GTE, LTE.

The source is Boyce, DiPrima chpt 8.

CSE, 255, Marmara University

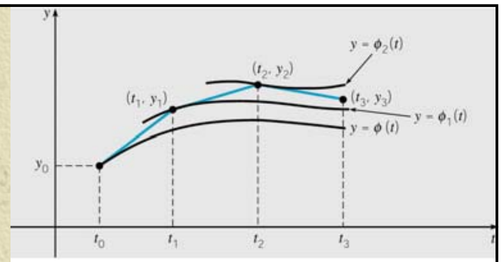
Dr. M. Sakalli

- HW
  - RLC, series and parallel, suppose homogeneous solutions in the form of r=s=σ+- j w. exp(s)..
  - Matlab codes in Coupled.pdf.
- Study chapter 7 boyce or Paul Dawkins.

Dr. M.Sakalli

Euler's Method

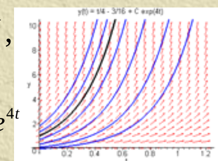
- 1^{st} and 2^{nd} order DE
  with integration
  by power series methds, in a week time.
- However, for **many problems** in eng. and science, especially nonlinear ones, these methods do not apply or become complicated.
- Apprach is Numerical approximation, where he concern is the how close to approach to the analytical (actual) solution if not known.
- An IVP prbl.
  if f and f_y are continuous then has a **unique solution** y in some interval about t_0. And Euler's method. Example.
  $y = \phi(t)$.
- $f_n = f(t_n, y_n)$. For a uniform step size $h = t_n - t_{n-1}$, Euler's method becomes y_{n-1}
- Forward Difference Quotient: $y = \phi(t)$



$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0,, y(0) = 1$$

$$y_{n+1} = y_n + f_n \cdot (t_{n+1} - t_n), \quad n = 0, 1, 2, \ldots$$

$$y(t) = \frac{1}{4}t - \frac{3}{16} + Ce^{4t},$$



$$y = \phi(t) = \frac{1}{4}t - \frac{3}{16} + \frac{19}{16}e^{4t}$$

$$\frac{\phi(t_{n+1}) - \phi(t_n)}{t_{n+1} - t_n} \cong f(t_n, \phi(t_n)) = \phi'$$

Euler's Method: Programming Outline Comparing Euler's Method
h = 0.05, 0.025, 0.01, 0.001, on the interval $0 \le t \le 2$.

- A computer program for Euler's method with a uniform step size will have the following structure.
  - Step 1. have function $f(t,y)$, and step size h
  - Step 2. Initialize values t0 and y0
  - Step 3. Limit the # of steps n
  - Step 5. for j =1: n,
  - Step 6. $k1 = f(t,y)$
    $y = y + h*k1$
    $t = t + h$
  - Step 7. Output t and y

$$\text{Relative Error} = \left| \frac{y_{exact} - y_{approx}}{y_{exact}} \right| \times 100$$

t	h = 0.05	h = 0.025	h = 0.01	h = 0.001	Exact	Rel Error h = 0.05	Rel Error h = 0.025	Rel Error h = 0.01	Rel Error h = 0.001
0.00	1.0000	1.0000	1.0000	1.0000	1.0000	0.00	0.00	0.00	0.00
0.10	1.5475	1.5761	1.5953	1.6076	1.6090	3.82	2.04	0.85	0.09
0.20	2.3249	2.4080	2.4646	2.5011	2.5053	7.20	3.88	1.63	0.17
0.30	3.4334	3.6144	3.7390	3.8207	3.8301	10.36	5.63	2.38	0.25
0.40	5.0185	5.3690	5.6137	5.7755	5.7942	13.39	7.34	3.12	0.32
0.50	7.2902	7.9264	8.3767	8.6771	8.7120	16.32	9.02	3.85	0.40
1.00	45.5884	53.8079	60.0371	64.3826	64.8978	29.75	17.09	7.49	0.79
1.50	282.0719	361.7595	426.4082	473.5598	479.2592	41.14	24.52	11.03	1.19
2.00	1745.6662	2432.7878	3029.3279	3484.1608	3540.2001	50.69	31.28	14.43	1.58

Backward Euler Formula

- The backward Euler formula is derived as follows. Let $y = \phi(t)$ be the solution of $y' = f(t,y)$. At $t = t_n$, we have

$$\phi' = f\left(t_n, \phi(t_n)\right)$$

- Using a backward difference quotient for ϕ', it follows that

$$\frac{\phi(t_n) - \phi(t_{n-1})}{t_n - t_{n-1}} \cong f\left(t_n, \phi(t_n)\right)$$

