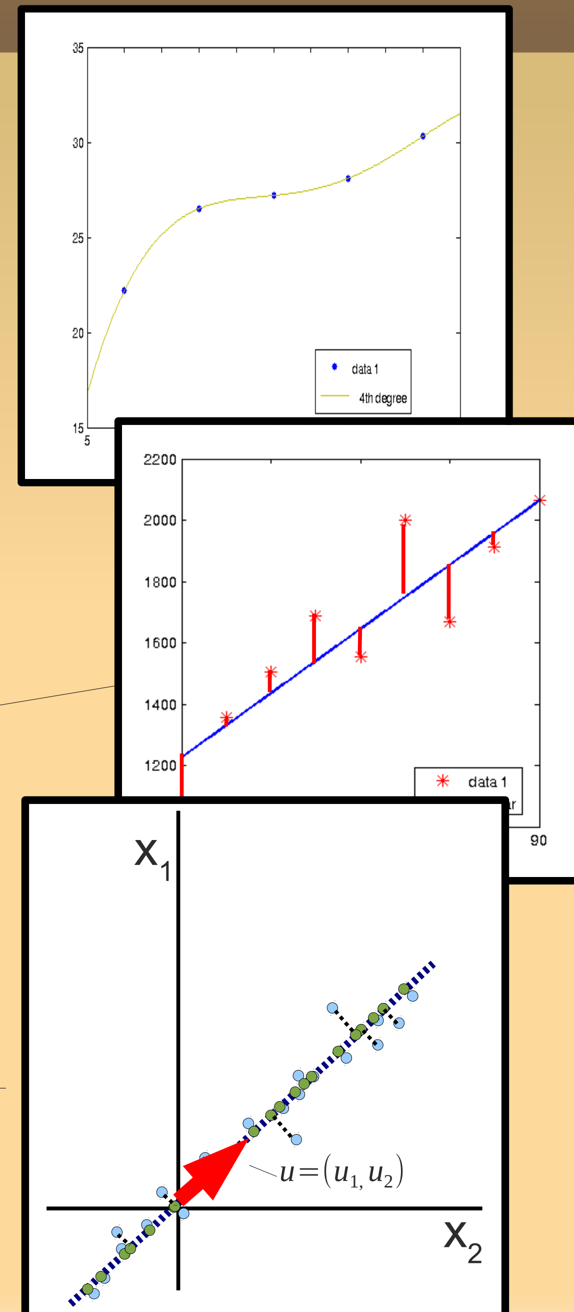
Scientific Computing
Maastricht Science Program

Week 4

Frans Oliehoek
<frans.oliehoek@maastrichtuniversity.nl>

Recap

- Matlab...!
- Supervised Learning
  - find *f* that maps $\{x_1^{(j)},\dots,x_D^{(j)}\} \rightarrow y^{(j)}$
  - Interpolation
    - *f* goes through the data points
  - linear regression
    - lossy fit, minimizes 'vertical' SSE
- Unsupervised Learning
  - PCA
  - We just have data points $\{x_1^{(j)},\dots,x_D^{(j)}\}$



Numerical Differentiation and Integration

Numerical Differentiation and Integration

- Finding derivatives or primitives of a function f
- not always easy or possible....
  - no closed form solution exists
  - the solution is a very complex expression that is hard to evaluate
  - we may not know f (as before!)

→ numerical methods

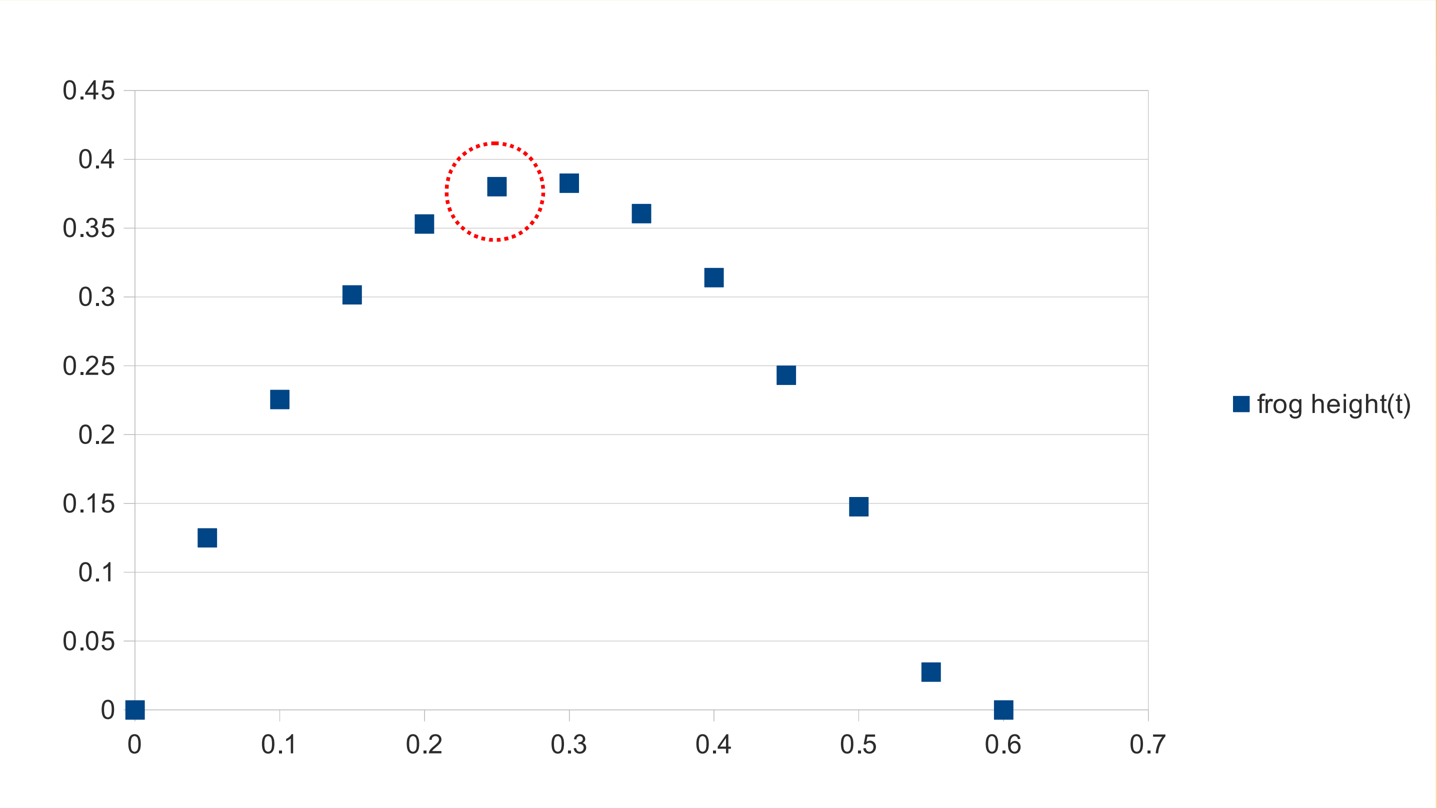
Numerical Differentiation

- If we want to know the rate of change...

