Chapter 1, Introduction

Extract from
“Introduction to Scientific Computing with Differential Equation Models”
Lennart Edsberg, KOD, KTH, SE-100 44 Stockholm

There is probably no exaggeration to say that differential equations are the most common and important mathematical model in science and engineering. Whenever we want to model a system where the state variables vary with time and/or space, differential equations are the natural tool for describing its behaviour. The construction of a differential equation model demands a thorough understanding of what takes place in the process we want to describe.

However, setting up a differential equation model is not enough, we must also invent a method for its solution. The effort of finding the solution of a differential equation is much a symbiosis of modeling, mathematics and choosing a method. Therefore, when you are requested to solve a differential equation problem from some application, it is useful to know facts both about its modeling background, its mathematical properties and its numerical solution. The last part involves choosing appropriate numerical methods, adequate software and appealing ways of visualizing the result.

This interaction between modeling, mathematics and numerical methods is nowadays referred to as scientific computing and its purpose is to perform simulations of processes in science and engineering.

1.1 What is a Differential Equation?

A differential equation is a relation between a function and its derivatives. If the function $u$ depends on only one variable $t$, i.e. $u = u(t)$, the differential equation is called ordinary. If $u$ depends on at least two variables $t$ and $x$, i.e. $u = u(t, x)$, the differential equation is called partial.

1.2 Examples of an ordinary and a partial differential equation.

An example of an elementary ordinary differential equation (ODE) is

$$\frac{du}{dt} = au$$

(1.1)

where $a$ is a parameter, in this case a real constant. It is frequently used to model e.g. the growth of a population ($a > 0$) or the decay of a radioactive substance ($a < 0$). The ODE (1.1) is a special case of differential equations called linear with constant coefficients, see chapter 2.
The differential equation (1.1) can be solved analytically, i.e. the solution can be written explicitly as an algebraic formula. The solution of the differential equation is

$$u(t) = Ce^{at}$$

In (1.2), $C$ is an arbitrary constant, hence the solution is not unique. The expression (1.2) is called the general solution. If $C$ is known to a certain value, however, we get a unique solution which, when plotted in the $(t, u)$-plane, gives a trajectory (solution curve). This solution is called the particular solution.

The constant $C$ can be determined e.g. by settling a point $(t_0, u_0)$ in the $(t, u)$-plane through which the solution curve shall go. Such a point is called an initial point and the demand that the solution shall go through this point is called the initial condition. A differential equation together with an initial condition is called an initial value problem (IVP).

Observe that the differential equation alone does not define a unique solution, we also need an initial condition or other conditions. A plot of all trajectories i.e. all solutions of an ODE in the $(t, u)$-plane will result in a graph that is totally black since there are infinitely many solution curves filling up the whole plane.

In general it is not possible to find the analytical solution of a differential equation that easily. The “simple” differential equation

$$\frac{du}{dt} = t^2 + u^2$$

cannot be solved in that way. If we want to plot trajectories of this problem we have to use numerical methods.
An example of an elementary partial differential equation (PDE) is
\[ \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0 \] (1.4)
where \( a \) is a parameter, in this case a real constant. The solution of (1.4) is a
function of two variables \( u = u(t, x) \). This differential equation is called the
advection equation. Physically it describes the evolution of a scalar quantity
such as temperature \( u(t, x) \) carried along the \( x \)-axis by a flow with constant
velocity \( a \). It is also known as the linear convection equation and is an example
of a hyperbolic PDE, see chapter 5.

The general solution of this differential equation is (see exercise 4)
\[ u(t, x) = F(x - at) \] (1.5)
where \( F \) is an arbitrary differentiable function in one variable. This is indeed
a large amount of possible solutions! The three functions
\[ u(t, x) = x - at, \quad u(t, x) = e^{-(x-at)^2}, \quad u(t, x) = \sin(x - at) \]
are just three solutions out of the infinitely many solutions of this PDE.

To obtain a unique solution for \( t > 0 \) we need an initial condition. If the
differential equation is valid for all \( x \), i.e. \(-\infty < x < \infty \) and \( u(t, x) \) is known
for \( t = 0 \), i.e. \( u(0, x) = f(x) \) where \( f(x) \) is a given function, the initial value
function, we get the particular solution
\[ u(t, x) = f(x - at) \] (1.6)
Physically (1.6) corresponds to the propagation of the initial function $f(x)$ along the $x$-axis with velocity $|a|$. The propagation is to the right if $a > 0$ and to the left if $a < 0$.

The graphical representation can alternatively be done in 3D:

![3D-graph of $u(t,x) = \exp(-(x-at)^2)$ when $a=1$](image)

When a PDE is formulated on a semi-infinite or finite $x$-interval, boundary conditions are needed in addition to initial conditions to obtain a unique solution.

Most PDEs can only be solved with numerical methods. Only for very special classes of PDE-problems it is possible to find an analytic solution, often in the form of an infinite series.

**Exercises**

1) If $a$ is a complex constant $a = \lambda + i\mu$ what is the real and imaginary part of $e^{at}$?

2) If $a$ is a complex constant what conditions are necessary to impose on $\lambda$ and $\mu$ if $e^{at}$ for $t > 0$ is to be
   
   a) exponentially decreasing, b) exponentially increasing,
c) oscillating with constant amplitude,
d) oscillating with increasing amplitude,
e) oscillating with decreasing amplitude?

3) If $a$ is a complex constant what condition on $\lambda$ is needed if $e^{at}$ is to be bounded for $t \geq 0$?

