A SERIES OF CLASS NOTES FOR 2005-2006 TO INTRODUCE LINEAR AND NONLINEAR
PROBLEMS TO ENGINEERS, SCIENTISTS, AND APPLIED MATHEMATICIANS

DE CLASS NOTES 1

A COLLECTION OF HANDOUTS ON
FIRST ORDER ORDINARY DIFFERENTIAL EQUATIONS (ODE's)

CHAPTER 4

Applied Problems,

Mathematical Modeling,

Mathematical Problem Solving,

and the Need for Theory

ODE's-I-4 REVIEW OF THE STEPS IN SOLVING
Handout #1 AN APPLIED MATH PROBLEM Professor Moseley

The need to develop a mathematical model begins with specific questions in a particular application area that the solution of the mathematical model will answer. Often the mathematical model developed is a mathematical "find" problem such as a scalar equation, a system of linear algebraic equations, or a differential equation. Finding all solutions of an ODE is a "find" problem. We wish to find all functions in a particular **function class** that satisfy the ODE. Usually there are an infinite number of such solutions parameterized by an integration constant. The function class we pick is the set Σ for the mathematical "find" problem where we look for solutions. Adding an initial condition gives an IVP with one solution so that this problem is **well posed**. We review the five steps to develop and solve any **applied math** or **application** problem, add three more, and apply the process to a simple autonomous linear model (IVP).

Step 1: UNDERSTAND THE CONCEPTS IN THE APPLICATION AREA. In order to answer specific questions, we wish to develop a mathematical model (or problem) whose solution will answer the specific questions of interest. Before we can build a mathematical model, we must first **understand** the concepts needed from the **application area** where answers to specific questions are desired. Solution of the model should provide answers to these questions. We start with a description of the phenomenon to be modeled, including the "laws" it must follow (e.g., that are imposed by nature, by an entrepreneurial environment or by the modeler). Recall that the need to answer questions about a ball being thrown up drove us to Newton's second law, F=MA. We also need to list all assumptions made. Also a list of the nomenclature developed should be given. A sketch which helps you to visualize the process is very helpful.

Step 2: UNDERSTAND THE MATHEMATICAL CONCEPTS NEEDED. In order to develop and solve a mathematical model, we must first be sure we know the appropriate mathematics. For this course, you should have previously become reasonably proficient in **high school algebra** including how to solve **algebraic equations** and **calculus** including how to compute **derivatives** and **antiderivatives**. We are developing the required techniques and understanding of **differential equations**. Our models will be initial value problems (IVP's) which, as we have said, are "find" problems. Additional required mathematics after first order ODE's (and solution of second order ODE's by first order techniques) is **linear algebra**. All of these must be mastered in order to understand the development and solution of mathematical models in science and engineering.

Step 3. DEVELOP THE MATHEMATICAL MODEL. The model must include those aspects of the application so that its solution will provide answers to the questions of interest. However, inclusion of too much complexity may make the model unsolvable and useless. To develop the mathematical model we use **laws** that must be followed, **diagrams** we have drawn to understand the process and **notation and nomenclature** we developed. Investigation of these laws results in a **mathematical model**. In this chapter our models are Initial Value Problems (IVP's) for a first order ODE that is a rate equation (dynamical system). This is indeed a mathematical "find"

problem. We wish to find a particular function in a prescribed **set of functions** with a common domain that satisfies the ODE and the IC. If the process evolves in time, we choose t as our independent variable and usually start it at t = 0. For our generic one state variable problem, we use u as our dependent variable (i.e., state variable).

MATHEMATICAL MODEL: In mathematical language the general (possibly nonlinear) model may be written as

$$\text{ODE} \qquad \frac{du}{dt} = f(t,u) \tag{1}$$

IVP

$$\text{IC} \qquad u(0) = u_0 \tag{2}$$

For specific applications, finding f(t,u) is a major part of the modeling process. For many (but not all) of the applications we investigate, the model is the simple linear **autonomous** (time-independent) model with one state variable given by

$$\text{ODE} \qquad \frac{du}{dt} + k\,u = r_0 \tag{3}$$

IVP

$$\text{IC} \qquad u(0) = u_0. \tag{4}$$

The parameters r_0, k and y_0 as well as the variables u and t are included in our nomenclature list.

Nomenclature

u = quantity of the state variable (the dependent variable),
t = time (the independent variable)
r_0 = the rate of flow for the source or sink (parameter number 1)
k = constant of proportionality (parameter number 2)
u_0 = the initial amount of our state variable (parameter number 3)

The model is specific in that we have selected a form for f(t,u). It is **general** in that we have not explicitly given the parameters r_0, k or u_0. These **parameters** are either given or found using specific (e.g., experimental) data. However, their values need not be known to solve this linear autonomous model. The model is linear since $f(t,u) = k\,u + r_0$ with $p(t) = k$ and $g(t) = r_0$. It is autonomous since $f(t,u) = k\,u + r_0$ is not dependent on time t. Thus if $k \neq 0$, then $u = -r_0/k$ (the zero of $f(u) = k\,u + r_0$) is an **equilibrium solution** of the system. (The constant solutions of the autonomous equation $\frac{du}{dt} = f(u)$ are the zeros of f(u).)

Step 4: SOLVE THE MATHEMATICAL MODEL Once correctly formulated, the **solver** of the mathematical model can **rely completely on mathematics** and need not know where the model came from or what the Nomenclature stands for. Solution of the model requires both practical ("how to") skills and theoretical ("why") skills.

