Computational Methods (PHYS 2030)

Instructors: Prof. Christopher Bergevin (cberge@yorku.ca)

Schedule: Lecture: MWF 11:30-12:30 (CLH M)

Website: http://www.yorku.ca/cberge/2030W2018.html

York University
Winter 2018
Lecture 5

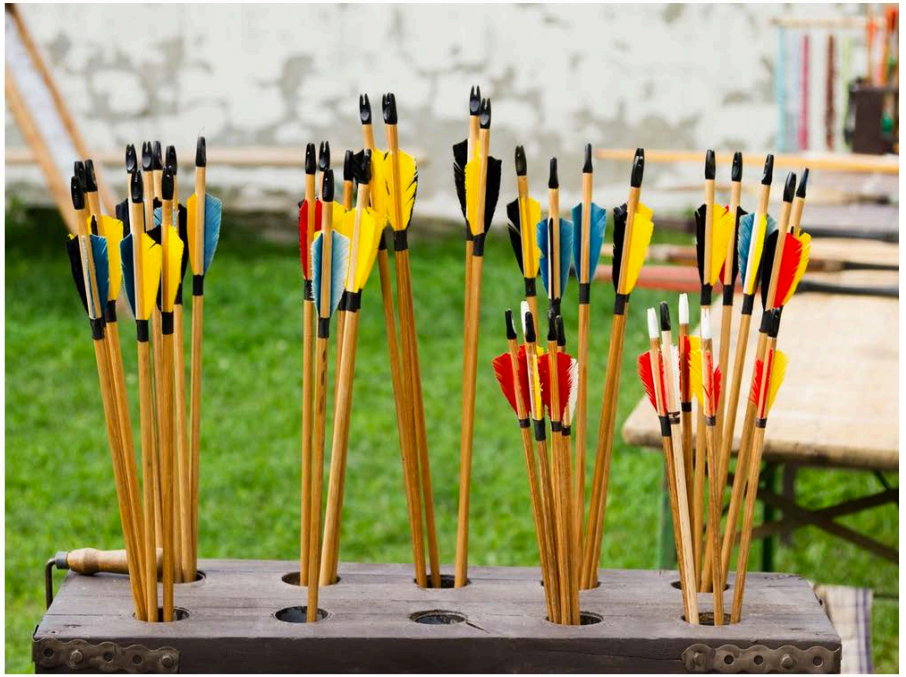
WIRED

RHETT ALLAIN SCIENCE 01.10.18 09:00 AM

CAN AN ARROW FIRED STRAIGHT UP FALL FAST ENOUGH TO KILL YOU?

main.py

```
GlowScript 2.6 VPython

#this just sets up a graph
gd=graph(xtitle="Time [s]", ytitle="Vertical Position [m]")
f1=gcurve(color=color.red)

#starting constants
g=9.8 #N/kg - gravitational constant

#change this
m=.019 #kg - mass of arrow

#change this
```



https://www.wired.com/story/can-an-arrow-fired-straight-up-fall-fast-enough-to-kill-you/

Newton's law of heating/cooling

"Newton proposed that the temperature of a hot object decreases at a rate proportional to the difference between its temperature and that of its surroundings. Similarly, a cold object heats up at a rate proportional to the temperature difference between the object and its surroundings."

$$\frac{dT}{dt} = \alpha(T_o - T)$$

When a murder is committed, the body, originally at 37°C, cools according to Newton's Law of Cooling. Suppose that after two hours the temperature is 35°C, and that the temperature of the surrounding air is a constant 20°C.

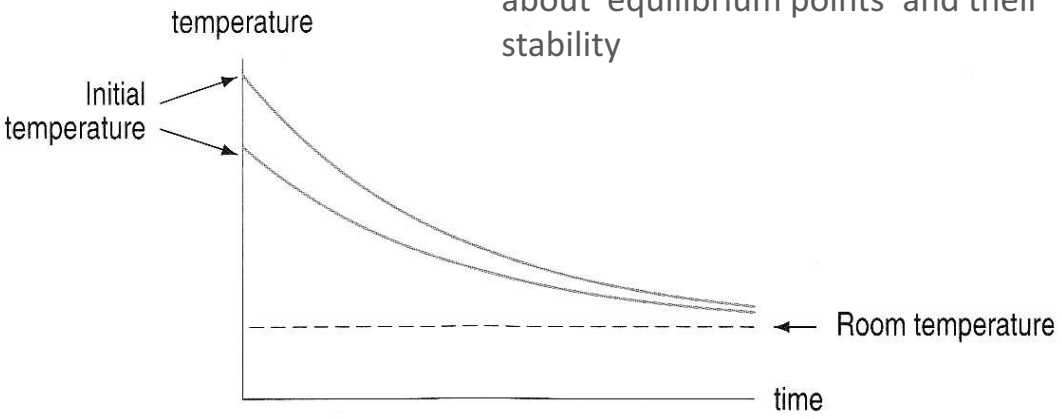
(a) Find the temperature, H, of the body as a function of t, the time in hours since the murder was committed.
(b) Sketch a graph of temperature against time.
(c) What happens to the temperature in the long run? Show this on the graph and algebraically.
(d) If the body is found at 4 pm at a temperature of 30°C, when was the murder committed?

Stability

Newton's law of heating/cooling

$$\frac{dT}{dt} = \alpha(T_o - T)$$

Note: Very natural place to think about 'equilibrium points' and their stability



Solution

$$T(t) = T_0 + Ce^{-\alpha t}$$

- An **equilibrium solution** is constant for all values of the independent variable. The graph is a horizontal line.
- An equilibrium is **stable** if a small change in the initial conditions gives a solution which tends toward the equilibrium as the independent variable tends to positive infinity.
- An equilibrium is **unstable** if a small change in the initial conditions gives a solution curve which veers away from the equilibrium as the independent variable tends to positive infinity.

Hughes-Hallett et al. (2005)

Some further common examples

Falling body: Terminal velocity

Assume air resistance is proportional to velocity, the Newton's 2nd Law leads to:

$$m\frac{dv}{dt} = mg - kv$$

$$\frac{dv}{dt} = -\frac{k}{m}\left(v - \frac{mg}{k}\right)$$

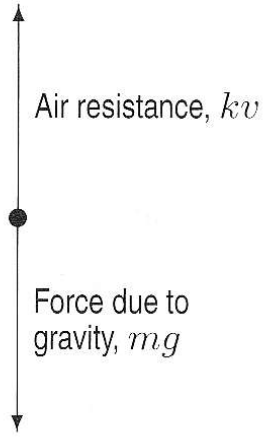


Figure 11.44: Forces acting on a falling object

Solution $$v = \frac{mg}{k}\left(1 - e^{-kt/m}\right)$$

Hughes-Hallett et al. (2005)

