FORMAL COMPUTATIONAL SKILLS COURSEWORK
Numerical Integration

Juan Pablo Calderón

November 15, 2002

1 Introduction

This paper is intended to explain the use of the Runge Kutta method for numerical integration of differential equations. The reader does not have to be familiarized with differential equations nor with the numerical integration method, any student who needs to solve a differential equation and doesn't know how, can use this coursework as a guide of how to solve it without knowing maths at all. (It is important that he knows what is that he's solving for.) At the end of this paper the reader would be able to use the method of Numerical integration to get an approximate solution of a great deal of differential equations. Some differential equations are just too difficult or simply impossible to solve analytically, in this paper is explained how to obtain a very accurate solution using Numerical Integration particularly with the fourth order Runge Kutta Method.

2 Numerical Integration

In numerical integration you want to generate an approximate solution of a differential equation, this solution is a discrete solution, which means you have a set of points which satisfy your equation (i.e. when plotting them you reproduce it's graph). The main idea is to increment one of the variables by a certain time step and add a small quantity to the other. In numerical integration you start from an initial condition and add small pieces to it as time increases, the amount you have to add is determined by the derivative at that point or the points around it, as we would see later on.

To begin I'm going to suppose you have a first order differential equation you want to solve of the form

$$\frac{dy}{dx} = f(x, y) \tag{1}$$

where $f(x, y)$ is an expression for the derivative (or rate of change) of y in terms of x and y, (sometimes you have a derivative in terms of the same variable). In this special case, we want to solve for y in terms of x, in more scientific words we are going to integrate y and the integration variable is x. If the differential equation is of a higher order you can always turn it into a series of differential equations of first order (see [1],[2]).

Before learning the Runge Kutta method I'm going to show you a simpler method called the Euler method. This method is really simple as you will see. It is not as approximate as the Runge Kutta but is much simpler to implement in a computer or even a calculator.

2.1 Euler Method

When you have a differential equation like eqn.1, you can make the approximation:

$$\frac{dy}{dx} \simeq \frac{\Delta y}{\Delta x} = f(x, y) \tag{2}$$

And from the definition of a derivative and expressing $\Delta y = yn - y_{n-1}$ you can write down your problem as:

$$y_n = f(x_{n-1}, y_{n-1})\Delta x + y_{n-1} \tag{3}$$

by changing subindex:

$$y_{n+1} = f(x_n, y_n)h + y_n \tag{4}$$

where y_n and x_n refer to the value of y or x in an iteration n. Δx is the time step which can be called h, the smaller it is, the more approximate the integration would be, but the longer it would take. This last equation expresses the value of y in an iteration $n+1$ in terms of the value of x and the value of y in the iteration n, this means that if you know the value of x in some initial condition (x_0) you can calculate recursively the value of y at each time step and end with a set of points that approximate the function y, by slowly incrementing x and calculating the corresponding value for y.

I think the most easier way to illustrate how the method works is by the use of examples, I'm going to show you with two different examples how to use this method and solve some simple differential equations.

2.1.1 Examples

1. The most simple example of a first order differential equation would be a linear equation. An example of a linear equation can be the one describing

a robot moving at constant velocity $v = 3.7\ \frac{m}{sec}$, and finding its displacement x given the initial condition that at $t = 0$ the initial position of the robot is $x_0 = 4\ metres$.

$$\frac{dx}{dt} = v \tag{5}$$

One thing about numerical methods is that you have to use real values, you can't solve the equation analytically, this means that for every initial condition or if you by any chance want to change one of the parameters of the problem you would have to start all over again, this may look like a disadvantage, but can easily be overcome by using a computer program. To solve the equation using the Euler method we just have to look at equation 4 and replace the appropriate values. First we want to chose an integration step suitable for this problem, for example if I want to know where would the robot be in a year, my time step h can't be in $msec$ because it can take to long to integrate, but it can't be so big either, then the solution wouldn't be so accurate. When implementing your solution using a programming language (such as C) you can always try several values and take the one that suits you best (this is one that converges rapidly, and gives you good accuracy, you can compare the time several values of h takes to complete the whole integration and see how accurate your solutions are, then choose a value of h that is accurate enough and doesn't take too long to integrate.)

So lets say we want to see the displacement of the robot in the first 10 seconds. Here I'm going to take $h = 0.1$ just for illustrating the procedure, later on I will show you the effect of the time step in the convergence of the solution.

