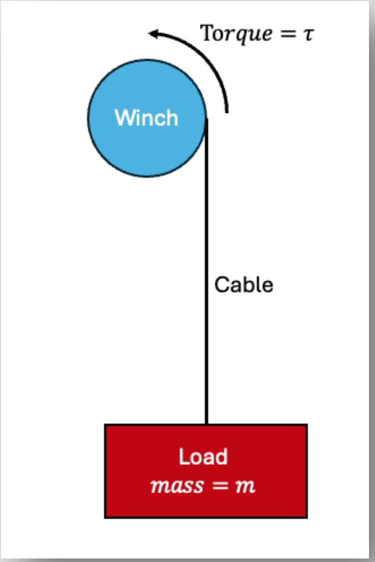
Simulating Continuous Systems Part 3

CMSC 326 Simulations

Simulating Continuous Systems

Ricker Model of Population Growth

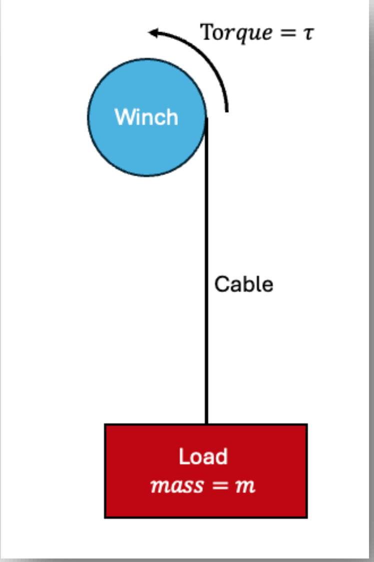
Rotational Motion



Simulating Continuous Systems

Ricker Model of Population Growth

Rotational Motion



Ricker Model of Population Growth

Time is modeled by discrete time steps:

$$\mathrm{x}(n) = \text{population at step } n$$

Ricker Model of Population Growth

The Ricker model is:

$$\mathrm{x}(n+1) = r\, x(n)\, e^{-x(n)}$$

Ricker Model of Population Growth

$$\underbrace{\mathrm{x}(n+1)}_{\text{The population time step } n+1} = r\, x(n)\, e^{-x(n)}$$

Ricker Model of Population Growth

$$\mathrm{x}(n+1) = \underbrace{r\, x(n)\, e^{-x(n)}}_{\text{A calculation including the current population } x(n)}$$

Ricker Model of Population Growth

$$\mathrm{x}(n+1) = \underbrace{r}_{\text{Growth rate constant}}\, x(n)\, e^{-x(n)}$$

Ricker Model of Population Growth

$$\mathrm{x}(n+1) = \underbrace{r\,x(n)}\,e^{-x(n)}$$

The **population growth** part of the model

Ricker Model of Population Growth

$$\mathrm{x}(n+1) = r\,x(n)\,\underbrace{e^{-x(n)}}_{\text{The }\textbf{population decline}\text{ part of the model}}$$

Ricker Model of Population Growth

One might expect the sequence

$$\mathrm{x}(1), x(2), x(3), \ldots$$

to converge to a “stable” population

called an attractor

Ricker Model of Population Growth

For each r, it seems plausible that we will get a different stable population for large enough n

$$\mathrm{x}(n+1) = r\, x(n)\, e^{-x(n)}$$

Ricker Model of Population Growth

```python
# Population grown factor
r = 2

# Initialize population x(0)
x = 1

for n in range(500):

    # Compute the Ricker model formula
    x = r * x * np.exp(-x)

    print(f"n = {n},  r = {r},  x = {x:0.4f}")
```

Ricker Model of Population Growth

```python
# Population grown factor
r = 2

# Initialize population x(0)
x = 1

for n in range(500):

    # Compute the Ricker model formula
    x = r * x * np.exp(-x)

    print(f"n = {n},  r = {r},  x = {x:0.4f}")
```

Ricker Model of Population Growth

```
# Population grown factor
r = 2

# Initialize population x(0)
x = 1

for n in range(500):

    # Compute the Ricker model formula
    x = r * x * np.exp(-x)

    print(f"n = {n},  r = {r},  x = {x:0.4f}")
```

Ricker Model of Population Growth

```python
# Population grown factor
r = 2

# Initialize population x(0)
x = 1

for n in range(500):

    # Compute the Ricker model formula
    x = r * x * np.exp(-x)

    print(f"n = {n},  r = {r},  x = {x:0.4f}")
```