Some further common examples

Falling body: Terminal velocity

$$m\frac{dv}{dt} = mg - kv \qquad v = \frac{mg}{k}\left(1 - e^{-kt/m}\right)$$

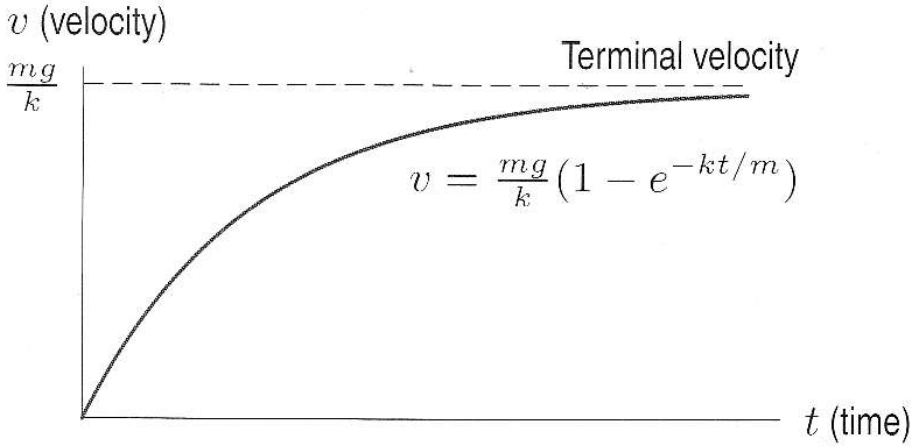
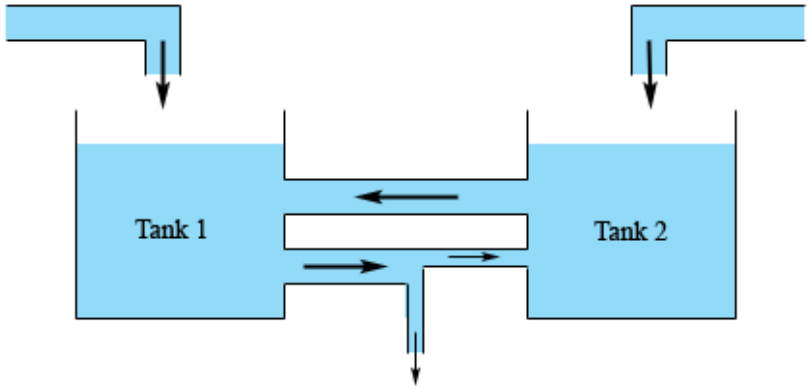


Figure 11.45: Velocity of falling dust particle assuming that air resistance is kv

Compartmental models
(e.g., salt in a reservoir)



http://tutorial.math.lamar.edu/

Some further common examples

Compartmental models (e.g., salt in a reservoir)

"A water reservoir holds 100 million gallons of water and supplies a city with 1 million gallons a day. The reservoir is partly refilled by a spring which provides 0.9 million gallons a day, and the rest of the water, 0.1 million gallons a day, comes from run-off from the surrounding land. The spring is clean, but the run-off contains salt with a concentration of 0.0001 pound per gallon. There was no salt in the reservoir initially and the water is well mixed (that is, the out-flow contains the concentration of salt in the tank at that instant)."

Think about units!

It is important to distinguish between the total quantity, Q, of salt in pounds, and the concentration, C, of salt, in pounds/gallon, where

$$\text{Concentration} = C = \frac{\text{Quantity of salt}}{\text{Volume of water}} = \frac{Q}{100\,\text{million}}\left(\frac{\text{lb}}{\text{gal}}\right).$$

$$\begin{pmatrix}\text{Rate of change of}\\ \text{quantity of salt}\end{pmatrix} = \text{Rate salt entering} - \text{Rate salt leaving}.$$

$$\text{Rate salt entering} = \text{Concentration} \cdot \text{Volume per day} \qquad \frac{dQ}{dt} = 10 - \frac{Q}{100}$$

Hughes-Hallett et al. (2005)

Some further common examples

Logistic equation (e.g., population growth, 'carrying capacity')

$$\frac{1}{P}\frac{dP}{dt} = k - aP$$

- Nonlinear

$$\frac{dP}{dt} = kP\left(1 - \frac{P}{L}\right)$$

- To solve, use separation of variables and partial fractions

Solution $$P = \frac{L}{1 + Ae^{-kt}}$$

where $$A = \frac{L - P_0}{P_0}$$

$$P_0 = P(0)$$

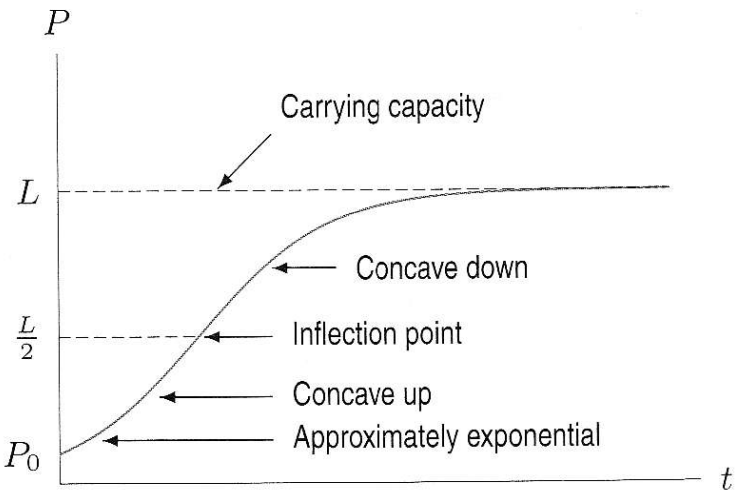


Figure 11.51: Logistic growth with inflection point

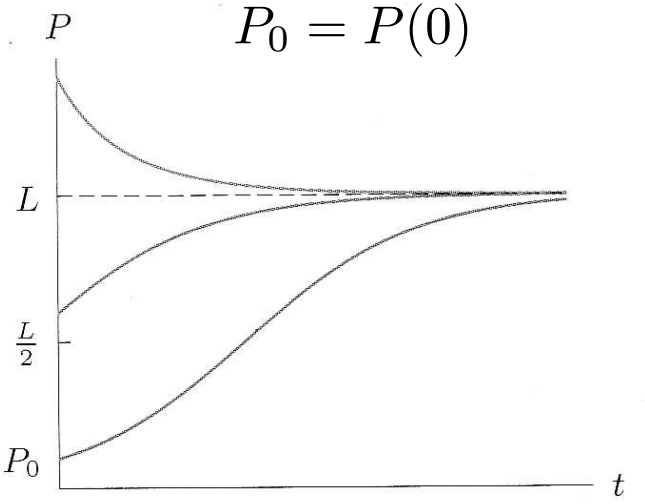


Figure 11.52: Solutions to the logistic equation

Hughes-Hallett et al. (2005)