- E.g.:
  - fluid in a cylinder with a hole in the bottom, measured every 5 seconds.
  - High-speed camera images of animal movements, (jumping in frogs and insects, suction feeding in fish, and the strikes of mantis shrimp)
    - determine speed
    - and acceleration

Numerical Differentiation

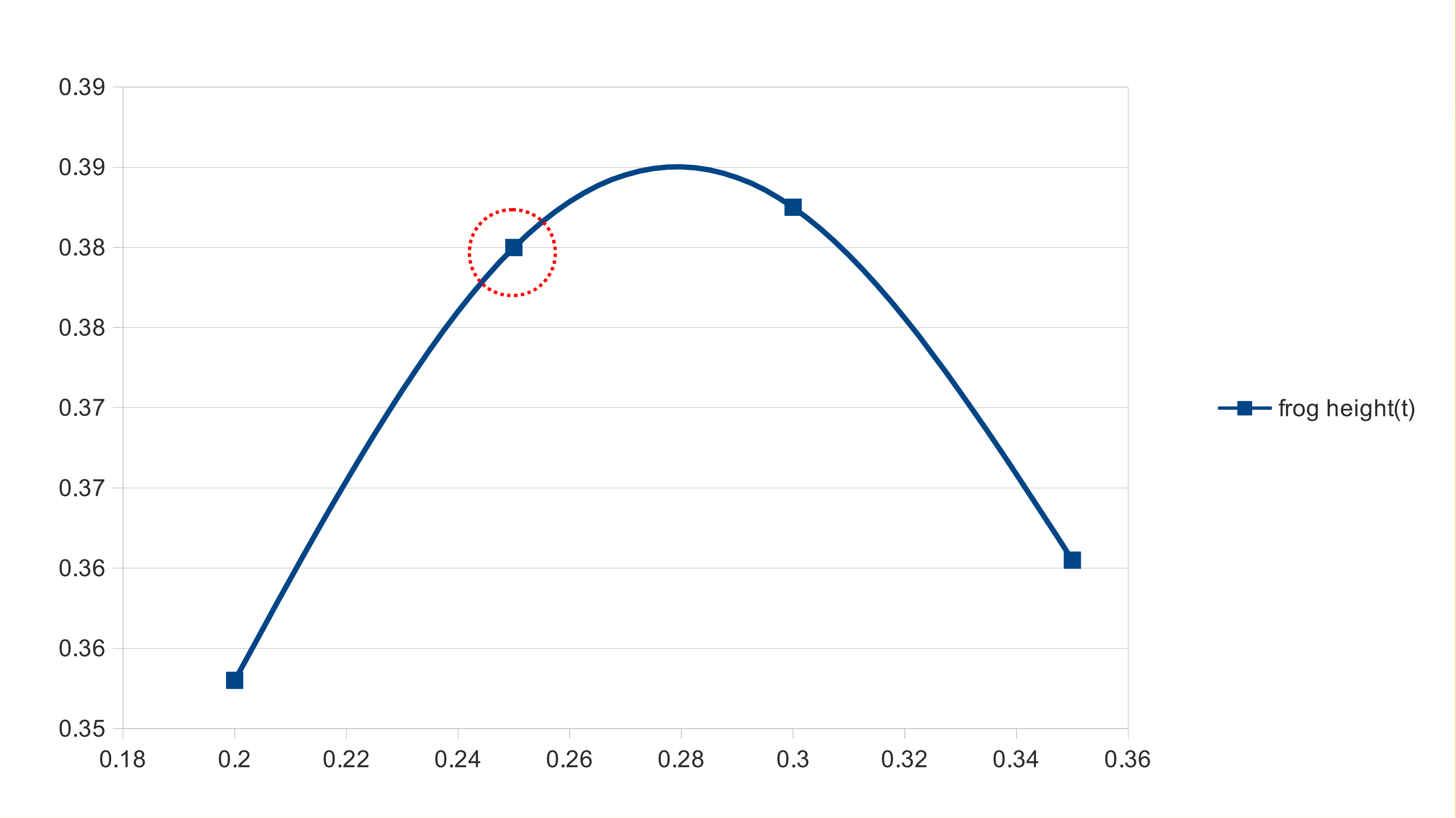
- Determine the vertical speed at t=0.25



- what would you do?

