Lecture 21: Numerical Solution of Differential Equations

Instructor: Professor Amos Ron Scribes: Yunpeng Li, Mark Cowlishaw, Nathanael Fillmore

1 Introduction

The last topic of this semester is numerical solution of differential equations. We will not be so much concerned with the theory of differential equations, but rather with algorithms for finding numerical approximations. Recall that differential equations are classified by order. For this discussion, we shall consider only first order differential equations.

In general, when trying to solve a first order differential equation, we are searching for a function $y(t)$ given some constraints expressed in terms of the independent variable t, the dependent variable y, and the derivative of y, y'. When we say that we are looking for a function $y(t)$, of course what we really mean is that we need to be able to determine the value of y at some point b (i.e. $y(b)$). To illustrate, consider the following problem:

EXAMPLE 1.1. *Determine $y(t)$ given the first order differential equation*

$$y'(t) = 2t \cdot y(t)$$

Solution: The trivial solution

$$y(t) = 0$$

is obvious, and it is not too difficult to find other solutions:

$$\begin{aligned} y(t) &= e^{t^2} \\ y(t) &= 2 \cdot e^{t^2} \end{aligned}$$

It turns out that there are infinitely many solutions to example 1.1, each of the form:

$$y(t) = k \cdot e^{t^2} \quad \text{for all } k \in \mathbb{R}$$

⊠

We can rewrite example 1.1 in terms of a function $f(t, y)$:

$$\begin{aligned} f(t, y) &= 2t \cdot y \\ y'(t) &= f(t, y(t)) \end{aligned}$$

As we noted, there are infinitely many solutions to this problem. In fact, the solutions to this problem set up a one-to-one correspondence for the entire 2-space defined by t and y. In other

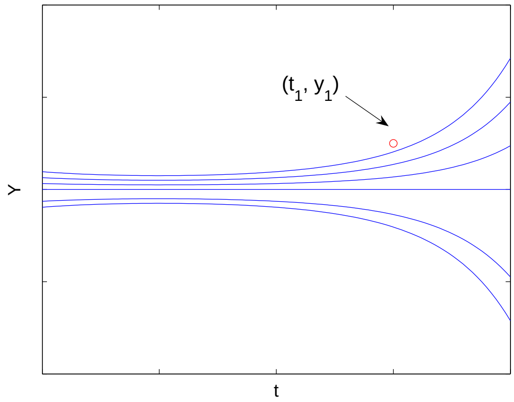

Figure 1: Solutions to $y'(t) = 2t \cdot y(t)$

words, for any point (t_1, y_1), there is a unique solution that passes through this point, as shown in Figure 1.

This means that example 1.1 poses an ill-formed problem, since infinitely many functions solve this differential equation, we have no way to pick one. Our analysis above shows the way to make this problem well-formed; since every point (t, y) has a unique solution that passes through it, if we provide a single data point with the differential equation, our problem will be well-formed.

In general, we will discuss problems of the form:

Problem 1.1 (Initial Value Problem). *Find $y(t)$ given*

$$\begin{aligned} y'(t) &= f(t, y(t)) \\ y(a) &= Y_a \end{aligned}$$

A problem of this form is called an *Initial Value Problem.* It is interesting to note that first order differential equations which are inherently difficult in terms of the theory of differential equations, such as:

$$f(t, y) = 2t^2y^{10} + 3t^3y$$

do not pose any special difficulty to numerical methods; since f is a polynomial, from a numerical perspective, this problem is not much different from a simpler problem such as example 1.1.

2 Solution of Initial Value Problems

To solve initial value problems, we will use numerical integration. Note that we can determine the difference between values of y using the definite integral of y'.

$$\int_a^b y'(t)dt = y(b) - y(a)$$

So, given an initial value problem with $y(a)$ defined and a differential equation for $y'(t)$, we can find $y(b)$ using:

$$y(b) = y(a) + \int_a^b y'(t)dt$$

We will use this idea to define an iterative method for estimating $y(b)$ given $y(a)$. As usual, we will split the interval between a and b into many, small, equal-sized subintervals, with partition points $(a = t_0, t_1, \ldots, t_N = b)$. Using numerical integration, we will generate values for y and y' at each partition point, until we reach the final value $y(t_n) = Y_n$. This process is depicted in Figure 2.