For the general linear model, we can obtain a general formula for its solution. Since $p(t) = k$, we have $\int p(t)\,dt = \int k\,dt = kt + c$. Letting $c = 0$, we obtain $\mu = e^{kt}$. Assuming $k \neq 0$, we obtain the following sequence of **equivalent equations** for the function u:

$$\frac{d(u\,e^{kt})}{dt} = r_0\,e^{kt}, \quad u\,e^{kt} = \int r_0\,e^{kt}dt = \frac{r_0}{k}e^{kt} + c, \quad u = \frac{r_0}{k} + c\,e^{-kt}. \tag{5}$$

Applying the initial condition we obtain the following sequence of equivalent algebraic equations for the scalar c:

$$u_0 = \frac{r_0}{k} + c, \qquad c = u_0 - \frac{r_0}{k}. \tag{6}$$

Hence

$$u = \frac{r_0}{k} + \left(u_0 - \frac{r_0}{k}\right) e^{-kt}. \tag{7}$$

is the general solution (i.e., a formula) for the model. If specific data is given, we can insert it into our formula. Note that if $u_0 = r_0/k$, then u is the equilibrium solution $u = r_0/k$. If $k > 0$, then all solutions approach the equilibrium solution so that it is **stable**. If $k < 0$, then all solutions diverge from the equilibrium solution so that it is **unstable**. We wish to extend this model to n state variables and indeed to an abstract state space.

Step 5: INTERPRETATION OF RESULTS. Interpretation of results can involve lots of things including the analysis for stability given above. In the current context where the general model has been solved, it usually means insert the specific data given in the problem into the formula and **answer the questions** asked with regard to that specific data. This may require additional solution of **algebraic equations**, for example, the **formula** that you derived as the general solution of the IVP. However, some applications may involve other equations. The term general solution is used here since arbitrary values of k, r_0, and y_0 are used. Recall that the term general solution is also used to indicate the (infinite) family of functions which are solutions to an ODE before a specific initial condition is imposed. We could argue that since the initial condition is arbitrary, we really have not imposed an initial condition, but again, general here means not only an arbitrary initial condition, but also an arbitrary value of k and r_0.

GENERAL AND SPECIFIC MODELS Once a general model has been formulated and solved, it can be applied using specific data. Alternately, the model can be written directly in terms of the specific data and then solved (again). If a general solution of the model has been

obtained, this is redundant. However, writing a specific model and resolving provides much needed practice in the process of formulating and solving models and hence is useful in understanding these processes. It is sometimes useful to remember a general model and its solution (e.g. the quadratic formula as the solution of the general quadratic equation), but this obviously can not be done for all general models. However, solutions to general models can be programmed for use by those not interested in their derivation. On the other hand, specific data may simplify the solution process and the formulas obtained. It may be easier to solve a simple problem with specific data rather than try to apply a complicated formula resulting from a complicated model.

Repeating, it is acceptable (and indeed desirable since it gives practice in formulating and solving models) to formulate and solve a model using specific data. The advantage of formulating and solving a model in a general context is that the solutions can be recorded in textbooks in physics, biology, etc. and programed on personal computers for those not interested in learning to solve differential equations. However, if the model assumptions change, a new model must be formulated and solved. Practice in formulating and solving specific models will help you to know when a different model is needed and in what generality a model can reasonably be developed. General models are useful when their results can be easily recorded or can be easily programmed. On the other hand, trying to use the results of a complicated model can unduly complicate a simple problem.

MORE STEPS IN MODELING. For a complicated model, the above process generally requires more that one person and is usually interdisciplinary in nature. Three additional steps are often needed to complete the process. These can be iterated.

6. Verification and Evaluation of the Model. For example by comparison with experimental results.
7. Implementation of the Model. For example, providing a user-friendly computer environment for use by non-experts.
8. Maintenance and Updating of the Model. For example, extending the model to cases not previously covered.
9. Iteration of all of the previous steps.

Recall that in Chapter 0-2 we considered math problems that fall into three categories:

1. Problems with an established algorithm for solution (e.g. computational problems). Such problems will be referred to as **evaluation problems**. They ask the question "How to find?" or "How do we compute?". Students can be trained (or train themselves by doing homework) to carry out these procedures. However many of these can be programed on a computer which can "get the answer" much faster and with far more accuracy then any human.
2. Problems defined by equations, inequalities or other properties. Such problems will be referred to as **find or locate problems**. They ask the question "If any, which ones?" There may or may not be a "How to find" algorithm associated with the problem. If there is, it can be applied (or appropriate software used). If not, the problem becomes developing such an algorithm. This may begin with showing that the problem is **well posed**, that is, showing that there is exactly one solution. If the solution algorithm requires an infinite number of steps, we need the concept of an **approximate solution**.
3. Theory problems. Such problems will be referred to as **think problems**. They ask the question "Why?" Why does a particular algorithm work for one problem, but not for a similar problem? What is the set of problems that a particular algorithm does work for and why? How can we develop solution procedures for all problems of interest. These results are often given in the development of a mathematical theory using a definition/theorem/proof format.

Learning to **solve evaluation problems** means training oneself to apply known processes or algorithms to particular examples. This may mean knowing all steps in a complicated process or simply substituting specific data into a known formula such as the quadratic formula. At the other extreme in problem solving is the **development of a mathematical theory** which may then lead to the development of algorithms for solving find problems (which then become evaluation problems). Theory development requires an understanding of what is already known (i.e. what has been proved) and hence an ability with proofs. We are considering problems (of the type useful to engineers, scientist, and applied mathematicians) between these two extremes by examining a framework which generalizes the problem of solving scalar equations; that is, we consider **find or locate problems**. This framework assumes in the problem formulation that you understand what is meant by solving an evaluation problem (i.e., that you can train yourself to carry out specific processes), but not that you can write proofs or develop a mathematical theory. Differential equations and initial value problems (IVP's) fall into this framework.

We say that a problem, call it Prob, is well-formulated in a mathematical or set theoretic sense if:

1. There is a clearly defined set, call it Σ, where, if there are any, we will find all **solutions** to the problem.
2. There is a clearly defined property or condition, call it C, that the solution elements in Σ and only the solution elements satisfy.

