

Model Simplification by Asymptotic Order of Magnitude Reasoning

Kenneth Man-kam Yip*
Department of Computer Science
Yale University
P.O. Box 2158, Yale Station
New Haven, CT 06520.
yip-ken@cs.yale.edu

Abstract

One of the hardest problems in reasoning about a physical system is finding an approximate model that is mathematically tractable and yet captures the essence of the problem. Approximate models in science are often constructed by informal reasoning based on consideration of limiting cases, knowledge of relative importance of terms in the model, and understanding of gross features of the solution. We show how an implemented program can combine such knowledge with a heuristic simplification procedure and an inequality reasoner to simplify difficult fluid equations.

Introduction

Many important scientific and technological problems – from life in moving fluids, to drag on ship hulls, to heat transfer in reentering spacecrafts, to motion of air masses, and to evolution of galaxies – arise in connection with fluid equations. In general, these equations form a system of coupled nonlinear partial differential equations, which presents enormous analytical and numerical difficulties.

We are interested in making computers to help scientists and engineers analyze difficult fluid problems. By this we do not mean the development of new computer technology for more machine cycles and memory nor clever numerical methods nor better turbulence models nor techniques for automatic grid generation or body definition. Advances in all these areas will no doubt enhance the applicability of direct numerical approaches to fluid problems. A thorough understanding of the physics involved, however, requires much more than numerical solutions. The present computers generate too much low-level output and that makes the process of discovering interesting flow phenomena and tracking important structures tedious and error-prone.

Our goal is to build a new generation of smart, expert machines that know how to represent – not just present – the important features of the solutions so

that they can talk about them, reason about them, and use them to guide further experiments or build simplified mathematical models. Our programs are not big number-crunchers; nor are they symbolic calculators like Macsyma. Rather we view them as models of what some scientists do when they are investigating physical phenomena. We want our computer programs to simulate how scientists analyze these phenomena; they should be able to formulate approximate models, to perform qualitative and heuristic analyses, to provide a high-level executive summary of these analyses, and to give meaningful information that helps a scientist in understanding the phenomena.

One of the most important skills in developing understanding of a physical phenomenon is the ability to construct approximate models that are mathematically tractable but yet retain the essentials of the phenomenon. The scientist must exercise judgment in choices of what idealizations or approximations to make. Making such judgement often requires an understanding of the gross features of the solution, knowledge of the relative importance of terms in the model, and consideration of limiting cases. The purpose of this paper is to demonstrate how this kind of knowledge can be embodied in a computer program to tackle the difficult problem of model approximation in fluid dynamics.

Related works in AI include research in model selection and model generation. Addanki's *graph of models* guides the selection of an appropriate model from a set of handcrafted models [Addanki *et al.*, 1991]. Weld's *model sensitivity analysis* provides an alternative but more general approach to model selection [Weld, 1992]. Falkenhainer and Forbus automate model generation by composing suitable model fragments [Falkenhainer and Forbus, 1991].

Another relevant line of work concerns order of magnitude reasoning. Raiman introduces *order of magnitude scales* to extend the power of qualitative algebra [Raiman, 1991]. Weld explores related ideas in a technique called *exaggeration* in the context of comparative analysis [Weld, 1990]. Mavrovouniotis and Stephanopoulos combines numerical and symbolic or-

*Supported in part by NSF Grant CCR-9109567.

der of magnitude relations in analyzing chemical processes [Mavrovouniotis and Stephanopoulos, 1988].

Our project differs from these works in two major aspects. First, whereas all the previous works deal with either qualitative models or models specified by algebraic or ordinary differential equations, we analyze systems of nonlinear partial differential equations (PDEs). Second, we base our programs on a theory of asymptotic order of magnitude of *functions*, which we believe is closer to what applied mathematicians or fluid dynamicists use.¹

The Task

We are interested in the task of model simplification, a part of a larger process of modeling-analysis-validation the purpose of which is to establish our confidence in the applicability of an approximate model in describing certain physical phenomenon. Model simplification takes three inputs: (1) a detailed model, (2) a description of the parameters, dependent variables, and independent variables of the model, and (3) essential physical effects to be included. Its output is one or more simplified models with constraints on parameters to represent the applicability of the models.