After replacing the variables of this specific problem in eqn.4 we end up with

$$f(t, x) = \frac{dx}{dt} = v = 3.7 \tag{6}$$

Note that we want to integrate x in terms of t, ($h = \Delta t$). The equation above can be expressed as

$$x_n = x_{n-1} + vh = x_{n-1} + (3.7\ m/sec)(0.1\ sec) \tag{7}$$

so if you start with $x_0 = 4$ then $x_1 = x_0 + vh = 4 + 0.37 = 4.37$ and $x_2 = x_1 + 0.37 = 4.37 + 0.37 = 4.74$ and so on. Now we have three points (t, x) to put in our graph: $(0, 4)$, $(0.1, 4.37)$ and $(0.2, 4.74)$. A C program which solves this example is included in the appendix at the end, make sure you understand what is going on before moving to the next example. I think this is an example which is more difficult to solve by numerical integration than by direct integration, but is an example of how the method works, if you look at equation 7 you will see that this is the same equation that describes the Newtonian mechanics of this movement. In this particular problem we have that $dy/dx = \Delta y/\Delta x$ because the differential equation is linear.

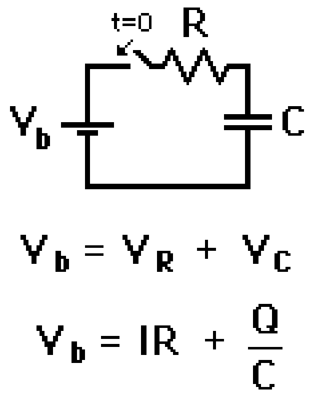

Figure 1: Circuit of example 2.

2. Another example involving differential equations, is to calculate the charge in a capacitor $C = 1\mu F$ with an initial voltage $V_o = 0\ V$ connected to a resistance $R = 1\ \Omega$ and a battery of $V_b = 10\ V$ in an electrical circuit like the one shown in figure 1. The equation describing this system is:

$$V_b = \frac{q(t)}{C} + I(t)R \tag{8}$$

which can be rearranged knowing that $I(t) = dq/dt$ to get

$$\frac{dq}{dt} = \frac{V_b}{R} - \frac{q(t)}{RC} \tag{9}$$

we know the analytical solution of this differential equation which is

$$q(t) = V_bC(1 - e^{-t/RC}) \tag{10}$$

and using the fact that $V(t) = q(t)/C$ we can rewrite equation 10 as

$$V(t) = \frac{q(t)}{C} = V_b(1 - e^{-t/RC}) \tag{11}$$

for an explanation of how to solve this equation analytically or where does the equation comes from, see [3].

Now we want to solve this equation using the Euler method and see if we get the same solution or something similar, comparing both solutions is let to the reader as an exercise.

To solve this problem we just need to replace the corresponding terms in equation 4 as we did for the previous example, to end up with something that looks like this:

$$q_{n+1} = q_n + f(q,t)h \tag{12}$$

By looking at equation 9 we can see that $f(q,t) = V_b/R - q(t)/C$, which is exactly what we need to solve this problem. Note that we want to solve for $q(t)$ in terms of t. To solve for $q(t)$ we just have to make one further step and it is to write the above equation as follows

$$q_{n+1} = q_n + h(\frac{V_b}{R} - \frac{q_n}{C}) \tag{13}$$

We can now start replacing values

$$q_1 = q_0 + 0.1(\frac{10\ V}{1\ \Omega} - \frac{q_0}{10^6 \mu F}) = 1\ V/\Omega \tag{14}$$

$$q_2 = q_1 + 0.1(\frac{10\ V}{1\ \Omega} - \frac{q_1}{10^6 \mu F}) \tag{15}$$

and so on. By replacing the value of q_0 we can find the value of q_1, which is needed to find the value of q_2. In the appendix is a C implementation of this solution. To get the voltage in the capacitor we can use the relation $V = q(t)/C$. Easy.

