Math3090 - Computational Mathematics

Asaf Jeff Dror
York University
(Dated: December 7, 2011)

CONTENTS

I. INRODUCTION

Website: `http://math.yorku.ca/~moghadas/Comp_Math_Fall_2011.htm`

A. Due Dates

Assignments: First (October 3); Second (October 24); Third (November 25); Fourth (due on final exam date)
Midterm Test: Monday October 24, 2011 (Time: 11:30 - 1:00)
Final Exam: TBD

II. SUMMARY OF LAST LECTURE

Def 1. An isolated critical point is a point such that there exists a neighborhood of this point in which the system has no other critical points.

Def 2. Stable Node: Isolated critical point x^* is called a stable node if there exists a neighborhood such that for any solutions $x(t)$ starting in this neighborhood at time $t = t_0$ then $x(t)$ is defined at all $t \geq t_0$ and remain in this neighborhood for all $t \geq t_0$.

Def 3. A stable node x^* is called locally asymptomatically stable if there exists a neighborhood of x^* such that for any solution $x(t)$ starting in this neighborhood at time t_0 then

$$\lim_{t\to\infty} x(t) = x^* \tag{II.1}$$

Example 1: Determine the type of critical point of the system:

$$\begin{cases} \dot{x} = y = 0 \\ \dot{y} = -x = 0 \end{cases} \tag{II.2}$$

The only critical point is $(0, 0)$. Solving the system gives:

$$x(t) = r\sin(t + \alpha) \tag{II.3}$$
$$y(t) = r\cos(t + \alpha) \tag{II.4}$$

Hence

$$x(t)^2 + y(t)^2 = r^2 \tag{II.5}$$
$$\tag{II.6}$$

The critical point, (0,0), is a stable node. The solutions $X(t) = (x(t), y(t)), x^2 + y^2 = r^2$ stays inside a given neighborhood but they do not go to $(0, 0)$ so they are not locally asymptotically stable.

$$\lim_{t\to\infty} x^2(t) + y^2(t) = r^2 \neq 0 \tag{II.7}$$

The only solution that this is equal to the critical point is the solution for $r = 0$ Example 2: Determine the type of critical point for the system: (note that prime here designates derivative with respect to t)

$$\begin{cases} x' = -x \\ y' = -y \end{cases} \tag{II.8}$$

The solutions are

$$\begin{cases} x(t) = c_1 e^{-(t-t_0)} \\ y(t) = c_2 e^{-(t-t_0)} \end{cases} \tag{II.9}$$

By equating x and y to 0 it is easy to find that the only solution to this system is (0,0). For x and y not equaling zero, one can write:

$$y(t) = \frac{c_2}{c_1} x(t) \tag{II.10}$$

This is clearly the equation of a line passing through the origin. Note that $\lim_{t\to\infty} x(t) = lim_{t\to\infty} y(t) = 0$. Hence a solution that is in a neighborhood will not leave that neighborhood making the point $(0, 0)$ a locally asymptotic

stable node. This is true for every combination of c_1 and c_2. *Note: If a point is locally asymptotically stable it enables it is a stable node but the converse is not true.*
Question: How do we determine the type of critical points of a system without having the general solution?
Recall the system:

$$X(t) = F(x_1(t), x_2(t), ...x_n(t)) \tag{II.11}$$

$$\mathcal{X} = (x_1(t), x_2(t), ..., x_n(t)) \tag{II.12}$$

$$\begin{pmatrix} x_1'(t) \\ x_2'(t) \\ ... \\ x_n'(t) \end{pmatrix} = \begin{pmatrix} f_1(x_1(t), ..., x_n(t)) \\ f_2(x_1(t), ..., x_n(t)) \\ ... \\ f_n(x_1(t), ..., x_n(t)) \end{pmatrix} \tag{II.13}$$

Def 4. The linearize system around any isolate critical point of equation II.12 is given by the Jacobian of the system is defined as

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & ... & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & ... & \frac{\partial f_2}{\partial x_n} \\ ... & ... & ... \\ \frac{\partial f_n}{\partial x_1} & ... & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \tag{II.14}$$

Consider the system

$$\begin{cases} x' = rx(1 - x/k) - axy = f_1(x, y) \\ y' = cxy - dy = f_2(x, y) \end{cases} \tag{II.15}$$

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{pmatrix} = \begin{pmatrix} r(1 - x/k) - rx/k - ay & -ax \\ cy & cx - d \end{pmatrix} \tag{II.16}$$

The linearized system at the critical point x^* is given by

$$\mathcal{X}' = J_{x^*}\mathcal{X} \tag{II.17}$$

The three critical points of the system are given by

$$\tag{II.18}$$

$$x_1^* = (0, 0) \tag{II.19}$$

$$x_2^* = (k, 0) \tag{II.20}$$

$$x_3^* = (d/c, \frac{r}{a}(1 - \frac{d}{ck})) \tag{II.21}$$

The linearized system associated with $x_1^* = (0, 0)$.

$$J(x_1^*) = \begin{pmatrix} r & 0 \\ 0 & -d \end{pmatrix} \tag{II.22}$$

Hence the linearized system is

$$\begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix} = \begin{pmatrix} r & 0 \\ 0 & -d \end{pmatrix} \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} \tag{II.23}$$

Linearized system of equation II.15 around $x_2^* = (k, 0)$

$$J(x_2^*) = \begin{pmatrix} -1 & -ak \\ 0 & ck - d \end{pmatrix} \tag{II.24}$$

Hence the linearized system is

$$\begin{pmatrix} x'(t) \\ y'(t) \end{pmatrix} = \begin{pmatrix} -1 & -ak \\ 0 & ck - d \end{pmatrix} \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} \tag{II.25}$$

The linearized system for the last point is given as homework.

Missed a lecture.

III. LAST LECTURE SUMMARY

A. Determining the Type of Critical Point

Suppose x^* is an isolated crtical point and the Jocobian of the system of x^*lies real eigenvalues then

- x^* is locally asymptotically stable if all the eigenvalues have a negative sign
- x^* is unstable if all the eigenvalues have positive sign
- x^* is a stable node if all the eigenvalues have a negative sign or positive sign, and atleast two eigenvalues have opposite sign

B. Field Direction

Suppose that x^* is an isolated critical point and eigenvalue of the Jacobian a x^* are complex numbers then

- x^* is a locally asymptotically stable if the real point of all the eigenvalues have a negative sign

---

Lecture ? - Sept 21, 2011
Last lecture considered the system:

$$\begin{cases} x' = 2x\left(1 - \frac{x}{10}\right) - 3xy \\ y' = (x-5)\,y \end{cases} \tag{III.1}$$

The critical points are

$$(0,0) \qquad (10,0) \qquad (5,1/3) \tag{III.2}$$

Suppose the eigenvalues of the Jacobian of the system at a critical point x^* are complex

1. The point x^* is a locally asymptotically stable if all the eigenvalues have negative real parts
2. The point x^* is an unstable node if all the eigenvalues have positive real parts
3. The point x^* is a saddle node if all the eigenvalues have positive or negative real parts and at least two eigenvalues have real parts with opposite signs.

The Jacobian is easy enough to find. It is given by

$$J(x,y) = \begin{pmatrix} 2\left(1 - \frac{x}{10}\right) - \frac{2x}{10} - 3y & -3x \\ y & x-5 \end{pmatrix} \tag{III.3}$$

The Jacobian evaluated at $(5,1/3)$ is given by

$$J(5,1/3) = \begin{pmatrix} -1 & -15 \\ 1/3 & 0 \end{pmatrix} \tag{III.4}$$

The eigenvalues of this matrix are given by

$$\lambda = -\frac{1}{2} \pm \frac{\sqrt{19}i}{2} \tag{III.5}$$

Since the real part is negative is implies that the point $(5,1/3)$ is asymptotically stable.
In a predator/prey system the first quadrant is the only *biologically feasible region.* The point (0,0) is called the extinction point. The point (k,0) corresponds to one - single prey species existing (no predator). The critical points that do not contain zeros are called survival points. If there are no prey, i.e. a point $(0,k)$ then it is inevitable that the system will tend to go to $(0,0)$, the extinction point.

IV. TWO-DIMENSIONAL SYSTEMS

Suppose $X^* = (x^*, y^*)$ is a critical point of the system

$$x' = f(x,y) \qquad y' = g(x,y) \tag{IV.1}$$

For general non-linear functions f and g. i.e.

$$f(x^*, g^*) = 0 \qquad g(x^*, y^*) = 0 \tag{IV.2}$$

Recall from calculus: Taylor expansion around x^* in the domain of f (x) is given by

$$f(x) \approx f(x^*) + (x - x^*)f'(x^*) + \frac{1}{2}(x - x^*)^2 f''(x^*) + ... \tag{IV.3}$$

The Taylor expansion around a point (x^*, y^*) for the function $f(x, y)$ is given by

$$f(x,y) \approx f(x^*, y^*) + (x - x^*)f_x(x^*, y^*) + (y - y^*)f_y(x^*, y^*) + f_{xy}(x^*, y^*)(x - x^*)(y - y^*) + ... \tag{IV.4}$$