Detailed fluid models are usually available from standard textbooks and so are the physical meanings of parameters and variables. The description of variables is problem-dependent; it often includes their boundary values and estimated maximum order of magnitude. Knowledge of which physical effects are essential can come from experimental observations concerning the phenomenon. For instance, a model that neglects viscosity will predict zero drag on a solid body in steady flow; results diverge from physical reality.

In general, the simplified model is valid only under a range of parameter values. For instance, the approximation may require the Reynolds number to be large and conditions like this are represented by symbolic constraints among the parameters.

As our model problem, we use Prandtl's boundary layer approximation for high Reynolds number flows, which is probably the single most important approximation made in the history of fluid mechanics. For ease of exposition, we consider the case of two-dimensional, steady, incompressible flow over a flat plate (Fig. 1). The same technique will work for three-dimensional, unsteady flow over arbitrary bodies.

The detailed model is the 2D steady incompressible Navier-Stokes equations (Fig. 2). Equations (1) and (2) are the momentum equations, while (3) is the equation of continuity (or conservation of mass). The model is a system of three coupled PDEs containing three unknowns u , v , and p . The objective is to simplify the model in the limit $Re \rightarrow \infty$.

¹The asymptotic theory is also commonly used in the analysis of algorithms.

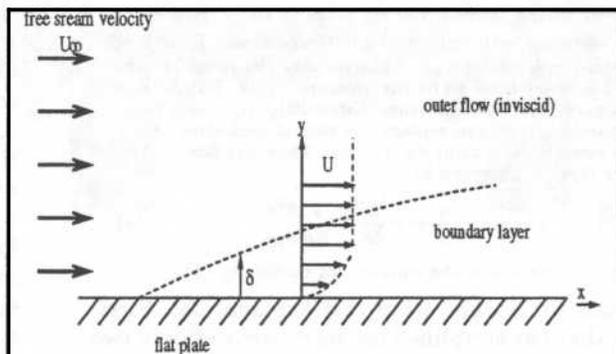


Figure 1: Boundary layer over a flat plate. The velocity gradient near the surface is large because of the no-slip condition. As one moves away from the surface, along the y -direction, the local velocity increases steadily as it approaches the free stream velocity.

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (1)$$

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \quad (2)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (3)$$

Figure 2: Two-Dimensional steady, incompressible Navier Stokes Equations: u and v are the horizontal and normal components of the velocity, p is the pressure, and Re is the Reynolds number.

Prandtl's idea is that at high Reynolds numbers viscosity remains important near the body surface even if it could be disregarded everywhere else. As long as the "no-slip" condition holds, i.e., that fluids do not slip with respect to solids, there will be a thin layer around the body where rapid changes of velocity produce notable effects, despite the small coefficient $\frac{1}{Re}$. The layer in question is called *boundary layer*.

To get a feel of the type of reasoning involved in the derivation of the boundary layer approximation, we will quote a passage, slightly edited for our purpose, from a standard fluid dynamics textbook [Yih, 1977]:

To start with we assume that δ^* , the width of the boundary layer, is small compared with L , the length of the flat plate if Re is large. That means $\delta = \frac{\delta^*}{L} \ll 1$, and the range of the boundary layer y is δ . Since u and x are all of order of unity, equation (3) states that v is of order δ . Now the convective terms in equation (1) are all of $O(1)$. A glance at the viscous terms in equation (1) reveals that $\frac{\partial^2 u}{\partial x^2} \ll \frac{\partial^2 u}{\partial y^2}$ so that the first can be neglected and the viscous terms can be replaced by $\frac{1}{Re} \frac{\partial^2 u}{\partial y^2}$. Since in the boundary layer the viscous terms are of the same order of magnitude as the inertial terms, $\frac{1}{Re} \frac{\partial^2 u}{\partial y^2} = O(1)$; this shows that:

$$Re = O\left(\frac{1}{\delta^2}\right) \quad (4)$$

To see how p varies, we turn to equation (2). Again the term $\frac{\partial^2 v}{\partial x^2}$ can be neglected since it is added to a much larger term

$\frac{\partial^2 v}{\partial y^2}$. Then all the terms involving v are of $O(\delta)$. Hence the pressure variation with respect to y in the boundary layer is of $O(\delta^2)$, and can be neglected. Thus we take the pressure outside the boundary layer to be the pressure inside. But outside the boundary layer, the pressure distribution $p(x)$ is a function of x only. So we can replace the partial derivative of the pressure term by the total derivative. Thus the flow in the boundary layer is governed by:

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{dp}{dx} + \frac{1}{Re} \frac{\partial^2 u}{\partial y^2} \quad (5)$$

to which must be added the equation of continuity (3).