2.2 Runge Kutta Method

The fourth-order Runge Kutta method is much more accurate than the Euler method because it not only uses the derivative at one point but takes a weighted

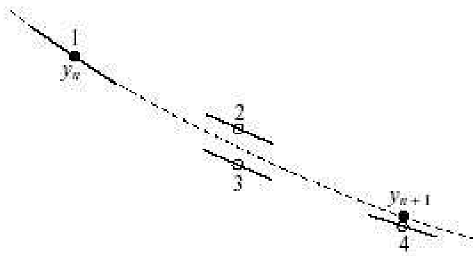

Figure 16.1.3. Fourth-order Runge-Kutta method. In each step the derivative is evaluated four times: once at the initial point, twice at trial midpoints, and once at a trial endpoint. From these derivatives the final function value (shown as a filled dot) is calculated. (See text for details.)

Figure 2: Figure borrowed from Numerical Recipes [1]

average of the derivatives at several points within the integration step. This is better explained below.

In the fourth-order Runge Kutta method you are going to evaluate your function $f(y, x)$ 4 times within your time step h and then take an average of this result. This is a much better approximation than the Euler method which only uses one evaluation within each time step.

Equations 16 to 19 are the four different evaluations within the integration you have to do when using Runge Kutta. Taking a closer look at these equations we see that eqn.16 gives a contribution to the integration equal to the one done when using the Euler method.

The other contributions $(k_2 - k_4)$ they have the same form as the equation for the Euler method but the derivatives are evaluated at different points, these points are displaced in both of the variables involved in the integration. For k_2 we see that the derivative is evaluated half a time step between x_n and x_{n+1} and half way between y_n and k_1. The case for k_3 is very similar than the case for k_2 replacing k_1 for k_2, finally k_4 uses $x_n + h$ and $y_n + k_3$ to calculate this contribution. Figure 2 illustrates this very well.

Equation 20 gives the value of y_{n+1} in terms of the previous value y_n, note that the terms in the middle of the interval (k_2 and k_3) have more weight than the terms at the edges of the integration interval (k_1 and k_4).

$$K_1 = hf(x_n, y_n) \tag{16}$$

$$K_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}) \tag{17}$$

$$K_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}) \tag{18}$$

$$K_4 = hf(x_n + h, y_n + k_3) \tag{19}$$

$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} \tag{20}$$

With these equations we can generate a set of points that reproduce the function almost perfectly, in a similar way that we did with Euler.

2.3 Example

1. Now we are going to solve the same circuit we solved with Euler using Runge Kutta. We already know $f(q,t) = V_b/R - q(t)/(RC)$, the first thing we have to do is calculate $k1 - k4$ for the first iteration to be able to calculate q_1. In the first iteration $q_0 = 0$ and $t = 0$, we can write these expressions as

$$K_1 = hf(t, q_0) = h(\frac{V_b}{R} - \frac{q_0}{C}) = 0.1(\frac{10\ V}{1\ \Omega} - \frac{q_0}{1\ \Omega 10^6 \mu F}) \tag{21}$$

$$K_1 = 0.1(\frac{10\ V}{1\ \Omega}) \tag{22}$$

$$K_2 = hf(t + \frac{h}{2}, q_0 + \frac{k_1}{2}) = h(\frac{V_b}{R} - \frac{q_0 + \frac{k_1}{2}}{RC}) = 0.1(\frac{10\ V}{1\ \Omega} - \frac{q_0 + \frac{k_1}{2}}{1\ \Omega 10^6 \mu F}) \tag{23}$$

$$K_2 = 0.1(\frac{10\ V}{1\ \Omega} - \frac{k_1}{2 * 10^6 \mu F \Omega}) \tag{24}$$

note we know the value of k_1 from eqn.22, this means we now know k_2.

$$K_3 = hf(t + \frac{h}{2}, q_0 + \frac{k_2}{2}) = h(\frac{V_b}{R} - \frac{q_0 + \frac{k_2}{2}}{RC}) = 0.1(\frac{10\ V}{1\ \Omega} - \frac{q_0 + \frac{k_2}{2}}{1\ \Omega 10^6 \mu F}) \tag{25}$$

$$K_3 = 0.1(\frac{10\ V}{1\ \Omega} - \frac{k_2}{2 * 10^6 \mu F \Omega}) \tag{26}$$

Using k_2 from eqn.24 we can now calculate k_3

$$K_4 = hf(t + h, q_0 + k_3) = h(\frac{V_b}{R} - \frac{q_0 + k_3}{RC}) = 0.1(\frac{10\ V}{1\ \Omega} - \frac{q_0 + k_3}{10^6 \mu F \Omega}) \tag{27}$$

$$K_4 = 0.1(\frac{10\ V}{1\ \Omega} - \frac{k_3}{10^6 \mu F \Omega}) \tag{28}$$

We now have the values for $k_1 - k_4$ and can replace them in equation 29 to obtain the value for q_1,

$$q_1 = q_0 + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} \tag{29}$$

having found q_1, you have to follow the same procedure to find q_2 and so on. This might look a bit complicated but it can implemented very simple in C or other programming language, see appendix.

