Lecture 29: Global Error Analysis

5.1.4. Global Error Analysis for One Step Methods.

The last lecture addressed the truncation error, T_k, of a one-step method. Consistency (i.e., $T_k \to 0$ as $h \to 0$) is an obvious necessary condition for the *global error*

$$e_k = x(t_k) - x_k$$

to converge as $h \to 0$. In this lecture, we wish to understand this key question:

Is consistency sufficient for convergence of the global error as $h \to 0$?

As before, consider the general one step method

$$x_{k+1} = x_k + h\Phi(t_k, x_k; h)$$

where the choice of $\Phi(t_k, x_k; h)$ defines the specific algorithm. We can rearrange the formula for truncation error,

$$T_k = \frac{x(t_{k+1}) - x(t_k)}{h} - \Phi(t_k, x(t_k); h),$$

to obtain an expression for $x(t_{k+1})$,

$$x(t_{k+1}) = x(t_k) + h\Phi(t_k, x(t_k); h) + hT_k.$$

This formula is comparable to the one-step method itself,

$$x_{k+1} = x_k + h\Phi(t_k, x_k; h).$$

Combining these expressions gives a formula for the global error,

$$\begin{aligned} e_{k+1} &= x(t_{k+1}) - x_{k+1} \\ &= x(t_k) - x_k + h\Big(\Phi(t_k, x(t_k); h) - \Phi(t_k, x_k; h)\Big) + hT_k \\ &= e_k + h\Big(\Phi(t_k, x(t_k); h) - \Phi(t_k, x_k; h)\Big) + hT_k. \end{aligned}$$

Recall the example $x'(t) = 1 + x^2$ from Lecture 27. That equation blew up in finite time, while the iterates of Euler's method were always finite. This is disappointing: for some equations, we can essentially have infinite global error! Thus, to get a useful error bound, we must make an assumption that the ODE is well behaved. Suppose we are integrating our equation from t_0 to some fixed t_{final}. Then assume there exists a constant L_Φ, *depending on the equation, the time interval, and the particular method* (but not h), such that

$$|\Phi(t, u; h) - \Phi(t, v; h)| \le L_\Phi |u - v|$$

for all $t \in [t_0, t_{\text{final}}]$ and all $u, v \in \mathbb{R}$. This assumption is closely related to the *Lipschitz condition* that plays an essential role in the theorem of existence of solutions given in Lecture 27. For 'nice' ODEs and reasonable methods Φ, this condition is not difficult to satisfy.

This assumption is precisely what we need to bound the difference between $\Phi(t_k, x(t_k); h)$ and $\Phi(t_k, x_k; h)$ that appears in the formula for e_k. In particular, we now have

$$\begin{aligned}
|e_{k+1}| &= \Big|e_k + h\Big(\Phi(t_k, x(t_k); h) - \Phi(t_k, x_k; h)\Big) + hT_k\Big| \\
&\le |e_k| + h\Big|\Phi(t_k, x(t_k); h) - \Phi(t_k, x_k; h)\Big| + h|T_k| \\
&\le |e_k| + hL_\Phi|x(t_k) - x_k| + h|T_k| \\
&= |e_k| + hL_\Phi|e_k| + h|T_k| \\
&= |e_k|(1 + hL_\Phi) + h|T_k|.
\end{aligned}$$

Suppose we are interested in all iterates from x_0 up to x_n for some n. Then let T denote the magnitude of the maximum truncation error over all those iterates:

$$T := \max_{0 \le k \le n} |T_k|.$$

We now build up an expression for e_n iteratively:

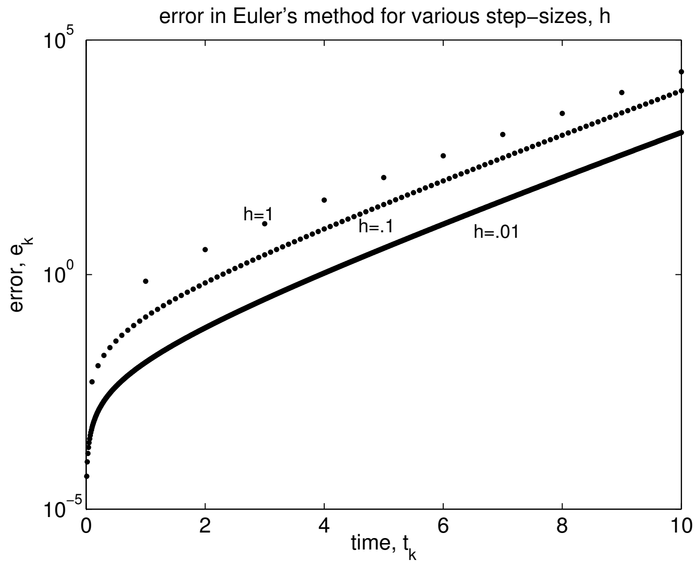
$$\begin{aligned}
|e_0| &= |x(t+0) - x_0| = 0 \\
|e_1| &\le h|T_0| \le hT \\
|e_2| &\le |e_1|(1 + hL_\Phi) + h|T_1| \le hT(1 + hL_\Phi) + hT \\
|e_3| &\le |e_2|(1 + hL_\Phi) + h|T_2| \le hT(1 + hL_\Phi)^2 + hT(1 + hL_\Phi) + hT \\
&\vdots \\
|e_n| &\le hT\sum_{k=0}^{n-1}(1 + hL_\Phi)^k.
\end{aligned}$$

Notice that this bound for $|e_n|$ is a finite geometric series, and thus we have the convenient formula

$$\begin{aligned}
|e_n| &\le hT\Big(\frac{(1 + hL_\Phi)^n - 1}{(1 + hL_\Phi) - 1}\Big) \\
&= \frac{T}{L_\Phi}((1 + hL_\Phi)^n - 1) \\
&\le \frac{T}{L_\Phi}(\mathrm{e}^{nhL_\Phi} - 1).
\end{aligned}$$

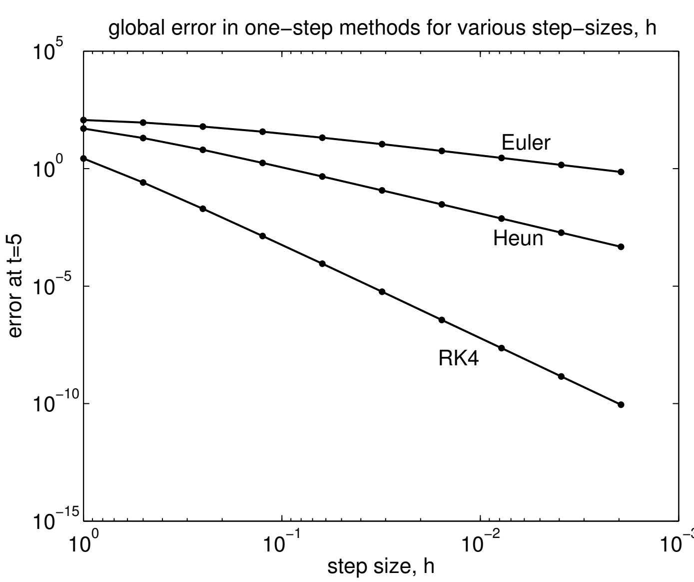
Here we have used the fact that the Taylor's series for e^γ implies that $1 + \gamma \le \mathrm{e}^\gamma$ for all $\gamma \ge 0$, with good agreement when $0 \le \gamma \ll 1$. (This result and proof are given as Theorem 12.2 in Süli and Mayers.)

There are two key lessons to be learned from this bound on $|e_n|$. First the bad news: As $n \to \infty$ and h is fixed, we expect the approximations from the one-step method to *exponentially* drift away from the true solution. This fact is illustrated in the plot below, where Euler's method has been applied to the model problem $x'(t) = x(t)$ for $t \in [0, 10]$ with various step sizes h.