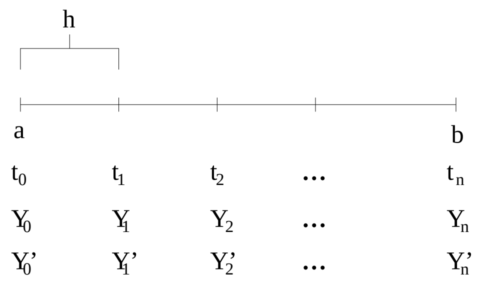

Figure 2: An Iterative Method for Solving Initial Value Problems

A generic step in this method consists of:

- Given $Y_0, Y_1, \cdots, Y_{i-1}$ and $Y'_0, Y'_1, \cdots, Y'_{i-1}$;
- Find (1) Y_i and (2) Y'_i.

Notice that, given Y_i, Y'_i can be easily computed using

$$Y'_i = f(t_i, Y_i)$$

Thus, our method really consists of finding each Y_i, then using this value to calculate Y'_i.

One way to solve this problem is to calculate each value Y_{i+1} based only on the previous point Y_i, that is:

$$y(t_i + 1) = y(t_i) + \underbrace{\int_{t_i}^{t_{i+1}} y'(t)dt}_{(*)}$$

but an obvious problem is how to calculate the definite integral $(*)$ when we only know the value of the function at the left endpoint $y'(t_i)$? Since the intervals are small, we may be able to use simple rules for approximating the definite integral to estimate this value. If we use the rectangle rule to approximate the definite integral $(*)$, then the resulting iterative process is known as *Euler's method*.

3 Euler's Method

Recall that the rectangle rule uses the value of the function at the left end-point of an interval to approximate the definite integral.

$$\int_{t_i}^{t_{i+1}} y'(t)dt \simeq Y_i' \cdot h$$

Euler's method applies the above idea to generate each Y_i for $0 \leq i \leq N-1$:

$$Y_{i+1} = Y_i + Y_i' \cdot h \quad \text{(for } i = 0, 1, \ldots, N-1\text{)}$$

then generates each Y_{i+1}' using the given differential equation:

$$Y_{i+1}' = f(t_{i+1}, Y_{i+1})$$

To illustrate this process, consider the following example.

EXAMPLE 3.1. *Given the following initial value problem*

$$\begin{aligned} y'(t) &= -3t^2 \cdot y(t) \\ y(0) &= 1 \end{aligned}$$

find $y(0.4)$.

Of course it is easy to see that $y(t) = e^{-t^3}$ is the solution. For the purpose of comparing methods of solving the IVP, we will solve this problem using an interval width of 0.1 and we will only compare the solution over the last interval (that is, given $Y_0 = y(0), Y_1 = y(0.1), Y_2 = y(0.2)$, and $Y_3 = y(0.3)$ find $Y_4 = y(0.4)$). We can calculate the initial values using $y(t) = e^{-t^3}$ and $y'(t) = -3t^2e^{-t^3}$:

$$\begin{aligned} Y_0 &= 1 & Y_0' &= 0 \\ Y_1 &= 0.999 & Y_1' &= -0.03 \\ Y_2 &= 0.992 & Y_2' &= -0.119 \\ Y_3 &= 0.973 & Y_3' &= -0.263 \\ Y_4 &= 0.938 \end{aligned}$$

Solution: Using Euler's method, we calculate $y(0.4)$ as:

$$\begin{aligned} y(0.4) &= y(0.3) + \int_{0.3}^{0.4} y'(t) \\ &\approx Y_3 + Y_3' \cdot 0.1 \\ &= 0.973 + (-0.263) \cdot 0.1 \\ &= 0.947 \end{aligned}$$

This yields an error of about 0.009, which is rather large for such a small, single step ⊠

4 Modified Euler's Method

How can we improve this estimate? Our knowledge of approximating the definite integral tells us that the midpoint rule does a far better job than the rectangle rule, so we might try applying the midpoint rule. However, there is a problem. The midpoint rule requires knowing the value of the function we are integrating at the *middle* of the interval, rather than at the end. This means that we need to know both the initial function value and the derivative value at the middle of the interval (or an approximation of it) to use the midpoint rule. The trick is to use an interval that is twice as large $[t_{i-1}, t_{i+1}]$, with t_i as the midpoint as our interval. Algorithms like this, that require values from more than one previous step to perform an iteration are called *multi-step* algorithms. Our estimate using the midpoint rule is given by:

$$\begin{aligned} Y_{i+1} &= Y_{i-1} + \int_{t_i}^{t_{i+1}} y(t)dt \\ &\approx Y_{i-1} + Y_i \cdot 2h \\ Y'_{i+1} &= f(t_{i+1}, Y_{i+1}) \end{aligned}$$

To illustrate, consider a solution to example 3.1 using the midpoint rule.