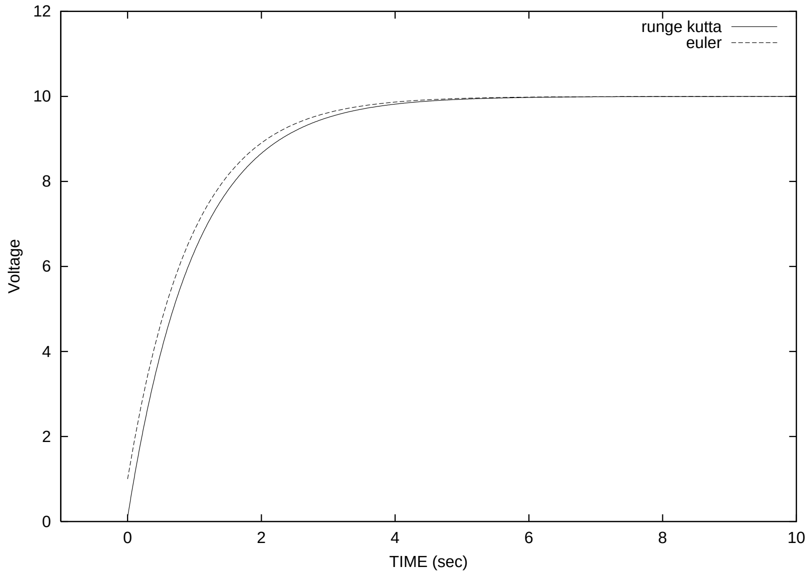

Figure 3: Comparison the Euler and Runge Kutta methods to solve the electric circuit example both with a time step $h = 0.1$

Figure 3 shows the convergence of both methods (Euler and Runge Kutta) using the same time step. Here we can see that the Runge Kutta method is much more precise than the Euler method having the same time step. Euler method can approximate Runge Kutta if the integration step used in Euler is much smaller than the one used in Runge Kutta. In figure 4 we can see clearly the effect of the time step in the convergence of the Runge Kutta method, we can see that as we decrease h the solution gets closer to the analytical solution, but you get to a point that decreasing the value of h results in huge computational time and the solution more accurate than what you really need. When plotting the analytical solution of the RC circuit problem shown in eqn.11 you can't tell the difference from the plot we obtained for $h = 0.001$.

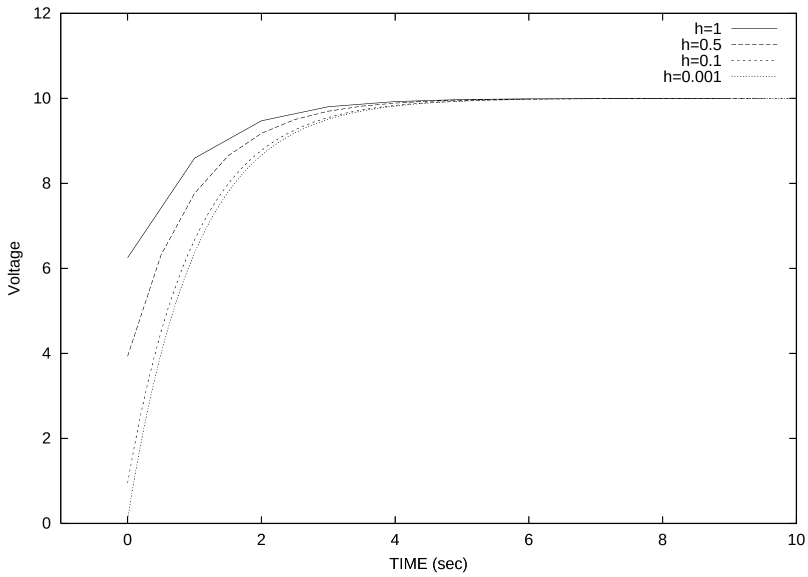

Figure 4: Comparing the convergence of the Runge Kutta method for different time steps.