Equilibria and stability

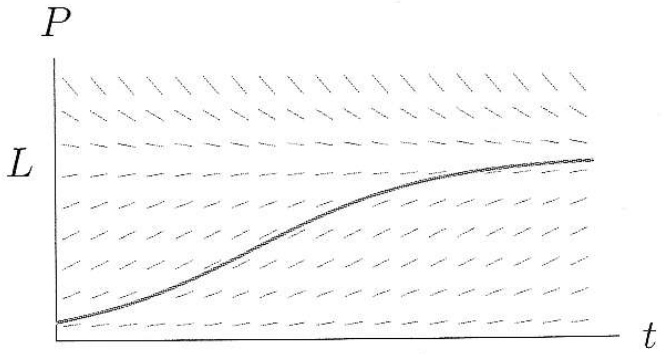


Figure 11.49: Slope field for $dP/dt = kP(1 - P/L)$

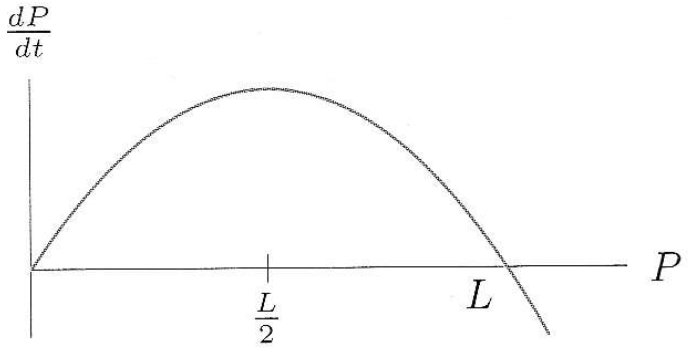
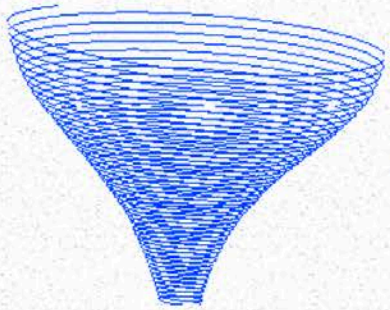


Figure 11.50: $dP/dt = kP(1 - P/L)$

- Plot *dP/dt* vs. P
- Where will the equilibrium points be?
- How can you infer their stability?

Hughes-Hallett et al. (2005)

Helpful (numerical) starting point http://math.rice.edu/~dfield/dfpp.html



dfield and pplane

the java versions

dfield and pplane

`dfield` (direction field) and `pplane` (phase plane) are software programs for the interactive analysis of ordinary differential equations (ODE). The software is described in detail in the manual Ordinary Differential Equations using MATLAB. Additionaly, several textbooks on differential equations refer to and use `dfield` and `pplane`. Among them are Differential Equations and Differential Equations with Boundary Value Problems by John Polking, Albert Boggess, and David Arnold.

This page contains download links to the latest Java versions of `dfield` and `pplane`. The original MATLAB versions can be found at: http://math.rice.edu/~dfield/index.html

Previously `dfield` and `pplane` were written as java applets meant to be run in a browser. These are being discontinued because of security reasons. The apps availalble here are the replacements.

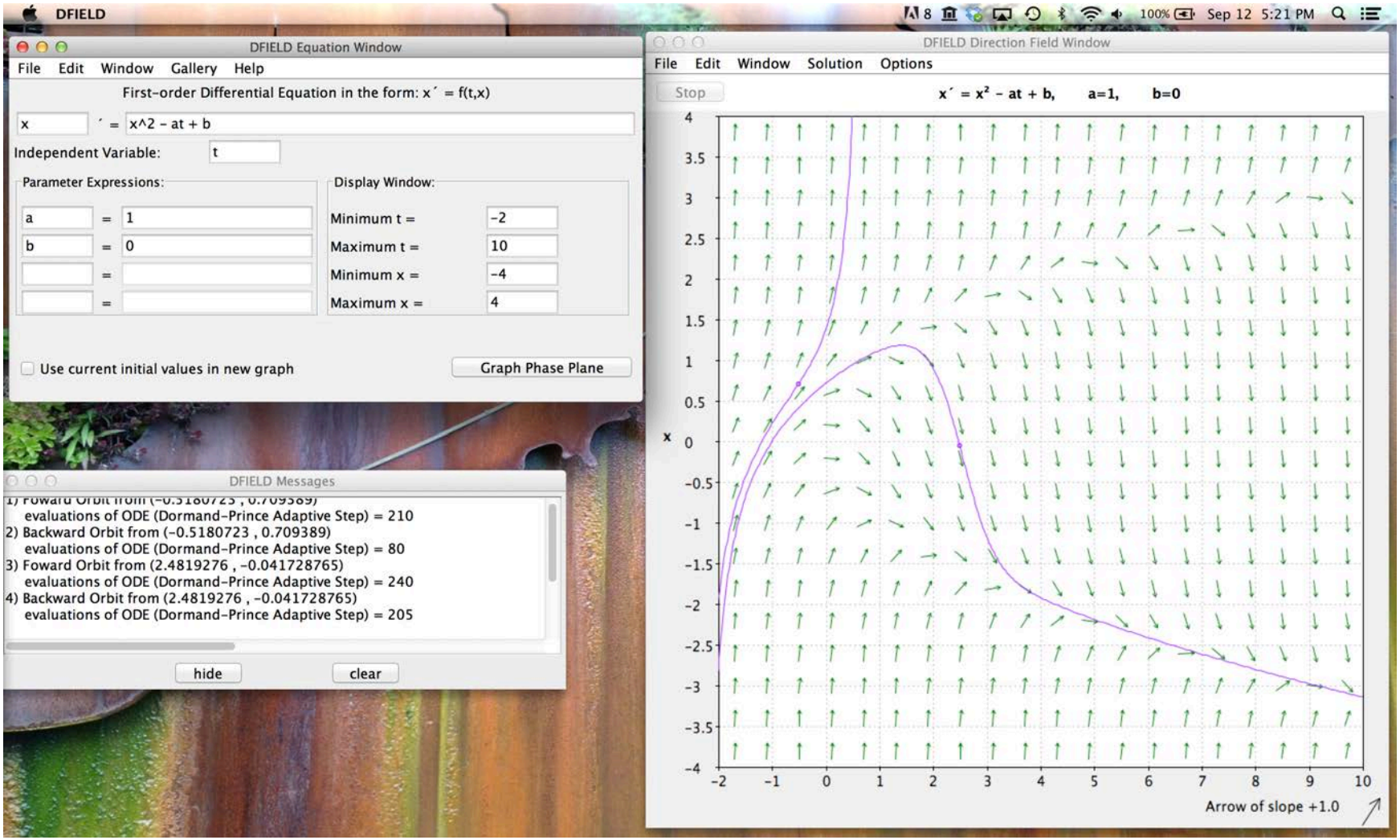
`dfield` and `pplane` are copyrighted in the name of John C. Polking, Department of Mathematics, Rice University. They are not in the public domain. However, they are being made available free for use in educational institutions. This offer does not extend to any application that is made for profit. Users who have such applications in mind should contact John C. Polking. The Java versions were written by Joel Castellanos in collaboration with John C. Polking.

- dfield – Useful for 1-D ODEs
- pplane – Useful for 2-D ODEs (e.g., 2nd order ODE, or system of two 1st order eqns.)