Suppose $(x(t), y(t))$ of the system that starts in a small neighborhood of $X^* = (x^*, y^*)$. Define $x(t), y(t)$ as

$$x(t) \equiv x^* + u(t) \tag{IV.5}$$
$$y(t) \equiv y^* + v(t) \tag{IV.6}$$

Equivalently:

$$u(t) \equiv x(t) - x^* \tag{IV.7}$$
$$v(t) \equiv y(t) - y^* \tag{IV.8}$$

Taking derivatives gives:

$$u'(t) = x'(t) \tag{IV.9}$$
$$v'(t) = y'(t) \tag{IV.10}$$

Now we can expand f and g as

$$f(x,y) \approx \cancelto{0}{f(x^*, y^*)} + f_x(x^*, y^*)\overbrace{(x - x^*)}^{u(t)} + f_y(x^*, y^*)\overbrace{(y - y^*)}^{v(t)} + \overbrace{f_{xy}(x^*, y^*)\overbrace{(x - x^*)}^{u(t)}\overbrace{(y - y^*)}^{v(t)}+...}^{\mathcal{O}(u^2, v^2)} \tag{IV.11}$$

$$g(x,y) \approx \cancelto{0}{g(x^*, y^*)} + g_x(x^*, y^*)\overbrace{(x - x^*)}^{u(t)} + g_y(x^*, y^*)\overbrace{(y - y^*)}^{v(t)} + \overbrace{g_{xy}(x^*, y^*)\overbrace{(x - x^*)}^{u(t)}\overbrace{(y - y^*)}^{v(t)}+...}^{\mathcal{O}(u^2, v^2)} \tag{IV.12}$$

Simplifying gives

$$u' = f(x,y) \approx f_x(x^*, y^*)u + f_y(x^*, y^*) + \mathcal{O}(u^2, v^2) \tag{IV.13}$$
$$v' = g(x,y) \approx g_x(x^*, y^*)u + g_y(x^*, y^*) + \mathcal{O}(u^2, v^2) \tag{IV.14}$$

The advantage of this system that according to a certain theorem (see later) we can ignore the second order terms and use the now simpler system to solve.

Lecture 7 - Sept 23, 2011 Consider the result discussed above. If we are looking in a neighborhood around (x^*, y^*) that is close to (x^*, y^*) then u and v are small. In this case we can ignore the terms second order in u and v. In this case the result above can be expressed in short-form notation as

$$u' \approx f_x^* u + f_y^* v \tag{IV.15}$$
$$v' \approx g_x^* + g_y^* v \tag{IV.16}$$

In matrix form this can be written as

$$\begin{pmatrix} u' \\ v' \end{pmatrix} = \overbrace{\begin{pmatrix} f_x^* & f_y^* \\ g_x^* & g_y^* \end{pmatrix}}^{J(x^*,y^*)} \begin{pmatrix} u \\ v \end{pmatrix} \tag{IV.17}$$

The matrix relating the derivatives of u and v to u and v is the Jacobian!

$$\mathcal{U}' = J(x^*, y^*)\mathcal{U} \tag{IV.18}$$

$$\det(\lambda I - J^*) = \det \begin{pmatrix} \lambda - f_x^* & -f_y^* \\ -g_x^* & \lambda - g_y^* \end{pmatrix} \tag{IV.19}$$

$$= (\lambda - f_x^*)(\lambda - g_y^*) - f_y^* g_x^* \tag{IV.20}$$

$$= \lambda^2 - \underbrace{(f_x^* + g_y^*)}_{p}\lambda + \underbrace{f_x^* g_y^* - f_y^* g_x^*}_{q} \tag{IV.21}$$

Setting the characteristic equation to 0 (with the definitions shown above) gives

$$\lambda - p\lambda + q = 0 \tag{IV.22}$$

$$\lambda_\pm = \frac{p}{2} \pm \frac{\sqrt{p^2 - 4q}}{2} \tag{IV.23}$$

a. *Case 1:* $q < 0$ Then

$$p^2 - 4q > p^2 \tag{IV.24}$$

$$\Rightarrow \lambda_- < 0 \qquad \lambda_+ > 0 \tag{IV.25}$$

Note this is true regardless of the sign of p. Since the eigenvalues are both positive and negative (x^*, y^*) is a saddle node.

b. *Case 2:* $0 \leq q \leq \frac{p^2}{4}$, $p > 0$ In this case $p^2 - 4q > 0$ but at the same time $\frac{p}{2} < \frac{\sqrt{p^2-4q}}{2}$. Since $p > 0$, the eigenvalues are both positive. This means that the critical point is unstable.

c. *Case 3:* $0 \leq q \leq \frac{p^2}{4}$, $p < 0$ In this case $p^2 - 4q > 0$ but at the same time $\frac{p}{2} < \frac{\sqrt{p^2-4q}}{2}$. Since $p < 0$, the eigenvalues are both negative. This means that the critical point is locally asymptotically stable.

d. *Case 4:* $q > \frac{p^2}{4}$, $p > 0$ In this case $\frac{\sqrt{p^2-4q}}{2}$ is imaginary. The only part that concerns us is the real part. Since the sign of p is positive, both eigenvalues are positive and hence the critical point is unstable.

e. *Case 5:* $q > \frac{p^2}{4}$, $p < 0$ In this case $\frac{\sqrt{p^2-4q}}{2}$ is imaginary. The only part that concerns us is the real part. Since the sign of p is negative, both eigenvalues are negative and hence the critical point is locally asymptotically stable
These ideas are summarized in figure 1

f. *Case 6:* $p = 0$ If $p = 0$ then we have a center that is in the boundary of stability regions. This is a difficult case to treat and is beyond the context of this course
Notice in figure 1 the boundaries of $p^2/4 = q$ ended up begin irrelevant.

V. EPIDEMIC SPREAD

Here we consider the spread of a disease in a population (Epidemic)
The assumptions in this model are:

1. Disease is incurable
2. Recovered individuals are immune against reinfection

The elements (variables) of the system are:

- Susceptible (S)

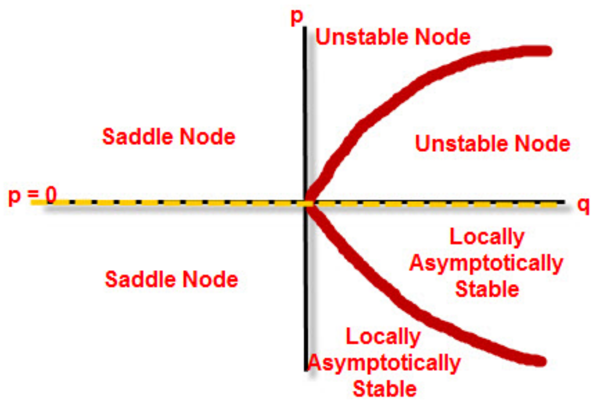

FIG. 1. Graph of the results

- Infected (I)
- Recovered (R)

The parameters of the system are

- Transmission rate of infection(β)
- Recovery rate (α)
- Death rate (d)
- Birth rate (b) (assume b is constant)

Lecture 7 - September 26th, 2011 S' is the rate of change in the susceptible population. For the transmission rate it is proportional to both the amount of people in the susceptible class and the amount of people in the infected class. The deaths are proportional to the number of people in the susceptible class. S' is given by

$$S' = \overbrace{b}^{\text{Births}} - \overbrace{\beta SI}^{\text{Transmission}} - \overbrace{dS}^{\text{Natural Deaths}} \equiv f(S, I, R) \tag{V.1}$$

$$I' = \overbrace{\beta SI}^{\text{Transmission}} - \overbrace{\alpha I}^{\text{Recovery}} - \overbrace{dI}^{\text{Natural Death}} \equiv g(S, I, R) \tag{V.2}$$

$$R' = \overbrace{\alpha I}^{\text{Recovery}} - \overbrace{dR}^{\text{Natural Death}} \equiv h(S, I, R) \tag{V.3}$$

Setting each of these terms to zero gives the critical points:

$$I' = 0 \tag{V.4}$$
$$\Rightarrow I\left(\beta S - (\alpha + d)\right) = 0 \tag{V.5}$$
$$\Rightarrow \begin{cases} I = 0 \Rightarrow R = 0; S = b/d \\ S = \frac{\alpha+d}{\beta}; I = \frac{\beta b - d(\alpha+d)}{\beta(\alpha+d)}; R = \frac{\alpha(\beta b - d(\alpha+d)}{\beta d(\alpha+d)} \end{cases} \tag{V.6}$$

We have two critical points. The critical point $\left(\frac{b}{d}, 0, 0\right)$ is an infection free or disease free state equilibrium (since $I = 0$). The other critical point is given by

$$\left(\frac{\alpha + d}{\beta}, \frac{\beta b - d(\alpha + d)}{\beta(\alpha + d)}, \frac{\alpha(\beta b - d(\alpha + d))}{\beta d(\alpha + d)}\right) \tag{V.7}$$

Since there are non zero values in each variable this is called the infection equilibrium or endemic state. To analyze this critical point we find the Jacobian of the system:

$$J = \begin{pmatrix} f_S & f_I & f_R \\ g_S & g_I & g_R \\ h_S & h_S & h_R \end{pmatrix} \tag{V.8}$$

$$= \begin{pmatrix} -\beta I - d & -\beta S & 0 \\ \beta I & \beta S - \alpha & 0 \\ 0 & \alpha & -d \end{pmatrix} \tag{V.9}$$

$$J\left(\frac{b}{d}, 0, 0\right) = \begin{pmatrix} -d & \frac{-\beta b}{d} & 0 \\ 0 & \frac{\beta b}{d} - \alpha - d & 0 \\ 0 & \alpha & -d \end{pmatrix} \tag{V.10}$$

This gives the eigenvalues:

$$\begin{aligned} \lambda_1 &= -d < 0 \\ \lambda_2 &= -d < 0 \\ \lambda_3 &= \tfrac{\beta b}{d} - (\alpha + d) \end{aligned} \tag{V.11}$$

In order to have a stable node we require all the eigenvalues to be negative this occurs if

$$\frac{\beta b}{d} - (\alpha + d) < 0 \tag{V.12}$$

$$\Rightarrow \underbrace{\frac{\beta b}{d(\alpha + d)}}_{R_0} < 1 \tag{V.13}$$

R_0 is called the basic reproduction number. If $R_0 > 1$ then an epidemic will start and if it is less than 1 then the epidemic will die out.
Homework: Show that if $R_0 > 1$ than the second critical point is locally asymptotically stable.