Much can be learned from this explanation. First, we notice that the simplified model consists of only two equations (5) and (3), and two unknowns u and v ; the momentum equation (2) is discarded. The pressure p becomes a known boundary term to be given by the solution to the outer flow, the farfield approximation, where viscosity can be totally ignored. Second, the explanation refers to physical meanings of the terms in the equations; we have *inertia terms*, *convective terms*, *viscous terms*, and *pressure terms*. Third, the reasoning makes heavy use of order of magnitude estimate to justify the elimination of small terms. Fourth, given a few basic order of magnitude estimates (such as those of δ , u , and x), estimates for more complicated quantities involving partial derivatives are automatically inferred. In particular, it derives the important conclusion that the dependency of the pressure on y , i.e., the variation across the thin boundary layer, can be neglected at this level of approximation. Finally, by balancing the inertia terms and the viscous terms, it obtains a quantitative condition on the range of parameter values Re , equation (4), for which the approximation is valid.

Characteristics of the Problem Domain

Some Terminology

Fluids obey Newton's laws of motion. The momentum equations (1) and (2) are just examples of Newton's 2nd Law ($F = ma$). In fluid mechanics, it is customary to have the acceleration or the inertia terms written on the left hand side of the equation, while the remaining force terms on the right. See Fig. 3.

inertia forces		applied forces	
local	convective	pressure	viscous
$\frac{\partial u}{\partial t}$	$+ u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y}$	$= -\frac{\partial p}{\partial x}$	$+ \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$
$\frac{\partial v}{\partial t}$	$+ u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y}$	$= -\frac{\partial p}{\partial y}$	$+ \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right)$

Figure 3: Meaning of terms in the 2D steady incompressible Navier-Stokes Equations.

Since the motion of a fluid particle can change with both time and space, the inertia consists of two parts:

the **local acceleration** (i.e., rate of change of velocity with respect to time), and the **convective acceleration** (i.e., product of velocity and the velocity gradient).

A **steady flow** is one in which the local acceleration is zero. The applied forces on the fluid can be divided into two types: (1) **surface forces**, caused by molecular attractions, include pressure and friction forces due to viscosity, and (2) **body forces** resulting from external force fields like gravity or magnetic field. It is often convenient to define the pressure term to include gravity (i.e., $p + \rho g y$, where ρ is density of fluid, g gravitational constant, and y is the vertical coordinate). When the divergence of the fluid velocity is zero (equation (3)), the flow is called **incompressible**, which just means that the mass of fluid inside a given volume is always conserved.

The momentum equations express a balance of opposing forces on the fluid: the inertia forces keep the fluid moving steadily against the effects of pressure gradient and viscous forces. **Reynolds number** is simply the ratio between the inertia and the viscous forces; it is an indication of the relative importance of viscosity - actually the unimportance since high Reynolds numbers are associated with slightly viscous flow.

Ontology

Description of fluid motion involves a variety of quantities: (1) the fundamental quantities: time, space, and mass, (2) the usual dynamical quantities from particle mechanics such as velocity, acceleration, force, pressure, and momentum, (3) quantities that are less familiar but can be easily derived from the more basic ones: velocity gradient and pressure gradient, convective acceleration, viscous shearing forces, and turbulent stress, (4) dimensionless parameters such as Reynolds number, and (5) scale parameters, such as δ , which determine the length, time, or velocity scale of interest.

