Numerical approximations of solutions of ordinary differential equations

Anotida Madzvamuse

Department of Mathematics
Pevensey III, 5C15,
Brighton, BN1 9QH,
UK

THE FIRST MASAMU ADVANCED STUDY INSTITUTE AND WORKSHOPS IN MATHEMATICAL SCIENCES
December 1 - 14, 2011
Livingstone, Zambia

Outline

Applications of ODEs

Ordinary differential equations (ODEs) are a fundamental tool in

- Applied mathematics,
- mathematical modelling

They can be found in the modelling of

- biological systems,
- population dynamics,
- micro/macroeconomics,
- game theory,
- financial mathematics.

They also constitute an important branch of mathematics with applications to different fields such as geometry, mechanics, partial differential equations, dynamical systems, mathematical astronomy and physics.

Applications of ODEs

The problem consists of finding $y : \mathbb{I} \longrightarrow \mathbb{R}$ such that it satisfies the differential equation

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

and the initial condition

$$y(x_0) = y_0. \tag{2}$$

The above is known as an initial value problem (IVP)

The need for computations

- Note that an analytical solution of (1)-(2) can be found only in very particular situations which are usually quite simple ones.
- In general, especially in equations that are of modelling relevance, there is no systematic way of writing down a formula for the function $y(x)$.
- Therefore, in applications where the quantitative knowledge of the solution is fundamental one has to turn to a numerical (i.e., digital or computer) approximation of $y(x)$. This is a computational mathematics problem. There are three main questions raised by a computational mathematics problem, such as ours.

Discretisation

- The first question is about our ability to come up with a computable version of the problem. For instance, in (1) there are derivatives that appear, but on a computer a derivative (or an integral) cannot be evaluated exactly and it needs to be replaced by some approximation.
- Also, there is a continuous infinity of time instants between x_0 and $x_0 + T > x_0$, it is not possible to determine (not even to approximate) $y(x)$ for each $x \in [x_0, x_0 + T)$ and one has to settle for a finite number of points $x_0 < x_1 < \cdots < x_N = T$.
- The process of transforming a continuous problem (which involves continuous infinity of operations) into a finite number of operations is called discretisation and constitutes the first phase of establishing a computational method.

Numerical Analysis

- The second important question regarding a computational method is to understand whether it yields the wanted results. We want to guarantee that the figures that the computer will output are really related to the problem.
- It must be checked mathematically that the discrete solution (i.e., the solution of the discretised problem) is a good approximation of the problem and deserves the name of approximate solution.
- This is usually done by conducting an error analysis, which involves concepts such as **stability, consistency** and **convergence**.

Efficiency and implementation

- Finally, the third important question regarding a computational method is that of efficiency and its actual implementation on a computer. By efficiency, roughly speaking, we mean the amount of time that we should be waiting to compute the solution. It is very easy to write an algorithm that computes a quantity, but it is less easy to write one that will do it effectively.
- Once a discretisation is found to be convergent and to have an acceptable level of efficiency, it can be implemented by using a computer language, and used for practical purposes.

Deliverables of the lecture series

- The main goal of this course is to derive, understand, analyse and implement effective numerical algorithms that provide (good) approximations to the solution y of problem (1)-(2).
- A theoretical stream in which we derive and analyse the various methods
- A practical stream where these methods are coded on a computer using easy progamming languages such as Matlab or Octave (a free and legal competitor of Matlab with very similar syntax) or Scilab (also free and very good quality software with a Matlab-equivalent syntax).
- The implementation of methods is fundamental in understanding and appreciating the methods and it provides a good feeling of reward once a numerical method is (successfully) seen in action.

Picard's theorem

In general, even if $f(x, y(x))$ is a continuous function, there is no guarantee that the initial value problem (1)-(2) possesses a unique solution. Fortunately, under a further mild condition on the function $f(x, y(x))$, the existence and uniqueness of a solution to (1)-(2) can be ensured: the result is encapsulated in the next theorem.

Picard's Theorem I

Picard's Theorem II

Theorem (Picard's Theorem)

Suppose that $f(\cdot,\cdot)$ is a continuous function of its arguments in a region U of the (x,y) plane which contains the rectangle

$$R = \Big\{(x,y) : x_0 \leq x \leq X_M, \quad |y - y_0| \leq Y_M\Big\},$$

where $X_M > x_0$ and $Y_M > 0$ are constants. Suppose that $\exists\ L > 0$:

$$|f(x,y) - f(x,z)| \leq L|y - z|,\ \forall (x,y),\ (x,z) \in R. \tag{3}$$

Suppose $M = max\Big\{|f(x,y)| : (x,y) \in R\Big\}$, with $M(X_M - x_0) \leq Y_M$. Then $\exists$ a unique continuously differentiable function $x \longrightarrow y(x)$, defined on the closed interval $[x_0, X_M]$, which satisfies (1) and (2).