3 Conclusion

I hope this can help anyone solve a differential equation, if they ever encounter one. I'm sure you will eventually find one, and solving it with a numerical method can be sometimes much faster than trying to solve it analytically which involves lots of maths skill you might not have. Make sure you understand what the equation means and what are you looking for before you try to solve it.

A Implementation of examples

In this section you are going to find some C programs that solve the examples above, I think this can illustrate in a better way how to solve recurrently with the help of a computer and a simple C program an integration of a differential equation problem.

A.1 EULER EXAMPLE

```
#include <stdio.h>

#define MAX 10 //integration time
#define h 0.1  //integration step


main()
{

        float x,t,v;

        v=3.7;
        x=4;

//printing nicely in a table
        printf("\t t \t x \n--------------------------------\n");

        for(t=0;t<MAX;t+=h) //main loop adds function f*h and increases t
        {

                x+=h*v;
                printf("\t%.2f\t%.2f\n",t,x);
        }

}
```

A.2 Euler-circuit

This is a C implementation of the example of the circuit mentioned before.

```
#include <stdio.h>

#define h 0.1 //time step

//problem parameters
//I have C=1 instead of 10^-6
// this changes the units of q from coulomb to micro Coulomb
#define C 1
#define R 1
```

```
#define Vb 10
#define TMAX 10


double derv(double ); //derivative function f(q,t) returns value f, receives q

main( )
{
  double t,q;
  FILE* solution;

  solution=fopen("euler.dat","w"); //this opens a file to store the solution

  t=0;//initial conditions
  q=0;//initial conditions

     for(t=0;t<TMAX;t+=h) // main loop increases time in the time step
        {
                        q+=h * derv(q);

        fprintf(solution,"%lf\t%lf\n",t,q); //prints this points in the file

/*********************************************************************************
 because V=q/c and c is constant we can say that 1 micro Coulonb = 1 Volt,
 so this means that in this particular case V=q, because q is given in micro
 coulomb and V can be given in volts.
*******************************************************************************/

        }
        fclose(solution);
}


double derv(double q) //function f(q,t)
{
                return  (Vb/R)-(q/(C*R));
}
```

A.3 Runge Kutta - circuit

Here is a C implementation of A Runge Kutta method to solve the circuit example.

```
#include <stdio.h>

#define h 0.01
#define C 1
 //I have C=1 instead of 10^-6
// this changes the units of q from coulomb to micro Coulomb
```

```
#define R 1
#define Vb 10
#define TMAX 10


double derv(double ); //function f(q,t)

main( )
{
  double k[4],t,q;
  FILE* solution;

  solution=fopen("rk.dat","w"); //this opens a file to store the solution

  t=0;//initial conditions
  q=0;//initial conditions

     for(t=0;t<TMAX;t+=h) // main loop increases time each time in the time step
        {

/* Here you calculate each of the runge kutta constants  with the previous value of q */

                        k[0]=h * derv(q);
                        k[1]=h * derv(q+k[0]/2);
                        k[2]=h * derv(q+k[1]/2);
                        k[3]=h * derv(q+k[2]);

                        q+=k[0]/6 + k[1]/3 + k[2]/3 + k[3]/6;    //increases q

        fprintf(solution,"%lf\t%lf\n",t,q); //prints this points in the file

/********************************************************************************
 because V=q/c and c is constant we can say that 1 micro Coulonb = 1 Volt,
 so this means that in this particular case V=q, because q is given in micro
 coulomb and V can be given in volts.http://www.q12.org/ode/ode.html
*******************************************************************************/


        }
        fclose(solution);
}


double derv(double q) //funtion f(q,t)
{
                return  (Vb/R)-(q/(C*R));
}
```

References

[1] Numerical Recipes Web Page. http://www.nr.com/

[2] Numerical Recipes Books online at Cornell chap.16, http://www.library.cornell.edu/nr/bookcpdf.html

[3] Good Solution for the RC circuit. http://circuitscan.homestead.com/files/ancircp/rcdiff1.htm

[4] Formal Computational Skills course notes. http://www.cogs.susx.ac.uk/courses/msc_fcs/notes.pdf

[5] Gear,C.W. 1971, Numerical Initial Value Problems in Ordinary Differential Equations (Englewood Cliffs, NJ: Prentice-Hall, Chapter 2).