Asymptotic Order of Magnitude of Functions

Flows often vary widely in character depending on the relative magnitude of certain parameters or variables. For instance, the flow near a jet may be highly irregular, but at a large distance the mean velocity profile may become quite regular; this is the so-called farfield approximation. Another example is the Reynolds number. Small Reynolds number are often associated with laminar (smooth) flow, whereas large Reynolds numbers flow are quite erratic. So it should not be surprising that most useful approximations in fluid mechanics (and in many other branches of physics) are dependent on a limit process, the approximation becoming increasingly accurate as a parameter tends to some critical value. In our model problem, for example, we would be interested in how the boundary layer velocities u and v behave as Re becomes large.

More generally, we will consider the **asymptotic behavior** of a function $f(\epsilon)$ as ϵ approaches some critical value ϵ_0 . Without loss of generality, we can assume $\epsilon_0 = 0$, since translation ($\epsilon - \epsilon_0$) and inversion ($\frac{1}{\epsilon}$) can be used to handle any non-zero finite and infinite limiting values.

There are several ways to describe the asymptotic behavior of a function with varying degrees of precision. For instance, we could describe the limiting value $f(\epsilon)$ as $\epsilon \rightarrow 0$ qualitatively, i.e., whether it is bounded, vanishing, or infinite. Or, we could describe the limiting value *quantitatively* by giving a numerical value for the bound. But it is most useful to describe the *shape of the function qualitatively* as a limit is approached. The description uses the order symbols O ("big oh"), o ("little oh"), and \sim ("asymptotically equal") to express the relative magnitudes of two functions.

Definition 1 $f(\epsilon) = O(g(\epsilon)), \epsilon \rightarrow 0$ if $\lim_{\epsilon \rightarrow 0} \frac{f(\epsilon)}{g(\epsilon)} = K$ where K is a finite number.

Definition 2 $f(\epsilon) = o(g(\epsilon)), \epsilon \rightarrow 0$ if $\lim_{\epsilon \rightarrow 0} \frac{f(\epsilon)}{g(\epsilon)} = 0$

Definition 3 $f(\epsilon) \sim g(\epsilon), \epsilon \rightarrow 0$ if $\lim_{\epsilon \rightarrow 0} \frac{f(\epsilon)}{g(\epsilon)} = 1$

Typically, we will use a convenient set of simple functions inside an order symbol; they are called the **gauge functions** because they are used to describe the shape of an arbitrary function in the neighborhood of a critical point. Common gauge functions include the powers and inverse powers of ϵ . For example, $\sin(\epsilon) = O(\epsilon)$ as $\epsilon \rightarrow 0$. For more complicated problems, logarithms and exponentials of powers of ϵ may also be used.

The asymptotic order of magnitude must be distinguished from the numerical order of magnitude. If $f = 10^6 g$, then f and g differ by 6 numerical orders of magnitude, but they are still of the same asymptotic order. However, in a physical problem the variables are normally scaled in such a way that the proportionality constant K will be close to 1.

Below we list some useful rules of operation on order symbols:

- $O(fg) = O(f)O(g)$
- $O(f + g) = \max(O(f), O(g))$
- $O(f) + o(f) = O(f)$
- $o(fg) = O(f)o(g) = o(f)o(g)$
- If $f = O(g)$, then $\int_0^\epsilon f(t)dt = O(\int_0^\epsilon |g(t)| dt)$ as $\epsilon \rightarrow 0$.

Order relations cannot in general be differentiated. That is, if $f = O(g)$, then it is not generally true that $f' = O(g')$. However, using the definition of the total differential of a function $f(x, y)$, $df = \underbrace{\frac{\partial f}{\partial x} dx}_{df-x} + \underbrace{\frac{\partial f}{\partial y} dy}_{df-y}$

where $df-x$ and $df-y$ are the partial differentials, we can derive some useful rules involving partial derivatives:

- $O(\frac{\partial f}{\partial x})O(dx) = O(df-x)$
- $O(\frac{\partial f}{\partial y})O(dy) = O(df-y)$

$$3. O(df) = \max(O(df-x), O(df-y))$$

Theory of Simplification

The basic idea in simplification is to identify small terms in an equation, drop these terms, solve the simplified equation, and check for consistency. But this does not always work. Consider the following simple polynomial:

$$3\epsilon^2 x^3 + x^2 - \epsilon x - 4 = 0$$

in the limit $\epsilon \rightarrow 0$. We might naively drop the cubic and the linear terms because their coefficients are small. But if we do that, we only get two roots $x = \pm 2$, losing the third root. Thus, the process of simplification leads to a loss of important information.