Ricker Model of Population Growth

```
# Population grown factor
r = 2

# Initialize population x(0)
x = 1

for n in range(500):

    # Compute the Ricker model formula
    x = r * x * np.exp(-x)

    print(f"n = {n},  r = {r},  x = {x:0.4f}")
```

Ricker Model of Population Growth

```
# Population grown factor
r = 2

# Initialize population x(0)
x = 1

for n in range(500):

    # Compute the Ricker model formula
    x = r * x * np.exp(-x)

    print(f"n = {n},  r = {r},  x = {x:0.4f}")
```

Ricker Model of Population Growth

```
n = 490,  r = 2,  x = 0.6931
n = 491,  r = 2,  x = 0.6931
n = 492,  r = 2,  x = 0.6931
n = 493,  r = 2,  x = 0.6931
n = 494,  r = 2,  x = 0.6931
n = 495,  r = 2,  x = 0.6931
n = 496,  r = 2,  x = 0.6931
n = 497,  r = 2,  x = 0.6931
n = 498,  r = 2,  x = 0.6931
n = 499,  r = 2,  x = 0.6931
n = 500,  r = 2,  x = 0.6931
```

Ricker Model of Population Growth

```
n = 490,  r = 2,  x = 0.6931
n = 491,  r = 2,  x = 0.6931
n = 492,  r = 2,  x = 0.6931
n = 493,  r = 2,  x = 0.6931
n = 494,  r = 2,  x = 0.6931
n = 495,  r = 2,  x = 0.6931
n = 496,  r = 2,  x = 0.6931
n = 497,  r = 2,  x = 0.6931
n = 498,  r = 2,  x = 0.6931
n = 499,  r = 2,  x = 0.6931
n = 500,  r = 2,  x = 0.6931
```

Ricker Model of Population Growth

```
n = 490,  r = 2,  x = 0.6931
n = 491,  r = 2,  x = 0.6931
n = 492,  r = 2,  x = 0.6931
n = 493,  r = 2,  x = 0.6931
n = 494,  r = 2,  x = 0.6931
n = 495,  r = 2,  x = 0.6931
n = 496,  r = 2,  x = 0.6931
n = 497,  r = 2,  x = 0.6931
n = 498,  r = 2,  x = 0.6931
n = 499,  r = 2,  x = 0.6931
n = 500,  r = 2,  x = 0.6931
```

Ricker Model of Population Growth

```
n = 490,  r = 2,  x = 0.6931
n = 491,  r = 2,  x = 0.6931
n = 492,  r = 2,  x = 0.6931
n = 493,  r = 2,  x = 0.6931
n = 494,  r = 2,  x = 0.6931
n = 495,  r = 2,  x = 0.6931
n = 496,  r = 2,  x = 0.6931
n = 497,  r = 2,  x = 0.6931
n = 498,  r = 2,  x = 0.6931
n = 499,  r = 2,  x = 0.6931
n = 500,  r = 2,  x = 0.6931
```

Ricker Model of Population Growth

```
n = 490,  r = 4,  x = 1.3863
n = 491,  r = 4,  x = 1.3863
n = 492,  r = 4,  x = 1.3863
n = 493,  r = 4,  x = 1.3863
n = 494,  r = 4,  x = 1.3863
n = 495,  r = 4,  x = 1.3863
n = 496,  r = 4,  x = 1.3863
n = 497,  r = 4,  x = 1.3863
n = 498,  r = 4,  x = 1.3863
n = 499,  r = 4,  x = 1.3863
n = 500,  r = 4,  x = 1.3863
```

Ricker Model of Population Growth

```
n = 490,   r = 4,   x = 1.3863
n = 491,   r = 4,   x = 1.3863
n = 492,   r = 4,   x = 1.3863
n = 493,   r = 4,   x = 1.3863
n = 494,   r = 4,   x = 1.3863
n = 495,   r = 4,   x = 1.3863
n = 496,   r = 4,   x = 1.3863
n = 497,   r = 4,   x = 1.3863
n = 498,   r = 4,   x = 1.3863
n = 499,   r = 4,   x = 1.3863
n = 500,   r = 4,   x = 1.3863
```

Ricker Model of Population Growth

```
n = 490,  r = 8,  x = 2.7726
n = 491,  r = 8,  x = 1.3863
n = 492,  r = 8,  x = 2.7726
n = 493,  r = 8,  x = 1.3863
n = 494,  r = 8,  x = 2.7726
n = 495,  r = 8,  x = 1.3863
n = 496,  r = 8,  x = 2.7726
n = 497,  r = 8,  x = 1.3863
n = 498,  r = 8,  x = 2.7726
n = 499,  r = 8,  x = 1.3863
n = 500,  r = 8,  x = 2.7726
```

