CS412: INTRODUCTION TO NUMERICAL ANALYSIS 12/9/10

Lecture 25: Numerical Solution of Differential Equations — Error Analysis

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

1 Error Analysis for Solving IVP

For the last several lectures, we have studied methods for solving initial value problems. In general, we are trying to find $y(b)$ given:

$$\begin{aligned} y'(t) &= f(t, y(t)) \quad \forall t; \\ y(a) &= y_a \end{aligned}$$

Recall that the general approach is to partition the interval $[a, b]$ into N equal-width sub-intervals using points $a = t_0, t_1, \cdots t_N = b$ and interval width $h = (b - a)/N$. We then estimate $y(t_{j+1})$ at each partition point using the definite integral:

$$y(t_{j+1}) = y(t_j) + \int_{t_j}^{t_{j+1}} y'(t)dt \tag{1}$$

Since we cannot compute the integral directly, we use a numerical integration rule to estimate the value of y at each partition point t_j. We can then use the function f to estimate y' at each new partition point. To end our discussion of numerical solution of differential equations, we will discuss the error in our solution method.

The error in our solution method has three fundamental sources:

1. The error caused by using approximate input values to the differential equation. We call this *propagation error*
2. The error caused by using approximate input values to our numerical integration methods.
3. The error of our numerical integration method (if we assume perfect input). This is known as the *truncation error* or *discretization error.*

A detailed quantitative treatment of the error is beyond the scope of this class. Instead, we shall provide some intuition for these sources of error and how significant they are.

1.1 Truncation Error

We have exact error formulas for each of our numerical integration methods. For example, Simpson's rule has error formula:

$$\epsilon_S = \underbrace{-\frac{f^{(4)}(c)}{2880}}_{(*)} \cdot h^5 \tag{2}$$

However, we have no way of estimating $(*)$ over each of the intervals in our solution method, so we will ignore this part and note our requirement that the function is well-behaved enough that this value is not too large. Instead, we shall concentrate on the dominant term, and so we shall estimate the error using big-O notation, $\epsilon_S = O(h^5)$. Thus, We say that Simpson's rule has $O(h^5)$ error over a single interval. Since this error occurs over N separate intervals, summing up these errors this gives a total error over $[a, b]$ of $O(h^4)$.

In general, a numerical integration method with error $O(h^a)$ will have a total truncation error of $O(h^{a-1})$ over the entire interval $[a, b]$.

Table 1 shows the local and global truncation error for the methods we have studied.

Method	**Local Error**	**Global Error**
Euler	$O(h^2)$	$O(h)$
Modified Euler	$O(h^3)$	$O(h^2)$
Trapezoid	$O(h^3)$	$O(h^2)$
Simpson, Milne, A-B, A-M	$O(h^5)$	$O(h^4)$
Runge-Kutta 4	$O(h^5)$	$O(h^4)$

Table 1: Local and Global Truncation Errors for IVP Solution Methods

1.2 Approximate Input to Numerical Integration Methods

To see the effect of approximate input on our numerical integration methods, assume that we are at some point t_j near the midpoint of the interval $[a, b]$. Assume further that we are using Simpson's rule for our calculations. This means that each of our input values will have an error of $O(h^4)$. How will these errors affect our calculations over the interval $[t_{j-1}, t_{j+1}]$? Consider the formula for Simpson's rule:

$$Y_{j+1} = Y_{j-1} + \frac{h}{6}\left(Y'_{j-1} + 4Y'_j + Y'_{j+1}\right)$$

Note that each of our calculated values (Y'_{j-1}, Y'_j, Y'_{j+1}) is multiplied by h, so that, if they have error $O(h^4)$, the error will be multiplied by a factor of h, becoming $O(h^5)$, *exactly the same as the truncation error*. Thus, the truncation error dominates the error from using approximate input, so that we may safely ignore this source of error.

1.3 Propagation Error

We can define the propagation error as the difference between using approximate and exact input to the differential equation:

$$\epsilon_{\text{propagation}}(t_j) = f(t_j, Y_j) - f(t_j, y(t_j))$$

This error will depend greatly on the function f. We measure the effect of f using the partial derivative of f with respect to y. For example, consider the functions f_1 and f_2:

$$y'(t) = f_1(t) = y \cdot e^{t^2} \quad (3)$$

$$y'(t) = f_2(t) = t \cdot e^{y^2} \quad (4)$$

Taking the partial derivative for f_1 (equation 3) with respect to y yields:

$$\frac{\partial f_1}{\partial y} \quad = \quad e^{t^2}$$

this expression does not depend at all on y, so that small perturbations in y should have little impact. However, if we take the partial derivative of f_2 (equation 4) with respect to y, we get:

$$\frac{\partial f_2}{\partial y} \quad = \quad 2tye^{y^2}$$

Clearly, small differences in y will have a huge impact on the output of f_2, meaning that this differential equation cannot be accurately solved using our method, no matter how good the numerical integration rule is.