There is some confusion as to what is meant by **the solution of an evaluation problem**. **The solution process or algorithm** is sometimes referred to as **the solution** whereas sometimes **the answer obtained** is referred to as **the solution**. In our framework, **a solution** is an element in Σ

that satisfies the property C(s). Thus the solution set is $S = \{s \in \Sigma: C(s)\}$ and **the solution process** is whatever algorithm is used to obtain an explicit description of S. Since Σ and C define the problem Prob, we let $Prob(\Sigma,C) = \{s \in \Sigma: C(s)\}$ and think of $Prob(\Sigma,C)$ as an implicit description of the **solution set**. We then use $Soln(\Sigma,C)$ to mean the explicit description of the solution set obtained by the solution process. Since as sets we have $Prob(\Sigma,C) = Soln(\Sigma,C)$, for brevity in working examples we usually just let $S = \{s \in \Sigma: C(s)\} \subseteq \Sigma$ be the solution set during the solution process.

If there is a clearly defined and implementable algorithm to check the condition C(s) for any given element $s \in \Sigma$ so that we may determine if it is indeed a solution to the problem, we say that solutions to the problem are **testable** (and that the problem Prob is testable). We denote this algorithm to test C(s) for possible solutions by T so that the operation T(s) results in a yes if s is a solution and in a no if s is not a solution. Thus the collection of elements s in Σ such that T(s) results in a yes is the solution set S for the problem Prob defined by the set Σ and the property C. The need for clearly defining the set Σ is illustrated by the equation $x^2 + 1 = 0$. The existence of a solution depends on whether we choose the real numbers **R** or the complex numbers **C** as the set which must contain the solution. Problems requiring the solution(s) to equations provide a testable algorithm T that defines a property C.

Normally Σ is large or infinite (e.g. **R** and **C**) so that it is not possible to use the algorithm T to test each element in Σ individually. Problems where Σ is small enough so that a check of its elements by hand is possible are considered to be trivial. On the other hand, some problems where Σ is large but not to large (e.g. Which students at a university have brown eyes?) yield to the technique of testing each element in Σ by using computers and data bases.

Examples of "find" problems were given in Chapter 0-2. We considered scalar algebraic equations where we looked for the unknown variable in an **algebraic field** such as **Q**, **R**, or **C**.. We considered not just a single scalar equation, but a system of scalar equations. For clarity, we restricted our attention to linear systems of the form $A\vec{x} = \vec{b}$, where A is a matrix, $\vec{x}$ and $\vec{b}$ are "column" vectors in a **vector space** such as $\mathbf{Q}^n$, $\mathbf{R}^n$, or $\mathbf{C}^n$ and $A\vec{x}$ is defined by matrix multiplication. For example, for two equations in two unknowns, the set Σ is the set of **ordered pairs** $\Sigma = \{\vec{x} = [x,y]^T ; x,y \in \mathbf{R}\} = \mathbf{R}^2$. (We use the transpose notation $\vec{x} = [x,y]^T$ to indicate that $\vec{x}$ is a "column" vector.) Possible solutions are no longer numbers, but ordered pairs which we refer to as (column) vectors. The solution set is $S = \{\vec{x} = [x,y]^T \in \mathbf{R}^2: A\vec{x} = \vec{b}\}$. The test algorithm $T(\vec{x})$ is effected by multiplying the matrix A by $\vec{x}$ and checking to see if this gives the vector $\vec{b}$. Similar to scalar equations, we define the **operator** $F(\vec{x}) = A\vec{x} - \vec{b}$ and reformulate our problem as the "vector" equation $F(\vec{x}) = \vec{0}$ so that $S = \{\vec{x} = [x,y]^T \in \mathbf{R}^2 : F(\vec{x}) = 0\}$. In addition to systems of equalities, the framework also includes systems of inequalities (e.g. $x+2y < 3$, $x - y < 5$). The solution set, instead of being a portion of the real line, is a portion of the plane, or more generally, a portion of $\mathbf{R}^n$.

Besides algebraic equations, differential equations also fit into our framework. The function f or the operator F is replaced by a **differential operator**, say L where $L[y] = y'' + 3y' + 2y$. Hence the "vector" equation $L[y] = 0$ is simply the differential equation $y'' + 3y' + 2y = 0$. The set Σ, instead of being a set of numbers or a set of ordered pairs, is now a set of functions, say the set $C^2(\mathbf{R})$ of all functions with domain **R** whose second derivatives are continuous. The solution set $S = \{ y \in C^2(\mathbf{R}) ; L[y\} = 0\}$ is the set of all functions in $C^2(\mathbf{R})$

which satisfy the differential equation. We note that algebraically, functions can be viewed as **vectors** and that our framework includes equations in any **vector space** or, for that matter, any algebraic structure (e.g. groups, rings, fields, and vector spaces). Hence we see that the framework is quite extensive.

Although it does not encompass all problem types, the framework discussed here provides a standard problem solving context for high school students and college undergraduates at the freshman and sophomore level. A clear understanding of this framework should help you toward a better understanding of why problems may have no solution (e.g. $3x-1=(6x+2)/2$ and the simultaneous equations $x+2y=3$, $2x+4y=5$), one solution (e.g. $3x-1 = 4x+2$, and the simultaneous equations $x+3y=3$, $x+4y=5$), more than one solution (e.g. $x^2-4 = 0$ and $x^5(x-2)(x-4) = 0$), or even an infinite number of solutions (e.g. $3x+1 = (6x+2)/2$, the inequality $|x-3|-4 < 0$, the simultaneous equations $x+2y=3$, $2x+4y=6$, and the differential equation $y''+3y'+2y = 0$). This should help you to understand that not every math problem has exactly one solution. It should also help you to begin to move from just focusing on learning algorithms for the solution of evaluation problems to the more advanced view of, given a problem that is well formulated, how does one find answers to the questions: Does a solution exist? Is it unique? How do we know? Can we develop algorithms to find all of the solutions? What other problems will our algorithms solve and why?. Hopefully, this will encourage you to spend time trying to understand the "why"s of solving problems in mathematics as well as the "how to"s.