Picard's Theorem: Conceptual Proof I

The essence of the proof is to consider the sequence of functions $\left\{y_n\right\}_{n=0}^{\infty}$, defined recursively through what is known as the Picard Iteration:

$$\begin{cases} y_0(x) = y_0, \\ y_n(x) = y_0 + \int_{x_0}^{x} f(\xi, y_{n-1}(\xi))d\xi, \quad n = 1, 2, \cdots, \end{cases} \tag{4}$$

and show, using the conditions of the theorem, that $\left\{y_n\right\}_{n=0}^{\infty}$ converges uniformly on the interval $[x_0, X_M]$ to a function y defined on $[x_0, X_M]$ such that

$$y_n(x) = y_0 + \int_{x_0}^{x} f(\xi, y_{n-1}(\xi))d\xi.$$

Picard's Theorem: Conceptual Proof II

This then implies that y is continuously differentiable on $[x_0, X_M]$ and it satisfies the differential equation (1) and the initial condition (2). The uniqueness of the solution follows from the Lipschitz condition.

Picard's Theorem: System of ODEs

Picards Theorem has a natural extension to an initial value problem for a system of m differential equations of the form

$$\begin{cases} \mathbf{y}' = \mathbf{f}(x, \mathbf{y}), \\ \\ \mathbf{y}(x_0) = \mathbf{y}_0, \end{cases} \tag{5}$$

where $y_0 \in \mathbb{R}^m$ and $\mathbf{f} : [x_0, X_M] \times \mathbb{R}^m \longrightarrow \mathbb{R}^m$.

Picard's Theorem: System of ODEs

On introducing the Euclidean norm $\|.\|$ on $\mathbf{R}^m$

$$\|\mathbf{v}\| = \left(\sum_{i=1}^{m} |v_i|^2 \right)^{\frac{1}{2}}, \quad \mathbf{v} \in \mathbf{R}^m$$

we can state the following result.

Picard's Theorem: System of ODEs

Theorem (Picard's theorem)

Suppose that $\mathbf{f}(\cdot,\cdot)$ *is a continuous function of its arguments in a region* U *of the* $(x,\mathbf{y})$ *space* $\mathbb{R}^{1+m}$ *which contains the parallelepiped* $R = \left\{(x,\mathbf{y}) : x_0 \leq x \leq X_M, \quad |\mathbf{y}-\mathbf{y}_0| \leq Y_M\right\}$, *where* $X_M > x_0$ *and* $Y_M > 0$ *are constants. Suppose also that there exists a positive constant* L *such that*

$$|\mathbf{f}(x,\mathbf{y}) - \mathbf{f}(x,\mathbf{z})| \leq L|\mathbf{y}-\mathbf{z}| \tag{6}$$

holds whenever $(x,\mathbf{y})$ *and* $(x,\mathbf{z})$ *lie in* R. *Finally, letting* $M = max\{|\mathbf{f}(x,\mathbf{y})| : (x,\mathbf{y}) \in R\}$, *suppose that* $M(X_M - x_0) \leq Y_M$. *Then there exists a unique continuously differentiable function* $x \longrightarrow \mathbf{y}(x)$, *defined on the closed interval* $[x_0, X_M]$, *which satisfies* (5).

Picard's Theorem: System of ODEs

A sufficient condition for (6) is that $\mathbf{f}$ is continuous on R, differentiable at each point $(x, \mathbf{y})$ in $int(R)$, the interior of R, and there exists $L > 0$ such that

$$\left\| \frac{\partial \mathbf{f}}{\partial \mathbf{y}}(x, \mathbf{y}) \right\| \leq L \quad \text{for all } (x, \mathbf{y}) \in int(R), \tag{7}$$

where $\frac{\partial \mathbf{f}}{\partial \mathbf{y}}$ denotes the $m \times m$ Jacobi matrix of $\mathbf{y} \in \mathbb{R}^m \longrightarrow \mathbf{f}(x, \mathbf{y}) \in \mathbb{R}^m$, and $\| \cdot \|$ is a matrix norm subordinate to the Euclidean vector norm on $\mathbb{R}^m$. Indeed, when (7) holds, the Mean Value Theorem implies that (6) is also valid. The converse of this statement is not true; for the function $\mathbf{f}(\mathbf{y}) = (|y_1|, \cdots, |y_m|)^T$, with $x_0 = 0$ and $\mathbf{y}_0 = 0$, satisfies (6) but violates (7) because $\mathbf{y} \longrightarrow \mathbf{f}(\mathbf{y})$ is not differentiable at the point $\mathbf{y} = 0$.

Picard's Theorem: System of ODEs

Definition (Stability of solutions)

A solution $\mathbf{y} = \mathbf{v}(x)$ to (5) is said to be stable on the interval $[x_0, X_M]$ if for every $\epsilon > 0$ there exists $\delta > 0$ such that for all $\mathbf{z}$ satisfying $|\mathbf{v}(x_0) - \mathbf{z}| < \delta$ the solution $\mathbf{y} = \mathbf{w}(x)$ to the differential equation $\mathbf{y}' = \mathbf{f}(x, \mathbf{y})$ satisfying the initial condition $\mathbf{w}(x_0) = \mathbf{z}$ is defined for all $x \in [x_0, X_M]$ and satisfies $|\mathbf{v}(x) - \mathbf{w}(x)| < \epsilon$ for all $x \in [x_0, X_M]$.
A solution which is stable on $[x_0, \infty)$ (i.e. stable on $[x_0, X_M]$ for each X_M and with δ independent of X_M) is said to be stable in the sense of Lyapunov. Moreover, if

$$lim_{x \longrightarrow \infty} |\mathbf{v}(x) - \mathbf{w}(x)| = 0,$$

then the solution $\mathbf{y} = \mathbf{v}(x)$ is called asymptotically stable.