VI. EPIDEMIC SPREAD (INFLUENZA)

We still have the same three variables: S,I, and R discussed above. For influenza a person can be infected but still not show any symptoms and can also affect others but with a lower transmission rate. Thus there is another variable A which corresponds to just such person. Use the same parameters used last time but now we have two transmission rates:

$$\beta_1 = \text{Transmission rate of infected population} \tag{VI.1}$$

$$\beta_2 = \text{ Transmission rate of infected with symptoms populations} \tag{VI.2}$$

$$\alpha_1 = \text{ Recovery rate of infected population} \tag{VI.3}$$

$$\alpha_2 = \text{ Recovery rate of infected with symptoms populations} p = \text{ Fraction of people that are immune to the disease (I think)} \tag{VI.4}$$

$$\begin{aligned} S' &= b - \beta_1 SI - \beta_2 SA - dS && \text{(VI.5)} \\ I' &= p\left(\beta_1 SI + \beta_2 SA\right) - \alpha_1 I - dI && \text{(VI.6)} \\ A' &= (1-p)\left(\beta_1 SI + \beta_2 SA\right) - \alpha_2 I - dA && \text{(VI.7)} \\ R' &= \alpha_1 I + \alpha_2 A - dR && \text{(VI.8)} \end{aligned}$$

Homework: Find the critical points of the system and determine their type of stability

---

Lecture 8 - Sept 28, 2011

VII. DISCRETIZATION USING NUMERICAL METHODS

Since many models can not be solved to find the solutions in close forms, we use numerical methods to illustrate the solutions or approximate them. The process by which the solutions are approximated is called **Discretization**. The idea is the find the solutions at different discrete points and interpolate the solution in between those points.

A. Euler Method

Suppose the system $x'(t) = f(t, x(t))$ is defined on the interval $t \in [a, b]$. Let $N > 0$ by an integer and $h = \frac{b-a}{N}$. The difference between neighboring points is h. i.e.

$$t_{i+1} - t_i = h \tag{VII.1}$$

Using Taylor expansion of $x(t)$ at t_{i+1} we have

$$x(t_{i+1}) = x(t_i) + \frac{\partial x(t_i)}{\partial t} \overbrace{(t_{i+1} - t_i)}^{h} + ... \tag{VII.2}$$

Euler method says that

$$x(t_{i+1}) = x(t_i) + hf(t_i, x(t_{i+1}) \tag{VII.3}$$

Euler method approximates the solutions with local error or truncation error of $\frac{h^2}{2}x''(\xi_i)$ for some $\xi_i(t_i, t_{i+1})$

$$x'(t) = f(t, x(t)) \tag{VII.4}$$

$$x'(t) = \lim_{h\to 0} \frac{x(t+h) - x(t)}{h} \tag{VII.5}$$

$$f(t_i, x(t_i)) \approx \frac{x(t_{i+1}) - x(t_i)}{h} \tag{VII.6}$$

B. Modification of Euler Method

The integral over a function $f(x)$ can be approximated using:

$$\int_{x_0}^{x_n} f(x)dx \leftrightarrow y_{i+1} = y_i + hf(x_i) \tag{VII.7}$$

Lecture 9 - Sept 30, 2011 Recall: Given

$$x' = f(t, x(t)) \tag{VII.8}$$

with $t \in [a, b]$. We can split the interval into equally spaced points such that

$$t_{i+1}t_i = h \tag{VII.9}$$

Euler method says that

$$x(t_{i+1}) = x(t_i) + hf(t_i, x(t_i)) + \mathcal{O}(h^2) \tag{VII.10}$$

Where the error term is given by

$$\frac{h^2}{2}x''(\xi_i); \qquad \xi_i \in (t_i, t_{i+1}) \tag{VII.11}$$

The Euler method has $\mathcal{O}(h^2)$ error. Develop a method to reduce $\mathcal{O}(h^2)$ to $\mathcal{O}(h^3)$.
Recall: given a function $f(x)$ defined on the domain (x_0, x_N). You can approximate the integral over this function by a Riemann sum. Let y_i represent the area from x_0 to x_i

$$y_{i+1} = y_i + hf(x_i) \tag{VII.12}$$

The method called **Trapezoidal** method is given by

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i) + f(x_{i+1})\right) \tag{VII.13}$$

Analogously to approximate the function at each point can be done by

$$x(t_{i+1}) = x(t_i) + \frac{h}{2}\left(f(t_i, x(t_i)) + f(t_{i+1}, x(t_{i+1}))\right) \tag{VII.14}$$

Notice the term $f(t_{i+1}, x(t_{i+1}))$ contains the variable we want to look for. So we need to find a way to get rid of this dependance. Suppose

$$x(t_{i+1}) = x(t_i) + ah \tag{VII.15}$$

For some a. If we Taylor expand $f(t_{i+1}, x(t_{i+1})$:

$$f(t_{i+1}, x(t_{i+1}) = f(t_i, x(t_i)) + f_t(t_i, x(t_i))\overbrace{(t_{i+1} - t_i)}^{h} + f_x(t_i, x(t_i))\overbrace{(x(t_{i+1}) - x(t_i))}^{ah} + \mathcal{O}\left(h^2\right) \tag{VII.16}$$

$$x(t_{i+1}) = x(t_i) + \frac{h}{2}f(t_i, x(t_i)) + \frac{h}{2}\left(f(t_{i+1}, x(t_{i+1}))\right) \tag{VII.17}$$

$$x(t_i) + hf(t_i, x(t_i)) + \frac{h^2}{2}\left(f_t(t_i, x(t_i)) + af_x(t_i, x(t_i))\right) + \mathcal{O}(h^3) \tag{VII.18}$$

The system was $x' = f(t, x(t))$. Lets write the solution $x(t)$ in Taylor expansion"

$$x(t_{i+1}) = x(t_i) + x'(t_i)\overbrace{(t_{i+1} - t_i)}^{h} + x''(t_i)\overbrace{\frac{(t_{i+1} - t_i)^2}{2}}^{h^2} + \mathcal{O}\left(h^3\right) \tag{VII.19}$$

$$x(t_{i+1}) = x(t_i) + hf(t_i, x(t_i)) + x''(t_i)\frac{h^2}{2} + \mathcal{O}(h^3) \tag{VII.20}$$

but knowing that $x' = f$ and using the chain rule:

$$x''(t_i) = (x'(t_i))' = \overbrace{\frac{\partial t}{\partial t}}^{1} f_t(t_i, x(t_i)) + \overbrace{\frac{\partial x}{\partial t}}^{f(t,x(t))}(t_i) f_x(t_i, x(t_i)) \tag{VII.21}$$

Now finally our equation can be simplified to

$$x(t_{i+1}) = x(t_i) + hf(t_i, x(t_i)) + \frac{h^2}{2}\left(f(t_i, x(t_i)) + f(t_i, x(t_i))f_x(t_i, x(t_i))\right) + \mathcal{O}(h^3) \tag{VII.22}$$

Comparing with equation VII.18 says that

$$a = f(t_i, x(t_i)) \tag{VII.23}$$

We now have an approximation of the exact solution that is of order h^3. Going back to the original system:

$$x'(t) = f(t, x(t)) \tag{VII.24}$$

The modified Euler method is given by

$$\boxed{x(t_{i+1}) = x(t_i)\frac{h}{2}\left(f(t_i, x(t_i)) + f(t_{i+1}, x(t_i) + hf(t_i, x(t_i)))\right)} \tag{VII.25}$$

Note that this is a two step method since we must first calculate the argument of f in the last term in order to calculate $x(t_{i+1})$.
For convenience define

$$y_{i+1} = x(t_i) + hf(t_i, x(t_i)) \tag{VII.26}$$

Then

$$x(t_{i+1}) = x(t_i) + \frac{h}{2}\left(f(t_i, x(t_i)) + f(t_{i+1}, y_{i+1})\right) \tag{VII.27}$$

Alternatively this can be written as

$$xt_{i+1} = \frac{1}{2}\left(x(t_i) + y_{i+1} + hf(t_{i+1}, y_{i+1})\right) \tag{VII.28}$$

Example: For the equation

$$x' = \overbrace{-x + t + 1}^{f(t,x(t))} \tag{VII.29}$$

with $x(0) = 1$ and $t \in [0, 5]$ calculate $x(2)$ using the Modified Euler Method with $h = 1$ and compare it with the exact solution $x = e^{-t} + t$ at $t = 2$ ($x(2) = 2.135335283$).