We extend our discussion of the **framework for find problems** (FFP's) by giving more examples illustrating the types of sets Σ and properties or conditions C that we can use, the number of solutions that the problem might have and possible techniques for solution. We have seen that Σ can be a number system such as $\mathbf{R}$ or $\mathbf{C}$. Since a solution of two equations in two unknowns, say x and y, is an ordered pair $[x,y]^T$, Σ can also be the set $\{[x,y]^T: x,y\in\mathbf{R}\}=\mathbf{R}^2$ of all ordered pairs of real numbers. Since solutions to problems could have any number of unknown variables, Σ can also be the set $\mathbf{R}^n = \{[x_1,x_2,...,x_n]^T: x_i\in\mathbf{R} \text{ for } i = 1, 2, ... , n\}$. For differential equations, the solution is a function so that Σ could be a function space like $C^1(I) = \{f:I \to \mathbf{R}: f(x)$ and $f'(x)$ are continuous on the interval $I\}$. All of the above are examples of **vector spaces**. Very often for science and engineering problems, Σ is a vector space.

A problem Prob(Σ,C) is **well-posed** in a set theoretic sense if it has exactly one solution. We give examples to show that all problems are not well-posed. Any number of solutions are possible. We use $S = \{x\in\Sigma:C(x)\}$ as the solution set for Prob(Σ,C).

EXAMPLES. $S = \{x\in\mathbf{R}: x + 3 = 2\}= \{-1\}$ well-posed
$S = \{x\in\mathbf{R}: x^2 - 4 = 0\} = \{2, -2\}$ two solutions, not well-posed
$S = \{x\in\mathbf{Q}: x^2 - 2 = 0\} = \varnothing$, no solution, not well-posed
$S = \{x\in\mathbf{R}: x^2 = 2\} = \{\sqrt{2}, -\sqrt{2}\}$ two solutions, not well-posed
$S = \{x\in\mathbf{R}: x^2 + 1 = 0\} = \varnothing$ no solution, not well-posed
$S = \{x\in\mathbf{C}: x^2 + 1 = 0\} = \{i, -i\}$ two solutions, not well-posed

$S = \{x\in\mathbf{R}: x + 3 < 2\} = \{x\in\mathbf{R}: x < -1\}$ an infinite number of solutions, definitely not well-posed.

The solution process gets us from the implicit definition of S (which we call Prob(Σ,C) on the left to the explicit description on the right (which we call Soln(Σ,C). For equations this process often (but not always) consists of a sequence of steps where we rewrite the given equation as an **equivalent equation** with the same solution. However, other processes such as factoring and theorems such as "if the product of two numbers is zero, then one of the numbers must be zero" can be of use. Deciding exactly what we mean by an explicit solution to a problem is really part of the problem. If the problem is well posed, and the unique solution has a name, we want to know it. If we can show that there exists exactly one solution and and it does not have a name, we can give it one. If $\Sigma = \mathbf{R}$ and the problem is well-posed, we may want a decimal approximation to the solution (i.e, the one element in the solution set). Suppose

$$S = \{x \in \mathbf{R}: x^2 = 2 \text{ and } x>0\} = \{ \sqrt{2} \}.$$

We have given a name of the element that we claim is the only element in the solution set. But how do we know there is such a number and how can we find a decimal approximation to it. If it is a rational number, we want one of its "fraction" names. We see that showing that a problem is well-posed and finding its name or the name of a close by neighbor are in fact two different processes and either task (or both) could be called the solution process. We indicate how to show that Prob($\mathbf{R}$, $x^2 = 2$ and $x>0$} is well-posed so that there exists exactly one positive number whose square is two and we may then call it $\sqrt{2}$. Obviously there is a process to obtain an approximation of $\sqrt{2}$ that your calculator has been programed to do when you punch the correct buttons. To show that Prob($\mathbf{R}$, $x^2 = 2$ and $x>0$} is well-posed, we need the axiomatic properties of $\mathbf{R}$ given in Chap 0-2. We assume that those interested in proofs of existence and uniqueness will reread these. We can show that

$S = \{x \in \mathbf{R}: x^2 = 2 \text{ and } x>0\} = S_1 \cap S_2 = S_3 \cap S_4 \cap S_2$ where

$S_1 = \{x \in \mathbf{R}: x^2 = 2 \}$, $S_2 = \{x \in \mathbf{R}: x>0\}$, $S_3 = \{x \in \mathbf{R}: x^2 \geq 2 \}$, and $S_4 = \{x \in \mathbf{R}: x^2 \leq 2 \}$.

We can show that $S_1 = S_3 \cap S_4$ by using the **trichotomy** property of $\mathbf{R}$. Since $\forall x \in S_4$, $x \leq 5$ we have that S_4 has an upper bound. Since $0^2 = 0 \leq 2$, $0 \in S_4$ so that S_4 is not empty. Hence by the **least upper bound axiom**, S_4 has a least upper bound (whether we can calculate it or not) and we call it $\sqrt{2}$. If we can show $\sqrt{2} \in S_3$ and $\sqrt{2} \in S_2$, we have existence of a solution. If we assume that there is another solution, say $s \in S$ and can show that $s = \sqrt{2}$, we have uniqueness. This is totally independent of having a process for computing an approximation for $\sqrt{2}$. But we probably should show that $\sqrt{2}$ exists and is the unique solution to our problem before trying to compute an approximation to it.