Ricker Model of Population Growth

```
n = 490,   r = 8,  x = 2.7726
n = 491,   r = 8,  x = 1.3863
n = 492,   r = 8,  x = 2.7726
n = 493,   r = 8,  x = 1.3863
n = 494,   r = 8,  x = 2.7726
n = 495,   r = 8,  x = 1.3863
n = 496,   r = 8,  x = 2.7726
n = 497,   r = 8,  x = 1.3863
n = 498,   r = 8,  x = 2.7726
n = 499,   r = 8,  x = 1.3863
n = 500,   r = 8,  x = 2.7726
```

Ricker Model of Population Growth

```
n = 490,   r = 8,   x = 2.7726
n = 491,   r = 8,   x = 1.3863
n = 492,   r = 8,   x = 2.7726
n = 493,   r = 8,   x = 1.3863
n = 494,   r = 8,   x = 2.7726
n = 495,   r = 8,   x = 1.3863
n = 496,   r = 8,   x = 2.7726
n = 497,   r = 8,   x = 1.3863
n = 498,   r = 8,   x = 2.7726
n = 499,   r = 8,   x = 1.3863
n = 500,   r = 8,   x = 2.7726
```

Ricker Model of Population Growth

```
n = 490,   r = 8,   x = 2.7726
n = 491,   r = 8,   x = 1.3863
n = 492,   r = 8,   x = 2.7726
n = 493,   r = 8,   x = 1.3863
n = 494,   r = 8,   x = 2.7726
n = 495,   r = 8,   x = 1.3863
n = 496,   r = 8,   x = 2.7726
n = 497,   r = 8,   x = 1.3863
n = 498,   r = 8,   x = 2.7726
n = 499,   r = 8,   x = 1.3863
n = 500,   r = 8,   x = 2.7726
```

Ricker Model of Population Growth

```
n = 490,   r = 8,   x = 2.7726
n = 491,   r = 8,   x = 1.3863
n = 492,   r = 8,   x = 2.7726
n = 493,   r = 8,   x = 1.3863
n = 494,   r = 8,   x = 2.7726
n = 495,   r = 8,   x = 1.3863
n = 496,   r = 8,   x = 2.7726
n = 497,   r = 8,   x = 1.3863
n = 498,   r = 8,   x = 2.7726
n = 499,   r = 8,   x = 1.3863
n = 500,   r = 8,   x = 2.7726
```

Ricker Model of Population Growth

```
n = 490,   r = 8,   x = 2.7726
n = 491,   r = 8,   x = 1.3863
n = 492,   r = 8,   x = 2.7726
n = 493,   r = 8,   x = 1.3863
n = 494,   r = 8,   x = 2.7726
n = 495,   r = 8,   x = 1.3863
n = 496,   r = 8,   x = 2.7726
n = 497,   r = 8,   x = 1.3863
n = 498,   r = 8,   x = 2.7726
n = 499,   r = 8,   x = 1.3863
n = 500,   r = 8,   x = 2.7726
```

Ricker Model of Population Growth

```
n = 490,   r = 8,   x = 2.7726
n = 491,   r = 8,   x = 1.3863
n = 492,   r = 8,   x = 2.7726
n = 493,   r = 8,   x = 1.3863
n = 494,   r = 8,   x = 2.7726
n = 495,   r = 8,   x = 1.3863
n = 496,   r = 8,   x = 2.7726
n = 497,   r = 8,   x = 1.3863
n = 498,   r = 8,   x = 2.7726
n = 499,   r = 8,   x = 1.3863
n = 500,   r = 8,   x = 2.7726
```

Ricker Model of Population Growth

```
n = 490,  r = 8,  x = 2.7726
n = 491,  r = 8,  x = 1.3863
n = 492,  r = 8,  x = 2.7726
n = 493,  r = 8,  x = 1.3863
n = 494,  r = 8,  x = 2.7726
n = 495,  r = 8,  x = 1.3863
n = 496,  r = 8,  x = 2.7726
n = 497,  r = 8,  x = 1.3863
n = 498,  r = 8,  x = 2.7726
n = 499,  r = 8,  x = 1.3863
n = 500,  r = 8,  x = 2.7726
```

