 1991] is closest to our. They focus on generating an initial model based on the requirement of a query. A domain theory is decomposed into a set of model fragments, where each model fragment contains a description of objects, processes associated with the objects, and the assumptions governing its uses. The system deduces a set of assumptions from the requirement of a query, and uses ATMS to find a minimum set of model fragments,

such that they fulfill the requirement of the query. Their system cover wider domains involving qualitative and ordinary differential equations. Our system aims at a more narrow but complex domain, involving models of partial differential equations. They take the compositional approach of forming a model from existing pieces of model fragments, while we focus on generating a model from its physical description. Overall, we view our research as complementary to each other.

Also related to our work is that by Gelsey[Gelsey, 1989], which deals with the problem of inferring the behavior of mechanical devices like simple clocks, starting with a CAD/CAM-like model of their structure. For this task, the artifact decomposition is given and the set of physical processes is known (they are just the contact forces between parts of the device), and the primary problem is to instantiate these forces. However, both the focus, the domain, and the models are quite different from ours.

Summary

We take the view that understanding the task of model generation is fundamental to our future research on approximate modeling in design. We focused on a single but broad domain of thermal manufacturing processes, and analyzed how mathematical models of physical systems can be derived from physical principles. Using an example of a distributed formulation involving partial differential equation, we illustrate the task of model generation as a sequential subtasks of: identifying an appropriate system boundary on the artifact, determining and instantiating relevant physical processes, and transforming the set of processes into an equation, and carrying out mathematical simplification.

Knowledge used at various stages of the process include: knowledge of choosing system boundary, the domain specific physical laws, knowledge of geometry, and knowledge of mathematical transformations.

Besides discussing the tasks and types of knowledge, we also discuss two types of approximations, order of magnitude approximation and approximation by looking for invariants, and show where these kinds of approximation can be made in the system.

Acknowledgments

We would like to thank Prof. Yogesh Jaluria, R.S. Viswanath and Steve Lee for teaching us the domain knowledge, and to Saul Amarel, Tom Ellman, Don Smith, Casimir Kulikowski, and members of the CAP project for their advice and support.

References

- S. Addanki, R. Cremonini, and J. S. Penberthy. Graphs of models. *Artificial Intelligence*, 51(1-3):145-177, 1991.
- Vedat S. Arpacı. *Conduction Heat Transfer*. Addison-Wesley Publishing Co., 1966.
- Brian Falkenhainer and Ken Forbus. Compositional modeling: Finding the right model for the job. *Artificial Intelligence*, 51(1-3):95-144, 1991.
- Andrew Gelsey. Automated physical modeling. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Detroit, Michigan USA, August 1989.
- Frank P. Incropera and David P. DeWitt. *Fundamentals of Heat and Mass Transfer-2nd Ed*. John Wiley and Sons, 1985.
- Yogesh Jaluria and Kenneth E. Torrance. *Computational Heat Transfer*. Hemisphere Publishing Corporation, 1986.
- Ringo Ling and Louis Steinberg. Automated Generation of Approximate models for Thermal Casting Processes-Some Preliminary Results. In *Working Notes of the AAAI-90 workshop on Automatic Generation of Approximations and Abstractions*, 1990.
- R. Viswanath and Y. Jaluria. Knowledge-based system for the computer-aided design of ingot casting processes. In *Proceedings of the Second International Conference on Design Theory and Methodology*, Chicago, Illinois, Sept. 1990. ASME.
- Daniel S. Weld and Sanjaya Addanki. Task-driven model abstraction. Technical Report 90-02-21, Computer Science Dept, University of Washington, 1990.
- Daniel S. Weld. Approximation reformulations. In *Proceedings of Eight National Conference on Artificial Intelligence*, pages 407-412, Cambridge, Ma, 1990. AAAI.