EXAMPLE 4.1. *Given the following initial value problem*

$$\begin{aligned} y'(t) &= -3t^2 \cdot y(t) \\ y(0) &= 1 \end{aligned}$$

find $y(0.4)$ *using the midpoint rule.*

Solution: Using the midpoint rule, we calculate Y_4 using an interval width 0.2:

$$\begin{aligned} y(0.4) &= y(0.2) + \int_{0.2}^{0.4} y'(t)dt \\ &\approx Y_2 + Y'_3 \cdot 2h \\ &= 0.992 + (-0.263)(2)(0.1) \\ &= 0.939 \end{aligned}$$

⊠

As we can see, this has error ≈ 0.001, an order of magnitude better than Euler's method.

5 Trapezoid Method

We might also try the trapezoid method. Recall that, using the trapezoid method, we interpolate the function we are integrating using a line between the endpoints, and estimate the integral as the area of the resulting trapezoid. Applying this idea to the initial value problem yields:

$$\begin{aligned} Y(t_{i+1}) &= y(t_i) + \int_{t_1}^{t_{i+1}} y'(t)dt \\ Y_{i+1} &\approx Y_i + \frac{Y'_i + Y'_{i+1}}{2} \cdot h \end{aligned}$$

However, this poses a problem. We do not know Y'_{i+1}, so how can we use it in our estimate? We might consider approximating Y'_{i+1} using a simpler method - if the error is not unreasonable, the multiplication by a small h should produce a small error. For example, we could use Euler's method to estimate Y'_{i+1}. Algorithms like this, which require us to estimate some of the input values before performing the calculation are called *closed* algorithms.

For example, consider solving example 3.1 using the trapezoid method, and using Euler's method to approximate Y'_4:

EXAMPLE 5.1. *Given the following initial value problem*

$$\begin{aligned} y'(t) &= -3t^2 \cdot y(t) \\ y(0) &= 1 \end{aligned}$$

find $y(0.4)$ *using the trapezoid rule.*

Solution: First, we need to calculate Y'_4 using Euler's method:

$$\begin{aligned} Y'_4 &= -3t^2 \cdot Y_4 \\ &= -3(0.4)^2(0.947) \\ &= -0.455 \end{aligned}$$

Using the trapezoid rule, we calculate Y_4 using :

$$\begin{aligned} y(0.4) &= y(0.3) + \int_{0.3}^{0.4} y'(t)dt \\ &\approx Y_3 + \frac{Y'_3 + Y'_4}{2} \cdot h \\ &= 0.973 + \frac{-0.263 + -0.455}{2}(0.1) \\ &= 0.937 \end{aligned}$$

⊠

As we can see, despite the error in the Euler's method approximation, the trapezoid rule produced a much smaller error of ≈ 0.001.

6 Summary of Methods

It should be clear from these examples that we can adapt each of our rules for approximating the definite integral into an iterative method for solving the IVP. These methods may be *closed*, as in the trapezoid rule, meaning that they require us to estimate some values at the right endpoint using another method, or *open* as in the rectangle rule, meaning that no values need to be estimated. Methods may also be *single step*, meaning that only values from the previous step are needed for the next step calculation, or *multi-step*, as in the midpoint rule, meaning that values from multiple previous steps are required for the next step calculation (giving us some problem in the first step). Here are some methods derived from the rules for approximating integrals, along with their categorization.

1. Euler's Method (rectangle rule) - Open, Single-step
2. Modified Euler's Method (midpoint rule) - Open, Multi-step
3. Trapezoid rule - Closed, Single Step
4. Simpson's Rule - Closed, Multi-Step

7 Iteration?

In class, a student asked whether we should perhaps iterate in order to get a better solution. Although this may seem like a good idea initially, one has to make a careful analysis of the costs of doing so, relative to just using a smaller step size. The cost of any method is dominated by the number of times we need to evaluate the differential equation. It turns out that it is better to shrink the step size than to iterate.