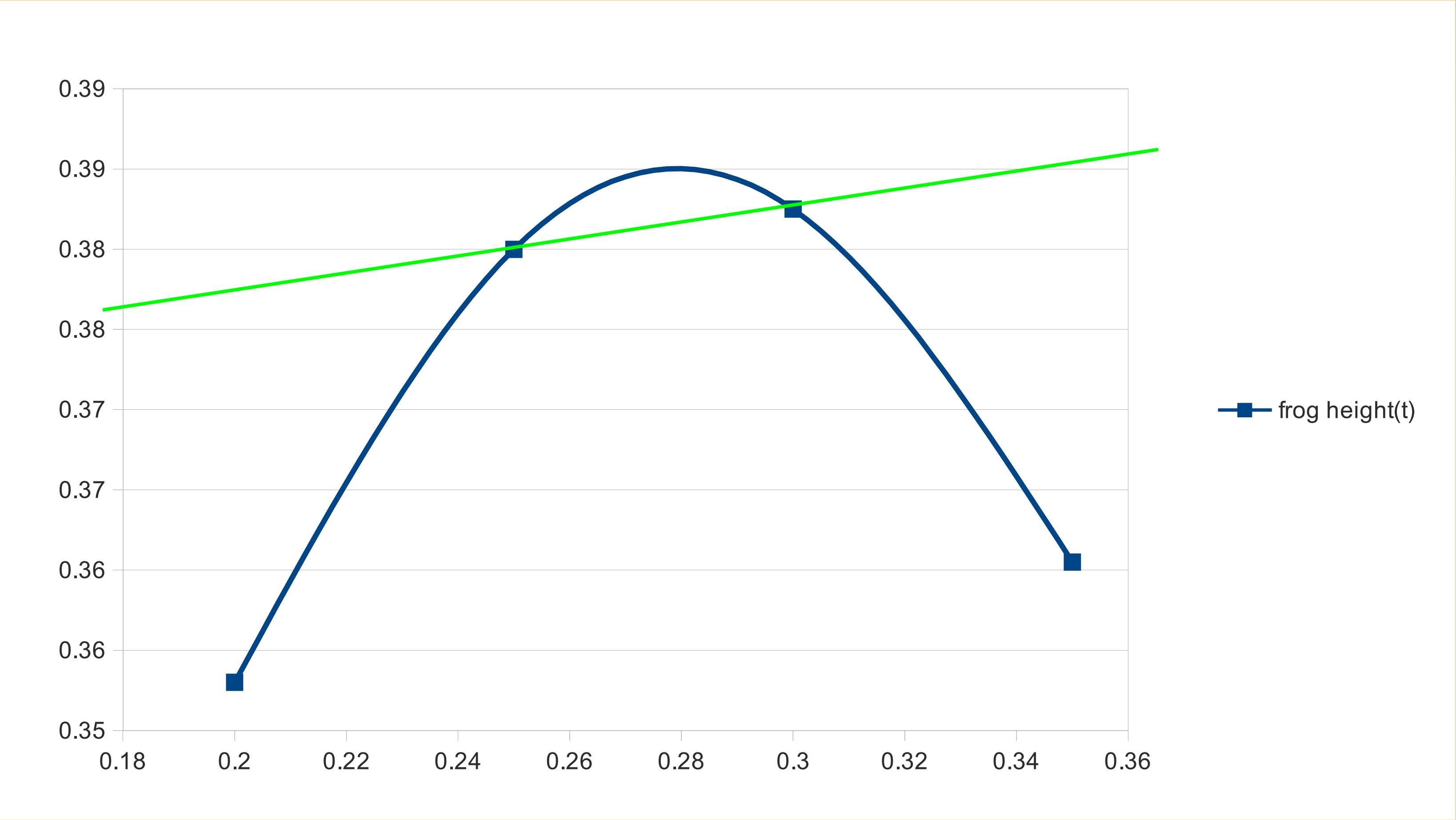
Numerical Differentiation

- Determine the vertical speed at t=0.25...
  - a few options...



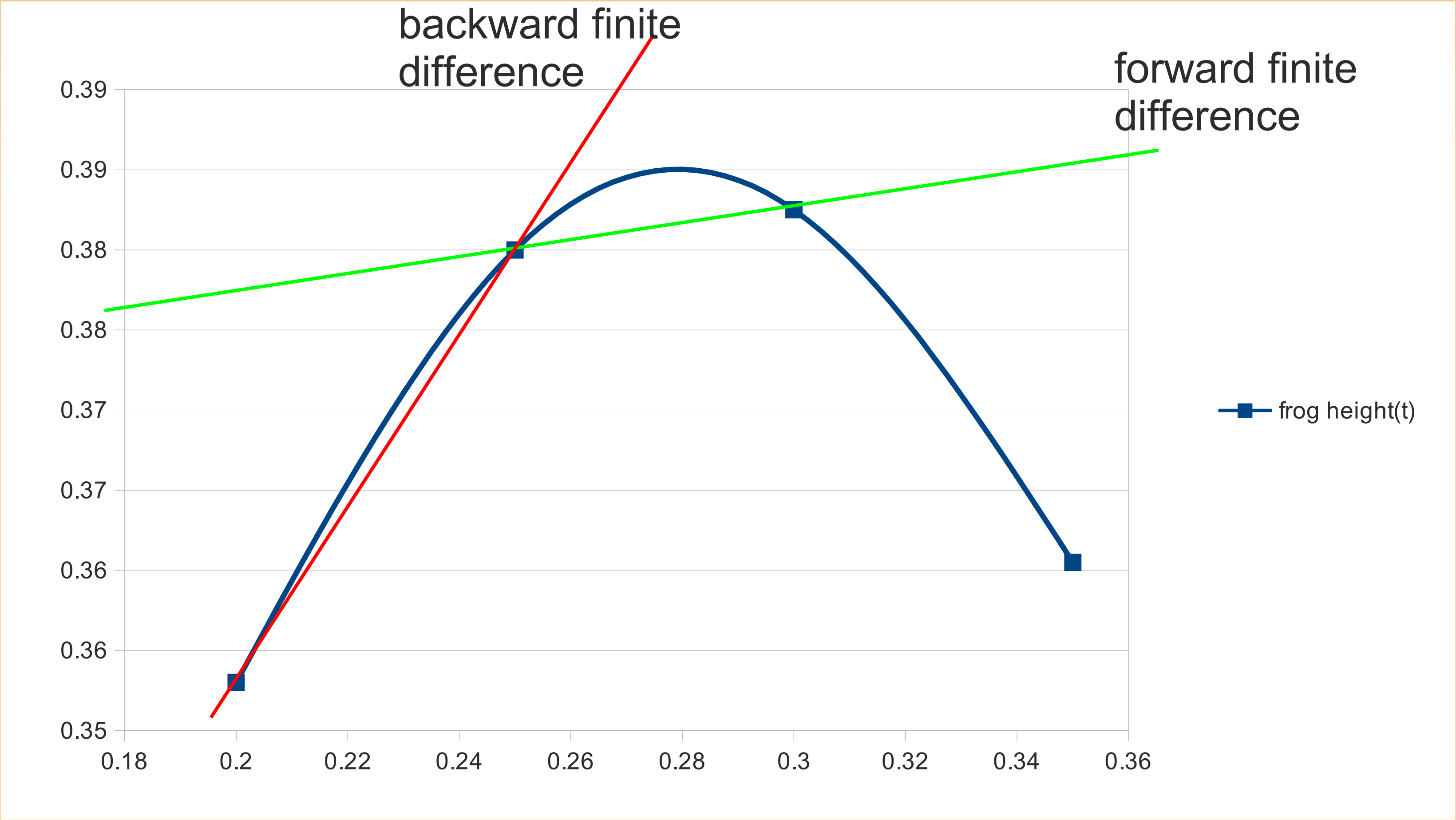
Numerical Differentiation

- Determine the vertical speed at t=0.25...
  - a few options...



