These are the lecture notes for CSC349A Numerical Analysis prepared by Rich Little and George Tzanetakis. They roughly correspond to the material covered in each lecture in the classroom but the actual classroom presentation might deviate significantly from them depending on the flow of the course delivery. They are provided as a reference to the instructor as well as supporting material for students who miss the lectures. They are simply notes to support the lecture so the text is not detailed and they are not thoroughly checked. Use at your own risk.

# 1 Course Logistics

Throughout the course I will try to give examples of how ideas from Numerical Analysis are used in all sorts of cool applications as well as how errors in the usage of Numerical Analysis algorithms have had significant cost.

I am going to experiment with using the graphics tablet in conjunction with the blackboard this term.

The use of the textbook is optional but it will definitely help you understand deeper the material. Older editions are ok.

I will NOT be recording the lectures or posting lecture recordings from previous classes.

The course outline can be found at:

`https://heat.csc.uvic.ca/coview/course/2023091/CSC349A`

The course outline has an overview of the material that will be covered and a rough schedule for the assignments and exams. There will be 5 to 10 assignments (worth a total of 30% of the final grade), two midterms worth 30%, and a final exam worth 40%. In addition to the book handouts with notes about the various topics covered will be provided online at the course webpage. A course pack containing the course handouts is available on demand at the UVic bookstore. Finally lecture notes such as this document will be provided for most of the lectures. These notes are by no means a substitute for attending the lectures and the material presented in a lecture might deviate from what is on the notes.

# 2 Importance of Numerical Analysis

A common view of Numerical Analysis is that it is boring, uninviting field mainly concerned with the study of various types of errors. I certainly felt that way when I took it for the first time as an undergraduate. Over time my view of Numerical Analysis has significantly changed and I have found that the ideas and concepts learned in this course are used in a large variety of different contexts and form the foundation of a lot of the software that we use daily. One of my goals for this course is to convey this importance of Numerical Analysis in addition to covering the actual methods themselves.

Although the analysis of errors is central to Numerical Analysis it is much more than that. Numerical analysis is the study of algorithms for the problems of continuous mathematics. "Continuous" means that real or complex numbers are involved. Because continuous numbers cannot be represented exactly on computers errors are inevitable but need to be controlled. "Continuous" problems are everywhere in Engineering and Computer Science including many areas that you probably never thought had any connection with Numerical Analysis.

There is an incredible number of applications of numerial methods in all fields of engineering and computer science. In many cases even practitioners familiar with the application might not realize that "under the hood" of their techniques lie numerical analysis methods. Throughout the course I will try to highlight some of these applications and draw connections between what we are learning and actual application areas. It is also interesting to look at the times when misunderstandings or errors in numerical simulations have resulted in enormous financial and in some ways human cost. Some examples of errors in numerical analysis can be found in the following link: `http://www.ima.umn.edu/~arnold/disasters/disasters.html`.

Here is a quick list of applications in which numerical methods play a fundamental role. It is by no means exhaustive and it is a mixture of more obvious applications and some that you probably never expected. The first applications of computers were in the military. They include discrete types of operations used in code breaking [1] as well as continuous mathematics in things like the computation of ballistic tables. The origins of numerical methods lie in large scale computer simulations initially for atomic bombs and later for things like climate modeling but as computers became increasing

[1] `http://en.wikipedia.org/wiki/AlanTuring`

smaller, faster and cheaper they have expanded to all sorts of areas. Many computer games rely on simulations of physics to create realistic experiences. Angry birds is a popular example of how ballistics and Newtonian physics of motion can be used to hit structures using angry birds. Animators can use tools to model limbs, muscles individually but when it comes to large number of objects interacting such as hair, water molecules or Orcs in the Lord of the Rings movies physics simulation typically using differential equations is utilized. Machine learning and data mining is used in all sorts of application areas such as face recognition, credit card ratings, hedge funds, business analytics and the list goes on. Many modern machine learning methods (for example support vector machines, neural networks, self-organizing maps) are based on optimization methods. Search engines exploit the structure of links between web page and use methods from numeric analysis such as singular value decomposition to analyze large graphs represented as matrices. Computer networks, image processing, digital signal processing, climate modeling all make heavy use of numerical methods. In fact any time a computer program utilizes floating point numbers for some computation it is using results from numerical methods.

Basically numerical analysis algorithms are under everything. However they tend to be rather invisible and hidden under layers of software complexity. Software libraries such as *Lapack* and *BLAS* are parts of all sorts of existing software packages that are built on top of them. A lot of this code was originally written in Fortran and although the number of programmers who write in Fortran is diminishing they are still kept around as they work well and are so widely used that to some extent Fortran compilers are simply kept around to keep these libraries and software packages going. Even if you never “use” directly this library chances are you have used some software that does and getting some glimpse of what they do which you will in this course is a great way to appreciate the complexity and interdependency of software.