Picard's Theorem: System of ODEs

Theorem (Stability of solutions)

Under the hypotheses of Picards theorem, the (unique) solution $\mathbf{y} = \mathbf{v}(x)$ *to the initial value problem* (5) *is stable on the interval* $[x_0, X_M]$, *(where we assume that* $-\infty < x_0 < X_M < \infty$*).*

Proof.

Exercise □

Euler's method and its relatives: the θ-method

Suppose that the IVP (1)-(2) is to be solved on $[x_0, X_M]$.

- We divide this interval by the mesh-points $x_n = x_0 + nh, n = 0, \cdots, N$, where $h = \frac{(X_M - x_0)}{N}$ and $N \in \mathbb{Z}^+$.
- The positive real number h is called the step size.
- Now let us suppose that, for each n, we seek a numerical approximation y_n to $y(x_n)$, the value of the analytical solution at the mesh point x_n. Given that $y(x_0) = y_0$ is known, let us suppose that we have already calculated y_n, up to some n, $0 \leq n \leq N - 1$; we define

$$y_{n+1} = y_n + hf(x_n, y_n), \quad n = 0, \cdots, N - 1. \tag{8}$$

Thus, taking in succession $n = 0, 1, \cdots, N - 1$, one step at a time, the approximate values y_n at the mesh points x_n can be easily obtained. This numerical method is known as the **Eulers method**.

Euler's method: Method II

Suppose the function $y(x)$ is twice continuously differential with respect to x. By Taylor's Theorem we have

$$y(x_{n+1}) = y(x_n) + hy'(x_n) + O(h^2) \tag{9}$$

hence if $h << 1$ then we can write

$$y_{n+1} \approx y_n + hf(x_n, y_n)$$

where we have neglected $O(h^2)$ and higher order terms.

Euler's method: Method III

Integrating (1) between two consecutive mesh points x_n and x_{n+1} to deduce that

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(x, y(x))dx, \quad n = 0, \cdots, N-1, \quad (10)$$

and then applying the numerical integration rule

$$\int_{x_n}^{x_{n+1}} g(x)dx \approx hg(x_n),$$

called the rectangle rule, with $g(x) = f(x, y(x))$, to get

$$y(x_{n+1}) \approx y(x_n) + hf(x_n, y(x_n)), \quad n = 0, \cdots, N-1, \quad y(x_0) = y0.$$

The Euler's Method: the θ-method

Replacing the rectangle rule in the derivation of Eulers method with a one-parameter family of integration rules of the form

$$\int_{x_n}^{x_{n+1}} g(x)dx \approx h[(1-\theta)g(x_n) + \theta g(x_{n+1})], \qquad (11)$$

with $\theta \in [0,1]$ a parameter. On applying this with $g(x) = f(x, y(x))$ we find that

$$\begin{cases} y(x_{n+1}) \approx y(x_n) + h[(1-\theta)f(x_n, y(x_n)) + \theta f(x_{n+1}, y(x_{n+1}))], & n = 0, \\ \\ y(x_0) = y_0. \end{cases} \qquad (12)$$

The Euler's Method: the θ-method

This then motivates the introduction of the following one-parameter family of methods: given that y_0 is supplied by (2), define

$$y_{n+1} = y_n + h[(1-\theta)f(x_n, y_n) + \theta f(x_{n+1}, y_{n+1})],\ n = 0, \cdots, N-1. \tag{13}$$

parametrised by $\theta \in [0, 1]$; (13) is called the θ-method. Now, for $\theta = 0$ we recover Explicit Eulers method.

The Implicit Euler's Method

For $\theta = 1$, and y_0 specified by (2), we get

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}),\ n = 0 \cdots, N-1, \qquad (14)$$

referred to as the **Implicit Euler Method** since, unlike Eulers method considered above, (14) requires the solution of an implicit equation in order to determine y_{n+1}, given y_n.

The Trapezium Rule method: $\theta = \frac{1}{2}$

The scheme which results for the value of $\theta = \frac{1}{2}$ is also of interest: y_0 is supplied by (2) and subsequent values y_{n+1} are computed from

$$y_{n+1} = y_n + \frac{1}{2}h[f(x_n, y_n) + f(x_{n+1}, y_{n+1})], \quad n = 0, \cdots N-1; \ (15)$$

this is called the **Trapezium Rule Method**.