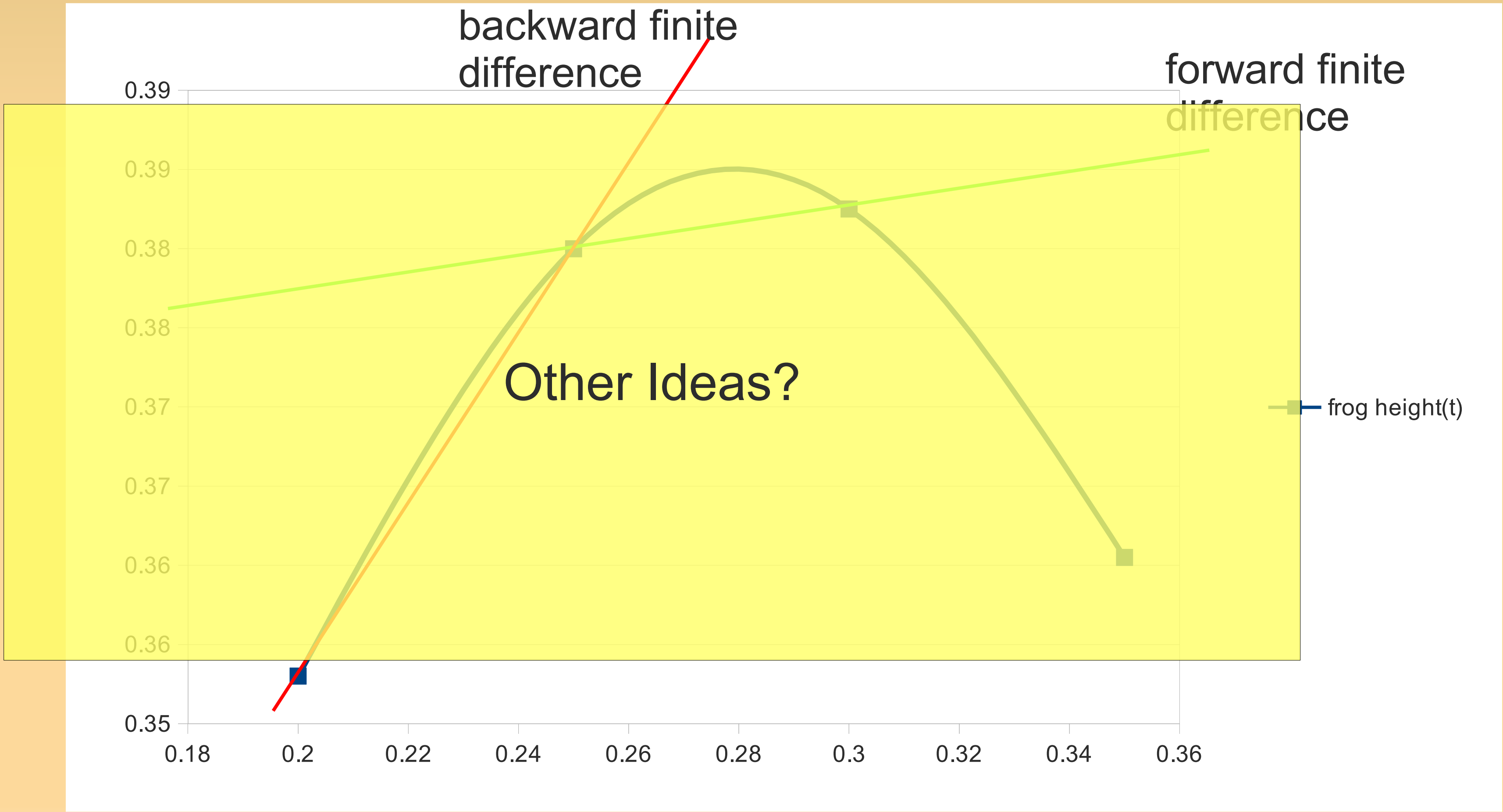
Numerical Differentiation

- Determine the vertical speed at t=0.25...
  - a few options...



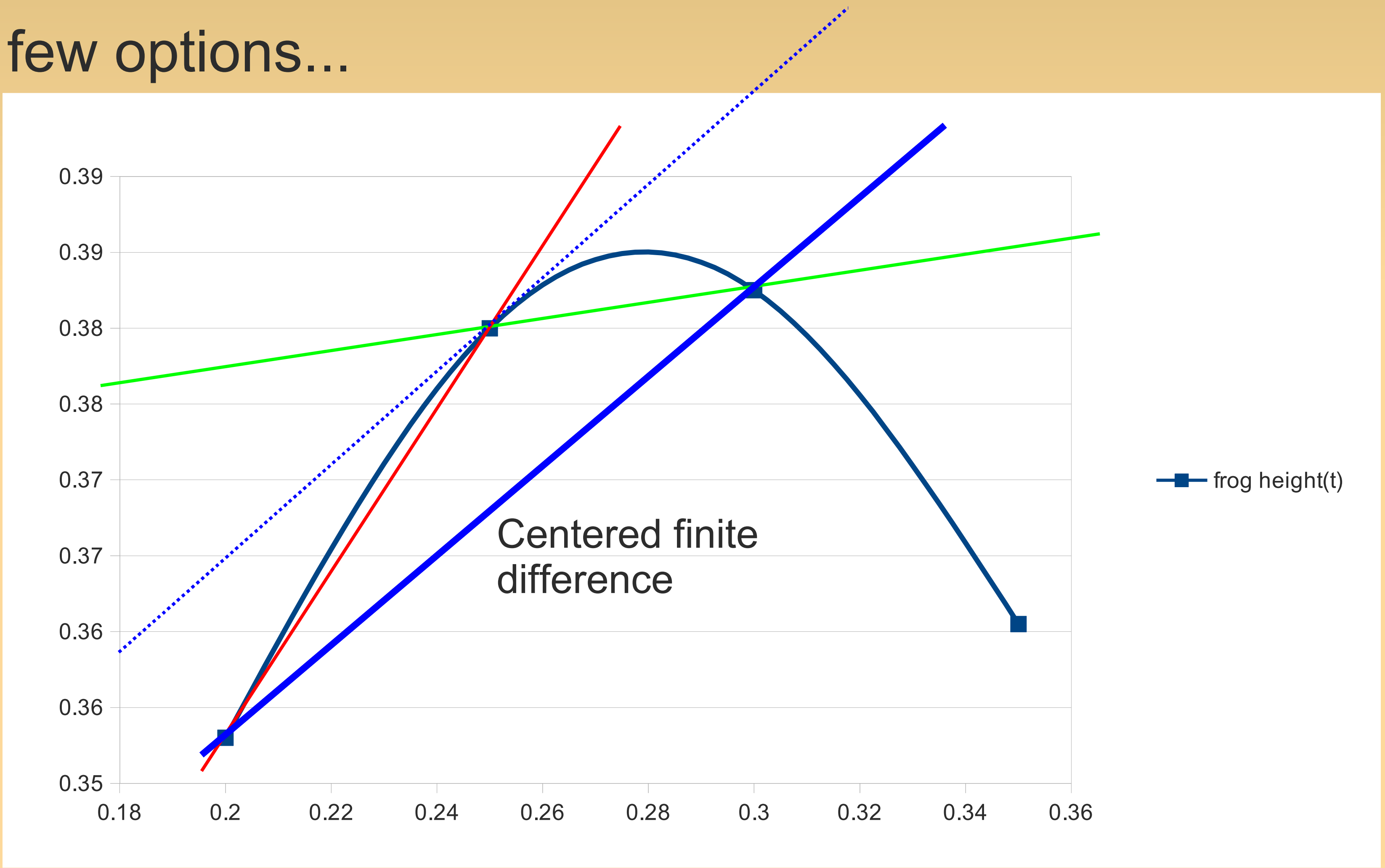
Numerical Differentiation

- Determine the vertical speed at t=0.25...
  - a few options...



Numerical Differentiation

- Determine the vertical speed at t=0.25...
  - a few options...



Numerical Integration

- Integration: the reversed problem...
- Suppose we travel in a car with a broken odometer
- Speedometer is working...