Lecture 10- Oct 3rd, 2011
Assignment 1 - Q2:

$$S_0 = 100kids \tag{VII.30}$$
$$\text{5 new kids/year} : n \tag{VII.31}$$
$$\beta = 0.0005/year/kid \tag{VII.32}$$
$$\text{Recovery rate: } \gamma = \frac{1}{26}/year \tag{VII.33}$$
$$\text{Graduation rate: } g \tag{VII.34}$$

Thus

$$S' = n - \beta SI - gS \tag{VII.35}$$
$$I' = \beta SI - \gamma I - gI \tag{VII.36}$$

All the parameters are given except for graduation rate. Setting the derivatives to zero gives two feasible critical points. One is the infection free situation. In the infection free equilibrium

$$(S, I) = \left(\frac{5}{g}, 0\right) = (100, 0) \tag{VII.37}$$

This equation sets the value of g.
For the last part of the second question consider a simple example: Suppose we have an initial number of susceptibles S_0 and an epidemic is started with I_0 initial number of infection. If infected individuals are recovered at the rate γ then how many infections we have at the end of epidemic? (i.e. what is the total number of infections during the epidemic). The system is given by

$$S' = -\beta SI \tag{VII.38}$$
$$I' = \beta SI - \gamma I \tag{VII.39}$$

There is only one feasible equilibrium in this system. The equilibrium with $I = 0$. Hence as $\lim_{t\to\infty} I = 0$. Dividing equation VII.38 by S gives:

$$\frac{S'}{S} = -\beta I \tag{VII.40}$$
$$\int_0^\infty \frac{S'}{S} = \int_0^\infty -\beta I \tag{VII.41}$$
$$\Rightarrow \ln\left(\frac{S(\infty)}{S(0)}\right) = -\beta \int_0^\infty I \tag{VII.42}$$

Now add equations VII.38 and VII.39:

$$S' + I' = -\gamma I \tag{VII.43}$$

$$\int_0^\infty (S' + I') = -\gamma \int_0^\infty I \tag{VII.44}$$

$$(S + I)\Big|_0^\infty = -\gamma \int_0^\infty I \tag{VII.45}$$

This result gives (for convenience define $S_\infty \equiv S(\infty)$)

$$\int_0^\infty I = \frac{S_0 + I_0 - S_\infty}{\gamma} \tag{VII.46}$$

Substituting this result into equation VII.42 gives

$$\ln\left(\frac{S_\infty}{S_0}\right) = -\beta\left(\frac{S_0 + I_0 - S_\infty}{\gamma}\right) \tag{VII.47}$$

This equation can be solved numerically for S_∞. The total number of infections during epidemic is given by

$$S_0 - S_\infty \tag{VII.48}$$

Continue with lecture:
Recall: The modified Euler method

$$y_{i+1} = x(t_i) + hf\left(t_i, x(t_i)\right) \tag{VII.49}$$

$$x(t_{i+1}) = x(t_i) + \frac{h}{2}\left(f(t_i)x(t_i) + f(t_{i+1}, y_{i+1})\right) \tag{VII.50}$$

Combining the equations gives

$$x(t_{i+1}) = \frac{1}{2}\left(x(t_i) + y_{i+1} + hf(t_{i+1}, y_{i+1})\right) \tag{VII.51}$$

Example:

$$x' = x + t + 1 \equiv f\left(t, x(t)\right); \qquad x(0) = 1 \tag{VII.52}$$

$$t \in [0, 5]; \qquad \text{find } x(2); h = 1 \tag{VII.53}$$

$$x(2) = \frac{1}{2}\left(x_1 + y_2 + hf(2, y_2)\right) \tag{VII.54}$$

$$y_2 = x(1) + hf(1, x(1)) \tag{VII.55}$$

but we need to know $x(1)$

$$x(1) = \frac{1}{2}\left(x(0) + y_1 + hf(1, y_1)\right) \tag{VII.56}$$

to get $x(1)$ we require y_1.

$$y_1 = x(0) + hf(t_0, x(0)) \tag{VII.57}$$

$$= 1 + (-1 + 1) \tag{VII.58}$$

$$= 1 \tag{VII.59}$$

This gives

$$x(1) = \frac{1}{2}\left(1 + 1 + f(1, 1)\right) \tag{VII.60}$$

$$= \frac{3}{2} \tag{VII.61}$$

Now we can get y_2

$$y_2 = 3/2 + f(1, 3/2) \tag{VII.62}$$
$$= 2 \tag{VII.63}$$

Finally we can get $x(2)$:

$$x(2) = \frac{1}{2}\left(3/2 + 2 + f(2, 2)\right) \tag{VII.64}$$
$$= 9/4 \tag{VII.65}$$

The error is

$$2.25 - 2.135 = 0.115 \tag{VII.66}$$

The error is of order h^3. If the procedure was repeated a set size of 0.5 then the result would be much closer to the exact value.

Lecture 11 - October 05, 2011
Continue from last lecture: Consider the system

$$x' = x + t - 1; \qquad x(0) = 1 \tag{VII.67}$$

Now solve using Modified Euler Method with $h = 1/2$. The iterations to reach the necessary values are as follows:

$$x(2) = \frac{1}{2}\left(x(3/2) + y_2 + hf(2, y_2)\right) \tag{VII.68}$$
$$y_2 = \left(x(3/2) + hf(3/2, x(3/2))\right) \tag{VII.69}$$
$$x(3/2) = \frac{1}{2}\left(x(1) + y_{3/2} + hf(3/2, y_{3/2})\right) \tag{VII.70}$$
$$y_{3/2} = \left(x(1) + hf(1, x(1))\right) \tag{VII.71}$$
$$x(1) = \frac{1}{2}\left(x(1/2) + y_1 + hf(1, y_1)\right) \tag{VII.72}$$
$$y_1 = \left(x(1/2) + hf(1/2, x(1/2))\right) \tag{VII.73}$$
$$x(1/2) = \frac{1}{2}\left(x(0) + y_{1/2} + hf(1/2, y_{1/2})\right) \tag{VII.74}$$
$$y_{1/2} = \left(x(0) + hf\,(0, x_0)\right) \tag{VII.75}$$

Finally inserted $x(0) = 1$ gives the solutions:

$$y_{1/2} = 1 \tag{VII.76}$$
$$x(1/2) = \frac{9}{8} \tag{VII.77}$$
$$y_1 = \frac{21}{16} \tag{VII.78}$$
$$x(1) = \frac{89}{64} \tag{VII.79}$$
$$y_{3/2} = \frac{217}{128} \tag{VII.80}$$
$$x(3/2) = \frac{893}{512} \tag{VII.81}$$
$$\vdots \quad \vdots \quad \vdots \tag{VII.82}$$

C. Midpoint Method

Recall when you take a Reimann sum you have a choice about where to discretize your function. One choice is such that your "boxes" are centered at the midpoint of the function.

$$y_{i+1} = y_i + hf(x_{i+h/2}) \tag{VII.83}$$

Midpoint method says: Suppose $x' = f(t,x)$ is defined on the interval $[a,b]$ with the interval split into $N > 0$ discrete points.

$$x(t_{i+1}) = x(t_i) + hf(t_i + h/2, x(t_i) + bh) \tag{VII.84}$$

For some b. Expanding $f(t_i + h/2, x(t_i) + bh)$ in a Taylor series:

$$x(t_{i+1}) = x(t_i) + h\left(f(t_i, x(t_i)) + f_t(t_i, x(t_i))(\cancel{t_i} + h/2 - \cancel{t_i}) + f_x(t_i, x(t_i))(\cancel{x(t_i)} + bh - \cancel{x(t_i)}) + \mathcal{O}(h^2)\right) \tag{VII.85}$$

$$= x(t_i) + h\left(f(t_i, x(t_i)) + \frac{h}{2} f_t(t_i, x(t_i)) + bhf_x(t_i, x(t_i)) + \mathcal{O}(h^2)\right) \tag{VII.86}$$

$$= x(t_i) + h\left(f(t_i, x(t_i))\right) + \frac{h^2}{2}\left(f_t(t_i, x(t_i)) + f(t_i, x(t_i))f_x(t_i, x(t_i))\right) + \mathcal{O}(h^3) \tag{VII.87}$$

$$\tag{VII.88}$$

The last step above used the previous results:

$$bh^2 f_x(t_i, x(t_i)) = \frac{1}{2} h^2 f_t(t_i, x(t_i)) f_x(t_i, x(t_i)) \tag{VII.89}$$

Finally the method gives

$$x(t_{i+1}) = x(t_i) + hf(t_i + h/2, y_i) \tag{VII.90}$$

$$\text{where } y_i = x(t_i) + \frac{1}{2} h f_i(t_i, x(t_i)) \tag{VII.91}$$

Using this new method the same example from before can be repeated:

$$x' = -x + t + 1; \qquad x(0) = 1; \qquad t \in [0,5] \qquad h = 1 \tag{VII.92}$$

$$x(2) = x(1) + hf(3/2, y_1) \tag{VII.93}$$

$$y_1 = x(1) + \frac{1}{2} f(1, x(1)) \tag{VII.94}$$

$$x(1) = x(0) + f(1/2, y_o) \tag{VII.95}$$

$$\tag{VII.96}$$

Now the answer can be found:

$$y_0 = 1 + 0 = 1 \tag{VII.97}$$

$$x(1) = 3/2 \tag{VII.98}$$

$$y_1 = 7/4 \tag{VII.99}$$

$$x(2) = 9/4 \approx 2.25 \tag{VII.100}$$

Notice that this result turned out to be identical to the result found using the modified Euler method. This is representative of the fact the two methods are of the same order ($\mathcal{O}(h^3)$).