Ricker Model of Population Growth

```
n = 489,  r = 14,  x = 0.4270
n = 490,  r = 14,  x = 3.9005
n = 491,  r = 14,  x = 1.1048
n = 492,  r = 14,  x = 5.1239
n = 493,  r = 14,  x = 0.4270
n = 494,  r = 14,  x = 3.9005
n = 495,  r = 14,  x = 1.1048
n = 496,  r = 14,  x = 5.1239
n = 497,  r = 14,  x = 0.4270
n = 498,  r = 14,  x = 3.9005
n = 499,  r = 14,  x = 1.1048
n = 500,  r = 14,  x = 5.1239
```

Ricker Model of Population Growth

```
n = 489,   r = 14,   x = 0.4270
n = 490,   r = 14,   x = 3.9005
n = 491,   r = 14,   x = 1.1048
n = 492,   r = 14,   x = 5.1239
n = 493,   r = 14,   x = 0.4270
n = 494,   r = 14,   x = 3.9005
n = 495,   r = 14,   x = 1.1048
n = 496,   r = 14,   x = 5.1239
n = 497,   r = 14,   x = 0.4270
n = 498,   r = 14,   x = 3.9005
n = 499,   r = 14,   x = 1.1048
n = 500,   r = 14,   x = 5.1239
```

Ricker Model of Population Growth

```
n = 489,   r = 14,   x = 0.4270
n = 490,   r = 14,   x = 3.9005
n = 491,   r = 14,   x = 1.1048
n = 492,   r = 14,   x = 5.1239
n = 493,   r = 14,   x = 0.4270
n = 494,   r = 14,   x = 3.9005
n = 495,   r = 14,   x = 1.1048
n = 496,   r = 14,   x = 5.1239
n = 497,   r = 14,   x = 0.4270
n = 498,   r = 14,   x = 3.9005
n = 499,   r = 14,   x = 1.1048
n = 500,   r = 14,   x = 5.1239
```

Ricker Model of Population Growth

```
n = 489,   r = 14,   x = 0.4270
n = 490,   r = 14,   x = 3.9005
n = 491,   r = 14,   x = 1.1048
n = 492,   r = 14,   x = 5.1239
n = 493,   r = 14,   x = 0.4270
n = 494,   r = 14,   x = 3.9005
n = 495,   r = 14,   x = 1.1048
n = 496,   r = 14,   x = 5.1239
n = 497,   r = 14,   x = 0.4270
n = 498,   r = 14,   x = 3.9005
n = 499,   r = 14,   x = 1.1048
n = 500,   r = 14,   x = 5.1239
```

Ricker Model of Population Growth

```
n = 489,   r = 14,   x = 0.4270
n = 490,   r = 14,   x = 3.9005
n = 491,   r = 14,   x = 1.1048
n = 492,   r = 14,   x = 5.1239
n = 493,   r = 14,   x = 0.4270
n = 494,   r = 14,   x = 3.9005
n = 495,   r = 14,   x = 1.1048
n = 496,   r = 14,   x = 5.1239
n = 497,   r = 14,   x = 0.4270
n = 498,   r = 14,   x = 3.9005
n = 499,   r = 14,   x = 1.1048
n = 500,   r = 14,   x = 5.1239
```

Ricker Model of Population Growth

```
n = 489,  r = 14,  x = 0.4270
n = 490,  r = 14,  x = 3.9005
n = 491,  r = 14,  x = 1.1048
n = 492,  r = 14,  x = 5.1239
n = 493,  r = 14,  x = 0.4270
n = 494,  r = 14,  x = 3.9005
n = 495,  r = 14,  x = 1.1048
n = 496,  r = 14,  x = 5.1239
n = 497,  r = 14,  x = 0.4270
n = 498,  r = 14,  x = 3.9005
n = 499,  r = 14,  x = 1.1048
n = 500,  r = 14,  x = 5.1239
```

Ricker Model of Population Growth

```
n = 489,  r = 14,  x = 0.4270
n = 490,  r = 14,  x = 3.9005
n = 491,  r = 14,  x = 1.1048
n = 492,  r = 14,  x = 5.1239
n = 493,  r = 14,  x = 0.4270
n = 494,  r = 14,  x = 3.9005
n = 495,  r = 14,  x = 1.1048
n = 496,  r = 14,  x = 5.1239
n = 497,  r = 14,  x = 0.4270
n = 498,  r = 14,  x = 3.9005
n = 499,  r = 14,  x = 1.1048
n = 500,  r = 14,  x = 5.1239
```