Numerical Integration

- maintain speeds, to figure out traveled distance

t	v(t) km/h
0	80
30	120
65	128
120	122
728	120
733	0
798	20
836	20
941	70
970	120
1350	123
1404	90

enter highway ramp

traffic jam

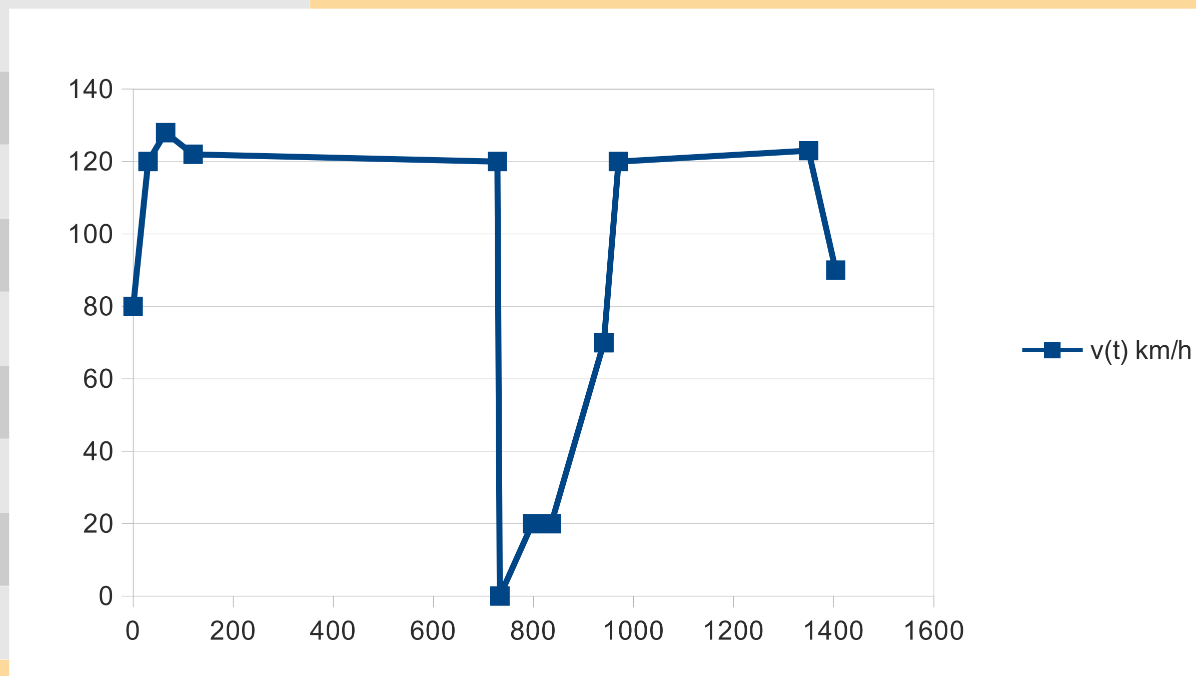
exit highway ramp

Numerical Integration

- maintain speeds, to figure out traveled distance

t	v(t) km/h
0	80
30	120
65	128
120	
728	
733	
798	
836	
941	
970	
1350	
1404	

enter highway ramp



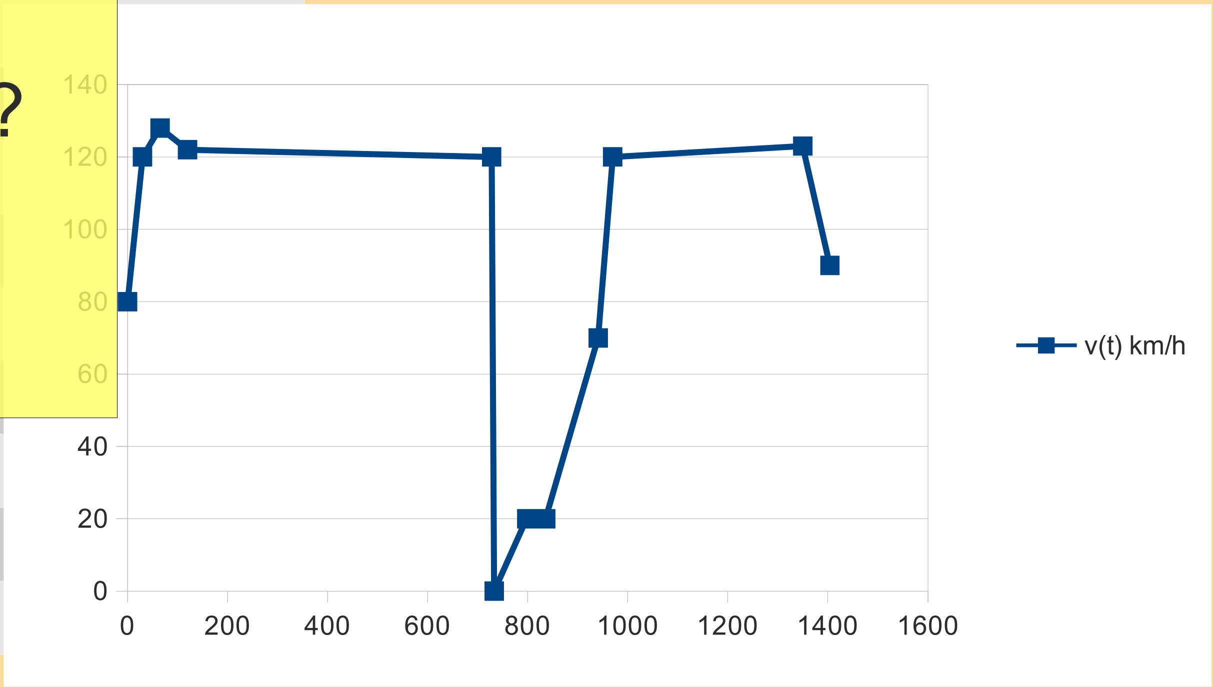
Numerical Integration

- maintain speeds, to figure out traveled distance

t	v(t) km/h
0	80
30	120
65	128
120	
[illegible]	
733	
798	
836	
941	
970	
1350	
1404	