Example 1

Given the IVP

$$y' = x - y^2, \quad y(0) = 0,$$

on the interval of $x \in [0, 0.4]$, we compute an approximate solution using the θ-method, for $\theta = 0$, $\theta = \frac{1}{2}$ and $\theta = 1$, using the step size $h = 0.1$. The results are shown in Table 1. In the case of the two implicit methods, corresponding to $\theta = \frac{1}{2}$ and $\theta = 1$, the nonlinear equations have been solved by a fixed-point iteration. For comparison, we also compute the value of the analytical solution $y(x)$ at the mesh points $x_n = 0.1 \times n$, $n = 0, \cdots, 4$. Since the solution is not available in closed form, we use a Picard iteration to calculate an accurate approximation to the analytical solution on the interval $[0, 0.4]$ and call this the *exact solution*.

Table 1

k	x_k	y_k for $\theta = 0$	y_k for $\theta = \frac{1}{2}$	y_k for $\theta = 1$
0	0	0	0	0
1	0.1	0	0.005	0.00999
2	0.2	0.01	0.01998	0.02990
3	0.3	0.02999	0.04486	0.05955
4	0.4	0.05990	0.07944	0.09857

Table: The values of the numerical solution at the mesh points.

Picard's Iteration

Given $y(0) = 0$ we compute the following sequence

$$y_k(x) = y(0) + \int_0^x (\xi - y_{k-1}(\xi))d\xi, \ k = 1, 2, \cdots,$$

to obtain

$$\begin{aligned}
y_0(x) &= 0, \\
y_1(x) &= \frac{1}{2}x^2, \\
y_2(x) &= \frac{1}{2}x^2 - \frac{1}{20}x^5, \\
y_3(x) &= \frac{1}{2}x^2 - \frac{1}{20}x^5 + \frac{1}{160}x^8 - \frac{1}{4400}x^{11}.
\end{aligned}$$

It is easy now to prove by induction that

$$y(x) = \frac{1}{2}x^2 - \frac{1}{20}x^5 + \frac{1}{160}x^8 - \frac{1}{4400}x^{11} + O(x^{14}).$$

Tabulating results

Tabulating $y_3(x)$ on the interval $[0, 0.4]$ with step size $h = 0.1$, we get the *exact solution* at the mesh points shown in Table 2.
The *exact solution* is in good agreement with the results obtained with $\theta = \frac{1}{2}$: the *error* $\leq 5 \times 10^{-5}$. For $\theta = 0$ and $\theta = 1$ the discrepancy between y_k and $y(x_k)$ is larger: *error* $\leq 3 \times 10^{-2}$.

k	x_k	$y(x_k)$
0	0	0
1	0.1	0.005
2	0.2	0.01998
3	0.3	0.04488
4	0.4	0.07949

Table: Values of the "exact solution" at the mesh points.

MAPLE: Exact Solution

We note in conclusion that a plot of the analytical solution can be obtained, for example, by using the MAPLE package by typing in the following at the command line:

First we have to explain what we mean by error.

- The exact solution of the initial value problem (1)-(2) is a function of a continuously varying argument $x \in [x_0, X_M]$, while the numerical solution y_n is only defined at the mesh points x_n, $n = 0, \cdots, N$, so it is a function of a *discrete* argument.
- We can compare these two functions either by extending in some fashion the approximate solution from the mesh points to the whole of the interval $[x_0, X_M]$ (say by interpolating between the values y_n), or by restricting the function y to the mesh points and comparing $y(x_n)$ with y_n for $n = 0, \cdots, N$.
- Since the first of these approaches is somewhat arbitrary because it does not correspond to any procedure performed in a practical computation, we adopt the second approach, and we define the global error e by

$$e_n = y(x_n) - y_n, \quad n = 0, \cdots, N.$$

$\theta = 0$

We wish to investigate the decay of the global error for the θ-method with respect to the reduction of the mesh size h.

- From the Explicit Euler's method

$$y_{n+1} = y_n + hf(x_n, y_n), \quad n = 0, \cdots, N, \quad y_0 \text{ given}$$

we can define the **Truncation erorr** by

$$T_n = \frac{y(x_{n+1}) - y(x_n)}{h} - f(x_n, y(x_n)), \tag{16}$$

obtained by inserting the analytical solution y(x) into the numerical method and dividing by the mesh size.

- Indeed, it measures the extent to which the analytical solution fails to satisfy the difference equation for Eulers method.