Recall the first order Initial Value Problem (IVP):

ODE $dy/dx = f(x,y)$

IVP

IC $y(x_0) = y_0$

To emphasize that we are now considering the mathematical solution to a mathematical problem, we choose to use y as a function of x and we allow the **initial condition** (IC) to be at an arbitrary point. It is perhaps better referred to as a **side condition** (SC) since an interpretation of x as time is not required and may cause false conclusions. We require that all of our logic must be mathematical and not temporal or spacial.

We discuss three fundamental mathematical problem solving contexts for an ODE or an IVP: Calculus (or Engineering), Classical, and Modern and then branch out to include others. It is the Calculus or Engineering context that we are most concerned with at an undergraduate level. But if you go to graduate school and develop new models, you may be interested in the others.

Calculus. In this context, f(x,y) is specified explicitly in terms of **elementary functions**. Calculus is then used to obtain an (infinite) parametric family of solutions to the ODE, one for each value of an (integration) constant, in terms of elementary functions Then the particular solution that satisfies the side condition is obtained by substituting these values into the formula. This context can be expanded to allow special functions and indeed to allow antiderivatives of any elementary function. This context can be further expanded to allow general forms where we are assured that for any specific f(x,y) having this form, a particular algorithm to obtain an infinite family of solutions will work. An example is when f(x,y) has the linear form $f(x,y) = -p(x)\,y + g(x)$ where $p,g \in C(I)$, then we know a procedure that will solve the problem. The solution process requires not only function algebra (addition, subtraction, multiplication and division of functions and multiplication and division of a function by a scalar) but also requires the operation of antiderivatives of functions that involve p and g. We say that the linear problem can be solved up to qradrature (i.e., up to finding antiderivatives of certain functions). We refer to this as solving in a **General context**..

A difficulty can result from the lack of consideration for the number of solutions. If we can guess a solution to a particular ODE, it can be checked by substituting into the ODE and checking to see if the side condition is satisfied. However, what about uniqueness? Is this the only solution or are there others? The solution process itself provides a proof of uniqueness since it assumes that a solution exists and goes through a sequence of equivalent problems (or properties) that the solution must satisfy, ending with one that actually gives the name of the (parametric family of) solution(s).

Another difficulty is that our solution processes (e.g., for nonlinear problems) often result in implicit, rather than explicit descriptions of the functions (i.e., curves rather than functions). Thus the interval of validity (i.e., the domain of the function we seek) is not self evident.

We might also add that there does not exist a procedure for all possible f(x,y)'s. For all these reasons and others, we need a second context.

Classical. Instead of specifying f(x,y) explicitly, we simply require it to satisfy certain conditions and then show that there exist exactly one solution in a particular function class Σ. This can be further subdivided based on the function class.
Classical I : Sufficient conditions are given so that there is exactly one solution in $\Sigma = C^1(I)$ where $x_0 \in I = (a,b)$.
Classical II : Sufficient conditions are given so that there is exactly one solution in $\Sigma = A(I)$ where $x_0 \in I=(a,b)$.

Modern. This is the same as classical except that the problem is reformulated to allow "weak" solutions, that is, things that, strictly speaking, are not functions. For example, solutions may be considered to be **distributions**, **functionals**, or **equivalence classes** of functions.

In addition to the reasons sighted above, if the traditional context does not yield an explicit solution in terms of elementary functions, it is very useful to know that exactly one solution exist. **Numerical techniques** such as **finite differences** and **finite elements** can then be used to find **approximate solutions** that can be shown to "close" to the actual solution. If we do not know that exactly one solution exists, there is no guarantee that the approximate solution obtained has any relevance to the problem.

ODE's-I-4 A FRAMEWORK FOR FIRST ORDER
Handout #4 DYNAMICAL SYSTEM MODELS Professor Moseley

Read Section 2.5. of Chapter 2 of text (Elem. Diff. Eqs. and BVPs by Boyce and Diprima, seventh ed.). Pay particular attention to the process of developing a mathematical model of a physical phenomenon. Also read Section 7.1. Try to understand the difference and similarities of a system of ODE's (a vector equation) and a scalar (one dimensional vector) equation.

The primary focus in Part 1 of these notes is for you to learn the "**how to**" for solving first order ODE's and IVP's. However, you should also understand "**why**" any given technique works. Of course we will also spend time learning how to use first order ODE's to develop **mathematical models** for applications. You will do much more of this in your engineering and science courses where it will be expected that you have already learned the "how to" and the "why" for solution techniques. An introduction to **mathematical modeling** is given for **first order scalar equations** in Chapter 1-5, for **second order scalar equations** in Chapter 3-4, for **discrete first order systems** in Chapter 5-1, and for **continuum first order systems** (the heat equation in one space dimension) in Chapter 7-1.

Not only do first order scalar ODE's arise directly in science and engineering, but study of these simply examples provides prerequisite understanding of the "how to" and "why" for more complicated models. But before continuing with more theory and techniques for first order ODE's, we introduce a **framework for applications**. Early exposure to this framework emphasizes the importance of the concept of a mathematical **vector space** (see Chapter 2-3) in understanding how to frame more complicated problems arising in science and engineering. It is hoped that this will inspire you to spend the time required to understand the concept of a vector space as an **abstract algebraic structure**. Not all application problems fall into this framework, (a framework for all problems does not exist), but a substantial number do.

