Homework 4 Nicole Degerl

① Derive the AB method having global accuracy $O(h^4)$

- general explicit LMM:

$$y_{n+1} = \sum_{j=0}^{p} a_j y_{n-j} + h \sum_{j=0}^{p} b_j f(t_{n-j}, y_{n-j})$$

- AB methods have $a_0 = 1$, $a_j = 0$ for $j > 0$:

$$y_{n+1} = y_n + h \sum_{j=0}^{p} b_j f(t_{n-j}, y_{n-j}) = y_n + \sum_{j=0}^{p} f_{n-j} \left[\int_{t_n}^{t_{n+1}} l_j(t)\, dt \right]$$

- so, $b_j = \int_{t_n}^{t_{n+1}} l_j(t)\, dt$, and $O(h^4) \Rightarrow 4$ coefficients, since AB-0 was $O(h)$ and had 1 coeff, AB-1 was $O(h^2)$ and had 2 coeffs, etc.

interpolating through (t_n, f_n), (t_{n-1}, f_{n-1}), (t_{n-2}, f_{n-2}), (t_{n-3}, f_{n-3})

$$l_0(t) = \frac{(t-t_{n-1})(t-t_{n-2})(t-t_{n-3})}{\underset{h}{(t_n-t_{n-1})}\underset{2h}{(t_n-t_{n-2})}\underset{3h}{(t_n-t_{n-3})}} = \frac{1}{6h^3} \underbrace{(t-t_{n-1})(t-t_{n-2})(t-t_{n-3})}_{f}$$

$$b_0 = \frac{1}{6h^3} \int_{t_n}^{t_{n+1}} (t-t_{n-1})(t-t_{n-2})(t-t_{n-3})\, dt = \frac{1}{6h^3}\left[\frac{h}{6}\left[f(t_n) + 4f(t_{n+1/2}) + f(t_{n+1})\right]\right]$$

by Simpson's rule being a perfect approximation for cubic polynomials.

$$= \frac{1}{36h^2}\left[h \cdot 2h \cdot 3h + 4 \cdot \frac{3h}{2} \cdot \frac{5h}{2} \cdot \frac{7h}{2} + 2h \cdot 3h \cdot 4h\right] = \frac{1}{36h^2}\left[6h^3 + \underbrace{\frac{420}{8}}_{=\frac{105}{2}} h^3 + 24h^3\right]$$

$$= \frac{1}{36h^2}\left(\frac{12 + 105 + 48}{2} h^3\right) = \frac{165}{72} h = \frac{55}{24} h$$

$t_n - h - t_n + 2h$

$$l_1(t) = \frac{(t-t_n)(t-t_{n-2})(t-t_{n-3})}{\underset{-h}{(t_{n-1}-t_n)}\underset{+h}{(t_{n-1}-t_{n-2})}\underset{+2h}{(t_{n-1}-t_{n-3})}} = \frac{-1}{2h^3} \underbrace{(t-t_n)(t-t_{n-2})(t-t_{n-3})}_{f}$$

$$b_1(t) = \frac{-1}{2h^3}\int_{t_n}^{t_{n+1}} (t-t_n)(t-t_{n-2})(t-t_{n-3})\,dt = \frac{-1}{2h^3}\left[\frac{h}{6}\left[f(t_n) + 4f(t_{n+1/2}) + f(t_{n+1})\right]\right]$$

by Simpson's rule

$$= \frac{-1}{12h^2}\left[0 + 4\cdot\frac{h}{2}\cdot\frac{5h}{2}\cdot\frac{7h}{2} + h\cdot 3h\cdot 4h\right] = \frac{-1}{12h^2}\left[\underbrace{\frac{140}{8}}_{\frac{35}{2}}h^3 + 12h^3\right]$$

$$= \frac{-1}{12h^2}\left[\frac{35+24}{2}h^3\right] = \frac{-59}{24}h$$

$$l_2(t) = \frac{(t-t_n)(t-t_{n-1})(t-t_{n-3})}{\underbrace{(t_{n-2}-t_n)}_{-2h}\underbrace{(t_{n-2}-t_{n-1})}_{-h}\underbrace{(t_{n-2}-t_{n-3})}_{+h}} = \frac{1}{2h^3}\underbrace{(t-t_n)(t-t_{n-1})(t-t_{n-3})}_{f}$$

$$b_2(t) = \frac{1}{2h^3}\int_{t_n}^{t_{n+1}} (t-t_n)(t-t_{n-1})(t-t_{n-3})\,dt = \frac{1}{2h^3}\left[\frac{h}{6}\left[f(t_n) + 4f(t_{n+1/2}) + f(t_{n+1})\right]\right]$$

