Adaptive time stepping

Key idea: Run 2 methods simultaneously with different truncation errors (orders):

e.g. 1) $y_{n+1}^A = \langle \text{method } A \rangle$, e.g. $O(h^4)$ LTE

2) $y_{n+1}^B = \langle \text{method } B \rangle$, e.g. $O(h^5)$ LTE.

Approximate the error as

$$err = |y_{n+1}^A - y_{n+1}^B|$$

If err > a user-chosen tolerance, reduce h (e.g. by half) and recompute the whole step, repeating until the bound is satisfied.

Q: Is this reasonable? Why?

Observe $y_{n+1}^A = y(t_{n+1}) + O(h^p) = y(t_{n+1}) + \underset{\substack{\text{unknown error}\\ \text{coefficient.}}}{C} h^p + O(h^{p+1})$

for some value C.

If method B is one order more accurate, $O(h^{p+1})$, then

$$y_{n+1}^B = y(t_{n+1}) + O(h^{p+1}).$$

Method A's true error is $|y_{n+1}^A - y(t_{n+1})| = Ch^p + O(h^{p+1})$

Our estimated error is $|y_{n+1}^A - y_{n+1}^B| = Ch^p + O(h^{p+1})$ also...

The dominant (lowest order) term of the error matches!

$\therefore$ Our estimated error is a reasonable approximation to use to adjust h, without knowing the true sol'n.

Predicting A Good Step Size

We can *estimate* the leading error coefficient C as

$$C \approx \frac{|y^A_{n+1} - y^B_{n+1}|}{(h_{old})^p}$$

where "h_{old}" is the current (most recent) step size.

Assuming C changes slowly/smoothly in time, we can *estimate* the next step's error as

$$err_{next} = |y^A_{n+2} - y(t_{n+2})| \approx C(h_{new})^p$$

next step size, to be determined.

Plugging in for C, we have:

$$err_{next} = \frac{|y^A_{n+1} - y^B_{n+1}|}{(h_{old})^p}(h_{new})^p = \left(\frac{h_{new}}{h_{old}}\right)^p |y^A_{n+1} - y^B_{n+1}|.$$

Given a desired error tolerance, "tol", set $err_{next} = tol$, and solve for h_{new}:

$$h_{new} = h_{old}\left(\frac{tol}{|y^A_{n+1} - y^B_{n+1}|}\right)^{\frac{1}{p}}.$$

To (roughly!) compensate for our approximations, we may be conservative by scaling our tolerance down by some factor α, e.g. $\alpha = \frac{1}{2}$ or $\alpha = \frac{3}{4}$, etc.

h_{new} will <u>not</u> be exact, but hopefully a good guess.