VIII. RUNGE-KUTTA METHODS

Suppose $x' = f(t, x(t)$ is defined on $t \in [a,b]$ with $h = \frac{b-a}{N}$. Then

$$x(t_{i+1}) = x(t_i) + (ak_1 + bk_2) \tag{VIII.1}$$

Where

$$k_1 = hf(t_i, x(t_i)); \qquad k_2 = hf(t_i + ch, x(t_i) + dk_1) \tag{VIII.2}$$

Note that in the particular case that $a = b = 1/2$ and $c = d = 1$ then we have the modified Euler method. Alternatively if $a = 0, b = 1, c = d = 1/2$ then we have the midpoint method.
Next consider a different method given by

$$x(t_{i+1}) = x(t_i) + (a_1k_1 + a_2k_2 + ...) \tag{VIII.3}$$

where

$$k_1 = hf(t_i, x(t_i)) \tag{VIII.4}$$
$$k_2 = hf(t_i + b_2h, x(t_i) + c_{21}k_1) \tag{VIII.5}$$
$$k_3 = hf(t_i + b_3h, x(t_i) + c_{31}k_1 + c_{32}k_2) \tag{VIII.6}$$
$$k_4 = hf(t_i + b_4h, x(t_i) + c_{41}k_1 + c_{42}k_2 + c_{43}k_3) \tag{VIII.7}$$
$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots \tag{VIII.8}$$

The more k's you use the lower the error of your method. If k_n is the maximum k used then the method is of order $\mathcal{O}(h^{n+1})$. In practice we need to find what the constants $a_1, a_2, ..., b_1, b_2, ..., c_{21}, ...$ are. This can be done by Taylor expansion as was done earlier. For the particular RK (Runge-Kutta) 4 method the result is:

$$x(t_{i+1}) = x(t_i) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{VIII.9}$$

where

$$k_1 = hf(t_i, x(t_i)) \tag{VIII.10}$$
$$k_2 = hf(t_i + \frac{1}{2}h, x(t_i) + \frac{1}{2}k_1) \tag{VIII.11}$$
$$k_3 = hf(t_i + \frac{1}{2}h, x(t_i) + \frac{1}{2}k_2) \tag{VIII.12}$$
$$k_4 = hf(t_i + h, x(t_i) + k_3) \tag{VIII.13}$$

This method is of order $\mathcal{O}(h^5)$. This is a four step method (modified Euler and midpoint are two step methods).
Project: Consider the system $x' = -x + t + 1$. Approximate $x(2)$. using RK4 and compare with approximation obtained from modified Euler method or midpoint method. For this approximation assume that $t \in [0, 5], h = 1, x(0) = 1$. Make a table such as shown in table Midterm is on the 24th. This assignment is due October 10th if you

	k_1	k_2	k_3	k_4	$x(t_i)$
Step 1					
Step 2					
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

want to use a computer.

Lecture 12: October 17th, 2011
The methods we have learned thus far in the course are summarized in table VIII

Method	Error Order
Euler Methods	$\mathcal{O}(h^2)$
Modified Euler Method	$\mathcal{O}(h^3)$
Midpoint Method	$\mathcal{O}(h^3)$
RK4	$\mathcal{O}(h^5)$

IX. USING DISCRETIZING TECHNIQUES

Discretizing systems of non-linear initial value problems:

$$x' = f(x, y) \tag{IX.1}$$
$$y' = g(x, y) \tag{IX.2}$$

Where f and g are non-linear functions and defined on $[a, b]$. Define a step size: $h = \frac{b-a}{N}$. Recall Euler method:

$$x(t_{i+1}) = x(t_i) + hf(x(t_i), y(t_i)) \tag{IX.3}$$
$$y(t_{i+1}) = y(t_i) + hg(x(t_i), y(t_i)) \tag{IX.4}$$

Example: Epidemic spread:

$$S' = -\beta SI \tag{IX.5}$$
$$I' = \beta SI - \alpha I \tag{IX.6}$$

The solution of S and I can be approximated by

$$S(t_{i+1}) = S(t_i) + h(-\beta S(t_i)I(t_i)) \tag{IX.7}$$
$$I(t_{i+1}) = I(t_i) + h(\beta S(t_i)I(t_i) - \alpha I(t_i)) \tag{IX.8}$$

Example: Predator prey system:

$$x' = rx\left(1 - \frac{x}{k}\right) - axy \tag{IX.9}$$
$$y' = (xc - d)\, y \tag{IX.10}$$

With

$$x(0) = x_0 \tag{IX.11}$$
$$y(0) = y_0 \tag{IX.12}$$

Again consider Euler method:

$$x(t_{i+1}) = x(t_i) + hrx(t_i)\left(1 - \frac{x(t_i)}{k}\right) - hax(t_i)y(t_i) \tag{IX.13}$$
$$y(t_{i+1}) = y(t_i) + h\,(cx(t_i) - d)\, y(t_i) \tag{IX.14}$$

X. NON-STANDARD FINITE-DIFFERENCE METHODS

Note these models describe a real system. For the predator prey system any solution must stay in the positive region. These is called the biologically feasible region. However there usually exists a choice of parameters or step size in which solutions move out of the biologically feasible region. In this case the method fails. This is summarized below:
Problem: In approximating many-real life problems, models may fail to provide accurate or incorrect approximation; The most common failure is the lack of preserving the physical property of the system.
Question: How to avoid the method's failure?

1. Reduce step size or use variable step size methods. This method works well, however it increases the number of computations required

2. Non-Standard Finite-Difference Method

Non-standard finite-difference methods are designed such that the physical properties of the system are preserved (e.g. the positivity of the system). Again, we'll consider Euler method. Now for the system:

$$x'(t) = f(t, x) \tag{X.1}$$

$$x(t_{i+1}) = x(t_i) + hf(t_i, x(t_i)) \tag{X.2}$$

Using the definition of the derivative:

$$x'(t) = \lim_{h\to 0} \frac{x(t+h) - x(t)}{h} \tag{X.3}$$

$$\lim_{h\to 0} \frac{x(\overbrace{t_i + h}^{t_{i+1}}) - x(t_i)}{h} = f(t_i, x(t_i)) \tag{X.4}$$

Rules to develop non-standard methods:

1. **Preserving positivity:** Eliminating negative terms of the right hand side associated with approximating variable. This can be done by a method called "approximation at an advanced stage". Hence every time we have a term with a negative sign we approximate that term at an advanced stage (hence t_{i+1} instead of t_i). Recall the endemic example:

$$S' = -\beta SI; \qquad I' = \beta SI - \alpha I \tag{X.5}$$

   The approximation to S using Euler method is given by

$$S(t_{i+1}) = S(t_i) - h\beta S(t_i)I(t_i) \tag{X.6}$$

   Approximate this result with the S on the right side evaluated at $S(t_{i+1})$.

$$S(t_{i+1}) = S(t_i) - h\beta S(t_i + 1)I(t_i) \tag{X.7}$$

   This can be rearranged:

$$S(t_{i+1})\left(1 + h\beta I(t_i)\right) = S(t_i) \tag{X.8}$$

$$\boxed{S(t_{i+1}) = \frac{S(t_i)}{1 + h\beta I(t_i)}} \tag{X.9}$$

Lecture 13: October 21st, 2011

2. (a) **Split 1:** This rule is used if there are negative terms of approximating variable with higher degree than 1. In this case the term is split into the product of two terms with the first term evaluated at an advanced point with the second evaluated at the current point. As an example, consider the system (not spread of disease),

$$S' = -\beta S^2 I; \qquad I' = \beta S^2 I - \alpha I \tag{X.10}$$

   To approximate this system using Euler method:

$$\frac{S(t_{i+1}) - S(t_i)}{h} = -\beta S(t_{i+1})S(t_i)I(t_i) \tag{X.11}$$

$$S(t_{i+1}) + h\beta S(t_{i+1})S(t_i)I(t_i) = S(t_i) \tag{X.12}$$

$$S(t_{i+1})\left(1 + h\beta S(t_i)I(t_i)\right) = S(t_i) \tag{X.13}$$

$$\boxed{S(t_{i+1}) = \frac{S(t_i)}{1 + h\beta S(t_i)I(t_i)}} \tag{X.14}$$

   More generally consider the case of

$$S' = -\beta S^{1+p} I \tag{X.15}$$

   In this case split the term as $S^{1+p} = S(t_i)S^p(t_{i+1})$.

(b) **Split 2:** This rule is used for negative terms of approximating variable with lower degree than 1. Suppose we have the particular system:

$$S' = -\beta S^{1-p} I; \qquad 0 < p < 1 \tag{X.16}$$

$$I' = \beta S^{1-p} I - \alpha I \tag{X.17}$$

If we assume that S never goes to zero:

$$\frac{S(t_{i+1}) - S(t_i)}{h} = -\beta S(t_{i+1}) S^{-p}(t_i) I(t_i) \tag{X.18}$$

$$S(t_{i+1}) + h\beta S(t_{i+1}) S^{-p}(t_i) I(t_i) = S(t_i) \tag{X.19}$$

$$S(t_{i+1}) = \frac{S(t_i)}{1 + h\beta S^{-p}(t_i) I(t_i)} \tag{X.20}$$

3. **Advancing:** If you have a problem with multiple variables that you interested in solving for (as above we have S and I). Then once we find the solution for the first variable at a point you can use that value (as opposed to the old value) as the input for the second variable. Suppose we have the system

$$S' = -\beta S I \tag{X.21}$$

$$I' = \beta S I - \alpha I \tag{X.22}$$

The solution for the S variable is

$$S(t_{i+1}) = \frac{S(t_i)}{1 + \beta h I(t_i)} \tag{X.23}$$

The solution for the I variable is normally given by

$$\frac{I(t_{i+1}) - I(t_i)}{h} = \beta S(t_i) I(t_i) - \alpha I(t_{i+1}) \tag{X.24}$$

(Where the last terms is evaluated at $i+1$ due to rule 1). A better approximation is found by

$$\frac{I(t_{i+1}) - I(t_i)}{h} = \beta S(t_{i+1}) I(t_i) - \alpha I(t_{i+1}) \tag{X.25}$$

Solving this result of $I(t_{i+1})$ gives:

$$I(t_{i+1}) = \frac{I(t_i)\left(1 + h\beta S(t_{i+1})\right)}{1 + \alpha h} \tag{X.26}$$

Example: Predator-prey system:

$$x' = rx\left(1 - \frac{x}{k}\right) - axy \tag{X.27}$$

$$y' = (cx - d)\, y \tag{X.28}$$

The solution to this system is strictly positive so non-standard methods can be used to cut down computation time. Simplifying we see that the this problem contains all the rules that we discussed.

