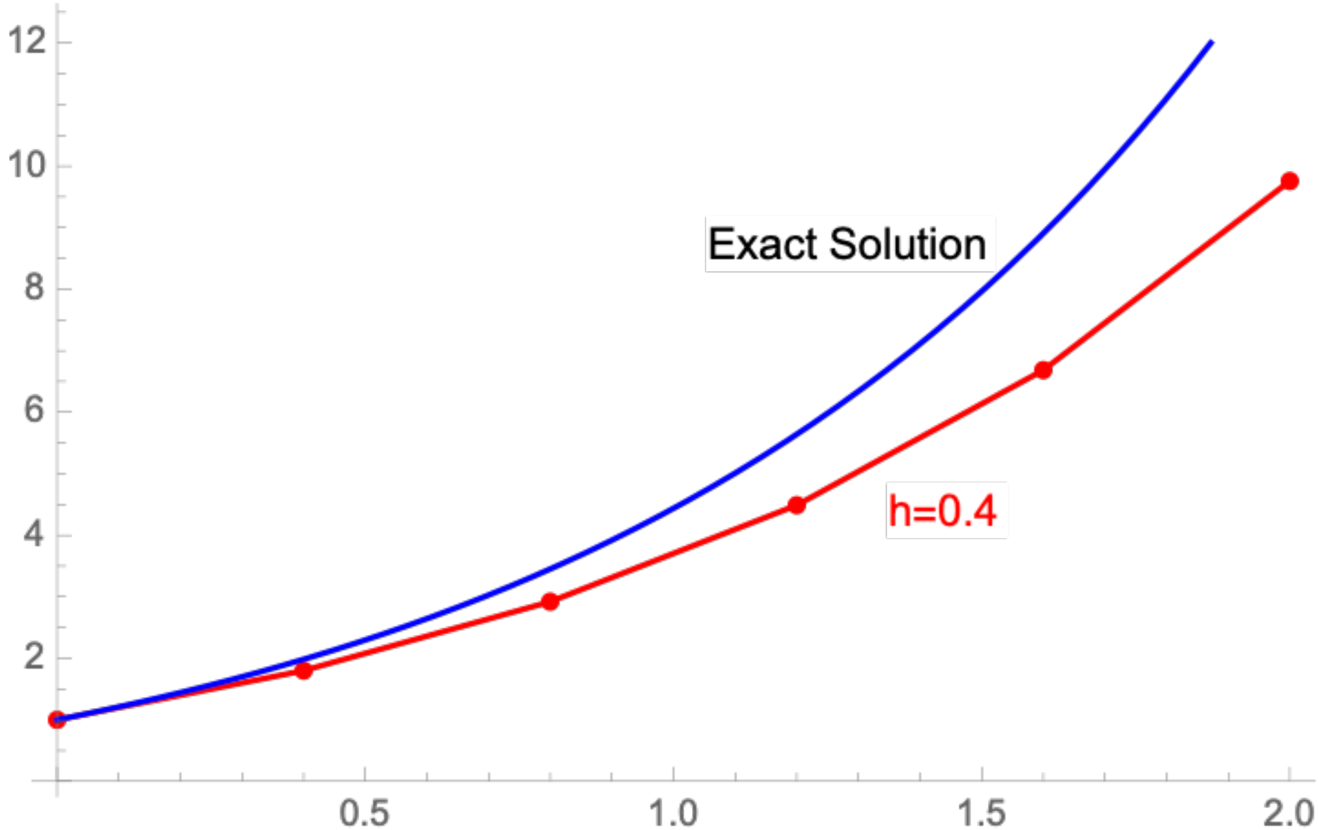
2.5 A Closer Look at the Euler Method

In the previous section, we talked about Euler method.

We have the



E

We have the **error** computed by

$$y_{\text{actual}} - y_{\text{approx}} = y(x_n) - y_n$$

The **error** becomes smaller when we take smaller step size h.

Can we improve this method so that the solution will be more accurate with less steps?

ALGORITHM The Improved Euler Method

Given the initial value problem

$$\frac{dy}{dx} = f(x, y), \qquad y(x_0) = y_0,$$

the improved Euler method with step size h consists in applying the iterative formulas