Ricker Model of Population Growth

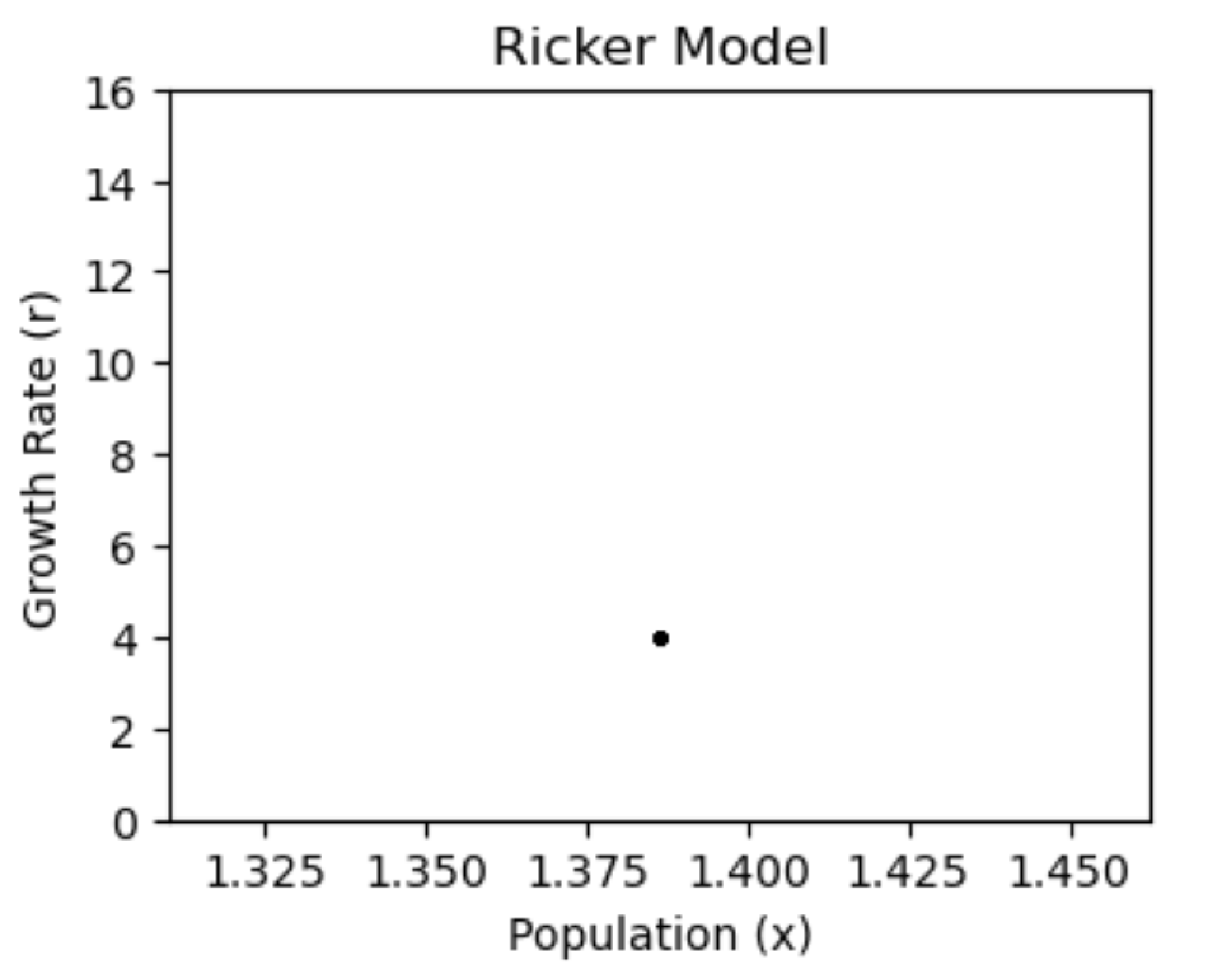
```
n = 489,   r = 14,   x = 0.4270
n = 490,   r = 14,   x = 3.9005
n = 491,   r = 14,   x = 1.1048
n = 492,   r = 14,   x = 5.1239
n = 493,   r = 14,   x = 0.4270
n = 494,   r = 14,   x = 3.9005
n = 495,   r = 14,   x = 1.1048
n = 496,   r = 14,   x = 5.1239
n = 497,   r = 14,   x = 0.4270
n = 498,   r = 14,   x = 3.9005
n = 499,   r = 14,   x = 1.1048
n = 500,   r = 14,   x = 5.1239
```