- Replacing $\phi(t_n)$ and $\phi(t_{n-1})$ by their approximations y_n and y_{n-1}, and solving for y_n, we obtain the backward Euler formula

$$y_n = y_{n-1} + f(t_n, y_n) \cdot h \quad \Leftrightarrow \quad y_{n+1} = y_n + f(t_{n+1}, y_{n+1}) \cdot h$$

- Note that this equation implicitly defines y_{n+1}, and must be solved in order to determine the value of y_{n+1}.

Dr. M.Sakalli

Investigating more accurate methods

Forward Difference Quotient: $y = \phi(t)$

$$\frac{\phi(t_{n+1}) - \phi(t_n)}{t_{n+1} - t_n} \cong f\left(t_n, \phi(t_n)\right) = \phi'$$

An integral equation since $y = \phi(t)$ is a solution of $y' = f(t, y)$, $y(t_0) = y_0$

$$\int_{t_n}^{t_{n+1}} \phi'(t)dt = \int_{t_n}^{t_{n+1}} f\left(t, \phi(t)\right)dt$$

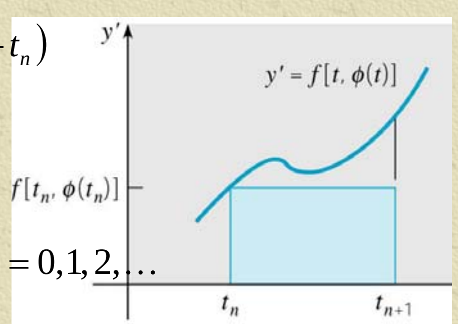
Approximating the above integral

$$\phi(t_{n+1}) \cong \phi(t_n) + f\left(t_n, \phi(t_n)\right)\left(t_{n+1} - t_n\right)$$

$$\phi(t_{n+1}) = \phi(t_n) + \int_{t_n}^{t_{n+1}} f\left(t, \phi(t)\right)dt$$

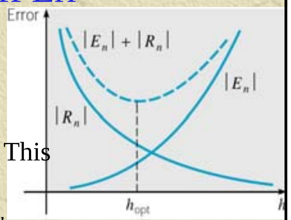
Replacing $\phi(t_{n+1})$ and $\phi(t_n)$ by their approximations y_{n+1} and y_n,:

$$y_{n+1} = y_n + f(t_n, y_n) \cdot \left(t_{n+1} - t_n\right), \quad n = 0,1,2,\ldots$$



Dr. M.Sakalli

Global and Local Truncation Error Round-off Err



- The **global truncation error** GTE is defined as

$$E_n = \phi(t_n) - y_n$$

Difference between actual and numerical solutions at t_n. This error arises from two causes:

  - Remember approximation of integration to determine y_{n+1}.
  - And the approximated $y_n =\sim \phi(t_n)$.

- **Local truncation error** e_n. LTE

Assume an accurate $y_n = \phi(t_n)$ at step n, the error at step $n+1$ is due to the approximation formula.

- Round-off error occurs

$$R_n = y_n - Y_n.$$

- $T_n = \phi(t_n) - Y_n$.
- From the triangle inequality,

$|a + b| \leq |a| + |b|$, it follows that

$$\begin{aligned}|\phi(t_n) - Y_n| &= |\phi(t_n) - y_n + y_n - Y_n| \\ &\leq |\phi(t_n) - y_n| + |y_n - Y_n| \\ &\leq |E_n| + |R_n|\end{aligned}$$

Dr. M.Sakalli

Taylor Series Analysis

- Assuming the solution $y = \phi(t)$ has a Taylor series about $t = t_n$.

$$\phi(t) = \phi(t_n) + \phi'(t_n)(t - t_n) + \frac{1}{2!}\phi''(t_n)(t - t_n)^2 + \cdots$$

- Since $h = t_{n+1} - t_n$ and $\phi' = f(t, \phi)$, it follows that

$$\phi(t_{n+1}) = \phi(t_n) + f(t_n, \phi(t_n))h + \frac{1}{2!}\phi''(t_n)h^2 + \cdots$$

- Take the 1st order apprx, and substitute y_{n+1} and y_n, for $\phi(t_{n+1})$ and $\phi(t_n)$, respectively. Again Euler's formula:

$$y_{n+1} = y_n + f(t_n, y_n) \cdot (t_{n+1} - t_n), \quad n = 0, 1, 2, \ldots$$

- Estimate the magnitude of error in this formula.