4) Show that the general solution of $u_t + au_x = 0$ is $u(x, t) = F(x - at)$ by introducing the transformation

$$\xi = x + at, \quad \eta = x - at$$

Transform the original problem to a PDE in the variables $\xi$ and $\eta$, and solve this PDE.

5) Show that a solution of (1.4) starting at $t = 0, x = x_0$ is constant along the straight line $x - at = x_0$. This means that the initial value $u(x_0, 0) = f(x_0)$ is transported unchanged along this line, which is called a characteristic of the PDE.

1.3 Numerical analysis, a necessity for scientific computing

In scientific computing the numerical methods used to solve the mathematical models should be robust, i.e. they should be reliable and give accurate values for a large range of parameter values. Sometimes, however, a method may fail and give unexpected results. Then it is important to know how to investigate why a wrong result has occurred and how it can be remedied.

Two basic concepts in numerical analysis are stability and accuracy. When choosing a method for solving a differential equation problem it is necessary to have some knowledge about how to analyse the result of the method with respect to these concepts. This necessity has been well expressed by late Prof Germund Dahlquist, famous for his fundamental research in the theory of numerical treatment of differential equations: “There is nothing as practical as a little good theory”.

As an example of unexpected results, choose the well-known vibration equation, occurring in the theory of e.g. mechanical vibrations, electrical vibrations and vibration of sound. The form of this equation with initial conditions is

$$m \frac{d^2 u}{dt^2} + c \frac{du}{dt} + ku = f(t), \quad u(0) = u_0, \frac{du}{dt}(0) = v_0$$

(1.7)
In mechanical vibrations \( m \) is the mass of the vibrating particle, \( c \) the damping coefficient, \( k \) the spring constant, \( f(t) \) an outer force acting on the particle, \( u_0 \) the initial position and \( v_0 \) the initial velocity of the particle. The five quantities \( m, c, k, u_0, v_0 \) are also referred to as the parameters of the problem.

Solving (1.7) numerically for a set of values of the parameters is an example of simulation of a mechanical process and it is necessary to choose a robust method, i.e. a method for which the results are reliable for a large range of values of the parameters. The following two examples based on the vibration equation show that unexpected results depending on unstability and/or bad accuracy may occur

1) Assume that \( f(t) = 0 \) (free vibrations) and the following values of the parameters: \( m = 1 \), \( c = 0.4 \), \( k = 4.5 \), \( u_0 = 1 \), \( v_0 = 0 \). Without too much knowledge about mechanics we would expect the solution to be oscillatory and damped, i.e. the amplitude is decreasing. If we use the simple Euler method with constant stepsize (see chapter 3) we obtain the following numerical solution, visualized together with the exact solution:

![Graph](fig.4)

The graph shows that the numerical solution gives an unstable result with increasing amplitudes, why? The answer is given in chapter 3. For the moment just accept that insight in stability concepts and experience in handling unexpected results are needed for successful simulations.
2) When the parameters in equation (1.7) are changed to \( m = 1, \ c = 10, \ k = 10^3, \ u_0 = 0, \ v_0 = 0 \) and \( f(t) = 10^{-4} \sin(40t) \) (forced vibrations) we obtain the following numerical result when using a method from a commercial software product for solving differential equations:

The graph shows that the numerical result is not correct, why? The answer is given in chapter 3. This time we have an accuracy problem. The default accuracy value used in the method is not enough; the corresponding numerical parameter must be tuned appropriately.

1.4 Outline of the contents of this book

After this introductory chapter, the text is organized so that ordinary differential equations, ODEs, are treated first followed by partial differential equations, PDEs. The aim of this book is to be an introduction to scientific computing. Therefore not only numerical methods are presented but also

1) how to set up a mathematical model in the form of an ODE or PDE,

2) give an outline of the mathematical properties of differential equation problems and explicit analytical solutions (when they exist)

3) show examples of how results are presented with proper visualization.

The ODE-part starts in chapter 2 presenting mathematical properties of ODEs, first the basic and important problem class of ODE-systems being
linear with constant coefficients applied to important ODE-system models from classical mechanics, electrical networks and chemical kinetics. This is followed by numerical treatment of ODE-problems in general, divided into the classical subdivision of \textit{initial value problems}, IVPs in chapter 3, and \textit{boundary value problems}, BVPs in chapter 4. For IVPs the \textit{finite difference method}, FD, is described starting with the elementary Euler method. Important concepts brought up for ODEs are accuracy and stability which is followed up also for PDEs in later chapters. For BVPs both FD and the \textit{finite element method}, FEM, are described.

Important application areas where ODEs are used as mathematical model are presented, selected templates are described in the chapters as the main theme and exercises, sometimes suitable for computer labs, are inserted into the text. This organization is continued in the chapters presenting PDEs.

PDEs are introduced in chapter 5 dealing with mathematical properties of their solutions and also a presentation of several of the important PDEs of science and engineering, such as the equations of Navier-Stokes, Maxwell and Schrödinger.

In chapter 6 an outline of mathematical modeling is brought up with the intention of giving a feeling of the principles used when a differential equation (ODE or PDE) is set up from constitutive and conservative equations.

The three chapters to follow are devoted to the numerical treatment of PDEs following the classical subdivision into \textit{parabolic}, \textit{elliptic} and \textit{hyperbolic} PDEs. Concepts from the ODE-chapters such as accuracy and stability are treated for time dependent (parabolic and hyperbolic) PDEs. For stationary problems (elliptic PDEs) sparse linear systems of algebraic equations are essential and hence presented.

Selected models presented in chapter 5 and 6 are used as illustrations for the different methods introduced.

The last chapter gives an overview of existing software for scientific computing with emphasis on the use of MATLAB for programming and FEMLAB for modeling and parameter studies.