Helpful (numerical) starting point

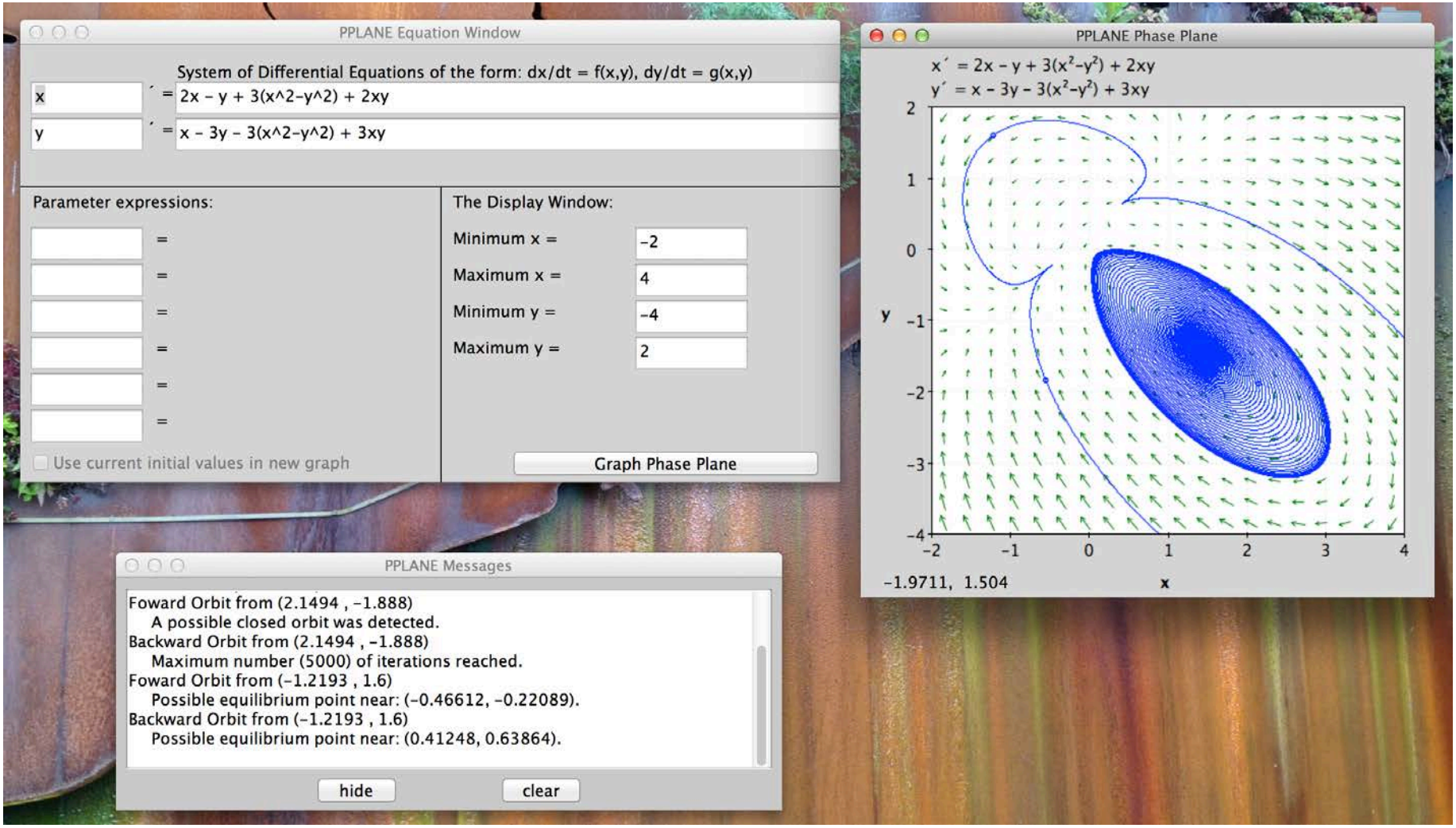
http://math.rice.edu/~dfield/dfpp.html

dfield



Helpful (numerical) starting point http://math.rice.edu/~dfield/dfpp.html

pplane → Phase plane analysis for 'systems' of ODEs (future lecture)

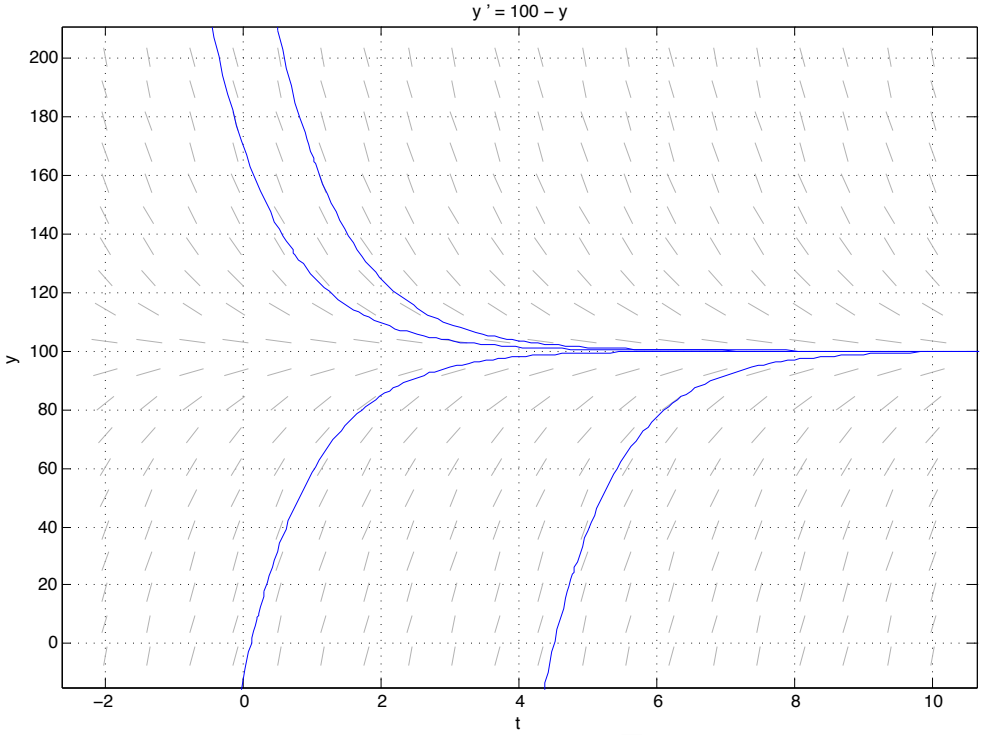


Ex.

$$\frac{dy}{dt} = 100 - y$$

- Slope field?