$\theta = 0$ Continued

By noting that $f(x_n, y(x_n)) = y'(x_n)$ and applying Taylors Theorem, it follows from (16) that there exists $\xi_n \in (x_n, x_{n+1})$ such that

$$T_n = \frac{1}{2} h y''(\xi_n) \tag{17}$$

where we have assumed that that f is a sufficiently smooth function of two variables so as to ensure that y'' exists and is bounded on the interval $[x_0, X_M]$. Since from the definition of Eulers method

$$0 = \frac{y_{n+1} - y_n}{h} - f(x_n, y_n),$$

on subtracting this from (16), we deduce that

$$e_{n+1} = e_n + h[f(x_n, y(x_n)) - f(x_n, y_n)] + hT_n.$$

$\theta = 0$ Continued

Thus, assuming that $|y_n - y_0| \leq Y_M|$ from the Lipschitz condition (3) we get

$$|e_{n+1}| \leq (1 + hL)|e_n| + h|T_n|, \ n = 0, \cdots, N-1.$$

Now, let $T = max_{0 \leq n \leq N-1}|T_n|$; then,

$$|e_{n+1}| \leq (1 + hL)|e_n| + hT, \ n = 0, ..., N-1.$$

By induction, and noting that $1 + hL \leq e^{hL}$,

$$\begin{aligned} |e_n| &\leq \frac{T}{L}[(1 + hL)^n - 1] + (1 + hL)^n|e_0| \\ &\leq \frac{T}{L}\left(e^{L(x_n - x_0)} - 1\right) + e^{L(x_n - x_0)}|e_0|, \ n = 1, \cdots, N. \end{aligned}$$

$\theta = 0$ Continued

This estimate, together with the bound

$$|T| \leq \frac{1}{2} h M_2, \text{ where } M_2 = max_{x \in [x_0, x_M]} |y''(x)|,$$

which follows from (17), yields

$$|e_n| \leq e^{L(x_n - x_0)} |e_0| + \frac{M_2 h}{2L} \left(e^{L(x_n - x_0)} - 1 \right), \ n = 0, \cdots, N. \quad (18)$$

Analogously, for the general θ-method we can prove that

$$|e_n| \leq |e_0| exp \left(L \frac{x_n - x_0}{1 - \theta L h} \right) + \frac{h}{L} \left\{ \left| \frac{1}{2} - \theta \right| M_2 + \frac{1}{3} h M_3 \right\} \left| \exp \left(L \frac{x_n - x_0}{1 - \theta L h} \right) - 1 \right|, \quad (19)$$

for $n = 0, \cdots, N$ where now $M_3 = max_{x \in [x_0, x_M]} |y'''(x)|$.

Orders of Convergence

W.L.O.G. suppose that $e_0 = y(x_0) - y_0 = 0$. Then it follows that

- $\theta = \frac{1}{2}$, then $|e_n| = O(h^2)$.
- $\theta = 0$ and $\theta = 1$ (or any $\theta \neq \frac{1}{2}$), then $|e_n| = O(h)$.

Hence at each time step:

- $\theta \neq \frac{1}{2}$: the mesh size h is halved, the truncation and global errors are reduced by a factor of 2,
- $\theta = \frac{1}{2}$: these are reduced by a factor of 4.
- **Price we pay**?

Improved Explicit Euler's Method

- It is less convenient, it requires the solution of implicit equations at each mesh point x_{n+1} to compute y_{n+1}.
- An attractive compromise is to use the forward Euler method to compute an initial crude approximation to $y(x_{n+1})$ and then use this value within the trapezium rule to obtain a more accurate approximation for $y(x_n + 1)$: the resulting numerical method is

$$\begin{cases} y_{n+1} = y_n + hf(x_n, y_n), \\ y_{n+1} = y_n + \frac{1}{2}h[f(x_n, y_n) + f(x_{n+1}, y_{n+1})], \end{cases} \tag{20}$$

  for $n = 0, \cdots, N,\ y_0 = \text{given}$.
- Frequently referred to as the **Improved Explicit Euler's method**.

General explicit one-step method

A general explicit one-step method may be written in the form:

$$y_{n+1} = y_n + h\Phi(x_n, y_n; h),\ n = 0, \cdots, N-1,\ y_0 = y(x_0), \quad (21)$$

where $\Phi(\cdot, \cdot; \cdot)$ is a continuous function of its variables. E.g. in the case of Explicit Euler's method:

$$\Phi(x_n, y_n; h) = f(x_n, y_n),$$

while for the Improved Explicit Euler's method

$$\Phi(x_n, y_n; h) = \frac{1}{2}[f(x_n, y_n) + f(x_n + h, y_n + hf(x_n, y_n))].$$

In order to assess the accuracy of the numerical method (21), we define the global error, by

$$e_n = y(x_n) - y_n.$$

Truncation error

$$T_n = \frac{y(x_{n+1}) - y(x_n)}{h} - \Phi(x_n, y(x_n); h. \tag{22}$$

A bound on the global error in terms of T_n

Theorem

Consider the general one-step method (22) where, in addition to being a continuous function of its arguments, Φ is assumed to satisfy a Lipschitz condition with respect to its second argument; namely, there exists a positive constant L_Φ such that, for $0 \leq h \leq h_0$ and for the same region R as in Picards theorem,

$$|\Phi(x, y; h) - \Phi(x, z; h)| \leq L_\Phi |y - z|, \text{ for } (x, y), (x, z) \text{ in } R. \quad (23)$$

Then, assuming that $|y_n - y_0| \leq Y_M$, it follows that

$$|e_n| \leq e^{L_\Phi (x_n - x_0)} |e_0| + \left[\frac{e^{L_\Phi (x_n - x_0)} - 1}{L_\Phi} \right] T, \; n = 0, \cdots N, \quad (24)$$

where $T = max_{0 \leq n \leq N-1} |T_n|$.

