

# Lecture 7

Austin Baird

UNC Department of Mathematics

UNC Department of Biology

February 26, 2014

# Summary

- Last week we learned:
  - Local truncation error.
  - Second Order v first order method.
  - Error analysis.
  - Global error.
- Today we will:
  - Stability analysis.
  - Equilibrium solution to ODEs.
  - Eigen values and eigenvectors.

# Introduction

We want to understand what an eigenvalue is and how it affects the behavior of our ODE and also the behavior of our numerical techniques.

An **eigenvalue**  $\lambda$  and corresponding **eigenvector**  $\hat{v}$  are defined to have the relation:

$$A\hat{v} = \lambda\hat{v}$$

For a linear operator  $A$ . We can also extend this to our differential operators and get the following relation:

$$\begin{aligned}\frac{dy}{dt} &= \lambda y(t) \\ y(0) &= y_0\end{aligned}$$

# Stability and Eigenvalues

Consider the ODE we just introduced, its simple to see the solution to be:

$$y(t) = y_0 e^{\lambda t}$$

Seeing this solution we can begin to consider the idea of **stability**:

A solution to the initial value ODE is considered **stable** if small perturbations to the initial value do not change solution trajectories as  $\lim t \rightarrow \infty$ .

Now that we know the solution to this toy problem we can perturb the initial condition:  $y_0 = y_0 + \delta y_0$ .

# Continued

We can now plug in this new initial condition to the solution:

$$y(t) = (y_0 + \delta y_0)e^{\lambda t}$$

**Do It!** Use Matlab to show that as time is taken to be greater and greater the initial condition and the perturbed solution diverge.

It becomes clear that as  $\lim t \rightarrow \infty$  this solution becomes infinitely different from the original solution (the speed of change is dependent upon  $\delta y_0$ ).

We can then require the following for this model problem (for a stable bounded solution):

$$\operatorname{Re}(\lambda) < 0$$

Why just the real portion?

# What does this mean for our numerical solvers?

We now want to extend this analysis to our numerical solvers.  
Forward Euler:

$$y_{n+1} = y_n + \Delta t f(t_n, y_n)$$

Plug in our model problem:

$$y_{n+1} = y_n + \lambda \Delta t y_n$$

...

$$y_{n+1} = (1 + \lambda \Delta t) y_n$$

...

$$y_{n+1} = (1 + \lambda \Delta t)^n y_0$$

# Numerical Stability

We can now define the idea of numerical stability for our model problem:

*Def:* An initial value problem is **Stable** if small perturbations in initial conditions do not cause the numerical approximation to diverge away from the true solution. Given a bounded solution.

What does this mean for the analysis we just did? Let  $\lambda < 0$ . The analytical solution is bounded, but the numerical solution can still be unbounded (depending upon  $h = \Delta t$ ), so we require:

$$|1 + \lambda \Delta t| \leq 1$$

...

$$-1 \leq (1 + \lambda \Delta t) \leq 1$$

...

$$h < \frac{-2}{\lambda}$$

# Numerical Stability

In the previous slide we were able to find a necessary condition for convergence of our numerical solution.

*Try It!* Take the forward Euler solver that you have programed (or the solution one) and experiment with the error for different  $h, \lambda$ , and  $t_{final}$ . Use the above formula to get a step size that garentees convergence.

$$\lambda = -100$$

$$h < \frac{-2}{-100}$$

$$h = 0.02$$

This shows that the time step needs to be taken quite small as  $\lim t \rightarrow \infty$ . This analysis can be extended to other methods, including rk2!



# Equilibrium Points

Equilibrium points of an ODE are  $y^*$  that satisfy the relation:

$$\frac{dy^*}{dt} = F(y^*) = 0$$

Basically, it is a point in which as time progresses no more change is occurring. Ex (logistic equation, a population model with a carrying capacity,  $\mathcal{K}$ ):

$$\frac{dy}{dt} = ry\left(1 - \frac{y}{\mathcal{K}}\right)$$

Find the equilibrium points of this solution by hand, then test yourself using matlab. Test different initial conditions and determine stability of points.

# Eigenvalues and Systems of ODEs

For our model problem  $\lambda$  is the eigenvalue of the equation and we were able to see some relations which made solutions bounded. Turns out eigenvalues can tell us a lot about the long term behavior of a system of ODEs (linear and non-linear).

Consider the system:

$$\begin{aligned}\frac{dy_1(t)}{dt} &= ay_1 + by_2 \\ \frac{dy_2(t)}{dt} &= cy_1 + dy_2\end{aligned}$$

Write this as a matrix equation!

# Eigenvalues and Systems of ODEs

Once we have a matrix equation of the form:

$$\begin{pmatrix} \frac{dy_1(t)}{dt} \\ \frac{dy_2(t)}{dt} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

We can solve for the eigenvalues  $\lambda_1, \lambda_2$  and eigenvectors  $\hat{b}_1, \hat{b}_2$  of  $\mathcal{A}$  (the 2x2 matrix). This will give us the solution of the form:

$$y(\hat{t}) = c_1 e^{\lambda_1 t} \hat{b}_1 + c_2 e^{\lambda_2 t} \hat{b}_2$$

We want to use matlab to do this, since these systems can get quite large!

» [eigvec, eigenval] = eig( $\mathcal{A}$ )

Now see what this does for a given  $\mathcal{A}$ ! Also, what can we say about the solutions for given  $\lambda_1, \lambda_2$ ?

# Revisiting Stability

We want to say something qualitative about the solution trajectories (plotting  $y_1$  v.  $y_2$  for a given initial value) given the eigenvalues  $\lambda_1, \lambda_2$ .

- **Asymptotic Stability:** Solutions  $y(t) \rightarrow 0$  as  $t \rightarrow \infty$ .  
Satisfied when:  $\text{Re}(\lambda) < 0$ . Negative real part eigenvalues.
- **Stability:** Solutions  $y(t)$  stay bounded as  $t \rightarrow \infty$ . Only imaginary eigenvalues.
- **Unstable:** Solutions  $y(t) \rightarrow \infty$  as  $t \rightarrow \infty$ .  $\text{Re}(\lambda) > 0$ . Real part of the eigenvalues greater than zero.

This gives us a good overview of how linear ODEs will behave depending on their eigenvalues.

# Plotting Trajectories Matlab

To complete this next homework we need to know how to plot trajectories of our ODEs to see their behavior. We did this once with the Lorenz attractor system. We now want to add many trajectories to the same graph:

- start a figure and hold at the begging of the script:
- `>> figure(1)`
- `>> hold on`
- `>> for i = 0:1:10,`
- `>> initial = [i+3 (i-2)/3]`
- `>> solve your system with ode45`
- `>> plot((y(:,1),y(:,2))`
- `>> end`
- `>> hold off`

# Homework6

We want to investigate different system of ODEs of the form:

$$\begin{pmatrix} \frac{dy_1(t)}{dt} \\ \frac{dy_2(t)}{dt} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

for various a,b,c,d's:

a=2, b=-5, c=1, d=-2

and

a=0.5, b=0.9, c=-1.2, d=-1.7

# Homework6

- Print out the eigen values for each system presented on the previous page.
- Print out how you'd expect the system to behave given the eigen values.
- Plot the exact solution for the system (using eigenvalues and eigenvectors) from  $t=[0,10]$ .
- For each example, plot out the trajectory for at least ten different initial values on the same graph.