We begin with the nonlinear vector model given by the **initial value problem** (IVP):

ODE $\dfrac{d\vec{u}}{dt} = \vec{F}(\vec{u}, t)$ (Recall that for a one-dimensional or (1)

IVP scalar equations where we have one

IC $\vec{u}(0) = \vec{u}_0$ state variable, we used $y' = f(x,y)$) (2)

where $\vec{u}$ is the **state vector** containing all of the **state variables** for the system (e.g., positions velocities, voltages, currents, chemical concentrations, money, etc.) which vary with time t. It is assumed that you have some knowledge of vectors. (See Chapter 2-3 for the definition of a **vector space** as a mathematical structure.) You should think of a **mathematical vector** as a collection of state variables that measure (physical) quantities and **not** as a **directed line segment**. It is very important that you realize that the state variables may, but need not have any geometrical interpretation. They are simply the measure of (physical) attributes such as voltage, temperature, and yes, position and velocity. Since these measures are often real numbers or scalars, examples of vector spaces of interest are $\mathbf{R}^n$ and the function spaces $C(\mathbf{R},\mathbf{R})$ and $A(\mathbf{R},\mathbf{R})$. If there are a finite number of state variables, we refer to the system as **discrete**. Since the real

world is considered to be a continuum, these are often **lumped parameter** systems (e.g., circuits, springs, and trusses). A **continuum system** (e.g., temperature in a rod, plate or three-dimensional object) has an infinite number of state variables, one for each point in space. We may think of the derivative $d\vec{u}/dt$ as the collection of (partial) derivatives of all of the state variables; that is, we compute the derivatives componentwise. Thus we may think of (1) as a collection of scalar equations.

For each t, the **operator** $\vec{F}(\cdot, t)$ (an operator is a mapping from one vector space to another) maps the **vector space** in which $\vec{u}$ resides into itself. It tells how the rate of change of each state variable depends on the current values of all of the state variables. That is, the ODE in our model is a **rate equation** that describes how the state of the system will change in time based on its **current state**. There is no memory of how it got there. In some cases, the model is a conservation law. For example, for a specific model (i.e., choice of $\vec{F}(\vec{u}, t)$) we may have

$$\vec{F}(\vec{u}, t) = \vec{R}_{in}(\vec{u}, t) - \vec{R}_{out}(\vec{u}, t) \tag{3}$$

where $\vec{R}_{in}(\vec{u}, t)$ gives the rate of increase of the state variables and $\vec{R}_{out}(\vec{u}, t)$ gives the rate of decrease. Hence we assume $\vec{u}$ is conserved. We discuss several special cases of the IVP (1) and (2).

Time Varying (Possibly Nonlinear) System

$$\frac{d\vec{u}}{dt} = \vec{T}(\vec{u}, t) + \vec{g}(t) \tag{4}$$

(For a one-dimensional scalar equation with one state variable we will use $y' = f(x,y) + g(x)$)

Although similar to (1) above, this model separates out explicitly the vector $\vec{g}(t)$ containing the external **sources** and **sinks** (e.g., voltage or current sources, external forces, or sources and sinks of chemicals).

Time Varying Discrete Linear System

$$\frac{d\vec{u}}{dt} = P(t)\,\vec{u} + \vec{g}(t) \tag{5}$$

P(t) is a square time varying matrix

(For a one-dimensional or scalar linear equation we used $y' = -p(x)y + g(x)$.)

This model is linear and has a finite number of state variables. Solution techniques for such systems where P is constant are considered in Part 5 of these notes.

Time Invariant (Autonomous) (Possibly Nonlinear) System

$$\frac{d\vec{u}}{dt} + \vec{T}(\vec{u}) = \vec{b} + \vec{g}(t) \qquad \text{(For a one-dimensional or scalar equation} \qquad (6)$$

we will use $y' + f(y) = b + g(x)$.)

This is a very important model since it is still general enough to model many real-world phenomena, but specific enough to allow us to begin a more rigorous analysis. Here the operator T (see Chapter 2-6) maps the (real or complex) **vector space** V back into itself, $T : V \to V$. (the definition of a vector space is given in Chapter 2-3.) Since a vector in V is a state of the system, the vector space V contains all possible states of the system. For example, V may be any of the real vector spaces $\mathbf{R}$, $\mathbf{R}^2$, $\mathbf{R}^n$, $C(\mathbf{R})$, $C^1(\mathbf{R})$, $\vec{C}^1(\mathbf{R})$.

Since we wish to model the rate of change of $\vec{u}$ (i.e., the derivative of $\vec{u}$) we need the vector space V to have a **topology**. However, we do not give V a topology directly. The vector spaces of interest in this course all have **inner products**. (The inner or **scalar product** in $\mathbf{R}^3$ is just the **dot product** with which you are familiar.) A vector space with an inner product is called an **inner product space** (see Chapter 4-6). If a vector space has an inner product, then a **norm** or **length** can be defined for each vector. Thus every inner product space is a **normed linear space**. (Mathematical vector spaces are often called **linear spaces** to emphasize that they are algebraic rather than geometric constructs.) Your physics teacher probably taught you that all vectors have **magnitude** and **direction**. Mathematical vectors in a vector space have direction but not magnitude (length). However, vectors in a normed linear space do. In addition, a norm generates a **metric** so that we can determine the distance between vectors (i.e., between states of the system. If $\|\vec{u}\|$ is the length of a vector $\vec{u}$ in V, then the metric $\rho(\vec{u}, \vec{v}) = \|\vec{u} - \vec{v}\|$ gives a measure of the distance between $\vec{u}$ and $\vec{v}$. (In $\mathbf{R}^3$, $\|\vec{u} - \vec{v}\|$ is the distance between the ends of the **position vectors** $\vec{u}$ and $\vec{v}$.) A metric in turn generates a topology (called the **metric topology**). Hence we can take **limits** and define a derivative. Hence if V is a normed liner space, the derivative of the vector $\vec{u}$ is defined using the norm topology. If V is finite dimensional, this will be equivalent to computing the derivative component wise.

We see from the model that the rate of change of (each variable in) the state of the system depends only on the current state of the system through the **operator** T (see Chapter 2-4). Thus our system has no memory of how it came to be in the state $\vec{u}$ and the future depends only on the present state and not on the past.