Proof

Proof.

Exercise □

Example 2: Consider

$$\begin{cases} y' = tan^{-1}y, \\ \\ y(0) = y_0. \end{cases} \tag{25}$$

The aim of the exercise is to apply (24) to quantify the size of the associated global error; thus, we need to find L and M_2. Here $f(x, y) = tan^{-1}y$, so by the Mean Value Theorem

$$|f(x, y) - f(x, z)| = \left|\frac{\partial f}{\partial y}(x, \eta)(y - z)\right|$$

where $y < \eta < z$.

Example 2 Cont'd

In our case

$$\left|\frac{\partial f}{\partial y}\right| = |(1+y^2)^{-1}| \leq 1,$$

and therefore $L = 1$. To find M_2 we need to obtain a bound on $|y''|$ (without actually solving the initial value problem!). This is easily achieved by differentiating both sides of the differential equation with respect to x:

$$y'' = \frac{d}{dx}(tan^{-1}y) = (1+y^2)^{-1}\frac{dy}{dx} = (1+y^2)^{-1}tan^{-1}y.$$

Therefore $|y''(x)| \leq M_2 = \frac{1}{2}\pi$. Inserting the values of L and M_2 into (18), have that

$$|e_n| \leq e^{x_n}|e_0| + \frac{1}{4}\pi\left(e^{x_n} - 1\right)h, \;\; n = 0, \cdots, N.$$

In particular if we assume that no error has been committed initially (i.e. $e_0 = 0$), we have that

$$|e_n| \leq \frac{1}{4}\pi(e^{x_n} - 1)h, \ n = 0, \cdots, N.$$

Thus, given a tolerance TOL specified beforehand, we can ensure that the error between the (unknown) analytical solution and its numerical approximation does not exceed this tolerance by choosing a positive step size h such that

$$h \leq \frac{4}{\pi}\left(e^{x_M} - 1\right)^{-1} TOL.$$

For such h we shall have $|y(x_n) - y_n| = |e_n| \leq TOL$ for each $n = 0, \cdots, N$, as required.

Consistency

Definition (Consistency)

The numerical method (21) is **consistent** with the differential equation (1) if the truncation error defined by (22) is such that for any $\epsilon > 0$ there exists a positive $h(\epsilon)$ for which $|T_n| < \epsilon$ for $0 < h < h(\epsilon)$ and any pair of points $(x_n, y(x_n))$, $(x_{n+1}, y(x_{n+1}))$ on any solution curve in R.

NOTE:

$$lim_{h \longrightarrow 0} Tn = y'(x_n) - \Phi(x_n, y(x_n); 0).$$

Therefore the one-step method (21) is consistent if and only if

$$\Phi(x_n, y_n; 0) \equiv f(x, y). \tag{26}$$

Convergence Theorem

Theorem

Suppose that the solution of the initial value problem (1)-(2) lies in R as does its approximation generated from (21) when $h \leq h_0$. Suppose also that the function $\Phi(\cdot, \cdot; \cdot)$ is uniformly continuous on $R \times [0, h_0]$ and satisfies the consistency condition (26) and the Lipschitz condition

$$\Phi(x, y; h) - \Phi(x, z; h)| \leq L_\Phi |y - z|; \quad \text{on } R \times [0, h_0]. \tag{27}$$

Then, if successive approximation sequences (y_n), generated for $x_n = x_0 + nh$, $n = 1, 2, \cdots, N$, are obtained from (21) with successively smaller values of h, each less than h_0, we have convergence of the numerical solution to the solution of the initial value problem in the sense that $|y(x_n) - y_n| \longrightarrow 0$ as $h \longrightarrow 0$, $x_n \longrightarrow x \in [x_0, X_M]$.

Order of accuracy

Definition (Order of accuracy)

The numerical method (21) is said to have order of accuracy p, if p is the largest positive integer such that, for any sufficiently smooth solution curve $(x, y(x))$ in R of the initial value problem (1)-(2), there exist constants K and h_0 such that

$$|T_n| \leq Kh^p \text{ for } 0 < h \leq h_0$$

for any pair of points $(x_n, y(x_n)), (x_{n+1}, y(x_{n+1}))$ on the solution curve.

R-stage Runge-Kutta family

RungeKutta methods aim to achieve higher accuracy by sacrificing the efficiency of Eulers method through re-evaluating $f(\cdot,\cdot)$ at points intermediate between $(x_n, y(x_n))$ and $(x_{n+1}, y(x_{n+1}))$. The general R-**stage Runge-Kutta family** is defined by

$$
\begin{aligned}
y_{n+1} &= y_n + h\Phi(x_n, y_n; h), \\
\Phi(x_n, y_n; h) &= \sum_{r=1}^{R} c_r k_r, \\
k_1 &= f(x, y), \\
k_r &= f\left(x + ha_r, y + h\sum_{s=1}^{r-1} b_{rs} k_s\right), \ r = 2, \cdots, R, \\
a_r &= \sum_{s=1}^{r-1} b_{rs}, \ r = 2, \cdots, R. \qquad (28)
\end{aligned}
$$