→ Useful starting point to connect what ODE is telling you and what 'solutions' look like



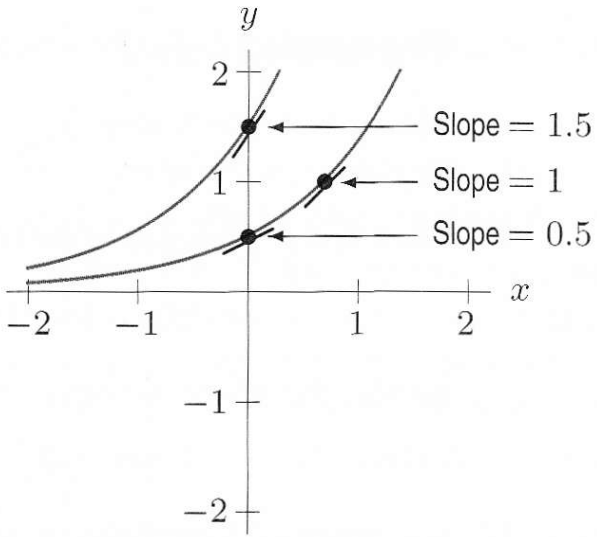


Figure 11.4: Solutions to $dy/dx = y$

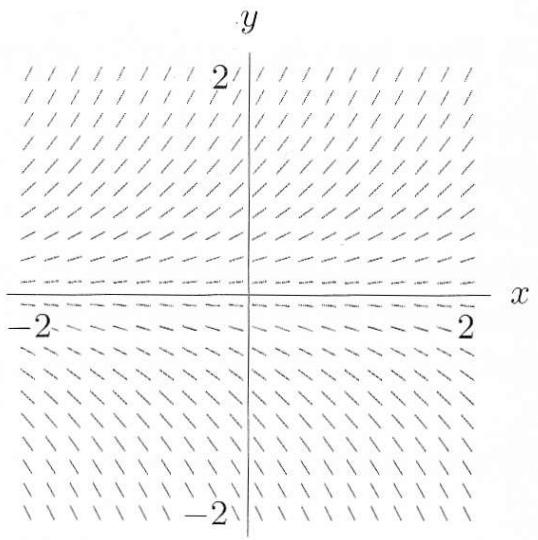


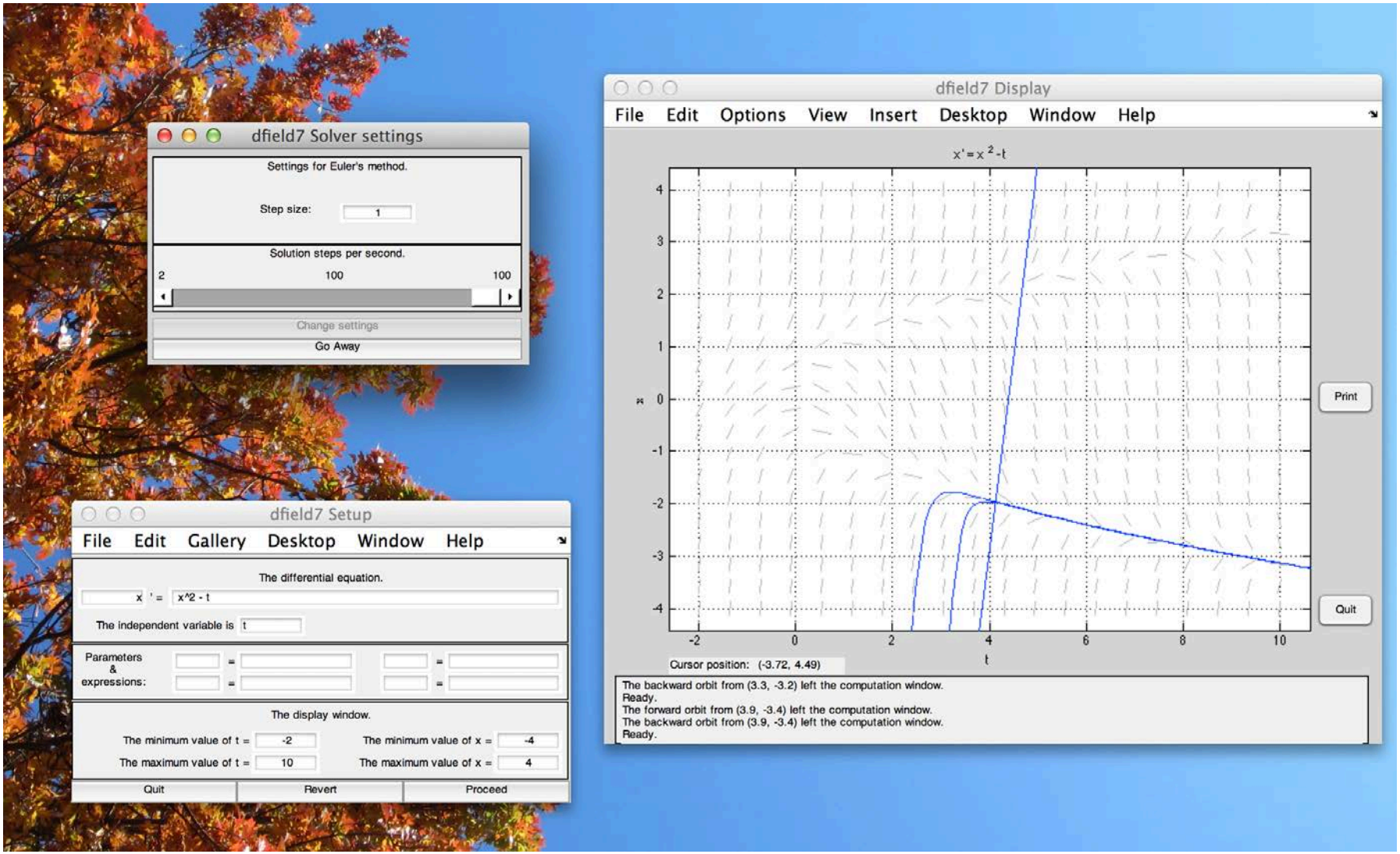
Figure 11.5: Slope field for $dy/dx = y$

→ But how is *dfield* computing solution curves?

Hughes-Hallett et al. (2005)

→ How is *dfield* computing solution curves?

Let's break it!



