Numerical Methods in MATLAB®

Z0966990

L2 Engineering Mathematics

April 23, 2017

1 Taylor Series Method

The p^{th} order Taylor's series method for ODEs finds numerical approximations to equations of the form $y'(t) = f(t, y)$. In n steps it approximates the solution y_n at some time T given an initial condition:

$$\begin{aligned} t_0 &= 0 & y_0 &= y(0) \\ t_{i+1} &= t_i + h & y_{i+1} &= y_i + \sum_{k=1}^{p} \frac{h^k}{k!} f^{(k-1)}(t_i, y_i) \\ t_n &= T & y_n &\approx y(T) \end{aligned} \tag{1}$$

where $h = T/n$ and is the step-size.

The use of successive approximations results in an error composed of the accumulation error and truncation error at each step. At time T, this is the final global error:

$$\epsilon_n = |y_n - y(T)| = O(h^p) = O(n^{-p}) \tag{2}$$

A MATLAB implementation of (1) with order 4 was applied to three ODEs of the form $y'(t) = f(t, y)$. These were defined as follows:

$$\begin{aligned} y'(t) &= \sin(t) & y(0) &= 0 & 0 \le t \le \pi && \text{(a)} \\ y'(t) &= 5y(t) & y(0) &= 1 & 0 \le t \le 0.1 && \text{(b)} \\ y'(t) &= 2 & y(0) &= 0 & 0 \le t \le 1 && \text{(c)} \end{aligned}$$

1.1 Results

The results of the fourth order Taylor series method were compared to the results of Taylor's second and Euler's method—which is a first order Taylor's series method. Table 1 lists y_n for different numbers of iterations n.

ODE (c) had no non-zero derivatives in $f(t, y)$, so Euler's method contained sufficiently many terms to find y_n when $T = 1$. In fact, ODE (c) converged superlinearly for all methods. For ODEs (b) and (c), many more iterations were required for the lower order methods to converge on a value for y_n as the higher order coefficients were larger.

The exact errors of the results in Table 1 are detailed in Table 2, in which ODEs (a) and (b) demonstrate how increasing the number of iterations reduces the final global error, as suggested in (2). Equation (2) also suggests increasing the order of the Taylor's series method reduces the final global error.

Table 1: ODE Verification Simulations

	n	**Euler**†	**Taylor 2nd**†	**Taylor 4th**†
ODE (a) at $t = \pi$	10	1.99	2.03	2.00
	20	2.00	2.01	2.00
	40	2.00	2.00	2.00
	80	2.00	2.00	2.00
ODE (b) at $t = 0.1$	10	1.63	1.65	1.65
	20	1.64	1.65	1.65
	40	1.64	1.65	1.65
	80	1.65	1.65	1.65
ODE (c) at $t = 1$	10	2.00	2.00	2.00
	20	2.00	2.00	2.00
	40	2.00	2.00	2.00
	80	2.00	2.00	2.00

† Accurate to 3 significant figures.

Particularly for ODE (b), doubling the order of the method halved the magnitude of the error.

Table 2: ODE Errors

	n	**Euler Error**†	**Taylor 2nd Error**†	**Taylor 4th Error**†
ODE (a) at $t = \pi$	10	1.65×10^{-2}	3.29×10^{-2}	1.62×10^{-4}
	20	4.11×10^{-3}	8.22×10^{-3}	1.01×10^{-5}
	40	1.03×10^{-3}	2.06×10^{-3}	6.34×10^{-7}
	80	2.57×10^{-4}	5.14×10^{-4}	3.96×10^{-8}
ODE (b) at $t = 0.1$	10	1.98×10^{-2}	3.31×10^{-4}	4.12×10^{-8}
	20	1.01×10^{-2}	8.43×10^{-5}	2.63×10^{-9}
	40	5.10×10^{-3}	2.13×10^{-5}	1.66×10^{-10}
	80	2.56×10^{-3}	5.34×10^{-6}	1.04×10^{-11}
ODE (c) at $t = 1$	10	2.22×10^{-16}	2.22×10^{-16}	2.22×10^{-16}
	20	4.44×10^{-16}	4.44×10^{-16}	4.44×10^{-16}
	40	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}
	80	3.11×10^{-15}	3.11×10^{-15}	3.11×10^{-15}

† Accurate to 3 significant figures.

