HW 7

Due in class on Wednesday, May 2

Remember that you may (and should!) talk about the problems amongst yourselves, or discuss them with me or the TA, providing attribution for any good ideas you might get – but your final write-up should be your own.

1: Testing Matlab's ODE suite Consider the initial value problem

$$y' = \frac{1}{1+t} + \lambda(y - \ln(1+t)), \quad y(0) = 0.$$

The following computational experiments illustrate that even with good solvers like those in MATLAB, innocuous-looking ODEs can cause spectacular problems if used without due caution.

- Verify that the general solution to the differential equation is

  $$y = \log(1+t) + y(0)\exp(\lambda t).$$

  For the initial condition $y(0) = 0$, therefore, the solution is simply $y = \log(1+t)$.

- Use `ode45` and `ode15s` to evaluate the solution for $\lambda = -10^4$ for $0 \leq t \leq 20$. You should turn on statistics reporting, i.e.

  ```
  opt = odeset('Stats', 'on');
  [t,y] = ode45(@hw7p1fun, [0 20], 0, opt);
  ```

  Explain the statistics that you observe. Also plot the error in the computed solution on a semi-logarithmic scale. What do you observe?

- Use `ode45` and `ode15s` to evaluate the solution for $\lambda = 1$ for $0 \leq t \leq 20$. In each case, plot the difference between the computed solution and the true solution $\log(1+t)$. Again, it may be illuminating to turn on statistics reporting. Based on our discussion of step size selection in class, can you guess what went wrong?

- By plotting the error, verify that `ode45` gives an adequate solution if we manually restrict the maximum time step by a modest amount:

```
opt = odeset('Stats', 'on');
opt = odeset('MaxStep', 0.1);
[t,y] = ode45(@hw7p1fun, [0 20], 0, opt);
```

What happens if we use `ode15s` with the same max step size of 0.1?

2: AN-stability To determine A-stability, we look at the constant coefficient linear test equation

$$y' = \lambda y.$$

We say the method is A-stable if the numerical solutions decay whenever the real part of λ is negative, regardless of the step size. The concept of *AN-stability* generalizes A-stability. A method is AN-stable if the numerical solutions to test problems of the form

$$y' = \lambda(t)y.$$

decay whenever $\lambda(t) < 0$ for all $t > 0$, regardless of step size.

Show that backward Euler is AN-stable, but the trapezoidal rule is not.

3: Investigating an RKC method The `chebode` function implements an implicit Runge-Kutta type method with no step size adaptation. Use the method to solve $y' = y$ with $y(0) = 1$ for $0 \leq t \leq 10$, and plot the error (in the ∞-norm) against h on a log-log plot. What is the apparent order of convergence?

4: Circles within circles The equation

$$z = iz, \quad z(0) = 1$$

has the solution $z(t) = \exp(it) = \cos(t) + i\sin(t)$. Argue that applying trapezoidal rule with a fixed step size h gives a numerical solution that is correct up to a change in frequency, i.e.

$$z_k = \exp(i\omega_h t_k) = \cos(\omega_h t_k) + i\sin(\omega_h t_k).$$

Write the second-order Taylor expansion for ω_h as a function of h. Does the numerical solution tend to overestimate or underestimate the rate of rotation?