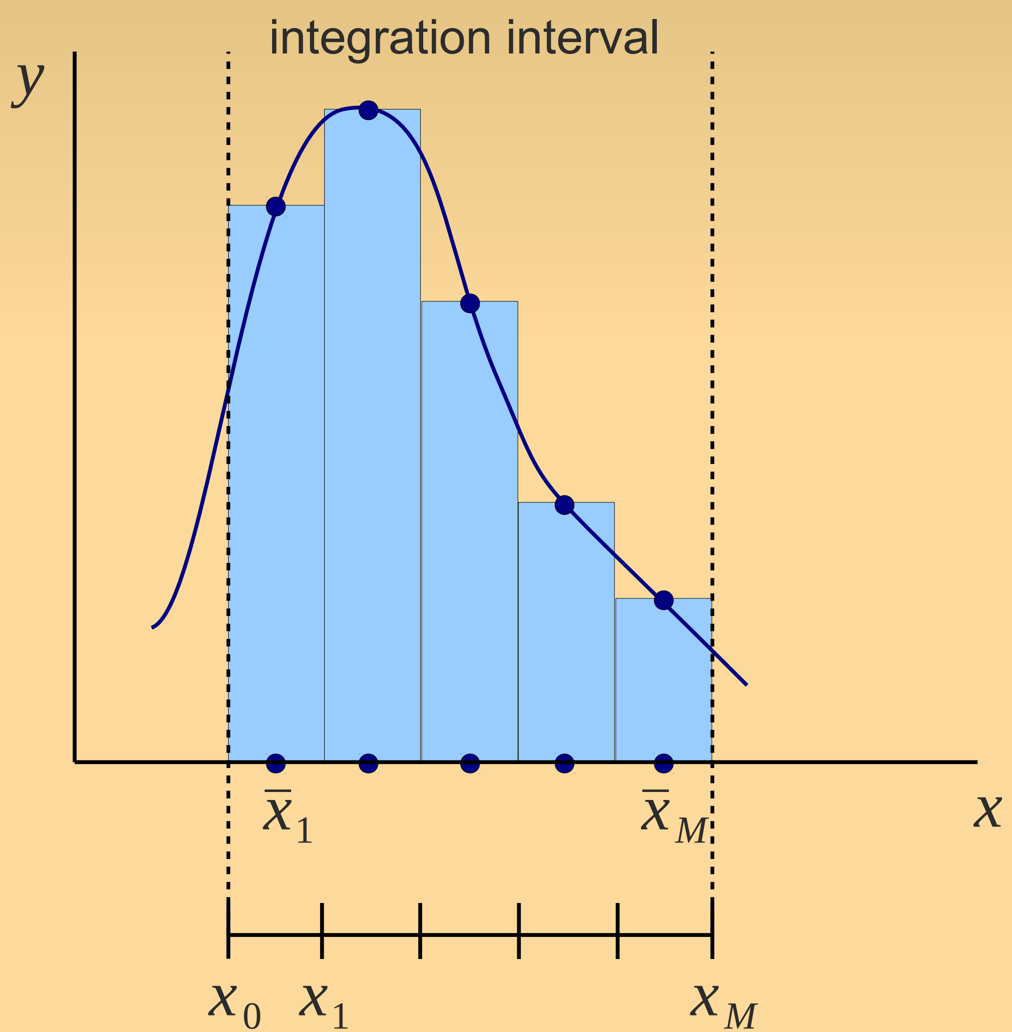
enter highway ramp

How far did we travel?