Dr. M.Sakalli

Local Truncation Error for Euler Method

- Assumption is that $y = \phi(t)$ is a solution to $\phi' = f(t, \phi)$, $y(t_0) = y_0$ and that f, f_t and f_y are continuous. Then ϕ'' is continuous, where

$$\phi''(t) = f_t(t,\phi(t)) + f_y(t,\phi(t))\,f(t,\phi(t))$$

- Using a Taylor expansion with a remainder to $\phi(t)$ about $t = t_n$,

$$\phi(t) = \phi(t_n) + \phi'(t_n)(t - t_n) + \frac{1}{2!}\phi''(\tau_n)(t - t_n)^2$$

where τ_n is some point in the interval $t_n < \tau_n < t_{n+1}$.

- Since $h = t_{n+1} - t_n$ and $\phi' = f(t, \phi)$, it follows that

$$\phi(t_{n+1}) = \phi(t_n) + f(t_n,\phi(t_n))h + \frac{1}{2!}\phi''(\tau_n)h^2$$

Recalling the Euler formula $y_{n+1} = y_n + f(t_n, y_n)h$,

- Error t_{n+1}

$$\phi(t_{n+1}) - y_{n+1} = [\phi(t_n) - y_n] + [f(t_n,\phi(t_n)) - f(t_n,y_n)]h + \frac{1}{2!}\phi''(\tau_n)h^2$$

- To compute the LTE e_{n+1}, take $y_n = \phi(t_n)$ and hence

$$e_{n+1} = \phi(t_{n+1}) - y_{n+1} = \frac{1}{2!}\phi''(\tau_n)h^2$$

Dr. M.Sakalli

Uniform Bound for LTE and h and Estimating GTE

Thus the LTE, e_{n+1} is proportional to the square of the step size h, and the proprtionlity constant depends on ϕ'', and hence is typically different for each step **(ie t dpnc)**.

- A uniform bound, valid on an interval $[a, b]$, which is the worst possible case, (may well be an overestimate in the interval of $[a, b]$. And **How to choose *h* for a desired LTE smaller than ε.**

$$|e_n| \le \frac{M}{2}h^2, \quad M = \max\{\phi''(t) : a < t < b\}$$

- This $\frac{M}{2}h^2 \le \varepsilon \;\Rightarrow\; h \le \sqrt{\frac{2\varepsilon}{M}}$
- It can be difficult estimating $|\phi''(t)|$ or M. However, the central fact is that LTE e_n is proportional to h^2. Thus *the x smaller the h, the x squared times the better the accuracy*.
- **GTE:** E_n for the Euler method - a **first order method**: Taking n steps, from t_0 to $T = t_0 + nh$, the error at each step is at most $Mh^2/2$, and hence error in n steps is at most $nMh^2/2$.

$$|E_n| \le \frac{nM}{2}h^2 = (T - t_0)\frac{M}{2}h$$

Dr. M.Sakalli

Example: LTE The same problem, $y' = 1 - t + 4y,\ \ y(0) = 1;\quad 0 \le t \le 2$

Using the solution $\phi(t)$, we have $\phi(t) = \left(4t - 3 + 19e^{4t}\right)/16 \Rightarrow \phi''(t) = 19e^{4t}$

The LTE e_{n+1} at step n + 1 is given by

$$e_{n+1} = \frac{\phi''(\tau_n)}{2}h^2 = \frac{19e^{4\tau_n}}{2}h^2,\ \ t_n < \tau_n < t_n + h$$

The presence of the factor 19 and the rapid growth of e^{4t} explains why the numerical approximations in this section with h = 0.05 were not very accurate.