# 3 Book Overview

The book has 8 parts and 32 chapters. Each parts begins with a Preface that has the motivation behind the problems covered and some background mathematics for that particular part.

Numerical Analysis is important in Science and Engineering.

- It provides powerful problem solving tools that can handle large amounts of data, non-linearities, and complex geometries (such problems can not be solved by any other means).
- Intelligent use of packaged software often requires knowledge of the basic underlying theory
- Package software cannot solve all real-world problems (you may have to design your own programs).

The following major topics will be covered in each part:

- Part 1: Modelling, Computers and Error Analysis
- Part 2: Roots of equations. Solve $f(x) = 0$ for $x$.
- Part 3: Linear algebraic equations:. Given the $a$'s and the $b$'s solve

$$a_{11}x_1 + a_{12}x_2 = c_1 \tag{1}$$
$$a_{21}x_1 + a_{22}x_2 = c_2 \tag{2}$$

  for the $x$'s.
- Part 5: Curve Fitting: Regression or Interpolation
- Part 6: Integration: $I = \int_a^b f(x)dx$ Find the area under the curve.
- Part 7: Ordinary differential equations: Given $\frac{dy}{dx} = f(x, y)$ solve for $y$ as a function of $x$.

# 4 A Motivating Example

Determine the terminal veloctiy of free-falling body (a parachutist) near the eart's surface. Mathematical Model: Newton's second law of motion:

$$F = ma \tag{3}$$

where $F$ is the force (measured in Newtons), $m$ is the mass (measured in kg), and $a$ is the acceleration (measured in $m/s^2$). Acceleration is by definition the time rate of change of the velocity:

$$a = \frac{F}{m} \tag{4}$$

Now consider a free falling object (a parachutist or an angry bird). There are two forces that operate on this object: the downward pull of gravity ($F_D$) and the upward force of air resistance ($F_U$).

$$F = F_D + F_U \tag{5}$$

These equations are typical of mathematical models of the physical world:

- Describes a natural process in mathematical terms
- It represents an idealization and simplification of reality
- It yields reproducible results and can be used for predictive purposes

$$F_D = mg \tag{6}$$

where $m$ is the mass and $g$ is the acceleration due to gravity which is approximately equal to 9.8 $m/s^2$. We will assume a simple model of air resistance in which it is linearly proportional to velocity and acts in an upward direction i.e:

$$F_U = -cv \tag{7}$$

where $c$ is the drag coefficient (a constant that accounts for the properties of the falling object such as as shape, type of surface etc). Therefore the net force is:

$$F = mg - cv \tag{8}$$

and by using the fact that the acceleration is the derivative of the velocity we can derive the following mathematical model:

$$\frac{dv}{dt} = g - \frac{cv}{m} \tag{9}$$

This is a differential equation that relates the acceleration of a falling object to the forces acting on it. It is possible to get an analytic solution to this differential equation using calculus. If the object is initially at rest i.e $v = 0$ at time $t = 0$ then:

$$v(t) = \frac{gm}{c}(1 - e^{-\frac{ct}{m}}) \tag{10}$$

| t | v |
|---|---|
| 0 | 0.0 |
| 2 | 16.40 |
| 4 | 27.77 |
| 6 | 35.68 |
| .. | .. |
| $\infty$ | 53.44 |

This analytical solution can be used directly to solve problems. [2]

For example a parachutist of mass $68.1kg$ jumps out of a stationary hot air balloon. The drag coefficient is equal to $12.5kg/sec$.

Inserting the parameters into the equation we get:

$$v(t) = \frac{9.81(68.1)}{12.5}(1 - e^{-(12.5/68.1)t}) \tag{11}$$

this equation can be used directly to compute the various values of velocity for different values of $t$. For example:

After a sufficiently long time a constant velocity, called *terminal* velocity of $53.44m/sec$ is reached as eventually the air resistance which keeps increasing as the velocity increases will become equal with the force of gravity which remains constant.