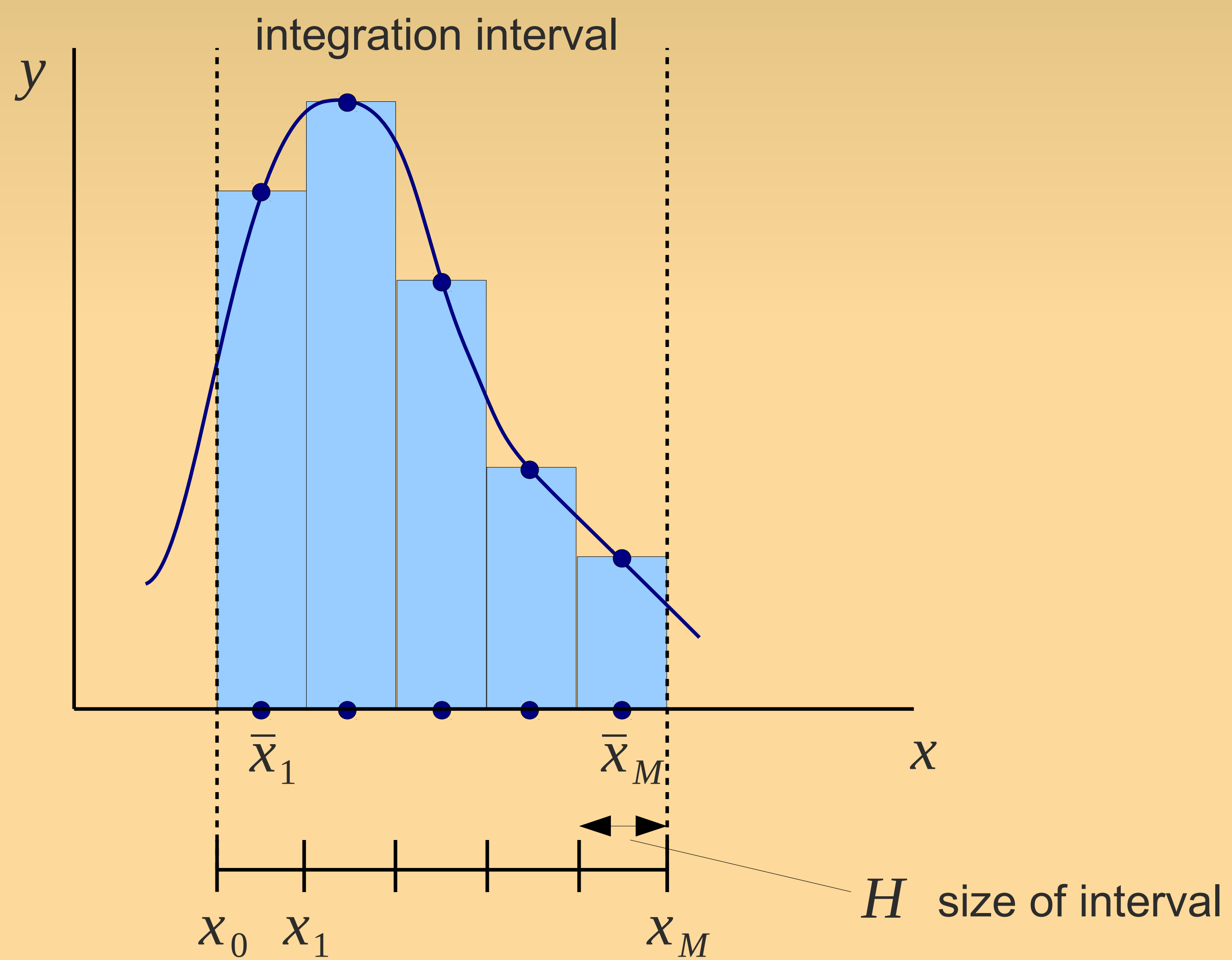


Midpoint Formula

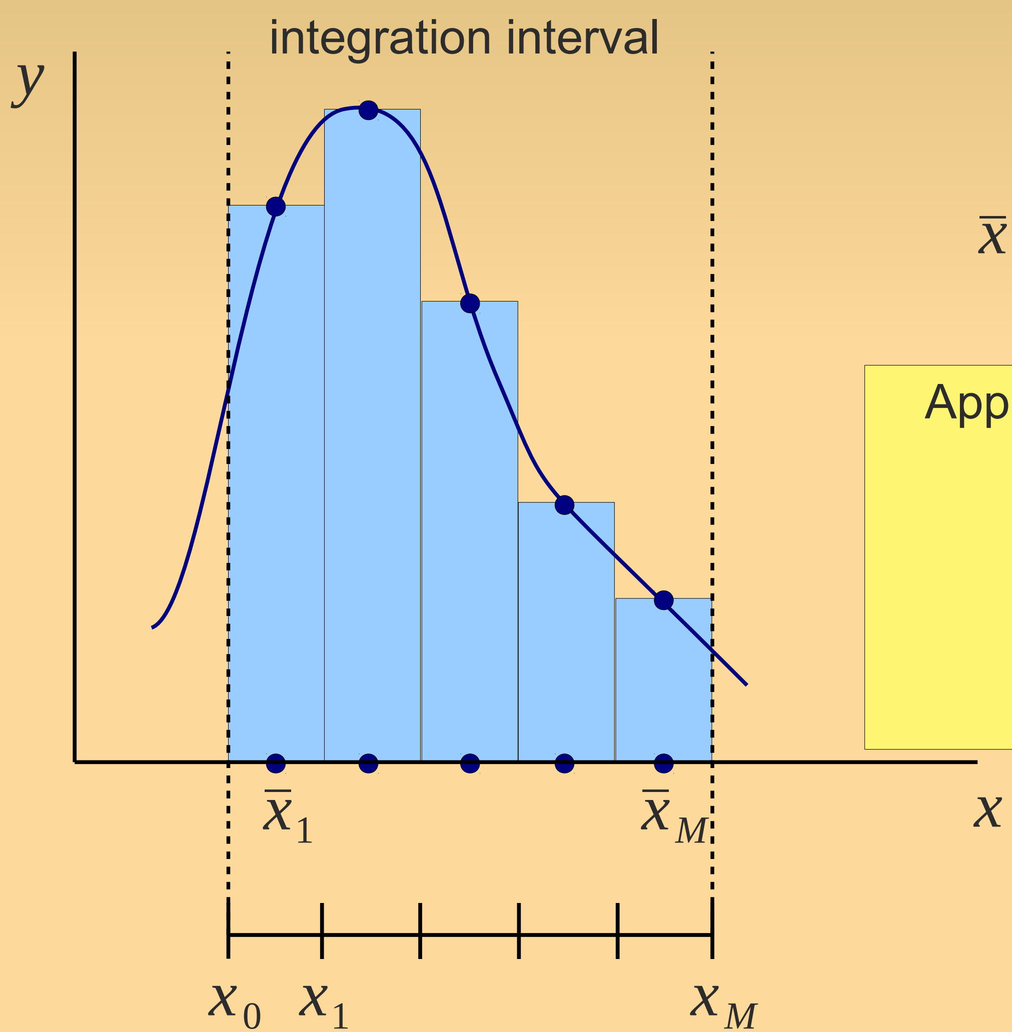
- Approximate the integral with a finite sum