For h = 0.05, the error in the first step is

$$e_1 = \frac{19e^{4\tau_0}}{2}(0.0025),\ \ 0 < \tau_0 < 0.05$$

Since $1 < e^{4\tau 0} < e^{4(0.05)} = e^{0.02}$, it follows that

$$0.02375 \cong \frac{19}{2}(0.0025) < e_1 < \frac{19e^{0.2}}{2}(0.0025) \cong 0.02901$$

It can be shown that the actual error is 0.02542.
Similar computations give the following bounds:

$$1.0617 < e_{20} < 1.2967\quad (0.95 < t < 1.0)$$
$$57.96 < e_{40} < 70.80\quad (1.95 < t < 2.0)$$

To reduce lte throughout $0 \le t \le 2$,
choose an h based on an analysis near t = 2.
To achieve $e_n < 0.01$, at $0 \le t \le 2$,
note that $M = 19e^{4(2)}$, and hence the required step size h is $h \le \sqrt{2\varepsilon/M} \cong 0.00059$

by Erica McEvoy, Using Matlab to integrate ODEs, coupledodes.pdf

y'=-5y; y(0)=1.43, realsolution = y=1.43 exp(-5t)

```
function dy = ilovecats(t,y)
    dy = zeros(1,1);
    dy = -5 * y;
[t,y] = ode45('ilovecats',[0 10], 1.43);
    plot(t,y,'-');
    xlabel('time');
    ylabel('y(t)');
    title('This plot dedicated to kitties everywhere');
error = abs(y - realsolution);
figure;
subplot(221)
plot(t,y,'-');
xlabel('time');
ylabel('y(t) computed numerically');
title('Numerical Solution');
subplot(222)
plot(t,realsolution,'-');
xlabel('time');
ylabel('y(t) computed analytically');
title('Analytical Solution');
subplot(223)
plot(t,error,'-');
xlabel('time');
ylabel('Error');
title('Relative error between numerical and analytical solutions');
subplot(224)
plot(0,0,'.');
xlabel('time')
ylabel('Kitties!!!!');
title('Kitties are SO CUTE!');
```

$d^2\Theta/dt^2=-w^2\sin(\Theta)$; w=sqrt(L/g);
$\Theta=y_1$, $d\Theta/dt=dy_1/dt=y_1'=y_2$;
$d^2\Theta/dt^2=dy_2/dt=y_2'=y_3$.. $dy_2/dt=-w^2\sin(y1)$;
$[Y']=A\ [Y]$; $[y_{11}'; y_2'] = [0\ 1; -w^2\sin(.)\ 0][y_1; y_2]$

```
function dy = pendulumcats(t, y, L, g)
    dy = zeros(2,1); w= 1;
    dy = [y(2);  -w^2*sin(y(1))];
return
[t,y] = ode45('pendulumcats', [0 25], [1.0 1.0]);
plot(t,y(:,1),'-');
xlabel('time');
ylabel('y_{1}(t)');
title('\theta (t)');
figure; plot(t,y(:,2),'-');
xlabel('time');ylabel('y_{2}(t)');
title('d \theta / dt (t)');
figure; plot(y(:,1),y(:,2),'-'); xlabel('\theta (t)');
ylabel('d \theta / dt (t)');
title('Phase Plane Portrait for undamped pendulum');
```

- "Strange attractor", or the "Lorenz attractor." Strange attractors appear in phase spaces of chaotic dynamical systems. Edward Lorenz is the first person to report such bizarre endings, and as such, he is often considered the father (or founder) of Chaos Theory. The "buttery effect“. The idea is that chaotic systems have a sensitive dependence on initial conditions { if you were to play around with the initial conditions for x(t), y(t) and z(t) in these equations and plot phase space portraits, the tiniest changes in initial conditions can lead to a crazy huge difference in position in phase space at some later time (which is not what you'd expect if the equations were
- considered "deterministic" {you'd expect that equations that were almost identical to give you almost identical trajectories and phase space portraits at any time! but). Because of this, chaotic systems like the weather are difficult to predict past a certain time range since you can't measure the starting atmospheric conditions completely accurately.
- Coupled equations: Edward Lorenz, a mathematician and weather forecaster for the US Army Air Corps, and MIT prof. Interested in solving a simple set of 3 coupled de because he wanted to estimate weather a week earlier. Equations of convection rolls??? rising in the atmosphere.
- x' =-p x + p y
- y' = r x – y – xz
- z' = x y – b z
- where P, r, and b are all constants (P represents the Prandtl number, and r is the ratio of Rayleigh number to the critical Rayleigh number), and x, y and z are all functions of time. (You can read more about what these equations represent in Lorenz's classic paper, Deterministic nonperiodic flow:J:Atmos:Sci:20 : 130


```
P = 10;r = 28;b = 8/3
function dy = lorenz(t,y,P,r,b)
    dy = zeros(3,1);
    dy(1) = P*(y(2) - y(1));
    dy(2) = -y(1)*y(3) + r*y(1) - y(2);
    dy(3) = y(1)*y(2) - b*y(3);
[t,y] = ode45('lorenz',[0 250], [1.0 1.0 1.0]');
subplot(221), plot(y(:,1),y(:,2),'-');
xlabel('x(t)');ylabel('y(t)');
title('Phase Plane Portrait for Lorenz attractor -- y(t) vs. x(t)');
subplot(222), plot(y(:,1),y(:,3),'-');
xlabel('x(t)');ylabel('z(t)');
title('Phase Plane Portrait for Lorenz attractor -- z(t) vs. x(t)');
subplot(223), plot(y(:,2),y(:,3),'-');
xlabel('y(t)');ylabel('z(t)');
title('Phase Plane Portrait for Lorenz attractor -- z(t) vs. y(t)');
subplot(224), plot(0,0,'.');
xlabel('Edward Lorenz')
ylabel('Kitties'); title('Kitties vs. Lorenz');
plot3(y(:,1),y(:,2),y(:,3),'-')
xlabel('x(t)'); ylabel('y(t)'); zlabel('z(t)');
title('3D phase portrait of Lorenz Attractor');
```