What went wrong? The problem is that terms that appear small are not really small. The missing root depends inversely on ϵ in such a way that the cubic term is not negligible even its coefficient becomes small. To fix this problem, we introduce three concepts: an **undetermined gauge**, a **significant gauge**, and a **maximal set**. To begin, we will assume $x = O(\epsilon^n)$ where n is still undetermined - hence the name undetermined gauge. The order of each term is then:

$$\underbrace{3\epsilon^2 x^3}_{O(\epsilon^{3n+2})} + \underbrace{x^2}_{O(\epsilon^{2n})} - \underbrace{\epsilon x}_{O(\epsilon^{n+1})} - \underbrace{4}_{O(1)} = 0$$

To determine the relative importance of terms, we use the heuristic that we only retain the smallest number of terms that will balance the equation. Since we must allow the situation where two or more terms may have the same asymptotic order, we group terms into equivalence classes by the relation \sim . A maximal set is any such class that is not smaller than any other classes. As an example, the cubic polynomial above has four maximal sets each containing one term. The heuristic can then be stated as follows:

Heuristic of minimal complication (or Method of Dominant Balance):

If the equation has two or more maximal sets, balance two of them; these two maximal sets are called dominant. Assume the remaining sets are negligible. Self-consistent choices of dominant maximal sets correspond to significant simplified equations.

Applying this heuristic to the polynomial, we get six cases to consider. For instance, one possibility is that the first two terms are dominant, i.e., $\epsilon^2 x^3 \sim x^2 \gg \epsilon x, 4$. Equating the two undetermined gauges, we get $3n + 2 = 2n$ and this implies $n = -2$. The remaining terms are $O(\epsilon^{-1})$ and $O(1)$, which is consistent with the assumption that the first two terms are dominant. So this possibility is included. On the other hand, if we assume $\epsilon^2 x^3 \sim \epsilon x \gg x^2, 4$, we get $n = -\frac{1}{2}$. But then $x^2 = O(\epsilon^{-1}) \gg O(\epsilon^{\frac{1}{2}})$, violating the assumption that it should be much smaller than the first term. This possibility must be excluded. A similar analysis

shows that only one more possibility, when the second and fourth terms are dominant, i.e., $n = 0$, is self-consistent. So the heuristic concludes that we should consider *two* simplified polynomials:

$$3\epsilon^2 x^3 + x^2 = 0 \Rightarrow x \sim \frac{1}{3\epsilon^2}$$

and

$$x^2 - 4 = 0 \Rightarrow x \sim \pm 2$$

The values of ϵ^n for which we get self-consistent dominant maximal sets are called **significant gauges**. The balancing of the dominant maximal sets produces simplified equations that correspond to qualitatively significant asymptotic behaviors.

Implementation: The Details

Our method has two main parts: (1) a preprocessor, which given the input specification of a model, creates internal representations of quantities, equations, and a constraint network connecting the quantities, and (2) a model-simplifier, which finds all the self-consistent approximate models by the heuristic of minimal complication. The model-simplifier relies on three procedures – a constraint propagator, a graph searcher, and an inequality bounder – to determine the order of magnitude of quantities and their relationships. We describe each of these five pieces in turn.

The Preprocessor

The problem specification is defined by the macro `defmodel`, which takes a name, a list of quantity descriptions, the momentum and continuity equations in infix form, relations defining external pressure and free stream velocities, and a list of estimated orders of magnitude.

```
(defmodel prandtl-boundary-layer-with-pressure-gradient
  (with-independent-variables
    ((x :lower-bound 0 :upper-bound 1
        :physical-features '(space streamwise))
     (y :lower-bound 0 :physical-features '(space transverse))))
  ...
  ;;similar descriptions for U, V, P, Re, etc.;;
  ...
  (with-essential-terms
    (viscous inertia)
    (with-equations
      ((streamwise-momentum-equation
        (U * (d U / d x) + V * (d U / d y)
         = - (d P / d x) + (d2 U / d2 x) / Re + (d2 U / d2 y) / Re))
       (transverse-momentum-equation
        (U * (d V / d x) + V * (d V / d y)
         = - (d P / d y) + (d2 V / d2 x) / Re + (d2 V / d2 y) / Re))
       (continuity
        ((d U / d x) + (d V / d y) = 0)))
      (with-relations
        (constant U 1)
        (constant x 1)
        (constant y 'delta)
        (constant P0 1))))))
```