Steady State or Equilibrium Problem for the

Time Invariant (Autonomous) (Possibly Nonlinear) System

We now look for equilibrium (or static) or steady state solutions which do not depend on time. Hence we assume $d\vec{u}/dt = \vec{0}$ and $\vec{g}(t) = \vec{0}$ (or $\lim_{t \to \infty} \vec{g}(t) = \vec{0}$ so that we are looking at steady state). Then we wish to solve

$$\vec{T}(\vec{u}) = \vec{b} \,. \tag{7}$$

This problem can be viewed as a **mapping problem**. The solutions are those vectors $\vec{u}$ that $-\vec{T}$ maps into the biasing vector $\vec{b}$. If $\vec{T}$ is a **one-to-one correspondence**, we can solve this problem for any $\vec{b}$ by finding the (compositional) **inverse** of the operator $\vec{T}$. However, in practice **inverse operators** are rarely ever actually computed. On the other hand (OTOH) the concept of an inverse operator provides an excellent theoretical tool if it is known that there always exists a unique solution (i.e., that T is invertible).

Nonlinear Dynamical System

Now let $N(\vec{u}) = d\vec{u}/dt - T(\vec{u})$ so that N:V(t)→V(t) where V(t) is the vector space of **time-varying vectors** in V. That is, V(t) = $\{\vec{u}:I\to V\}$ where I = (a,b) is an open interval in **R** to be determined as the **interval of validity** of the solution $\vec{u}$. As indicated, V(t) is also a vector space. Now let D = $\{\vec{u}(t)\in V(t): \vec{u}(0) = \vec{u}_0\}$ and N_0 be the restriction of N to D so that N_0:D→V(t). Theoretically, to solve the problem $N_0[u] = g(t)$, we simply wish to invert the operator N_0. Thus the dynamics problem can also be viewed as a mapping problem. However, to solve this problem numerically we would discretize and solve using **finite difference** or **finite element** methods as a "marching" problem. (See Chapter 1.6.)

Time Invariant (Autonomous) Linear System

$$\frac{d\vec{u}}{dt} + \vec{T}(\vec{u}) = \vec{b} + \vec{g}(t) \tag{8}$$

(For a linear one-dimensional or scalar equation we use $y' + a\,y = b + g(x)$)

Now assume that T:V→V is a linear operator. (The definition of a linear operator is given in Chapter 2-6.) V is the (real or complex) vector space of all possible states of the system. For example, V may be any of the real vector spaces **R**, $\mathbf{R}^2$, $\mathbf{R}^n$, C(**R**), C^1(**R**), $\vec{C}^1$ (**R**).

Steady State or Equilibrium Problem for the Time Invariant (Autonomous) Linear System

Assume that $d\vec{u}/dt = \vec{0}$ and that $\vec{g}(t) = \vec{0}$ so that the source is just the biasing vector $\vec{b}$. Again we have

$$\vec{T}(\vec{u}) = \vec{b} \,. \tag{9}$$

Again we wish the inverse of the operator T where T:V→V. For discrete (lumped parameter) linear problems T becomes the operator defined by multiplication by the matrix A (see Chapter 2-6), $\vec{u}$ becomes the column vector $\vec{x}$, and we wish to solve $A\vec{x} = \vec{b}$. (i.e., We wish to invert

the square matrix A. Numerically, we rarely actually invert A, but rather obtain its LU decomposition. (See Chapter 5-3.) The point is that **matrix algebra**, **solution to linear algebraic equations**, and **abstract linear algebra** all become important tools in our effort to solve linear application problems. We review these concepts in Part 2 of these notes. More development is done in Parts 4 and 5.

Linear Dynamical System

First let $\vec{u}_{ss}$ be the solution to the steady state problem. Now replace $\vec{u}$ with $\vec{u}_{ss}+\vec{u}$ so that $\vec{u}$ is now the displacement from equilibrium. Since T is linear we have $T[\vec{u}_{ss}+\vec{u}\,]= T[\vec{u}_{ss}\,] + T[\vec{u}\,]$. Also $d(\vec{u}_{ss}+\vec{u}\,)/dt = d\,\vec{u}/dt$. Substituting we let $T(\vec{u}_{ss}) - \vec{b} = \vec{0}$ to obtain:

$$\frac{d\vec{u}}{dt}+\vec{T}(\vec{u})=\vec{g}(t) \qquad \text{(For a one-dimensional or scalar equation we use } y'= a\,y + g(x).\text{)} \qquad (10)$$

Now let $L(\vec{u}) = d\vec{u}/dt + T(\vec{u})$ so that L:V(t)→V(t) where V(t) is the vector space of time-varying vectors in V. That is, V(t) = $\{\vec{u}\,(t)$: $\vec{u}\,(t)$: I→V$\}$ where I=(a,b) is an open interval in **R** to be determined as the interval of validity of the solution $\vec{u}\,(t)$. Now let D = $\{\vec{u}\,(t)\in V(t)$: $\vec{u}\,(0) = \vec{u}_{ss\,0}\}$ and L_0 be the restriction of L to D so that L_0:D→V(t). Theoretically, to solve $L_0[\vec{u}\,] = \vec{g}(t)$, we simply wish to invert the operator L_0. However, numerically we would discretize and solve using **finite difference** or **finite element** methods and solve as a marching problem. An introduction to numerical techniques for first order systems is given in Chapter 1-6 by using Euler's method on a scalar equation. We also give an introduction to **error analysis** for the mapping problem. This is directly applicable to equilibrium or steady state problems. For dynamical systems, we wish to estimate the increase in the error in $\vec{u}$ as a function of time rather than just obtaining an overall estimate.

EXERCISES on A Framework for First Order Dynamical System Models.