We can use ode45 to and solutions of the scale factor, a(t), to the Friedmann equations:

- $(a'/a)^2 = [(8\pi G)/3]*Þ – k/a^2$
- $(a''/a) = [(-4\pi G)/3]$ (Þ – 3(Pbar))

$m_1d^2x_1/dt^2=[-Gm_1m_2(r_{12}^3)]\ (x_1-x_2)$; $dx_1/dt=u_1$, $du_1/dt=-Gm_2(r_{12}^3)]\ (x_1-x_2)$;
$m_1d^2y_1/dt^2=[-Gm_1m_2(r_{12}^3)]\ (y_1-y_2)$; $dy_1/dt=v_1$, $dv_1/dt=-Gm_2(r_{12}^3)]\ (y_1-y_2)$;
$m_2d^2x_2/dt^2=[Gm_1m_2(r_{12}^3)]\ (x_1-x_2)$; $dx_2/dt=u_2$, $du_2/dt=Gm_1(r_{12}^3)]\ (x_1-x_2)$;
$m_2d^2y_2/dt^2=[Gm_1m_2(r_{12}^3)]\ (y_1-y_2)$; $dy_2/dt=v_2$, $dv_2/dt=Gm_1(r_{12}^3)]\ (y_1-y_2)$;

```
function dz = twobody(t,z)
        dz = zeros(8,1);
        G = 2;
        m1 = 2;
        m2 = 2;
        dz(1) = z(2);
        dz(2) = ((G*m2)/(((z(1) - z(5)).^2 + (z(3) - z(7)).^2).^(3/2)))*(z(5) - z(1));
        dz(3) = z(4);
        dz(4) = ((G*m2)/(((z(1) - z(5)).^2 + (z(3) - z(7)).^2).^(3/2)))*(z(7) - z(3));
        dz(5) = z(6);
        dz(6) = ((G*m1)/(((z(1) - z(5)).^2 + (z(3) - z(7)).^2).^(3/2)))*(z(1) - z(5));
        dz(7) = z(8);
        dz(8) = ((G*m1)/(((z(1) - z(5)).^2 + (z(3) - z(7)).^2).^(3/2)))*(z(3) - z(7));
```

- where z(1), z(2), ... through z(8) represent the functions $x_1(t)$, $u_1(t)$, $y_1(t)$, $v_1(t)$, $x_2(t)$, $u_2(t)$, $y_2(t)$, and $v_2(t)$, respectively. (So that $r_{12}^2 = (z(1) - z(5))^2 + (z(3)-z(7))^2$

```
[t,z] = ode45('twobody',[0 25], [-1 0 0 -1 1 0 0 1]);
        [t,z] = ode45('twobody',[0 25], [-1 0 0 -1 1 0 0 1]);
        plot(z(:,1),z(:,3),'-');
        xlabel('x_{1}(t)'); ylabel('y_{1}(t)');
        title('Particle 1 orbit in xy space -- first 25 seconds');
        figure;
        plot(z(:,5),z(:,7),'-');
        xlabel('x_{2}(t)'); ylabel('y_{2}(t)');
        title('Particle 2 orbit in xy space -- first 25 seconds');
```

Dr. M.Sakalli