One-stage Runge-Kutta methods

Suppose that $R = 1$. Then, the resulting one- stage RungeKutta method is simply Eulers explicit method:

$$y_{n+1} = y_n + hf(x_n, y_n).$$

Two-stage Runge-Kutta methods

Next, consider the case of $R = 2$, corresponding to the following family of methods:

$$y_{n+1} = y_n + h(c_1 k_s + c_2 k_2),$$

where

$$\begin{aligned} k_1 &= f(x_n, y_n), \\ k_2 &= f(x_n + a_2 h, y_n + b_{21} h k_1) \end{aligned}$$

and where the parameters c_1, c_2, a_2 and b_{21} are to be determined.

Consistency condition

By the consistency condition

$$\Phi(x_n, y_n; 0) \equiv f(x, y).$$

we have

$$c_1 f(x_n, y_n) + c_2 f(x_n, y_n) = f(x_n, y_n) \Longrightarrow c_1 + c_2 = 1.$$

Order of accuracy

Further conditions on the parameters are obtained by attempting to maximise the order of accuracy of the method. Indeed, expanding the truncation error in powers of h, after some algebra we obtain

$$T_n = \frac{1}{2}hy''(x_n) + \frac{1}{6}h^2y'''(x_n) - c_2h\left[a_2f_x + b_{21}f_yf\right] - c_2h^2\left[\frac{1}{2}a_2^2f_{xx} + a_2b_{21}f_{xy}f + b_{21}^2f_{yy}f^2\right] + O(h^3)$$

Noting that

$$y'' = f_x + f_yf$$

it follows that $T_n = O(h^2)$ for any f provided that

$$\frac{1}{2}hy'' - c_2h\left[a_2f_x + b_{21}f_yf\right] = 0 \Longrightarrow a_2c_2 = b_{21}c_2 = \frac{1}{2}.$$

Two-stage Runge-Kutta methods

- This implies that if $b_{21} = a_2$, $c_2 = \frac{1}{2a_2}$ and $c_1 = 1 - \frac{1}{2a_2}$ then the method is second-order accurate; while this still leaves one free parameter, a_2
- It is easy to see that no choice of the parameters will make the method generally third-order accurate. There are two well-known examples of second-order RungeKutta methods:
  - **The modified Euler Method**: Taking $a_2 = \frac{1}{2}$ we have

$$y_{n+1} = y_n + hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hf(x_n, y_n)\right)$$

  - **The improved Euler method**: By choosing $a_2 = 1$ we have

$$y_{n+1} = y_n + \frac{1}{2}h\Big[f(x_n, y_n) + f(x_n + h, y_n + hf(x_n, y_n))\Big].$$

Truncation errors of these two methods

For these two methods it is easily verified by Taylor series expansion that the truncation error is of the form, respectively,

$$T_n = \frac{1}{6}h^2\left[f_y F_1 + \frac{1}{4}F_2\right] + O(h^3),$$
$$T_n = \frac{1}{2}h^2\left[f_y F_1 - \frac{1}{2}F_2\right] + O(h^3),$$

where

$$F_1 = f_x + f\, f_y, \text{ and } F_2 = f_{xx} + 2f\, f_{xy} + f^2 f_{yy}.$$

Exercise

Let α be a non-zero real number and let $x_n = a + nh$, $n = 0, \cdots, N$, be a uniform mesh on the interval $[a, b]$ of step size $h = \frac{b-a}{N}$. Consider the explicit one-step method for the numerical solution of the initial value problem $y' = f(x, y)$, $y(a) = y_0$, which determines approximations y_n to the values $y(x_n)$ from the recurrence relation

$$y_{n+1} = y_n + h(1-\alpha)f(x_n, y_n) + h\alpha f\left(x_n + \frac{h}{2\alpha}, y_n + \frac{h}{2\alpha}f(x_n, y_n)\right).$$

Show that this method is consistent and that its truncation error, $T_n(h, \alpha)$, can be expressed as

$$T_n(h, \alpha) = \frac{h^2}{8\alpha}\left[\left(\frac{4}{3}\alpha - 1\right)y'''(x_n) + y''(x_n)\frac{\partial f}{\partial y}(x_n, y(x_n))\right] + O(h^3).$$

This numerical method is applied to the initial value problem $y' = -y^p$, $y(0) = 1$, where p is a positive integer. Show that if $p = 1$ then $T_n(h, \alpha) = O(h^2)$ for every non-zero real number α. Show also that if $p \geq 2$ then there exists a non-zero real number α_0 such that $T_n(h, \alpha_0) = O(h^3)$.