$$x' = rx - \frac{rx^2}{k} - axy \tag{X.29}$$

The Euler approximation for this system is

$$\frac{x(t_{i+1}) - x(t_i)}{h} = rx(t_i) - \frac{r}{k} x(t_i) x(t_{i+1}) - ax(t_{i+1}) y(t_i) \tag{X.30}$$

$$x(t_{i+1})\left(\frac{rh}{k} x(t_i) - ahy(t_i)\right) = x(t_i)\left(1 + rh\right) \tag{X.31}$$

$$x(t_{i+1}) = \frac{x(t_i)\left(1 + rh\right)}{\frac{rh}{k} x(t_i) - ahy(t_i)} \tag{X.32}$$

The solution for y can be approximate as:

$$\frac{y(t_{i+1}) - y(t_i)}{h} = cx(t_{i+1}) y(t_i) - dy(t_{i+1}) \tag{X.33}$$

$$y(t_{i+1}) = \frac{\left(1 + chx(t_{i+1})\right) y(t_i)}{1 + dh} \tag{X.34}$$

XI. MATLAB

Matlab is a software package that can be used for approximation or graphic simulations. Matlab commands:

Symbol	Action
+	Addition
-	subtraction
*	multiplication
/	division

Lecture 14: October 23, 2011
Test will concern all material from beginning of course up to RK4 method. For test memorize formulas for all numerical methods. Calculators are permitted.
A $m \times n$ matrix can be encoded in Matlab by

$$A = [row_1; row_2; ...; row_m] \tag{XI.1}$$

Here semi colon is used to separate rows while commas (spaces are alternatively permitted) are used to separate columns. Another way to encode a vector is

$$x = a : h : b \tag{XI.2}$$

Will create a vector that takes the form

$$[a, a + h, a + 2h, ..., b - h, b] \tag{XI.3}$$

Multiplication of two square matrices is done by

$$A_{n\times n} * B_{n\times n} = C_{n\times n} \tag{XI.4}$$

Another way to multiply the two matrices is

$$A. * B = C \tag{XI.5}$$

But in this case Matlab multiplies each of the corresponding elements (it doesn't do proper matrix multiplication). i.e. if $A = [a_{ij}]\,, B = [b_{ij}]$, then $C = [a_{ij}b_{ij}]$. The dot notation is extended to other operations such as exponentiating.
Functions and creating vectors/matrices:

$$ones(m, n) \to \begin{pmatrix} 1 & ... & 1 \\ \vdots & \ddots & \vdots \\ 1 & ... & 1 \end{pmatrix}_{m\times n} \tag{XI.6}$$

Similarly there is a zero matrix function:

$$zero(m, n) \to \begin{pmatrix} 0 & ... & 0 \\ \vdots & \ddots & \vdots \\ 0 & ... & 0 \end{pmatrix}_{m\times n} \tag{XI.7}$$

We will use vectors to save the output from the calculations at different steps. Another useful function is the plot function. The input for Matlab is numeric so suppose we have a list y where x is the input to the list and y is some operation on x then

$$plot(x, y) \tag{XI.8}$$

Plots y as a function of x. As an example of this is $x = [0 : 0.01 : 1]\,,\ y = x.\hat{}2$. To plot two functions as a function of a single input variable we use (take t to be the input variable and S and I to be the functions)"

$$plot(t, [S, I]) \tag{XI.9}$$

Plot can have different options. For example:

$$plot(x, y,' -') \quad (XI.10)$$

Gives a single thin line and

$$plot(x, y,' --') \quad (XI.11)$$

Gives a dashed line plot.

$$plot(x, y,' .g') \quad (XI.12)$$

Then it will plot the curve with dots in green.
In the lab we will consider a predator prey system:

$$x' = rx\left(1 - \frac{x}{k}\right) - a \overbrace{f(x)}^{\text{Predator functional response}} y \quad (XI.13)$$

$$y' = (cx - d)\, y \quad (XI.14)$$

Consider predator prey system with Holling-type II response:

$f(x)$	Name
x	Holling-type I
$\frac{x}{1+x}$	Holling -type II
$\frac{x^2}{1+x^2}$	Holling-type III
$\frac{x^b}{1+x^b}$; $p > 2$	Holling-type ?

$$x' = rx\left(1 - \frac{x}{k}\right) - \frac{ax}{1+x} y \quad (XI.15)$$

$$y' \left(\frac{cx}{1+x} - d\right) y \quad (XI.16)$$

First one needs to identify the parameter set (this is how you would type it into Matlab):

$$\%\, parameters \quad (XI.17)$$

$$r = r_o; \quad (XI.18)$$

$$a = a_o; \quad (XI.19)$$

$$k = k_o; \quad (XI.20)$$

$$c = c_o; \quad (XI.21)$$

$$d = d_o; \quad (XI.22)$$

```
 \% Method's parameters
 t _n = T; \qquad \% End point of time;
 t _o = t _o ^*; \qquad \% Beginning of time;
 h = h _o; \qquad \% Step size;

t = \left(t _o:t:t _n \right);
 nt = length \left(t \right); \% \text{(number of points)}
 x = zeros (nt,1)
 y = zeros(nt, 1)

 for j = 1:nt
 x (j) = x (j -1) +h \left(r x (j -1) \left(1 - \frac{x (j -1) }{k} - \frac{a x (j -1) }{1 +x (j -1) } y (j -1)
 y (j) = y (j -1) + h \left( \frac{c x(j -1) }{1 + x (j -1) } - d \right) y (j -1)
```

Next lecture on Wednesday we will go to Gauss lab.

Lecture 15 - Oct 31, 2011

XII. PHASE PORTRAIT

Recall that in Matlab we used plot(t,x) and plot(t,y) to see how prey and predator behave with time. Alternatively we can use plot(x,y) this says how y behaves with respect to x and is called the phase portrait of the system.
Recall the eigenvalues of the predator prey system were:

$$\lambda^2 - p\lambda + q = 0 \tag{XII.1}$$

Where p and q were defined as

$$p \equiv \mathrm{tr}\,(J)\,; \qquad q = \det\,(J) \tag{XII.2}$$

The graphical result of this is shown in figure 2. However we only know the stability of all points at which $q \neq 0$. What happens at $q = 0$? If real parts of eigenvalues are zero then we don't know whether the point is stable of unstable. Another example of this problem is

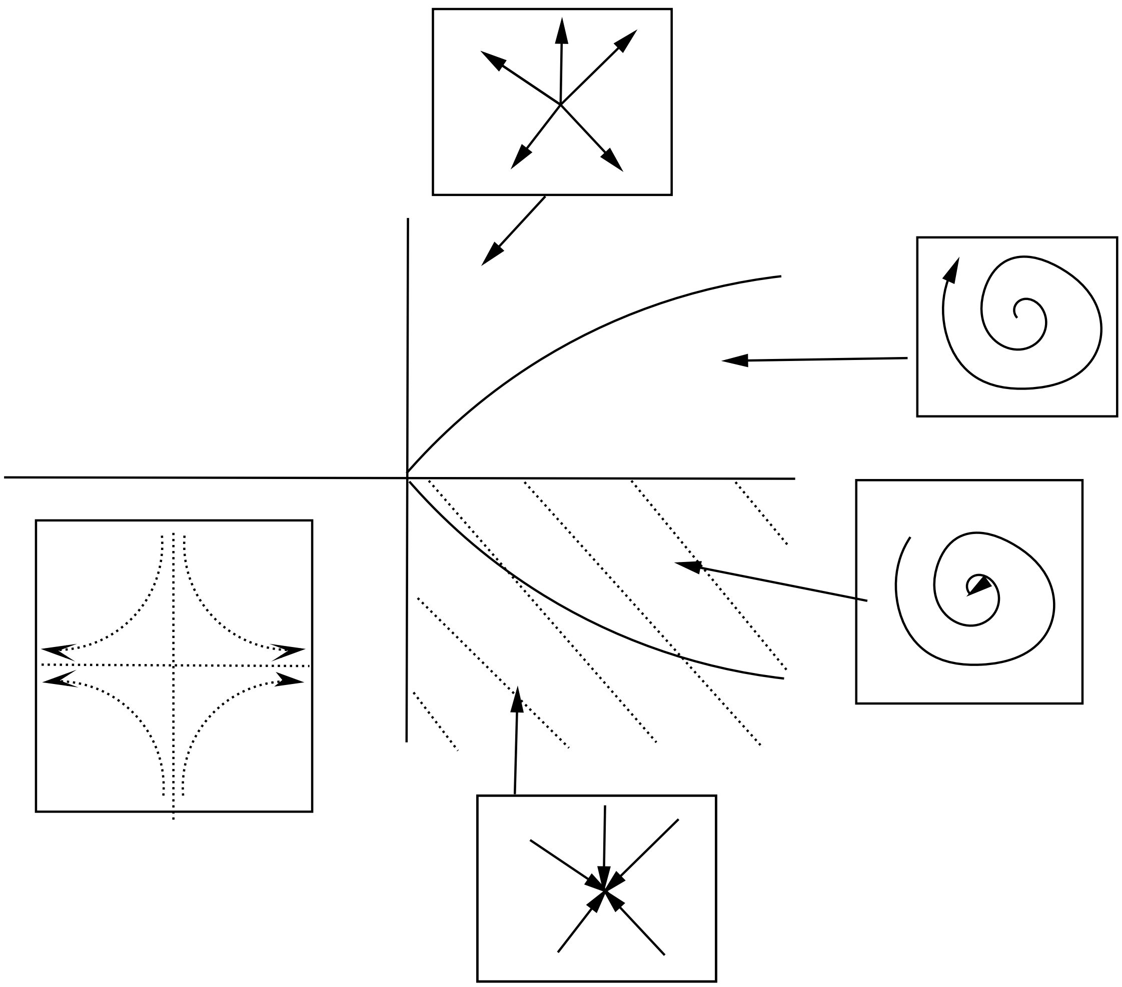

FIG. 2. Stability of predator prey system

$$x' = 2 - x - y; \qquad y' = x^2 - y \tag{XII.3}$$

Solving for the critical points gives

$$E_1 = (1, 1); \qquad E_2 = (-2, 4) \tag{XII.4}$$

$$J = \begin{pmatrix} -1 & -1 \\ 2x & -1 \end{pmatrix} \tag{XII.5}$$

$$J(E_1) = \begin{pmatrix} -1 & -1 \\ 2 & -1 \end{pmatrix} \tag{XII.6}$$

We can use the results for the predator prey system if we define p and q as . Alternatively we can consider the characteristic equation.