For ODE (c), higher order methods behave as Euler's method, so the order of the method has no effect on accuracy. Contrary to (2) the error increased with the number of iterations. This can be explained by the round off error in h. When $n = 10$, binary64 format cannot encode every bit of the recurring binary fraction $h = 0.0\dot{0}01\dot{1}_2$. As the number of iterations n increase, more round off error accumulates.

2 Finite Difference Method

The convection–diffusion–reaction equation can be used to describe how mass concentration u varies in space $\mathbf{x}$ with time t. The parabolic equation includes convection $\mathbf{v} \cdot \nabla u$, wherein the mass has a net velocity $\mathbf{v}$; diffusion $c^2\nabla^2 u$, where random motion disperses mass over time; and reaction du, a sink or source of mass—present when chemical reactions are taking place.

The equation can also be applied to heat transfer, where u represents heat.

In 1D, the convection–diffusion–reaction equation is defined as follows:

$$\begin{aligned} &\forall x, t \in (a, b) \times (0, T] \\ &u_t(x,t) = c^2 u_{xx}(x,t) + v u_x(x,t) + d u(x,t) \end{aligned} \tag{3}$$

with homogenous Dirichlet boundary conditions and initial condition $u(x, 0) = f(x)$ applied.

A FDM schema was devised to numerically solve (3), second order in space with interval h, first order in time with time-step k, and implemented in MATLAB as follows:

$$\begin{aligned} u^i_{j+1} = u^i_j + dku^i_j + \frac{vk}{h}(u^{i+1}_j - u^i_j) \\ + \frac{c^2k}{h^2}(u^{i+1}_j - 2u^i_j + u^{i-1}_j) \end{aligned} \tag{4}$$

Errors in FDM schema can grow or be damped out at each time-step, leading to a stable and accurate, or unstable solutions. Von Neumann stability analysis assumes these errors are amplified by a constant factor [1]. An expression for amplification factor is determined by expressing the errors as a Fourier series in space. Stability necessitates the amplification factor is not much greater than one for all frequencies resolvable in the grid.

Applying von Neumann stability analysis to (4) yielded the following stability criterion, and expression for the amplification factor:

$$G = \max \left| \begin{array}{r} 1 + R + (2S + C)(\cos\theta - 1) \\ + \mathrm{i}\, C \sin\theta \end{array} \right| \le 1 + \epsilon k \tag{5}$$

$$R = dk \quad C = \frac{vk}{h} \quad S = \frac{c^2k}{h^2}$$

where R, C and S are dimensionless quantities: the reaction, Courant and diffusion numbers. ϵ is a constant assumed to be smaller than 800×10^{-3}.

The model was tested on the following:

$$\begin{array}{lll} c = 1 & a = 0 & m, n = \{10, 100, 200, 400, 800\} \\ v = 0.5 & b = 1 & \qquad \times \{4, 16, 32\} \\ d = 0.02 & T = 0.2 & f(x) = 4x - 4x^2 \end{array}$$

With these coefficients fixed, the amplification factor in (5) varies as follows:

$$G = O(k/h^2) = O(n^2/m) \tag{6}$$

2.1 Results

Table 3 lists the value of (3) at the midpoint of the interval at time T, for each n and m.

Table 3: PDE Verification Simulations

n	m	$u(0.5, 0.2)$†	G†
4	10	0.111	1.00
16	10	-1.32×10^{7}	19.8
32	10	22.7	81.6
4	100	0.137	1.00
16	100	0.102	1.08
32	100	-3.71×10^{80}	7.26
4	200	0.139	1.00
16	200	0.138	1.00
32	200	-3.65×10^{93}	3.13
4	400	0.139	1.00
16	400	0.138	1.00
32	400	-5.65×10^{4}	1.06
4	800	0.140	1.00
16	800	0.139	1.00
32	800	0.140	1.00

Unstable: unable to satisfy (5).

† Accurate to 3 significant figures.

The highlighted simulations with a gain greater than unity did not appear to converge on similar values to those determined stable by (5). This occurred when there were too many intervals in space for a given number of time-steps, as predicted by (6).

References

[1] J. G. Charney, R. Fjörtoft, and J. v. Neumann, “Numerical integration of the barotropic vorticity equation”, *Tellus*, vol. 2, no. 4, pp. 237–254, 1950.