Quantities

Quantities are represented by CLOS objects. They are divided into four types: (1) independent variables (space and time), (2) dependent variables (e.g.,

pressure, velocity), (3) controllable parameters (e.g., Reynolds number), and (4) scale parameters (e.g., length scale δ). Each quantity has slots for its upper bound, lower bound, boundary values, physical features, and relations which other quantities. A dependent variable contains additional information about its dependency on the independent variables. For example, the dependent variable U depends on both x and y .

The input specifies nine quantities – $x, y, U, V, U\text{-inf}, P0, P, Re$, and δ . But a total of 60 quantities will be created. The reason is that for each dependent variable, quantities corresponding to its total differential, partial differentials, and derivatives are also automatically generated. For instance, the dependent variable U generates 5 additional quantities: $dU, dU\text{-}x, dU\text{-}y, \frac{\partial U}{\partial x}$, and $\frac{\partial U}{\partial y}$. Quantities are also generated for each term in the equations and relations. An example would be the dependent variable $d2Udx2/Re$ corresponding to the viscous term $\frac{1}{Re}(\frac{\partial^2 u}{\partial x^2})$.

Input quantities have associated physical features such as space, velocity, and pressure. These features are used to determine the physical meaning of derived quantities by simple rewrite rules. For instance, a velocity quantity differentiated by a space quantity gives a velocity-gradient quantity. The physical meaning of a term in the equation is determined in a similar fashion. For example, a term that is the product of a velocity quantity and a velocity gradient represents the convective inertia term.

A Constraint Language

Equations involving quantities are represented as constraints so that when all but one quantities are known the value of the remaining one can be computed in terms of the others. Our constraint language has 6 primitives:

1. The equality constraint, (`== q1 q2`), asserts that $O(q1) = O(q2)$. Example: the continuity equation (3) is represented by (`== dudx dvdy`).
2. The multiplier constraint, (`multiplier q1 q2 q3`), specifies that the quantities $q1, q2$ and $q3$ must be related by the equation $O(q1) \times O(q2) = O(q3)$. Example: (`multiplier u dudx ududx`).
3. The maximum constraint, (`maximum q1 q2 q3`), specifies that $O(q3) = \max(O(q1), O(q2))$. Example: (`maximum du-x du-y du`).
4. The variation constraint, (`variation f x df-x`), captures the inference that when the partial differential of a function $f(x,y)$ with respect to x is much less than the value of f at its outer boundary, then f is asymptotically equal to its boundary value. Symbolically, $df\text{-}x = o(f_0) \Rightarrow O(f) = O(f_0)$, where f_0 is the value of f at its outer boundary in the x -direction.
5. The total-variation constraint, (`total-variation f df`), specifies: $O(df) = O(\text{upperbound}(f) - \text{lowerbound}(f))$.
6. The constant constraint, (`constant q v`), just says that $O(q) = v$.

The constraint language allows simple inferences about quantities to be made. For instance, using the continuity equation (3) and the known order of magnitudes for the quantities U , x , and y , the value for V is automatically deduced.

Qualitative Order Relations

An important type of inference is the determination of the ordering relationship between two quantities. For instance, in order to drop a term A , the system has to show that A is much smaller than another quantity B in the equation. For models involving a few scale parameters, such as our model problem, the relationship can be determined by relatively simple algebraic manipulations. But for quantities involving three or more scale parameters, the algebra can be quite complicated.

A simpler inference technique is to represent the order relationships explicitly in a directed graph whose nodes are quantities and edges are labeled order relations, and to use a breadth-first search to find paths between quantities. The idea is similar to Simmons' graph search in a quantity lattice [Simmons, 1986], but we generalize it to include symbolic factors in the order relations. Let's look at an example (Fig. 4a). We have 4 quantities: A , B , C , and D . Assume δ is a small parameter. The following relations are also known: (1) $O(A) = O(B)$, (2) $O(B) = \delta O(D)$, and (3) $O(A) = \delta O(C)$. To show that $O(C) = O(D)$, we find the shortest path between them, collecting the symbolic factor of each edge of the path. The symbolic factors are divided into two groups: the \ll -factors, and the \gg -factors depending on whether the edge is labeled \ll or \gg . In the example, the \ll -factors consists of one factor δ , while the \gg -factors consists of one factor $\frac{1}{\delta}$.