Three-stage RungeKutta methods

Let us now suppose that $R = 3$ to illustrate the general idea. Thus, we consider the family of methods:

$$y_{n+1} = y_n + h\left[c_1 k_1 + c_2 k_2 + c_3 k_3\right],$$

where

$$\begin{aligned}
k_1 &= f(x, y),\\
k_2 &= f(x + ha_2, y + hb_{21}k_1),\\
k_3 &= f(x + ha_3, y + hb_{31}k_1 + hb_{32}k_2),\\
a_2 &= b_{21}, a_3 = b_{31} + b_{32}.
\end{aligned}$$

Three-stage Runge-Kutta methods

Writing $b_{21} = a_2$ and $b_{31} = a_3 - b_{32}$ in the definitions of k_2 and k_3 respectively and expanding k_2 and k_3 into Taylor series about the point (x, y) yields:

$$\begin{aligned} k_2 &= f + ha_2(f_x + k_1 f_y) + \frac{1}{2}h^2 a_2^2(f_{xx} + 2k_1 f_{xy} + k_1^2 f_{yy}) + O(h^3) \\ &= f + ha_2({}_f x + f\, f_y) + \frac{1}{2}h^2 a_2^2(f_{xx} + 2f\, f_{xy} + f^2 f_{yy}) + O(h^3) \\ &= f + ha_2 F_1 + \frac{1}{h}^2 a_2^2 F_2 + O(h^3), \end{aligned}$$

where

$$F_1 = f_x + f\, f_y \text{ and } F_2 = f_{xx} + 2f\, f_{xy} + f^2 f_{yy},$$

Three-stage Runge-Kutta methods

and

$$\begin{aligned} k_3 &= f + h\{a_3 f_x + [(a_3 - b_{32})k_1 + b_{32}k_2]f_y\} \\ &\quad + \frac{1}{2}h^2\left\{a_3^2 f_{xx} + 2a_3[(a_3 - b_{32})k_1 + b_{32}k_2]f_{xy}\right. \\ &\quad \left.+[(a_3 - b_{32})k_1 + b_{32}k_2]^2 f_{yy}\right\} + O(h3) \\ &= f + ha_3 F_1 + h^2\left(a_2 b_{32} F_1 f_y + \frac{1}{2}a_{23}F_2\right) + O(h^3). \end{aligned}$$

Substituting these expressions for k_2 and k_3 with $R = 3$ we find that

$$\begin{aligned} \Phi(x, y, h) &= (c_1 + c_2 + c_3)f + h(c_2 a_2 + c_3 a_3)F_1 \\ &\quad + \frac{1}{2}h^2\left[2c_3 a_2 b_{32} F_1 f_y + \left(c_2 a_2^2 + c_3 a_{23}\right)F_2\right] + O(h^3). \end{aligned}$$

Three-stage Runge-Kutta methods

We match this with the Taylor series expansion:

$$\begin{aligned}\frac{y(x+h)-y(x)}{h} &= y'(x)+\frac{1}{2}hy''(x)+\frac{1}{6}h^2y'''(x)+O(h^3)\\ &= f+\frac{1}{2}hF_1+\frac{1}{6}h^2(F_1f_y+F_2)+O(h^3).\end{aligned}$$

This yields:

$$\begin{aligned}c_1+c_2+c_3 &= 1,\\ c_2a_2+c_3a_3 &= \frac{1}{2},\\ c_2a_2+c_3a_{23} &= \frac{1}{3},\\ c_3a_2b_{32} &= \frac{1}{6}.\end{aligned}$$

Solving this system of 4 equations with 6 unknowns, results in a 2-parameter family of 3-stage Runge-Kutta methods.

Linear multi-step methods

While Runge-Kutta methods present an improvement over Eulers method in terms of accuracy, this is achieved by investing additional computational effort; in fact, Runge-Kutta methods require more evaluations of $f(\cdot,\cdot)$ than would seem necessary. For example, the fourth-order method involves four function evaluations per step. For comparison, by considering three consecutive points x_{n-1}, $x_n = x_{n-1} + h$, $x_{n+1} = x_{n-1} + 2h$, integrating the differential equation between x_{n-1} and x_{n+1}, and applying Simpsons rule to approximate the resulting integral yields

$$y(x_{n+1}) = y(x_{n-1}) + \int_{x_{n-1}}^{x_{n+1}} f(x, y(x))dx$$

$$\approx y(x_{n-1}) + \frac{1}{3h}\Big[f(x_{n-1}, y(x_{n-1})) + 4f(x_n, y(x_n)) + f(x_{n+1}, y(x_{n+1}))\Big]$$

Linear multi-step methods

which leads to the method

$$y_{n+1} = y_{n-1} + \frac{1}{3}h[f(x_{n-1}, y_{n-1}) + 4f(x_n, y_n) + f(x_{n+1}, y_{n+1})]. \quad (29)$$

Linear multi-step methods

- **Adams7- Bashforth method**: an explicit linear four-step method:

$$y_{n+4} = y_{n+3} + \frac{1}{24}h[55f_{n+3} - 59f_{n+2} + 37f_{n+1} - 9f_n]$$

- **Adams - Moul- ton method**: an implicit linear four-step method:

$$y_{n+4} = y_{n+3} + \frac{1}{24}h[9f_{n+4} + 19f_{n+3} - 5f_{n+2} - 9f_{n+1}].$$