by Simpson's rule

$$= \frac{1}{12h^2}\left[0 + 4\cdot\frac{h}{2}\cdot\frac{3h}{2}\cdot\frac{7h}{2} + h\cdot 2h\cdot 4h\right] = \frac{1}{12h^2}\left[\underbrace{\frac{84}{8}}_{\frac{21}{2}}h^3 + 8h^3\right]$$

$$= \frac{1}{12h^2}\left[\frac{21+16}{2}h^3\right] = \frac{37}{24}h$$

$$l_4(t) = \frac{(t-t_n)(t-t_{n-1})(t-t_{n-2})}{\underbrace{(t_{n-3}-t_n)}_{-3h}\underbrace{(t_{n-3}-t_{n-1})}_{-2h}\underbrace{(t_{n-3}-t_{n-2})}_{-h}} = \frac{-1}{6h^3}\underbrace{(t-t_n)(t-t_{n-1})(t-t_{n-2})}_{f}$$

$$b_4(t) = \frac{-1}{6h^3}\int_{t_n}^{t_{n+1}} (t-t_n)(t-t_{n-1})(t-t_{n-2})\,dt = \frac{-1}{6h^3}\left[\frac{h}{6}\left[f(t_n) + 4f(t_{n+1/2}) + f(t_{n+1})\right]\right]$$

by Simpson's Rule

$$= \frac{-1}{36h^2}\left[0 + 4\cdot\frac{h}{2}\cdot\frac{3h}{2}\cdot\frac{5h}{2} + h\cdot 2h\cdot 3h\right] = \frac{-1}{36h^2}\left[\underbrace{\frac{60}{8}}_{\frac{15}{2}}h^3 + 6h^3\right]$$

$$= \frac{-1}{36h^2}\left[\frac{15+12}{2}h^3\right] = \frac{-27}{72}h = -\frac{3}{8}h = \frac{-9}{24}h$$

$\Rightarrow$ The $O(h^4)$ AB method is given by

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

idea: after loop over h's

do 3 steps

ERK (t0, ...) → y3

ERK (t0+h, ...) → y2

ERK (t0+2h, ...) → y1 < update tspan

Then while (tcur < ...)

AB3Stepper (y, y1, y2, y3)

- eval 3 old f's
- enter timestep iteration

    eval 1 "new" old f

    shift others over

---

(3) Cost of ERK 4 vs ERK 4 followed by AB3 in terms of fcalls for N steps.

1 step of ERK 4:

4 stages: $z_1 = y \rightarrow f(t, y) = f0$

$$z_2 = y + h a_{11} f0 \rightarrow f(t + c_1 h, z_2) = f1$$

$$z_3 = y + h a_{21} f0 + h\, a_{22} f1 \rightarrow f(t + c_2 h, z_3) = f2$$

$$z_4 = y + h a_{31} f0 + h a_{32} f1 + h a_{33} \rightarrow f(t + c_3 h, z_4) = f3$$

Then $y_{n+1} = y_n + h\, b0\, f0 + h\, b1\, f1 + \dots + h\, b3\, f3$

$\Rightarrow$ 4 calls to f for each step.

1 Step of AB3:

- Send in 3 IC's $(y_{n-3}, y_{n-2}, y_{n-1})$ and current value y_n
- initially, do 3 fevals, $f(t, y_{n-3})$, $f(t, y_{n-2})$, $f(t, y_{n-1})$ to get started
- Then, 1 feval at current step $f(t, y_n)$
- For every step after this first step, we only need to do 1 feval at the current time, and shift over our old f's, that is $f(t, y_{n-3})$ now equals $f(t, y_{n-2})$, and so on, at the next step.

So, total cost for ERK4: 4 f-calls per step

$\Rightarrow$ cost $= 4N$

Total cost for AB3 using ERK4-ICs:

$3 \cdot 4$ f-calls for 3 IC's from ERK4

$+ 3$ f-calls to get started and find $f_{n-3}, f_{n-2}, f_{n-1}$

$+ 1$ f-call per step

$12 + 3 + N$

$\Rightarrow$ cost $= 15 + N$

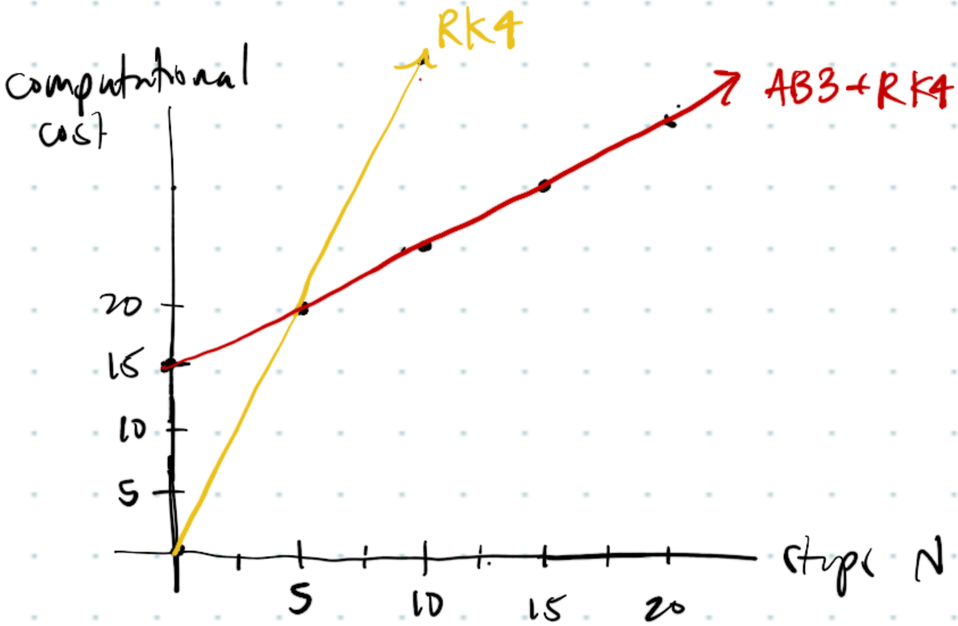
For $N = 3$: RK4 cost $= 12 \leq 18 =$ AB3 + RK4 cost

$N = 4$: RK4 cost $= 16 \leq 19 =$ AB3 + RK4 cost

$N = 5$ RK4 cost $= 20 = 20 =$ AB3 + RK4 cost

$N = 6$ RK4 cost $= 24 > 21 =$ AB3 + RK4 cost

So, for $N \geq 6$, AB3 + RK4 is more efficient than using RK4, and infact for $N \geq 6$, AB3 + RK4 is much more efficient than RK4, since both methods provide the same order of accuracy, but the number of computations for AB3 + RK4 grow by a factor of 1, whereas the number of computations (cost) for RK4 grows by a factor of 4.



④ Our numerical results reflect the theoretical expectation. RK4, which is what I used as the answer y_n for each step, is $O(h^4)$ accurate, whereas forward euler has $O(h)$ accuracy. Since we are doing adaptive methods, we don't really care to see each h value that was used and check the order of error in each step. Instead, I can see from my results that the order of relative error from my adaptive RKF45 is the same order as the relative error from adaptive euler. <u>But</u>, adaptive RKF45 takes 1/50th as many steps as adaptive euler. This indicates 2 things:

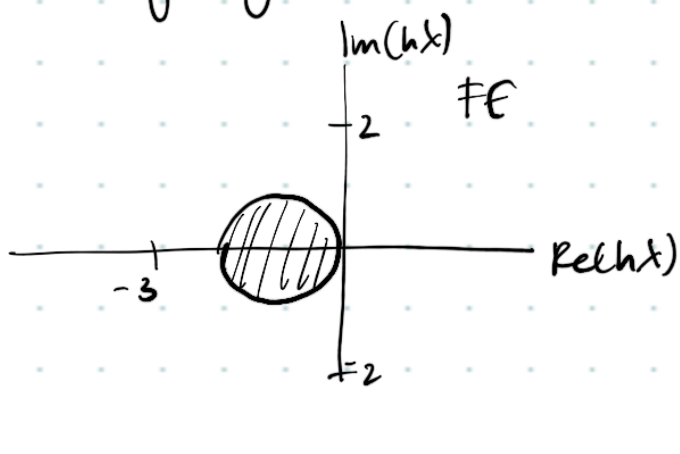
1) Adaptive RKF45 matches the order of error of adaptive euler while using much larger steps (we get from $t_0 = 0$ to $t_f = 10$ in 1/50th as many steps as for adapt_euler).

$\Rightarrow$ Adaptive RKF45 is using larger h-values, but matching the order of error

$\Rightarrow$ Adaptive RKF45 is much more accurate than adaptive euler

2) Adaptive RKF45 is much more efficient than adaptive euler. We are able to match the accuracy of adaptive euler while taking <u>much</u> fewer steps; although we fail more with adpative RKF45, the fails + successful steps still add up to less attempts than adaptive euler.

Two final things to note are that although Adaptive RKF45 takes much fewer steps than adaptive euler, RKF45 makes 5 calls to f (4 for RK4 + 1 more for RK5), whereas adaptive forward euler only makes 2 calls to f (since we're using Richardson extrap.) for each step. At the same time, RK4 (which is what we use as our soln update) has a better stability region than forward euler.



So, there is a give and take between the methods.

In summary:

- Adaptive RKF45 takes way less steps than adaptive Forward Euler to achieve the same order of error
- RKF45 makes 2.5 as many calls to f in each step
- RKF45 is more stable than forward euler.