The inference procedure can also handle partial information. For instance, in the graph shown in Fig. 4b, it will correctly conclude that $E \gg H$ even it is not told what the symbolic factor of edge $F \gg G$ is.

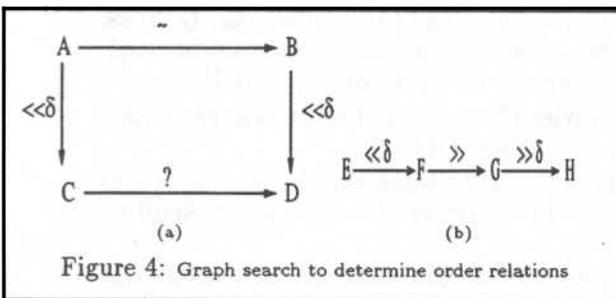


Figure 4: Graph search to determine order relations

Inequality Bounder

The constraint propagator and the graph searcher are fast but they cannot determine more subtle ordering relationships. For instance, given $\delta^2 = O(\frac{1}{Re})$ and $\delta \ll 1$, they can't deduce that $\frac{1}{Re} \times \frac{1}{\delta} \ll 1$. This problem in its general form is equivalent to the satisfiability of a set of inequality constraints. To solve this

problem, we use a version of the sup-inf bounding algorithm first proposed by [Bledsoe, 1975] and extended by [Brooks, 1981] and [Sacks, 1987] to deal with non-linear inequalities. Our algorithm is simpler because there is no need to deal with nonmonotonic functions such as the trigonometric functions.

Simplification Algorithm

The purpose of the simplification algorithm is to search for all self-consistent simplified models corresponding to a detailed input model. A simplified model is self-consistent if the terms neglected are consistent with the dominant balance assumptions, and it contains the essential terms specified by the input. The algorithm determines the maximal sets for each momentum equation, balances all possible pairs of maximal sets, and eliminates the inconsistent ones. It terminates when each momentum equation has only one maximal set.

The principal steps of simplification are:

1. If the model has no unsimplified momentum equation, then return the model.
2. Otherwise, pick the first unsimplified momentum equation and consider all possible pairwise dominant balances.
3. Propagate the effects of the dominant balance and record any assumptions made on parameters due to the balance.
4. If the resulting model is self-consistent, call simplification recursively on it. Otherwise, return nil.

The algorithm will terminate because during each call of simplification, the number of maximal sets is reduced by at least one. So each recursive call will return either a simplified model or nil if the partially simplified model is not self-consistent.

Performance Trace

The following script shows how the program produces the boundary layer approximation for our model problem. The problem generates 60 quantities and 65 constraints; it takes about 60 secs real time on a Sparc 330.

The program builds model-1 according to the input description. Each momentum equation has three maximal sets. The program simplifies the transverse momentum equation by balancing its maximal sets; there are three possible balances. The first choice - balancing viscous stress and pressure gradient - is not consistent.

```
> (search-simplifications *model*)
```

```
Making <MODEL-2: PRANDTL-BOUNDARY-LAYER> from
<MODEL-1: PRANDTL-BOUNDARY-LAYER>...
Balancing two terms:
D2VDY2/RE (VISCOUS STRESS TRANSVERSE)
DPDY (PRESSURE-GRADIENT)
in TRANSVERSE-MOMENTUM-EQUATION
with 1 parameter assumption:
```

(<< RE (^ DELTA -2))

The model is not self-consistent because the simplified equations do not contain the essential INERTIA term.

The second choice - balancing viscous stress and inertia - generates a consistent model model-3. Since model-3 is not completely simplified, the program goes on to simplify its streamwise equation, which now has two maximal sets. So there is only one balancing choice; the result is a consistent model model-4. The program also finds the correct condition on the Reynolds number.