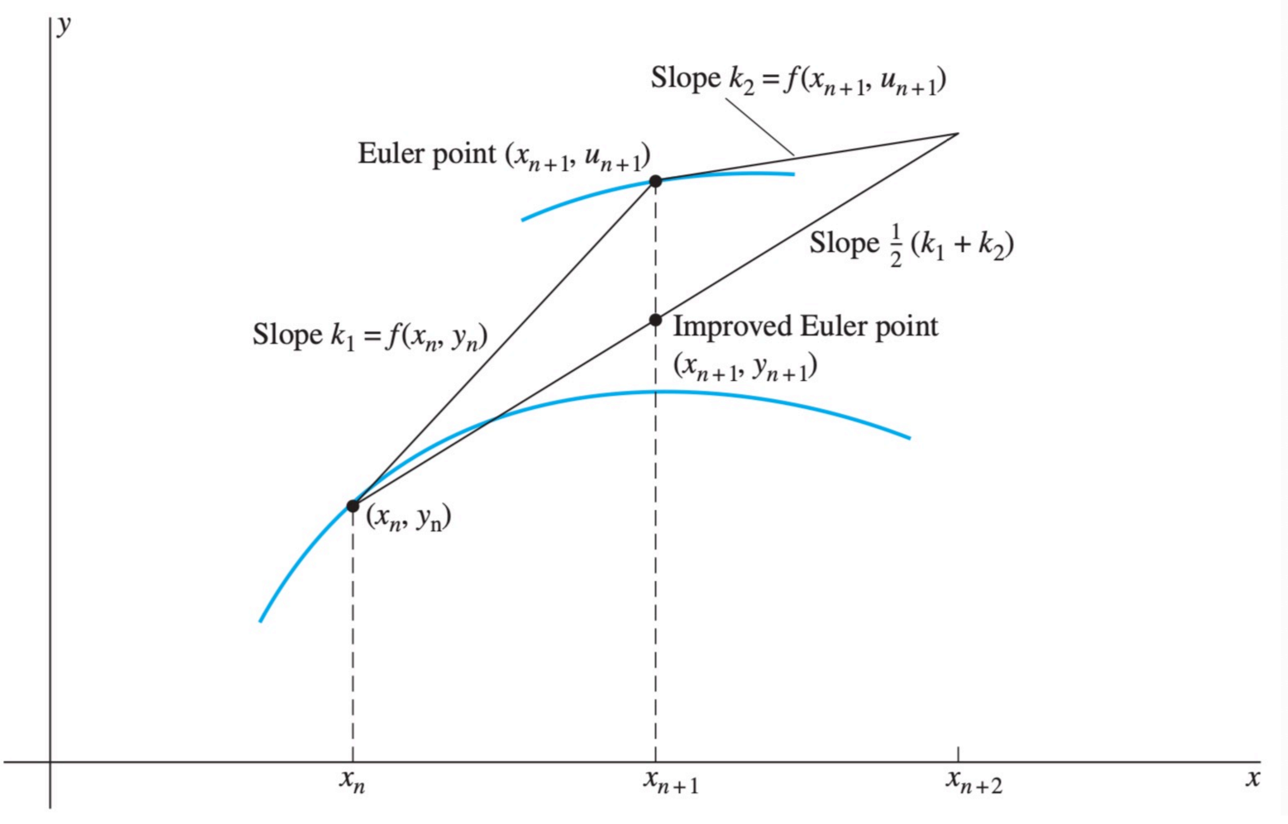
$$k_1 = f(x_n, y_n),$$

$$u_{n+1} = y_n + h \cdot k_1,$$

$$k_2 = f(x_{n+1}, u_{n+1}),$$

$$y_{n+1} = y_n + h \cdot \frac{1}{2}(k_1 + k_2)$$

to compute successive approximations y_1, y_2, y_3 to the [true] values $y(x_1), y(x_2), y(x_3), \cdots$ of the [exact] solution $y = y(x)$ at the points x_1, x_2, x_3, respectively.



Example 1 Use the improved Euler method with a computer system to find the desired solution values in the following problem. Start with step size $h = 0.1$, and then use successively smaller step sizes until successive approximate solution values at $x = 2$ agree rounded off to four decimal places.

$$y' = \frac{1}{2}x^2 + y^3 - 3, y(0) = 0; \qquad y(2) = ?$$

In **Matlab**, we write two .m files like the following. You can also find this in your text book Page 115.

1. First we define the function impeuler.m like the following. We need to save the file name as impeuler.m

```
function [X,Y] = impeuler(x,y,x1,n)
h = (x1 - x)/n; % step size
X = x; % initial x
Y = y; % initial y
for i = 1:n % begin loop
k1 = f(x,y); % first slope
k2 = f(x+h,y+h*k1); % second slope
k = (k1 + k2)/2; % average slope
x = x + h; % new x
y = y + h*k; % new y
X = [X;x]; % update x-column
Y = [Y;y]; % update y-column
end % end loop
```

2. Then we define our function yp.m below. We save this as f.m

```
function yp = f(x,y)
yp = (1/2)*x^2+y^2-3; %yp=y'
```

In the command window, we type

```
>> [X,Y]=impeuler(0,0,2,20)
```

Similarly to the last example in section 2.4. By changing the value of n, we make the following table

h	0.1	0.01	0.001	0.0001
n	20	200	2000	20000
$y(2)$	-1.2538	-1.2556	-1.2556	-1.2556

It shows that $y(2) = -1.2556$ rounded off accurate to 4 decimal places.