Another possibility is to try to use a numercial (in constrast to an analytical approach) for solving this mathematical model. Even though our solution will not be exactly the same depending on how much accurate we want to be we can make it arbitrarily close. There are several reasons to do so:

- Numerical methods can be applied to functions for which we can not easily find an analytical solution through Calculus
- As any equations is to some degree an approximation of reality and therefore errors are inevitable it is possible depending on the application that the errors introduced by the numeric approximation are neglible in the context of the desired application

[2] For those of you who are interested it is straightforward to show that the analytical solution satisfies the differential equation by substituting in the left hand side and differentiating and substituting in the right hand side. After a few straightforward algebraic manipulations you can show the two sides are equal. To derive the analytical solution you need to express the derivative and then take the limit of the resulting series.

| **t** | **v** |
|---|---|
| 0 | 0.0 |
| 2 | 19.62 |
| 4 | 32.04 |
| 6 | 39.90 |
| .. | .. |
| $\infty$ | 53.44 |

- It's extremely simple to write a computer program to do the job for us although the number of mathematical operations involved is much larger than the analytical solution

The idea is rather simple. We will approximate the derivative of the function by a "finite divided difference" effectively discretizing time. If we denote the discrete time steps we take as $t_i$ (we can decide what sampling rate is appropriate depending on the application) then we can approximate the derivative at time $t_i$ as follows:

$$\frac{dv}{dt}(t_i) \approx \frac{\Delta v(t)}{\Delta t} = \frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i} \tag{12}$$

We can substitute this equation into our parachutist model to give:

$$\frac{v(t_{i+1}) - v(t_i)}{t_{i+1} - t_i} \approx g - \frac{c}{m} v(t_i) \tag{13}$$

this can be rearranged to yield:

$$v(t_{i+1}) \approx v(t_i) + (g - \frac{c}{m} v(t_i))(t_{i+1} - t_i) \tag{14}$$

Notice that this equation gives us a way to compute the velocity value at time $t_{i+1}$ based on the previous value of the velocity. This approach is called Euler's method can be verbally expressed as: *New value = old value + slope x step size*. For example we can use it to compute the velocity values with a step size of 2 seconds.

At the start of the computation we use an initial velocity value $v_0 = 0$ for $t_0 = 0$ plug the numbers into the equation and get the value of $v_1$. Then we can use the value at $v_1$ to compute the value at $v_2$ and so forth.

Figure 1 shows a plot comparing the results of the numerical approximation using Euler's method and the exact analytical solution. In engineering

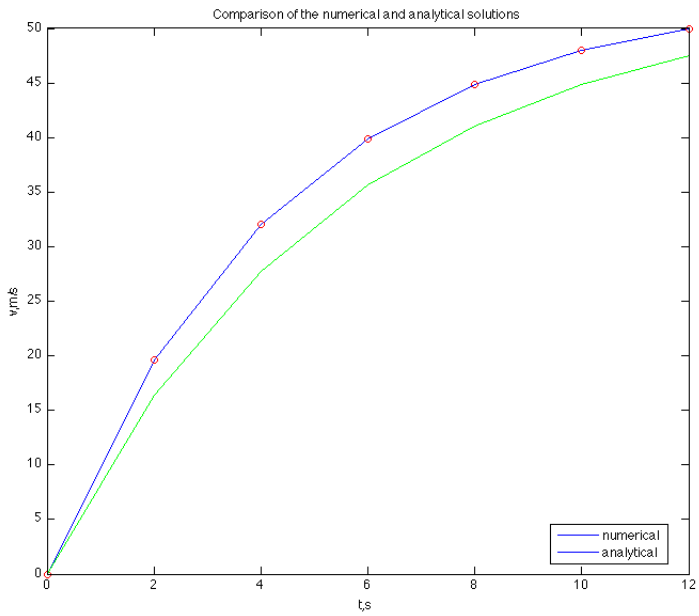


Figure 1: Comparison of numerical and analytical solution

there is frequently a tradeoff between accuracy and computational resources. The Euler method and more generally numerical methods allow you to control that tradeoff for example by adjusting the step size appropriately for the program at hand. There are several questions one could ask at this point that we will be exploring in the rest of the course. For example can we show that the error between the exact solution and the numerical approximate solution will always decrease with smaller step size ? Can we compare how long different approaches to approximating the derivative of a function take ? Are there any functions for which this would not work ? Could we reduce the number of multiplication/additions needed to perform each step of the iteration ?

# 5 Reading and Further Reading

The material covered in this lecture corresponds to Chapter 1 of the textbook. The article "The Definition of Numerical Analysis" by Lloyd N. Trefethen makes a good case for Numerical Analysis being a far more interesting field that simply the study of errors and defines it as the study of algorithms for continuous mathematics.

I also mentioned Alan Turing whose efforts during World War II to break the codes used by the Germans led to the development of electronic computers. The Wikipedia article on him is thorough and definitely worth reading: `ttp://en.wikipedia.org/wiki/Alan_Turing}.\bibliograp`ystyleIEEEbib