In general, the best we can hope for is that the propagation error will be of the same order as the truncation error of our numerical integration method (or better). Given a reasonably well behaved differential equation, this will be the case.

2 Predictor-Corrector Methods

Given this analysis, how do we determine the error of predictor-corrector methods? Assuming that propagation error is under control, and assuming that the predictor method is *good enough*, it is logical that the error is determined by the error of the corrector method. The question is, just how good does the predictor method have to be? In other words, what relationship do we require between the error of the predictor method and the error of the corrector method?

To illustrate, consider the predictor-corrector method with Euler's method as the predictor and Trapezoid as the corrector. Considering the interval $[t_j, t_{j+1}]$, we calculate an estimate of Y_{j+1} using Euler's method:

$$\tilde{Y}_{j+1} \quad = \quad Y_j + hY'_j$$

Based on the error for Euler's method, $\tilde{Y}_{j+1}$ will have error $O(h^2)$. We use this value to estimate Y'_{j+1} using the differential equation:

$$\tilde{Y}'_{j+1} \quad = \quad f(t_{j+1}, \tilde{Y}_{j+1})$$

Assuming that propagation error is under control, our estimate of the derivative $\tilde{Y}'_{j+1}$ will also have error $O(h^2)$. When we plug this value into the Trapezoid corrector, we get the final approximation Y_{j+1}:

$$Y_{j+1} \quad = \quad Y_j + \frac{h}{2}\left(Y'_j + \tilde{Y}'_{j+1}\right)$$

Since $\tilde{Y}'_{j+1}$ has error $O(h^2)$ and we multiply the value by h, the resulting error is $O(h^3)$, which is the same order as the truncation error for the Trapezoid rule. Therefore, Euler's method and the Trapezoid rule are compatible as predictor and corrector.

In general, if a corrector rule has error $O(h^a)$, the associated predictor rule must have error $O(h^{a-1})$ or better.

(One may wonder: why do we match $O(h^5)$ with $O(h^5)$ instead of $O(h^4)$ with $O(h^5)$, like we do in the Adams-Bashforth–Adams-Moulton pair? The answer is that we might as well do so, since it comes for free. Recall that the main cost to any method is the number of function evaluations, and using the same evaluations we used to get Adams-Moulton to $O(h^5)$, we can also get Adams-Bashforth to $O(h^5)$ for free. Doing so helps to ensure that the input error is negligible.)

3 Adaptive Methods

In practice, when solving an IVP, the value of h is often changed on the fly, adapting to the amount of error in the calculation. So, how can we determine the error as we calculate? If we use a predictor-corrector method, such as the AB-AM predictor-corrector, then we can use the difference between the prediction and correction at a point t_j:

$$\epsilon \quad \approx \quad \left|Y_j^{AB} - Y_j^{AM}\right|$$

If this difference is large, then the error is large and we must shrink h. If this difference is small, we can conclude that the error is small.

This is the primary reason that people use predictor-corrector methods like Adams-Bashforth–Adams-Moulton in preference to Runge-Kutta methods: both classes of methods give similar error bounds, but the predictor-corrector methods also give you a *practical* way to control the error.

4 Boundary Value Problems

Finally, we return to our discussion of second (and higher) order differential equations. Recall that we introduced two correct formulations for solving a second order differential equation, the initial value problem and the *boundary value problem.* A boundary value problem is defined as follows:

Problem 4.1. *Given a second order differential equation and boundary values:*

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

Find $y(c)$, with $c \in [a, b]$.

One method for solving boundary value problems is the *shooting* method:

1. Guess a value Y'_a for $y'(a)$.

2. Using this guess, use IVP methods to find Y_b.

3. If Y_b is sufficiently close to $y(b)$, take the value calculated at c (Y_c) as the answer.

4. Otherwise, correct Y'_a and start again.

This method clearly shows the relative difficulty of solving a boundary value problem — the solution of one boundary value problem has similar complexity to solving many initial value problems.