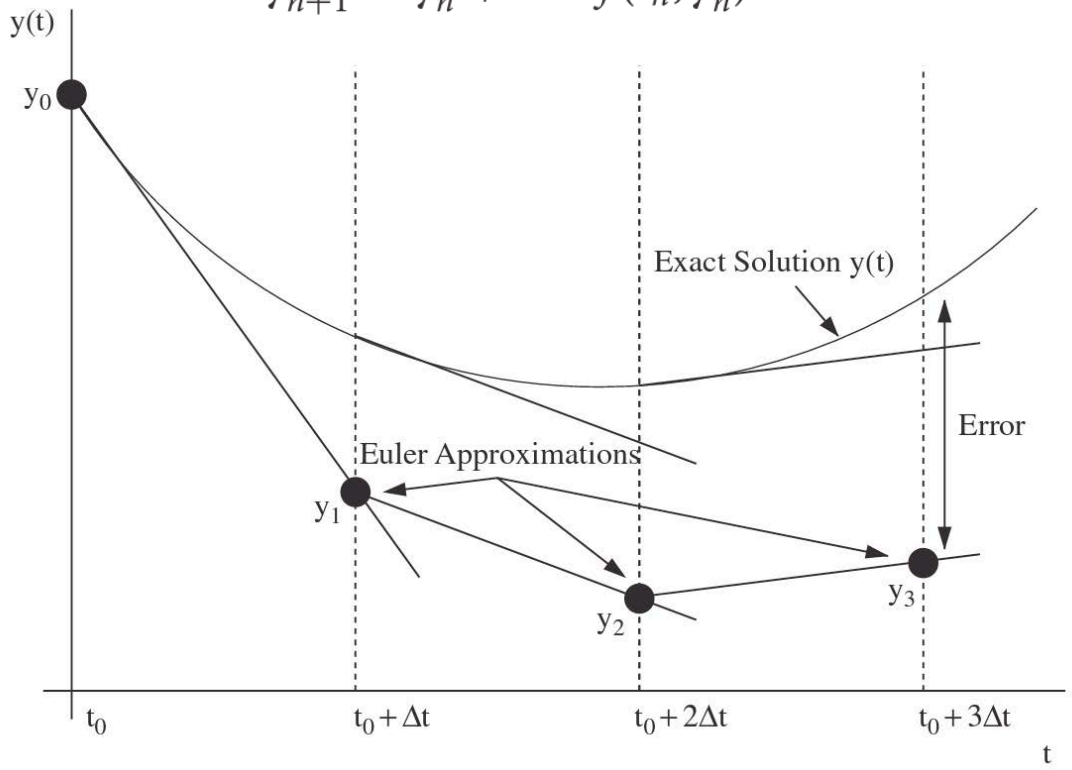
Starting point: Euler's method

Idea: Since equations tell us how things change, numerically integrate to find solution(s)

$$\frac{d\mathbf{y}}{dt} = f(t, \mathbf{y})$$

$$\frac{d\mathbf{y}}{dt} = f(t, \mathbf{y}) \quad \Rightarrow \quad \frac{\mathbf{y}_{n+1} - \mathbf{y}_n}{\Delta t} \approx f(t_n, \mathbf{y}_n).$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \cdot f(t_n, \mathbf{y}_n).$$



Note the 'boring' case: f is a function of t alone (in which case we simply integrate)

Reality check: Choice of variable names may change a bit throughout slides, but don't be confused!

$$y'(x) = f(x, y)$$

$$y(x) = \int^{x} f(\chi)\, d\chi$$

→ Note that while we can propagate a solution forward (or backward), there can be some associated ***error***

Kutz (2013)

Euler's method

→ Looking under the hood.....

Use Euler's method for $dy/dx = y$. Start at the point $P_0 = (0, 1)$ and take $\Delta x = 0.1$.

Table 11.2 *Euler's method for* $dy/dx = y$, *starting at* $(0, 1)$

	x	y	$\Delta y = (\text{Slope})\Delta x$
P_0	0	1	$0.1 = (1)(0.1)$
P_1	0.1	1.1	$0.11 = (1.1)(0.1)$
P_2	0.2	1.21	$0.121 = (1.21)(0.1)$
P_3	0.3	1.331	$0.1331 = (1.331)(0.1)$
P_4	0.4	1.4641	$0.14641 = (1.4641)(0.1)$
P_5	0.5	1.61051	$0.161051 = (1.61051)(0.1)$

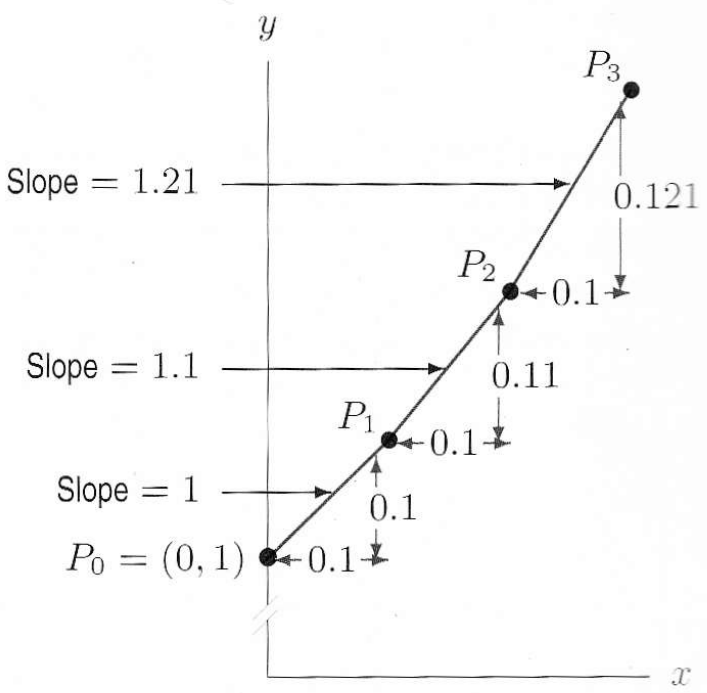


Figure 11.25: Euler's approximate solution to $dy/dx = y$

Hughes-Hallett et al. (2005)

Euler's method

Idea: Since equations tell us how things change, numerically integrate to find solution(s)

Connect back to Taylor series:

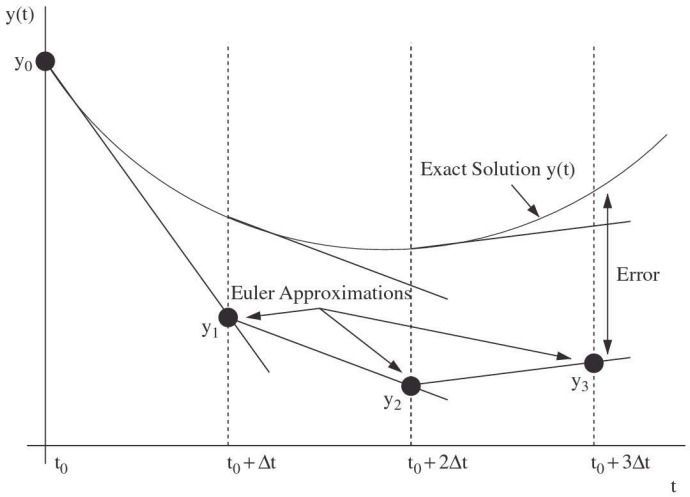
$$y(x) = y(x_0) + y'(x_0)(x - x_0) + \frac{y''(x_0)}{2!}(x - x_0)^2 + \frac{y^{(3)}(x_0)}{3!}(x - x_0)^3 + \cdots$$

Or rewrite as: $x \equiv x_0 + \Delta x$

$$y(x_0 + \Delta x) = y(x_0) + y'(x_0)\Delta x + \frac{1}{2!}y''(x_0)(\Delta x)^2 + \frac{1}{3!}y^{(3)}(x_0)(\Delta x)^3 + \cdots$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \cdot f(t_n, \mathbf{y}_n).$$

→ So Euler's method is a 'first order approximation' as higher order terms are ignored



Error in Euler's method

$$y(x_0 + \Delta x) = y(x_0) + y'(x_0)\Delta x + \frac{1}{2!}y''(x_0)(\Delta x)^2 + \frac{1}{3!}y^{(3)}(x_0)(\Delta x)^3 + \cdots$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \cdot f(t_n, \mathbf{y}_n).$$

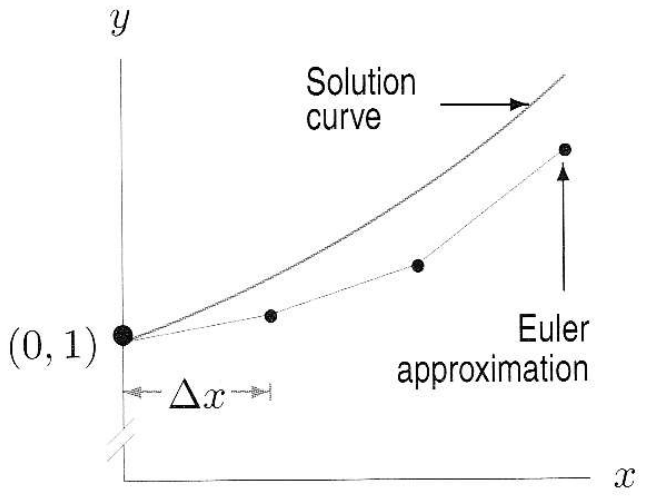


Figure 11.27: Euler's approximate solution to $dy/dx = y$

→The error in Euler's method is the difference between approximate and exact value. If the number of steps used is n (think about how n will be related to Δt), the error is approximately *proportional* to $1/n$

- 'higher order' methods reduce such error (e.g., Runge-Kutta) → We'll return to this

- Question: How can we know what the higher order derivatives are, since the ODE only tells us the first derivative?