Our objective is to not just to simply learn known algorithms for solving differential equations, but to also understand **why we use the methods we choose**. That is, we not only want to train ourselves in techniques and be able to apply them to problems where we are told they work, but also to know what techniques apply to what problems and to understand why these techniques work and when and how they can be extended to other problems. That is, we wish to **understand the theory** behind the methods. We are motivated by the fact that differential equations are used as mathematical models of scientific and other phenomena, particularly systems that change with time and space. To understand differential equations, we must **understand the theory** behind the methods.

After we finish first order scalar ODE's (and second order ODE's using first order techniques), we will only be concerned with **linear** problems. This means that it is very important to understand **linear** theory. One can understand the solution process for **linear** algebraic equations using high school algebra without understanding **linear theory**. Likewise, one can understand the solution technique using an integrating factor that we learned for solving scalar first order **linear** ODE's using calculus, again without understanding **linear theory**. But to justify the solution techniques for scalar second and higher order **linear** ODE's, first (and second) order **linear** systems of ODE's, and the heat equation (a **linear** PDE), **an understanding of linear theory is mandatory**.

Most, if not all, of the problems you have solved so far in mathematics have been of the evaluate or locate (find) type. For **evaluation problems**, you learned (i.e., trained yourself by doing homework after seeing an example) a well defined algorithm or **computational skill** such as addition, multiplication, raising a number to a power, extraction of roots, and evaluation of algebraic functions. However, for **locate problems** you were asked to find all objects (e.g., numbers) satisfying a given property (e.g., an algebraic equation). The fundamental "plan of attack" or philosophy used for such problems is to reformulate the problem as an **equivalent problem** (e.g., an equivalent algebraic equation) that has the same **solution set**. This process is repeated until an explicit description of the solution set is found. However, the exact steps in the solution process are not preordained, but are instead left up to the problem solver. (The fun is to see who can find the shortest route to the answer.)

For an algebraic equation such as $3x - 2 = 7$ that has exactly one solution, the "strategy" is to use **equivalent equation operations** (EEO's) to isolate, if possible, the **unknown** (i.e., the variable) on one side of the equation. The value on the other side is then the solution. If the number of solutions is finite (and small), **an explicit description of the solution set** consisting of the (names of) the solutions should be found. This same idea of isolating the unknown was used for solving **first order linear ODE's** when we used the integrating factor (as well as calculus) to isolate the **unknown function** on one side of the equation. However, the **solution set** for the ODE consisted of an **infinite number of functions** parameterized by an arbitrary integration constant.

The fundamental philosophy of **reformulating the problem** not only applies to equations where there is only one solution but extends to problems such as $x^2 + 3x + 2 = 0$ where there are

two solutions and to inequalities such as $3x - 2 < 7$ where there are an infinite number of solutions. However, more theory (i.e., more properties of **R**) and/or a clearer definition of what is meant by "**an explicit description of the solution set**" are needed to effect a solution algorithm. For the problem $3x - 2 < 7$, the solution set is $S = \{x \in \mathbf{R}: 3x - 2 < 7\} = \{x \in \mathbf{R}: x < 3\}$ so that $x < 3$ is a more explicit description of S then $3x - 2 < 7$ since the unknown has been isolated.

The fundamental philosophy of **reformulating the problem** to get a more **explicit description of the solution set** applies to any problem where it is not easy to precisely identify all of the elements in a predetermined set Σ that satisfy a given property (e.g., an equation or an inequality). If the philosophy succeeds, an explicit (or at least a more explicit) description of the solution set is obtained. If all steps are reversible, the **solution process** gives the **solution set** exactly. However, some **equation operations** such as "squaring both sides of the equation" can introduce **extraneous roots** i.e., result in a new problem whose solution set includes other elements in addition to solutions of the original problem. However, if the new problem (and hence the old problem) has only a finite number of solutions, these can be **checked individually** to see which are solutions and which are extraneous. (In fact, solutions to algebraic equations should always be checked to determine if errors have been made. Solutions to differential equations should likewise be checked.)

The process for solving differential equations studied so far as well as the process for solving a system of **linear algebraic equations** follow this same fundamental philosophy. For a system of linear algebraic equations, we start with the system and apply **Elementary Equation Operations** (EEO's) to reformulate the problem. For a first order ODE, we start with the ODE and apply **ODE Operations** to reformulate the equation. For example, for **separable ODE's**, we start with the ODE, separate variables and integrate both sides. However, since the equations may be nonlinear and the solutions obtained are implicit, care is required to make sure no solutions are lost and no extraneous solutions appear. When we studied **exact equations** and **substitutions** this same philosophy prevailed. However, when we study **second order linear ODE's**, the philosophy changes dramatically. We leave it to the theory to establish **existence and uniqueness** for an IVP, and use the **linear theory** to develop a solution process.

For problems in **dynamics** modeled by a linear ODE (or a linear PDE), this means first using the linear theory to find the **general solution** of the ODE parameterized by arbitrary constants (or, in the case of PDE's, at least a large family of solutions parameterized by arbitrary constants or arbitrary functions) and then applying the initial condition(s) to determine these constants (or functions) and hence obtaining the unique solution. For linear **equilibrium systems** with a finite number of **state variables**, this means solving a system of linear algebraic equations. However, **continuum equilibrium problems** defined as BVP's for ODE's or PDE's are solved similar to IVP's by first using the linear theory to find the general solution of the differential equation and then applying the boundary conditions.

The big difference in philosophy comes in finding the **general solution** of the ODE or PDE. We treat these problems as **linear mapping problems** and rely on the theory to tell us as much as possible about the **dimension** of the **null space** of the **linear operator**. (We explain these terms later.) For an nth order linear ODE's where the null space is **finite dimensional**, we try to guess the form of **basis functions** and then use algebra to compute parameters. The linear

theory then guarantees us that we have the **general solution** (i.e., all solutions). For a linear PDE's where the null space is **infinite dimensional** (even when the boundary conditions are included), we use the same approach, but whether we have all solutions is not as clear and convergence of the infinite sum requires some additional consideration.