$$\det\left(\lambda I - J\left(E_1\right)\right) = \left(\lambda + 1\right)^2 + 2 = 0 \tag{XII.7}$$

$$\left(\lambda^2 + 1\right)^2 = -2 \tag{XII.8}$$

$$\lambda = -1 - \pm\sqrt{2}i \tag{XII.9}$$

Since the eigenvalues are strictly negative the solution E_1 is asymptotically stable.

$$J\left(E_2\right) = \begin{pmatrix} -1 & -1 \\ -4 & -1 \end{pmatrix} \tag{XII.10}$$

The eigenvalues in this case are

$$\lambda = -1 \pm 2 \tag{XII.11}$$

Since E_1 has both real and imaginary eigenvalues it is a saddle node. The phase portrait is shown in figure 3 Matlab

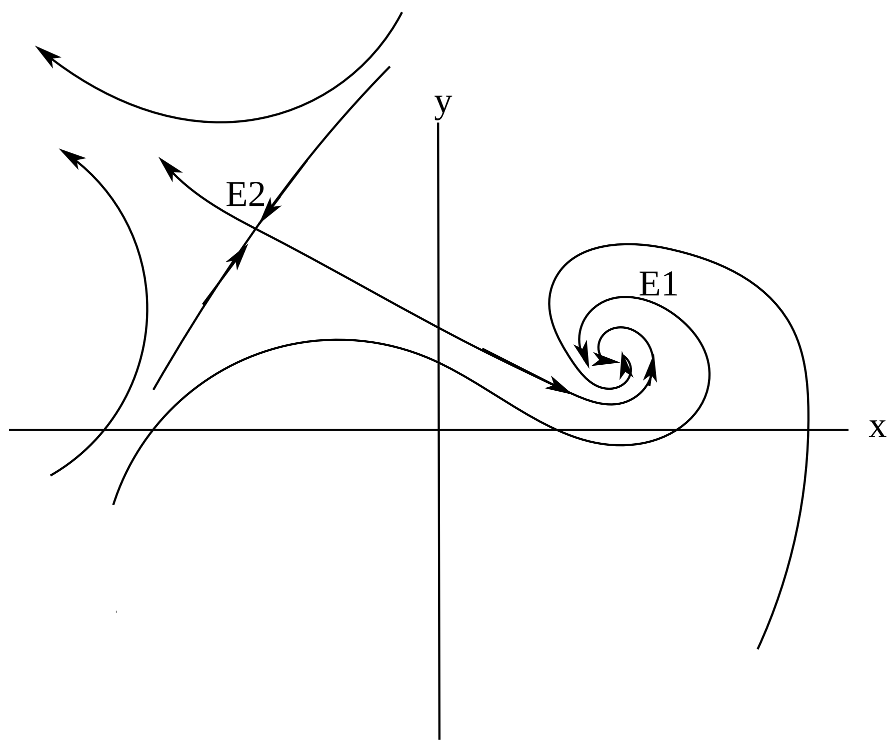

FIG. 3. Phase portrait of system

note: in order to plot two different plots on top of one another use the hold function in between outputs. E.g.

```
plot(x,y) %some initial condition
hold
plot(x,y) %some other initial condition
```

Def 5. If a critical point is a saddle point in a two dimensional system. Then the point has a saddle manifold of dimension 1, and an unstable manifold of dimension 1.

Def 6. If a critical point is a asymptotically stable in a two dimensional system, then the system has two manifolds, each one dimension and both stable.

Def 7. If a critical point is an unstable in a two dimensional system, then the system has two manifolds, both unstable.

Lecture 16 - November 7th, 2011

XIII. ROUTH-HURWITZ CRITERION

For a two dimensional system the Jacobian is straightforward to obtain and hence the critical points are easy to analyze. However if we have higher dimensional systems then solving the characteristic equation becomes a difficult question. In general the characteristic equation is of the form

$$a_n\lambda^n + a_{n-1}\lambda^{n-1} + ... + a_1\lambda + a_o = 0 \tag{XIII.1}$$

Routh-Hurwitz criterion is a method of determining the stability of a linear system by examining the eigenvalues of the Jacobian. Let's assume that

$$c(\lambda) = a_n\lambda^n + a_{n-q}\lambda^{n-1} + ... + a_1\lambda + a_o \tag{XIII.2}$$

is the characteristic equation of the system. Then

1. Two necessary but **NOT** sufficient conditions for all roots to have negative real parts (hence asymptotically stable point)
   (a) All the coefficients of $c(\lambda)$ (i.e. $a_o, a_1, ..., a_{n-1}, a_n$) have the same sign.
   (b) All the coefficients of must be **non zero**.
2. If these conditions are satisfied then we compute Routh-Herwitz array as

$$\left[\begin{array}{c|ccccc} \lambda^n & a_n & a_{n-2} & a_{n-4} & a_{a-6} & \cdots \\ \lambda^{n-1} & a_{n-1} & a_{n-3} & a_{n-5} & a_{n-7} & \cdots \\ \lambda^{n-2} & b_1 & b_2 & b_3 & b_4 & \cdots \\ \lambda^{n-3} & c_1 & c_2 & c_3 & c_4 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lambda^1 & \vdots & \vdots & \vdots & \vdots & \vdots \\ \lambda^0 & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}\right]$$

Where

$$b_1 = \frac{-1}{a_{n-1}} \det \begin{pmatrix} a_n & a_{n-2} \\ a_{n-1} & a_{n-3} \end{pmatrix} \tag{XIII.3}$$

$$b_2 = \frac{-1}{a_{n-1}} \det \begin{pmatrix} a_n & a_{n-4} \\ a_{n-1} & a_{n-5} \end{pmatrix} \tag{XIII.4}$$

$$b_3 = \frac{-1}{a_{n-1}} \det \begin{pmatrix} a_n & a_{n-6} \\ a_{n-1} & a_{n-7} \end{pmatrix} \tag{XIII.5}$$

etc. The c's are

$$c_1 = \frac{-1}{b_1} \det \begin{pmatrix} a_{n-1} & a_{n-3} \\ b_1 & b_2 \end{pmatrix} \tag{XIII.6}$$

$$c_2 = \frac{-1}{b_1} \det \begin{pmatrix} a_{n-1} & a_{n-5} \\ b_1 & b_3 \end{pmatrix} \tag{XIII.7}$$

$$c_3 = \frac{-1}{b_1} \det \begin{pmatrix} a_{n-1} & a_{n-7} \\ b_1 & b_4 \end{pmatrix} \tag{XIII.8}$$

3. The necessary condition that all roots have negative real parts is that all the elements of the first column of the Routh-Hurwitz (RH) array have the same sign. The number of changes in the sign equals the number of roots with positive real parts

Now consider the special case of a $3D$ system.

$$c(\lambda) = a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 = 0 \tag{XIII.9}$$

Since this equation is third order $a_3 \neq 0$. Hence we can divide by it

$$\frac{c(\lambda)}{a_3} = \lambda^3 + \frac{a_2}{a_3}\lambda^2 + \frac{a_1}{a_3}\lambda + \frac{a_o}{a_3} = 0 \tag{XIII.10}$$

Redefining our variables we have

$$c(\lambda) = \lambda^3 + a_1\lambda^2 + a_2\lambda + a_3 = 0 \tag{XIII.11}$$

$$\left[\begin{array}{c|ccccc} \lambda^3 & 1 & a_2 & 0 & 0 & \ldots \\ \lambda^2 & a_1 & 0 & 0 & 0 & \ldots \\ \lambda^1 & b_1 & 0 & 0 & 0 & \ldots \end{array}\right]$$

The coefficients are calculated below:

$$b_1 = -\frac{1}{a_1} \det \begin{pmatrix} 1 & a_2 \\ a_1 & a_3 \end{pmatrix} = -\frac{1}{a_1}\left(a_3 - a_1a_2\right) \tag{XIII.12}$$

$$b_2 = -\frac{1}{a_1} \det \begin{pmatrix} 1 & 0 \\ a_1 & 0 \end{pmatrix} = 0 \tag{XIII.13}$$

$$c_1 = -\frac{1}{b_1} \det \begin{pmatrix} a_1 & a_3 \\ b_1 & 0 \end{pmatrix} \tag{XIII.14}$$

$$= \frac{1}{b_1}\left(-a_3b_1\right) \tag{XIII.15}$$

$$= a_3 \tag{XIII.16}$$

Now we are concerned with the column

$$\begin{bmatrix} 1 \\ a_1 \\ -\frac{1}{a_1}\left(a_3 - a_1a_2\right) \\ a_3 \end{bmatrix}$$

Since 1 is positive we require the entire row be positive in order to have all negative eigenvalues. This requires

$$\boxed{a_1 > 0; \qquad a_3 > 0; \qquad a_3 < a_1 a_2} \tag{XIII.17}$$

For an example lets consider epidemic spread (Measles).

$$S' = B - \beta SI - \mu S \tag{XIII.18}$$
$$E' = \beta SI - \sigma E - \mu E \tag{XIII.19}$$
$$I' = \sigma E - \gamma I - \mu I \tag{XIII.20}$$