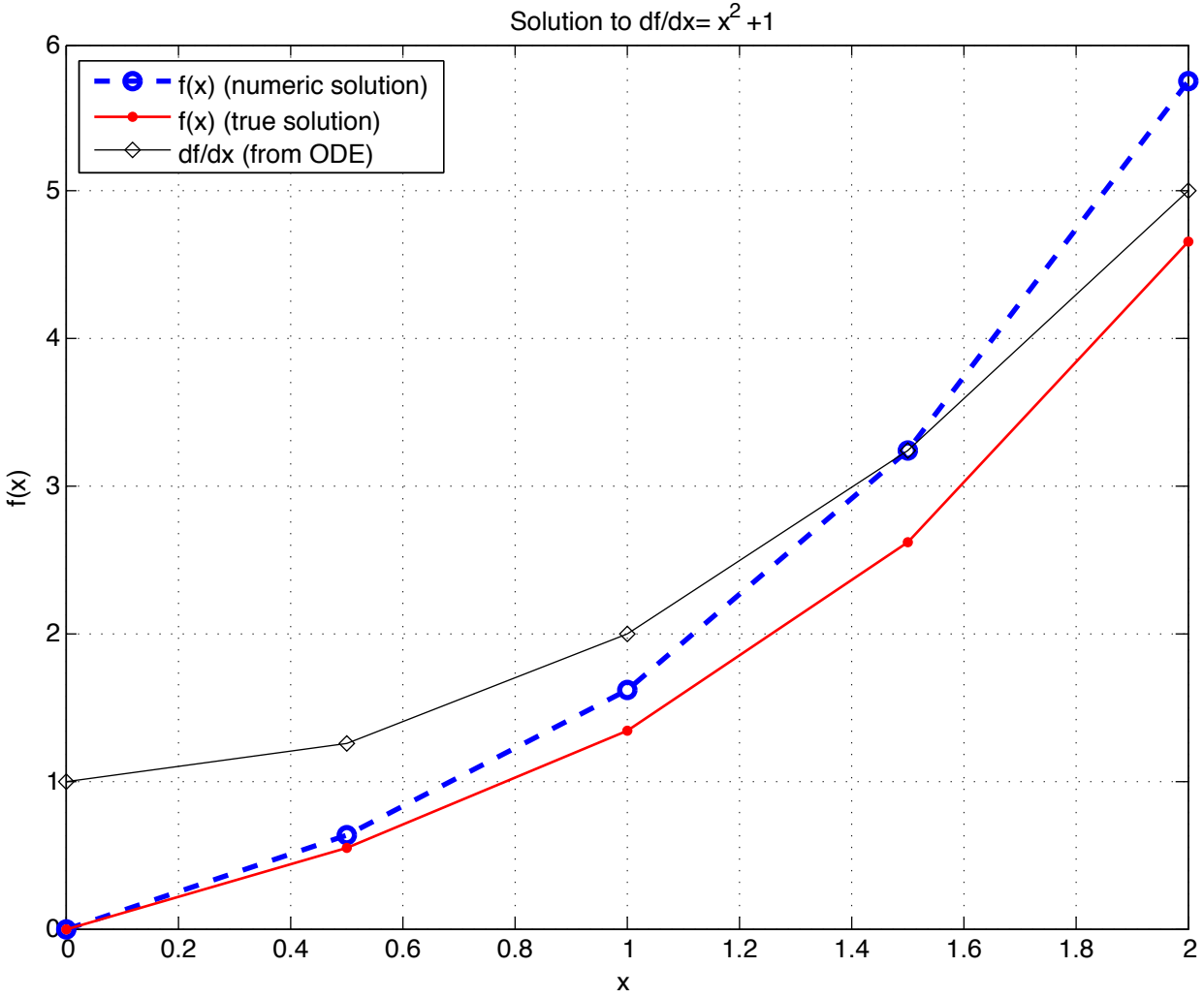
Hughes-Hallett et al. (2005)