Midpoint Formula



Midpoint Formula

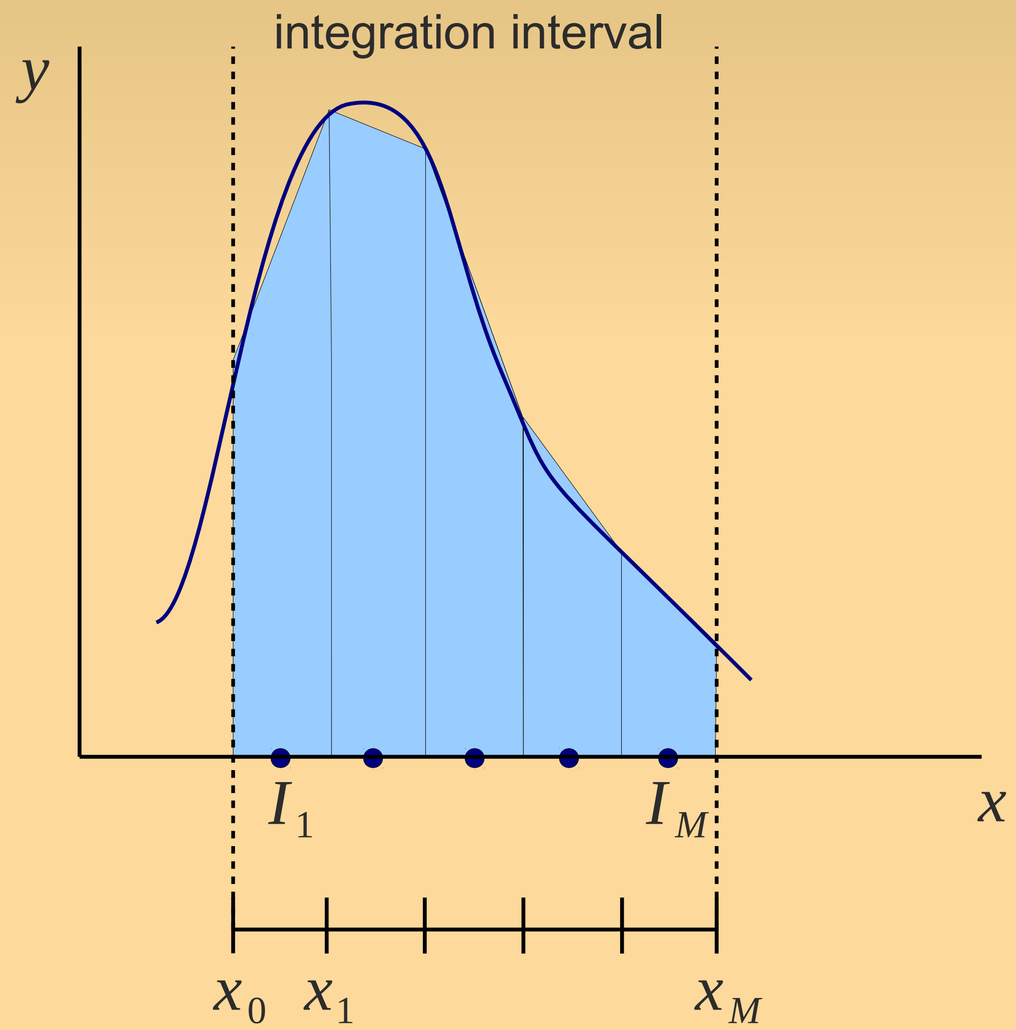


$$\bar{x}_k = \frac{x_{k-1} + x_k}{2}$$

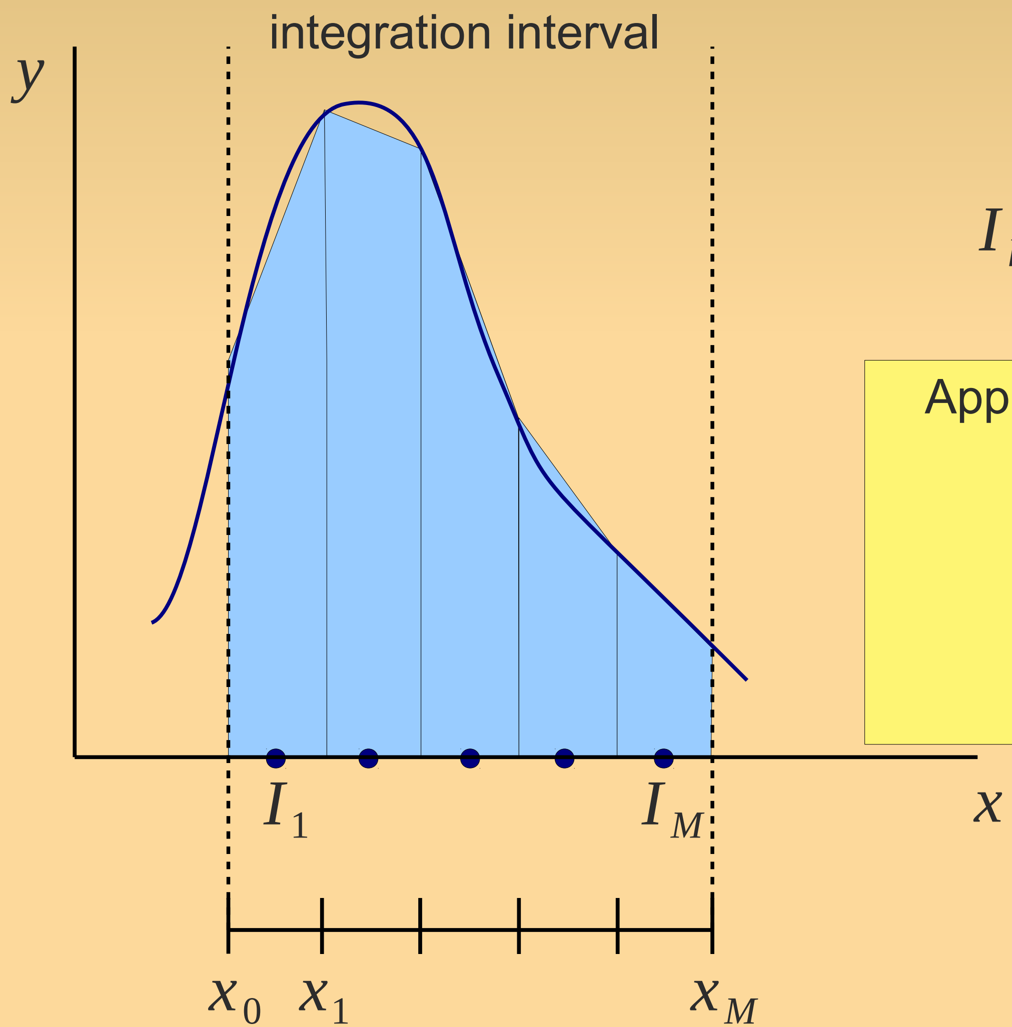
Approximation of the integral:

$$I_{MP}(f) = H \sum_{k=1}^{M} f(\bar{x}_k)$$

Trapezoid Formula



Trapezoid Formula



$$I_k = H \frac{f(x_{k-1}) + f(x_k)}{2}$$

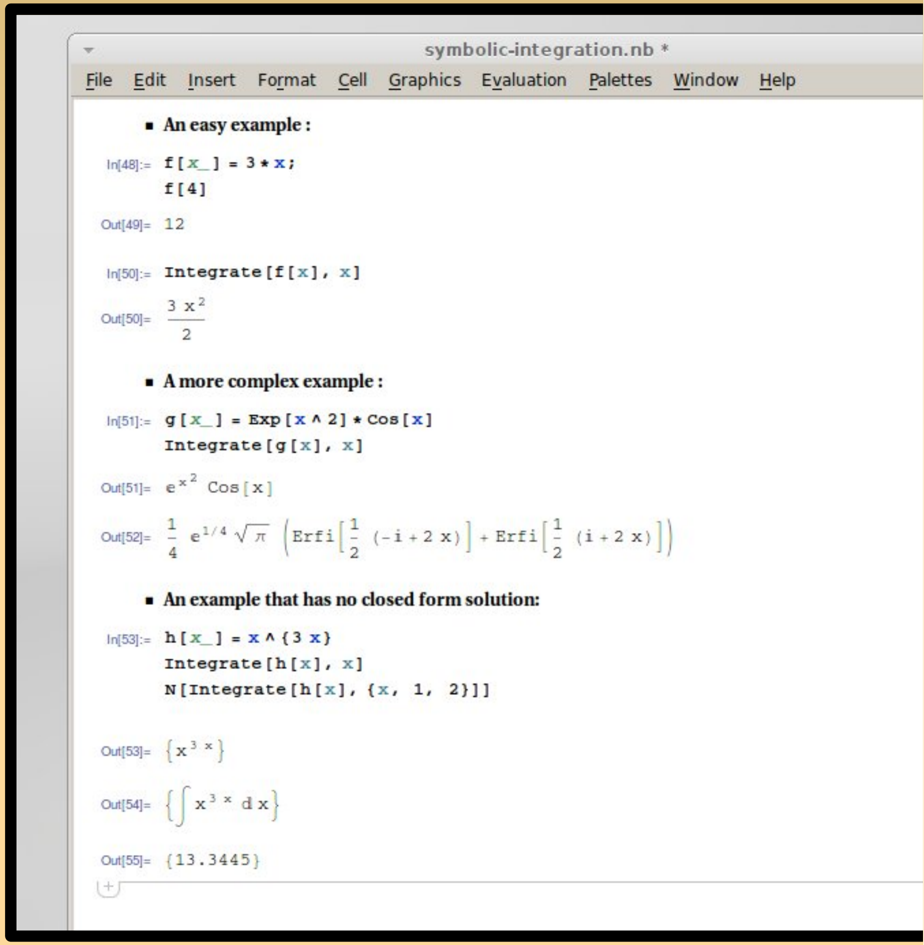
Approximation of the integral:

$$I_{MP}(f) = \sum_{k=1}^{M} I_k$$

Symbolic Integration

- Finally: when faced with a difficult integral...

  → try 'symbolic' packages!



symbolic-integration.nb *

File Edit Insert Format Cell Graphics Evaluation Palettes Window Help

- **An easy example :**

In[48]:=
```
f[x_] = 3 * x;
f[4]
```

Out[49]= 12

In[50]:=
```
Integrate[f[x], x]
```

Out[50]= $\frac{3\,x^2}{2}$

- **A more complex example :**

In[51]:=
```
g[x_] = Exp[x ^ 2] * Cos[x]
Integrate[g[x], x]
```

Out[51]= $e^{x^2}\,\mathrm{Cos}[x]$

Out[52]= $\frac{1}{4}\,e^{1/4}\,\sqrt{\pi}\,\left(\mathrm{Erfi}\left[\frac{1}{2}\,(-i+2\,x)\right]+\mathrm{Erfi}\left[\frac{1}{2}\,(i+2\,x)\right]\right)$

- **An example that has no closed form solution:**

In[53]:=
```
h[x_] = x ^ {3 x}
Integrate[h[x], x]
N[Integrate[h[x], {x, 1, 2}]]
```

Out[53]= $\{x^{3\,x}\}$

Out[54]= $\left\{\int x^{3\,x}\,dx\right\}$

Out[55]= {13.3445}