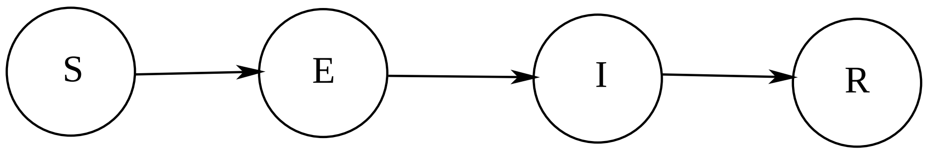

FIG. 4. Measles Transmission

E stands for exposed class. This class has the disease but doesn't show symptoms and can't transmit the disease yet. A person that catches the disease must pass through the exposed class to go to the infected class.
The disease free equilibrium is

$$\left(\frac{B}{\mu}, 0, 0\right) \tag{XIII.21}$$

The infection state is

$$(S^*, E^*, I^*) \tag{XIII.22}$$

Solving the system for the critical points gives

$$S^* = \frac{(\mu + \sigma)(\mu + \gamma)}{\beta} \tag{XIII.23}$$

Subbing this result back in gives

$$B - (\mu + \rho)(\mu + \gamma) I - \frac{\mu}{\beta}(\mu + \sigma)(\mu + \gamma) = 0 \tag{XIII.24}$$

$$I = \frac{B - \frac{\mu(\mu+\sigma)(\mu+\gamma)}{\beta}}{(\mu + \sigma)(\mu + \gamma)} \tag{XIII.25}$$

$$E^* = \frac{(\mu + \gamma) I^*}{\sigma} \tag{XIII.26}$$

The Jacobian of the system at this point is

$$\begin{pmatrix} -\beta I^* - \mu & 0 & -\beta S^* \\ \beta I^* & -(\mu + \sigma) & \beta S^* \\ 0 & \sigma & -(\mu + \sigma) \end{pmatrix} \tag{XIII.27}$$

The characteristic equation can now be found and we can use the RH criterion.

Lecture 17 - November 14th, 2011

XIV. MATLAB

A. Functions

So far in Matlab we have used scripts. However there is another way to program called function (also called subroutine in other programming languages). Function is designed to be used as a small tool during the course of a more elaborate project. There are three major elements in writing code:

$$Input \to tasks \to Output$$

In function they may be mixed up. Any function must start with the following line:

```
function[y_1,y_2,y_3, ..., y_m ] = my_function_name(x_1,x_2,..., x_n)
```

This function must be saved as

```
my_function_name.m
```

In function we use % to describe your program. After the first line, include " $\langle statements \rangle$ " that will perform task requested by the function.
Example: Create a function that calculates the mean and standard deviation of your term marks.

```
function[m,s]=marks(x)
% x denotes the list of marks
% m and s denote the mean and standard deviation.
n=length(x);
m=sum(x)/n;  %sum is a built in function in Matlab
s=sqrt(sum((x-m).^2)/n) %sqrt is a built-in square root function in Matlab. The dot makes things scalar.
%The standard deviation takes the vector x and subtracts m from each value of the vector.
%The .^ takes each value in the vector and squares it.
```

Your function name is "marks.m". Now you go to your command line

```
>>x=[80 70 85 70 75 90];
>>[m,n] = marks(x)
  m=76.4
  s=8.3299
```

B. Branch Statement

Branch statements are used when tasks are subject to some conditions.

```
if <conditions>
   <statement>
end
```

For more then one condition we use:

```
if <conditions>
   <statemnt>
else
   <statement>
end
```

For statements of more than two conditions:

```
if <conditions1>
   <statements1>
elseif <conditions2>
   <statements2>
elseif <conditions3>
   <statements3>
else <statement>
end
```

equal	==
not equal	~ =
and	&
or	—

Mathematical expressions in Matlab are shown in table XIV B Based on the previous example of function. **Script**

```
x=input('Enter your term marks as a vector:'); %The program will ask you for the marks when you run the script
disp(m)
if m>=90
disp('Your final grade is A+. Well done!')
elseif (m>=80)&(m<90)
disp('Your final grade is A')
else
disp('You did horribly!')
end
```

Command line Suppose you named your script lettergrade.

```
>>lettergrade
  Enter your term marks as a vector:[50 61 70 56 80 75]
    65.3333
  You did horribly!
```

C. Loops

Loops are used for performing tasks for a number of counts

```
for <counts>
    <statements>
end
```

Note that counts must be a positive integer.
Example: Create a function that to calculate the sum of squares of the first N integers. i.e.

$$1^2 + 2^2 + 3^2 + ...$$

```
function s=sum_squares(N)
%...
s=0;
for i=1:N
   s=s+i^2;
end
```

Command Line:

```
>>sum_squares(1000)
     ans=333833500
```

RK4 method for multiple variables needs to be modified. For two variables it is given by b

$$k_{1x} = hf(t_i, x(t_i), y(t_i)) \tag{XIV.1}$$

$$k_{2y} = hg(t_i + \frac{h}{2}, x(t_i), y(t_i)) \tag{XIV.2}$$

$$k_{2x} = hf(t_i + \frac{h}{2}, x(t_i) + \frac{1}{2}k_1x, y(t_i) + \frac{1}{2}k_1y) \tag{XIV.3}$$

$$k_{2y} = hg(t_i, \frac{h}{2}, x(t_i) + \frac{1}{2}k_1x, y(t_i) + \frac{1}{2}k_1y)\vdots \tag{XIV.4}$$

MATLAB
Suppose we have the system

$$-6x = 2y - 2z + 15 \tag{XIV.5}$$
$$4y - 3z = 3x + 13 \tag{XIV.6}$$
$$2x + 4y - 7z = -9 \tag{XIV.7}$$

This can be rearranged into

$$\begin{pmatrix} -6 & -2 & 2 \\ -3 & 4 & -3 \\ 2 & 4 & -7 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 15 \\ 13 \\ -9 \end{pmatrix} \tag{XIV.8}$$

In Matlab we would write

```
>>A=[-6 -2 2; -3 4 -3; 2 4 -7];
>>B =[15 14 -9]'
```

The prime is necessary to transpose the vector into a column vector. Recall from linear algebra. If A is invertible then A^{-1} exists and we can solve the system by

$$AX = B \tag{XIV.9}$$
$$A^{-1}AX = A^{-1}B \tag{XIV.10}$$
$$X = A^{-1}B \tag{XIV.11}$$

If the determinant is not zero than A^{-1} exists. In command line we can write

```
>>\det(A)
```

We can then write

```
x = A\B
```

Alternatively we can solve the system by writing

```
>>inv(A)
```

The output is some matrix.
Suppose we want to use interpolation. If we know $X_1 = (x_1, y_1)$ and $X_2 = (x_2, y_2)$ then some point X_3 at a given x_3 is in between these two and is estimated by

$$y_3 = y_1 + \frac{(y_2 - y_1)(x - x_1)}{x_2 - x_1} \tag{XIV.12}$$

If we want to code this into Matlab we use

```
function y _3 = interpol(X _1,X _2, x _3)
y _3 = ...
```

As an example consider the following case

```
>>X _1 = [60, 15.56];
>>X _2 = [90, 32.22];
>>x _3 = 73;
interpol(X _1, X _2, x _3)
```

Useful commands in Matlab:

```
abs(x); %Absolute value
acos(x); %Arccos(x)
asin(x); %Arcsin(x)
atan(x); %Arctan(x)
exp(x); % exponential of x
log(x); %Natural logarithm of x
log10(x); %Logarithm base 10
fix(x); %round towards zero
floor(x); %rounds toward negative infinity
```

If you have some complex number $z = a + ib$. Matlab can handle complex numbers. If we write

```
>>z =a + ib
>>real (z)
         a
>>imag(z)
         b
```

Given z we can also write

```
conj(z); %gives the complex conjugate of z
```

Lecture - November 30th, 2011

XV. VARIABLE STEPSIZE

Methods that we have discussed so far (Euler, Modified Euler, ...) had a fixed step size. However there are methods which the step size changes. An example of this is ode45.

Consider if we have the system

$$X' = \begin{pmatrix} x_1'(t) \\ x_2'(t) \\ \vdots \\ x_n'(t) \end{pmatrix} = \begin{pmatrix} f_1(t, x_1, ...x_n) \\ f_2(t, x_1, ...x_n) \\ \vdots \\ f_n(t, x_1, ...x_n) \end{pmatrix} \tag{XV.1}$$

Matlab has built in functions to solve systems. You need to provide a function. Define

$$\begin{aligned} x1 &= x(1) \\ x2 &= x(2) \\ &\vdots \\ xn &= x(n) \end{aligned}$$

```
function dX = mysystem(t,X)
dx (1) = x1;
dx (2) = x2;
...
dx (n) = xn;

dX =[dx(1);dx (2); ...;dx (n)]
```

In the command line:

```
>>[t,x] = ode45(@mysystem,[t _0, t _1],[x _1 (t _0);x _2 (t _0);...;x _n (t _n)])
```

This will produce a vector

```
x1 x2 ... xn \\
x1(t_0) x2(t_0) ... \\
... ... ... \\
x1(t_1) x2(t_1) ... x3(t_1)
```

Then we can plot the point x_i this as such

```
>>plot(t,x( : ,i))
```

The colon recognizes that there are a lot of points in the interval.
Given the system

$$X' = AX \tag{XV.2}$$

Where X and A are vectors and matrices. The same procedure can be used:

```
function dx =mysystem(t,x,A)
 dx=A*x;
```

In the command line:

```
>>A =[-3 1 0; -2 -1 1; 3 3 -6];
>>x0=[-1;2;0.5];
>>[t,x] = ode45(@mysystem,[t0,t1],x0,[],A)
```

The empty square brackets are required to give the possibility of adding options.