$$\frac{df}{dx} = x^2 + 1$$

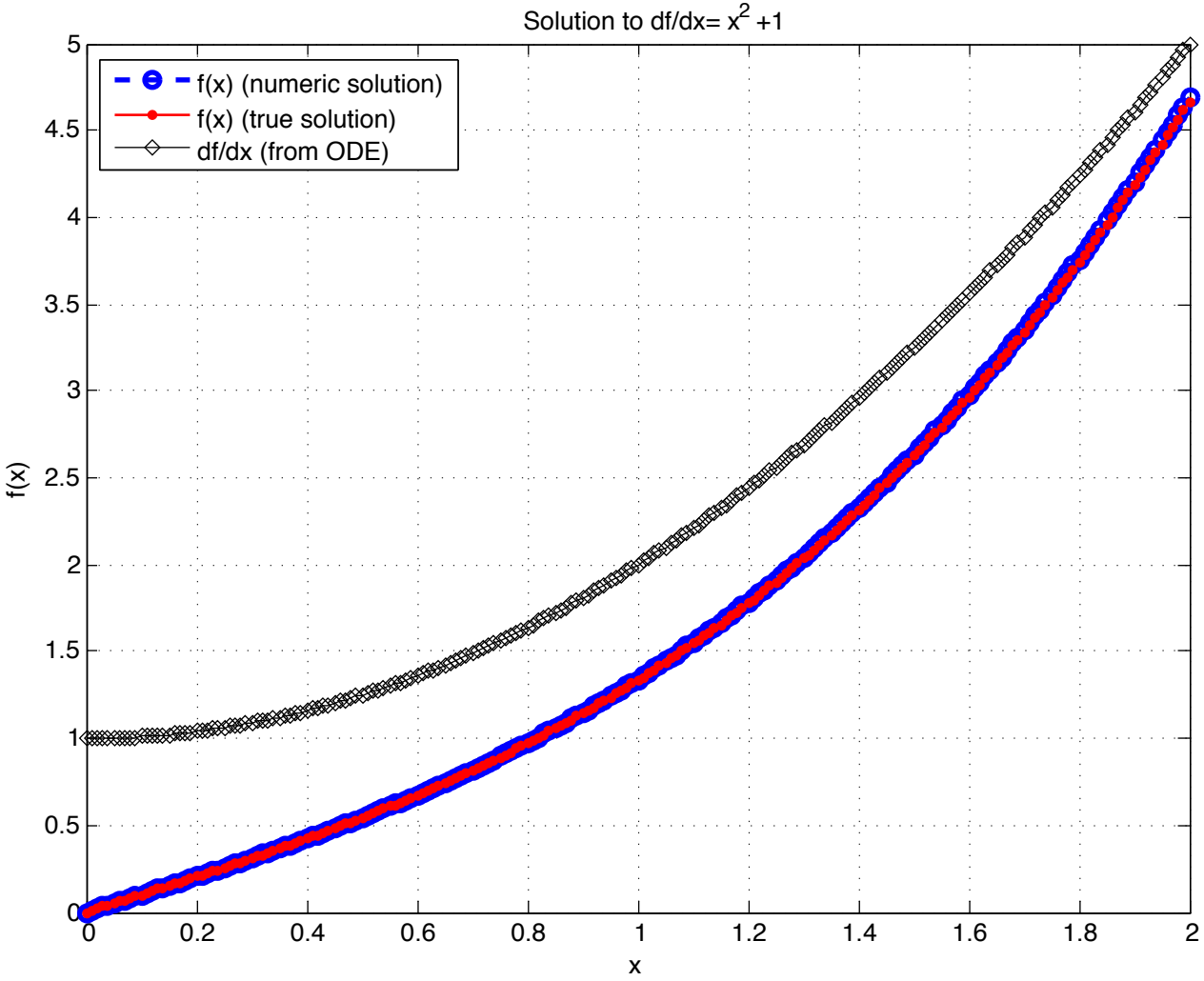
```
% York U PHYS 2030  (09.15.14)
% use basic Euler method to solve the following differential equation
% f'(x)= x^2 + 1

clear
% ************************
% User Inputs
Fprime = @(x)(x.^2+1); % function describing the ODE
Fsol= @(x,IC)(1/3*x.^3 + x + IC); % solution (known; IC is the initial condition, f0)
stepsize= 0.003;
f0= 0.0;   % initial condition at xI [i.e. f(xI)]
xB= [0 2];  % interval to solve over (i.e., boundaries)

% ************************

F(1)= f0;  %initialize first value
k=1;  % counter

for j=xB(1):stepsize:xB(2)
 if (j == xB(1)),
     F(k)= f0;  % handle initial condition
 else
     F(k)= F(k-1) + stepsize*(Fprime(j));    % apply Euler's method
 end
 x(k)= j;  % keep track of x for plotting
 k= k+ 1;  % increment counter
end

% visualize
figure(1); clf;
plot(x,F, 'o--','LineWidth',2);    grid on;    hold on
xlabel('x');    ylabel('f(x)'); title('Solution to df/dx= x^2 +1')

% now plot exact solution and derivative
plot(x,Fsol(x,f0), 'r.-','LineWidth',1);
plot(x,Fprime(x), 'kd-');
legend('f(x) (numeric solution)', 'f(x) (true solution)', 'df/dx (from ODE)','Location','NorthWest')
```

Where are the key steps occurring?

```
stepsize= 0.5;
f0= 0.0
```



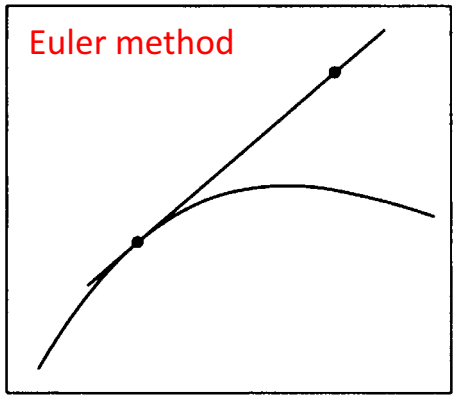
```
stepsize= 0.01;
f0= 0.0
```



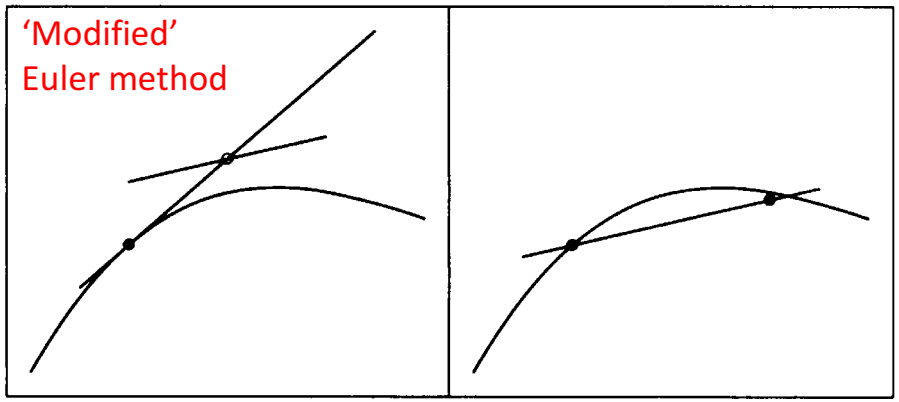
Improve Euler's method?

- Can we 'guess' the solution at a mid-point?

$$x_{mid} = x_0 + \frac{h}{2} \qquad y(x_{mid}) = y_0 + \frac{h}{2}y_0' = y_0 + \frac{h}{2}f_0$$



$$y(x_0 + h) = y(x_0) + hf\left(x_{mid}, y_{mid}\right)$$



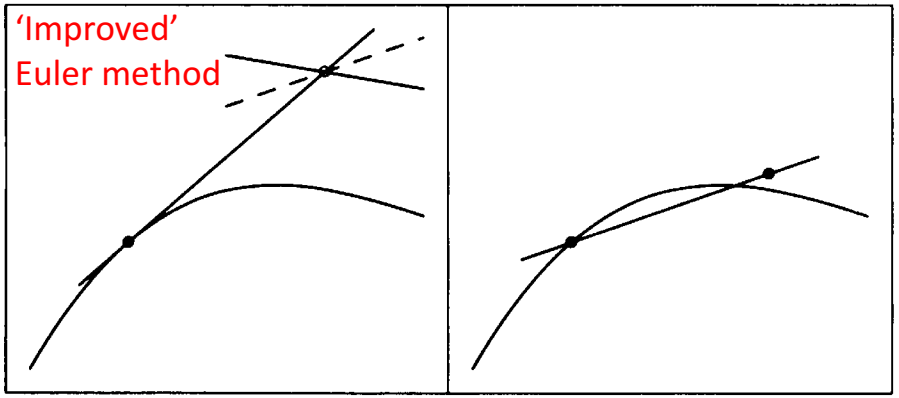
- Change in the associated error?
- How does this differ from changing our step-size?

$$f(x_{mid}, y_{mid}) = y'(x_{mid}) \approx \frac{y(x_0 + h) - y(x_0)}{h}$$

Devries (1994)

Improve Euler's method?

- What if we tried to use a *mean* value of the derivative?



$$y(x_0 + h) = y(x_0) + h\frac{f_0 + f(x_0 + h, y_0 + hf_0)}{2}$$