The numerical solution of Newton's equations relies on breaking up the motion into small time steps, $\Delta t$. For example, if you know $x(t=0)$ and $v(t=0)$, you can get them at $t=\Delta t$ by first finding $x(t=0.01\Delta t)$, $v(t=0.01\Delta t)$, then at $0.02\Delta t$, and so on.

The solution will always be approximate.

Tools to find a prescription that takes $x(t)$, $v(t)$ to $x(t_0)$, $v(t_0)$ include calculus and the Taylor series expansion.

Solving Newton's equation by stepping in time

Direct use of 2nd deriv of position

$$\ddot{x}(t) = \frac{F(x(t), t)}{m} = a(x(t), t)$$

1. Use the Taylor series expansion to show

$$\frac{x(t+\Delta t) - x(t) - x(t-\Delta t)}{2\Delta t^2} = \dot{x}(0) + O(\Delta t^2)$$

2. Use Taylor series to show

$$\frac{x(t+2\Delta t) - 2x(t) + x(t-2\Delta t)}{4\Delta t^2} = \ddot{x}(t) + O(\Delta t^4)$$

Algorithm: Use Newton's equation to get an expression for $\ddot{x}(t)$

3. Show

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) + \Delta t^2 a(x(t), t) + O(\Delta t^4)$$

Algorithm: start with $x(0)$ and $x(-\Delta t)$

use $x(0)$ to compute $a(x(0), 0)$

compute $x(\Delta t) = 2x(0) - x(-\Delta t) + \Delta t^2 a(x(0), 0)$

use $x(\Delta t)$ to compute $a(x(\Delta t), \Delta t)$

compute $x(2\Delta t) = 2x(\Delta t) - x(0) + \Delta t^2 a(x(\Delta t), \Delta t)$

Advantage - high order accuracy for little work
Disadvantage - some forces depend on $v(t)$, fixed time step need $x(0), x(-\Delta t)$ but usually have $x(0), v(0)$
Show Newton's equation is the same as
\[ \dot{x}_1(t) = x_1(t) \quad \text{and} \quad \dot{x}_2(t) = a(x_1(t), t) \]
where \( x_1(t) \equiv X(t) \) and \( x_2(t) \equiv V(t) \)

Show \( \frac{f(\delta t)}{\delta t} = f(0) + O(\delta t) \)

Show \( \frac{f(t+\delta t)}{\delta t} = f(t) + O(\delta t) \)

Euler's method

This approximation can be used to find a new algorithm

\[
\begin{align*}
X_1(t+\delta t) &= X_1(t) + X_2(t) \delta t + O(\delta t^2) \\
X_2(t+\delta t) &= X_2(t) + a(x_1(t), t) \delta t + O(\delta t^2)
\end{align*}
\]

Algorithm: Start with \( x_1(0) = X(0) \) and \( x_2(0) = V(0) \)

Use \( x_1(0) \) to compute \( a(x_1(0), 0) \)
Compute \( X_1(\delta t) = x_1(0) + x_2(0) \delta t \)
Compute \( X_2(\delta t) = x_2(0) + a(x_1(0), 0) \delta t \)

Use \( X_1(\delta t) \) to compute \( a(x_1(\delta t), \delta t) \)
Compute \( X_1(2\delta t) = X_1(\delta t) + X_2(\delta t) \delta t \)
Compute \( X_2(2\delta t) = X_2(\delta t) + a(x_1(\delta t), \delta t) \delta t \)

Advantage - will work if \( a \) also depends on \( V \)
Starts with \( X(0), V(0) \) - only uses 1st derivative
Disadvantages - low order method, energy will drift

You want to compute \( x(t_f), V(t_f) \) [\( t_f \) is the final time]
given an initial \( t_0 \).

Show that the accuracy is proportional to \( 1/N \) where
\( N \) is the number of time steps. (Hint: how does the accuracy
of 1 time step depend on \( N \).)
2nd Order Runge-Kutta

A method with higher order accuracy uses two calculations per time step.

\[ X(t) = F(X(t), t) \]

1. Suppose you have \[ X(t) = F(X(t), t) \]

2. Show that the following sequence has the stated accuracy

\[
\begin{align*}
X(t + \Delta t) &= X(t) + \Delta t \cdot F(X(t), t) + O(\Delta t^2) \\
X(t) &= X(t) + \frac{\Delta t}{2} \cdot F(X(t), t) + O(\Delta t^3)
\end{align*}
\]

Algorithm: Start \[ X_1(0) = X_0(0) \] and \[ X_2(0) = V(0) \]

use \[ X_1(0) \] to compute \[ a(X_1(0), 0) \]
compute \[ X_{\text{temp}} = X_1(0) + \frac{\Delta t}{2} \cdot a(X_1(0), 0) \]
use \[ X_{\text{temp}} \] to compute \[ a(X_{\text{temp}}, \frac{\Delta t}{2}) \]
compute \[ X_1(\Delta t) = X_1(0) + \Delta t \cdot a(X_{\text{temp}}, \frac{\Delta t}{2}) \]
compute \[ X_2(\Delta t) = X_2(0) + \Delta t \cdot a(X_{\text{temp}}, \frac{\Delta t}{2}) \]

3. Show that the accuracy of \[ X(t), V(t) \] is proportional to \( 1/N^2 \) where \( N \) is the number of time steps.