Making <MODEL-4: PRANDTL-BOUNDARY-LAYER> from <MODEL-3: PRANDTL-BOUNDARY-LAYER>...

Balancing two terms:

D2UDY2/RE (VISCIOUS STRESS TRANSVERSE)

DPDX (PRESSURE-GRADIENT)

in STREAMWISE-MOMENTUM-EQUATION

with 1 parameter assumption:

(= RE (^ DELTA -2))

<MODEL-4: PRANDTL-BOUNDARY-LAYER> is self-consistent.

The final choice of balance for the transverse equation is inconsistent. Let's check that model-4 has the correct boundary layer equations (equations (5) and (3)):

> (model-simplified-equations model-4)

```
((U * (D U / D X)) + (V * (D U / D Y)) =  
  - (D P / D X) + ((D2 U / D2 Y) / RE))  
((D U / D X) + (D V / D Y) = 0)
```

Evaluation

The program has been tested on several problems including ODEs and PDEs representing flows in turbulent wake and turbulent jet. The turbulent wake problem, for instance, has 89 quantities and 112 constraints; it takes the program about 90 secs real time to find two simplified models.

When does the simplification heuristic fail?

There are equations for which balancing two maximal sets does not give any self-consistent approximations. For instance, the ODE $\frac{dy}{dx} - \frac{y}{x} = \frac{\cos x}{x^2}$ requires a 3-term balance because all the pairwise balances are inconsistent. Our algorithm incorporates a systematic search starting from 2-term balance until a self-consistent model is found.

How good are the approximate models?

There is no simple answer to this question. It is known that solutions to a self-consistent approximate model derived by dominant balances can be grossly inaccurate. A simple example is an ill-conditioned set of linear algebraic equations, in which a small change in the coefficients can lead to a large change in the solution vector. The situation for PDEs is much worse because, except in rare cases, it is not known whether the approximate model has a solution at all or whether the solution if exists will be unique. The strongest claim one can made seems to be this: An approximate model

that is *not* self-consistent is certainly a poor approximation. In practice, an approximate model is validated by subjecting its predictions to experimental and numerical checks. In fact, there still exists no theorem which speaks to the validity and accuracy of Prandtl's boundary layer approximation, but ninety years of experimental results leave little doubt of its validity and its value.

Conclusion

We have demonstrated how a heuristic simplification procedure can be combined with knowledge of asymptotic order of functions, relative importance of terms, and gross physical features of the solution to capture certain aspects of the informal reasoning that applied mathematicians and fluid dynamicists use in finding approximate models - informal because the approximation is done without firm error estimates. The key to the simplification method is to examine limiting cases where the model becomes singular (i.e., when the naively simplified model has a different qualitative behavior from the original model). This idea of simplification by studying the most singular behaviors is very general: it comprises the core of many powerful approximation and analysis techniques that have proven to be extremely useful in reasoning about behaviors of complicated physical systems.

References

- Addanki, S; Cremonini, R; and Penberthy, J.S. 1991. Graphs of models. *Artificial Intelligence* 51.
- Bledsoe, W.W. 1975. A new method for proving certain presburger formulas. In *Proceedings IJCAI-75*.
- Brooks, R.A. 1981. Symbolic reasoning among 3d models and 2d images. *Artificial Intelligence* 17.
- Falkenhainer, B. and Forbus, K.D. 1991. Compositional modeling: finding the right model for the job. *Artificial Intelligence* 51.
- Mavrovouniotis, M.L. and Stephanopoulos, G. 1988. Formal order-of-magnitude reasoning in process engineering. *Computer Chemical Engineering* 12.
- Raiman, Olivier 1991. Order of magnitude reasoning. *Artificial Intelligence* 51(1).
- Sacks, Elisha P. 1987. Hierarchical reasoning about inequalities. In *American Association for Artificial Intelligence*.
- Simmons, Reid 1986. Commonsense arithmetic reasoning. In *Proceedings AAAI-86*.
- Weld, D.S. 1990. Exaggeration. *Artificial Intelligence* 43.
- Weld, D.S. 1992. Reasoning about model accuracy. *Artificial Intelligence* 56.
- Yih, Chia-shun 1977. *Fluid Mechanics*. West River Press.