However, this plot also hints at the good news to come: At any fixed time (say, t_{final}), the error gets smaller with decreasing h: this is the essential lesson to draw from this global error analysis. In typical situations, we are interested in the convergence of the global error at some fixed time t_{final} as the step size is reduced, $h \to 0$. In that case, set $h = (t_{\text{final}} - t_0)/n$, implying that $hn = t_{\text{final}} - t_0$ is fixed. Since L_Φ is independent of the step size h, if the truncation error converges, $T \to 0$ as $h \to 0$, then the global error at t_{final} will also converge. Moreover, if $T_k = O(h^p)$, then the global error at t_{final} will also be $O(h^p)$. This is a beautiful fact: the global error reduces at the same rate as the truncation error for one-step methods!

The plot below confirms this observation. Again for the model problem $x'(t) = x(t)$ with $(t_0, x_0) = (0, 1)$, we investigate convergence of Euler's method ($T_k = O(h)$), Heun's method ($T_k = O(h^2)$), and the four-stage Runge–Kutta method ($T_k = O(h^4)$) at the fixed time $t_{\text{final}} = 5$.



Note that the slopes of these global error curves agree with the order of the truncation error for each method – just as predicted by our global error analysis.

5.1.5. Adaptive Time-Step Selection.

One-step methods make it very to change the time-step h at each iteration. For complicated nonlinear problems, it is quite natural that some regions (especially when x' is large) will merit a small time-step h, yet other regions, where there is less change in the solution, can easily be handled with a large value of h.

In the 1960s, Erwin Fehlberg suggested a beautiful way in which the step-size could be automatically adjusted at each step. There exist Runge–Kutta methods of order 4 and order 5 that can both be generated with the same *six* evaluations of f. (Recall that any fifth-order Runge–Kutta method requires at least six function evaluations.) First, we define the necessary f evaluations for this method:

$$\begin{aligned}
k_1 &= f(t_k, x_k)\\
k_2 &= f(t_k + \tfrac{1}{4}h, x_k + \tfrac{1}{4}hk_1)\\
k_3 &= f(t_k + \tfrac{3}{8}h, x_k + \tfrac{3}{32}hk_1 + \tfrac{9}{32}hk_2)\\
k_4 &= f(t_k + \tfrac{12}{13}h, x_k + \tfrac{1932}{2197}hk_1 - \tfrac{7200}{2197}hk_2 + \tfrac{7296}{2197}hk_3)\\
k_5 &= f(t_k + h, x_k + \tfrac{439}{216}hk_1 - 8hk_2 + \tfrac{3680}{513}hk_3 - \tfrac{845}{4104}hk_4)\\
k_6 &= f(t_k + \tfrac{1}{2}h, x_k - \tfrac{8}{27}hk_1 + 2hk_2 - \tfrac{3544}{2565}hk_3 + \tfrac{1859}{4104}hk_4 - \tfrac{11}{40}hk_5).
\end{aligned}$$

The following method has $O(h^5)$ truncation error:

$$x_{k+1} = x_k + h\Big(\tfrac{16}{135}k_1 + \tfrac{6656}{12825}k_3 + \tfrac{28561}{56430}k_4 - \tfrac{9}{50}k_5 + \tfrac{2}{55}k_6\Big).$$

The f evaluations used to compute these k_j values can be combined in a different manner to obtain the following approximation, which only has $O(h^4)$ truncation error:

$$\widehat{x}_{k+1} = x_k + h\Big(\tfrac{25}{216}k_1 + \tfrac{1408}{2565}k_3 + \tfrac{2197}{4104}k_4 - \tfrac{1}{5}k_5\Big).$$

Why would one be interested in an $O(h^4)$ method when an $O(h^5)$ approximation is available? By inspecting $x_{k+1} - \widehat{x}_{k+1}$, we can see how much the extra order of accuracy changes the solution. A significant difference signals that the step size h may be too large; software will react by reducing the step size before proceeding. This technology is implemented in MATLAB's `ode45` routine. (The `ode23` routine is similar, but based on a pair of second and third order methods.)

Another popular fifth-order method, designed by Cash and Karp (1990), uses six carefully chosen function evaluations that can be combined to also provide $O(h)$, $O(h^2)$, $O(h^3)$, and $O(h^4)$ approximations.