Ricker Model of Population Growth

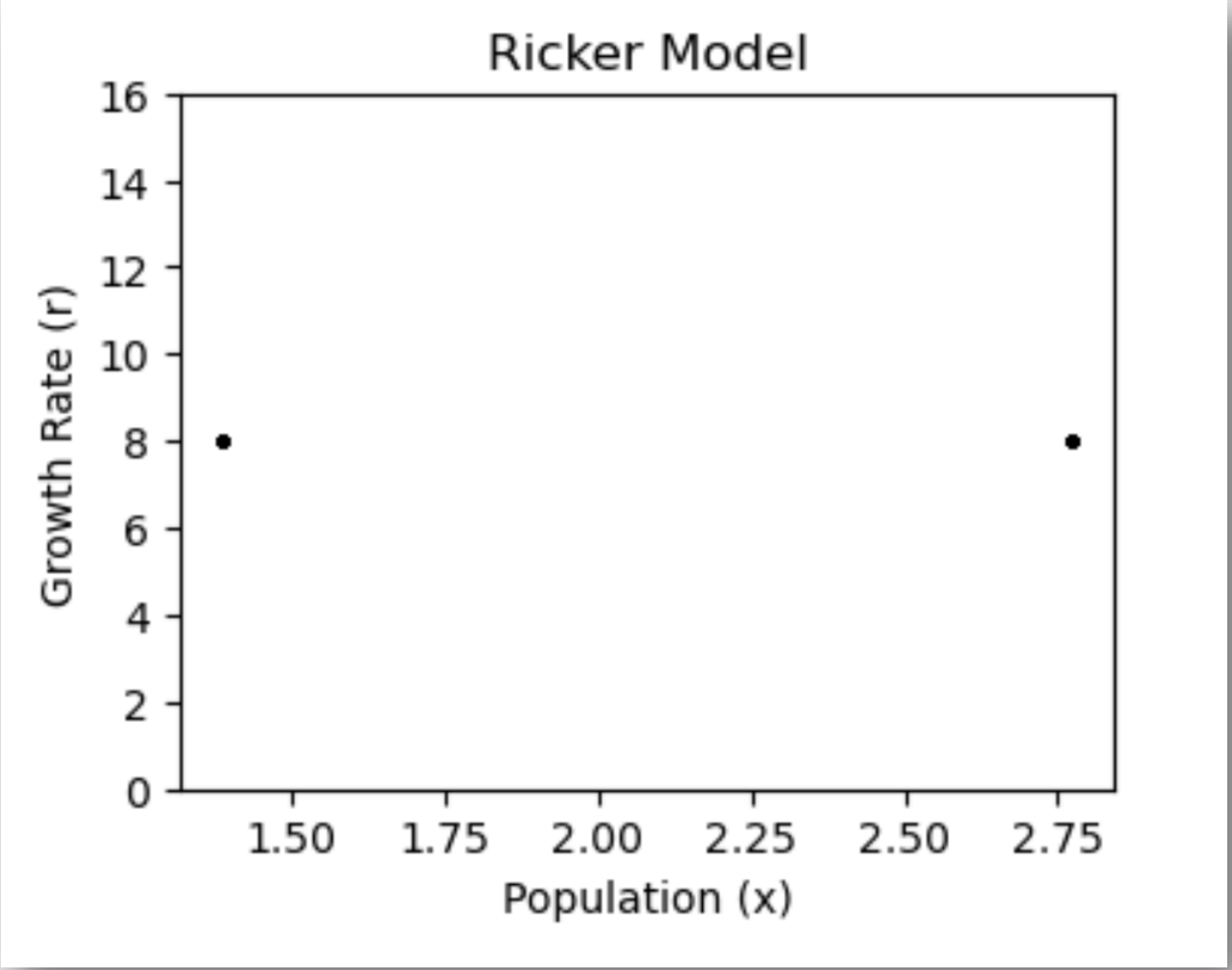
What we'll do:

1. For a given r, run the sequence and compute a histogram of all the $\mathrm{x}(\mathrm{n})$ values obtained
2. If the sequence does in fact stabilize, we should see a single convergence point show up in the histogram
3. Plot any convergence point that shows up more than once, after ignoring $x(1), \ldots, x(50)$
(to allow convergence)

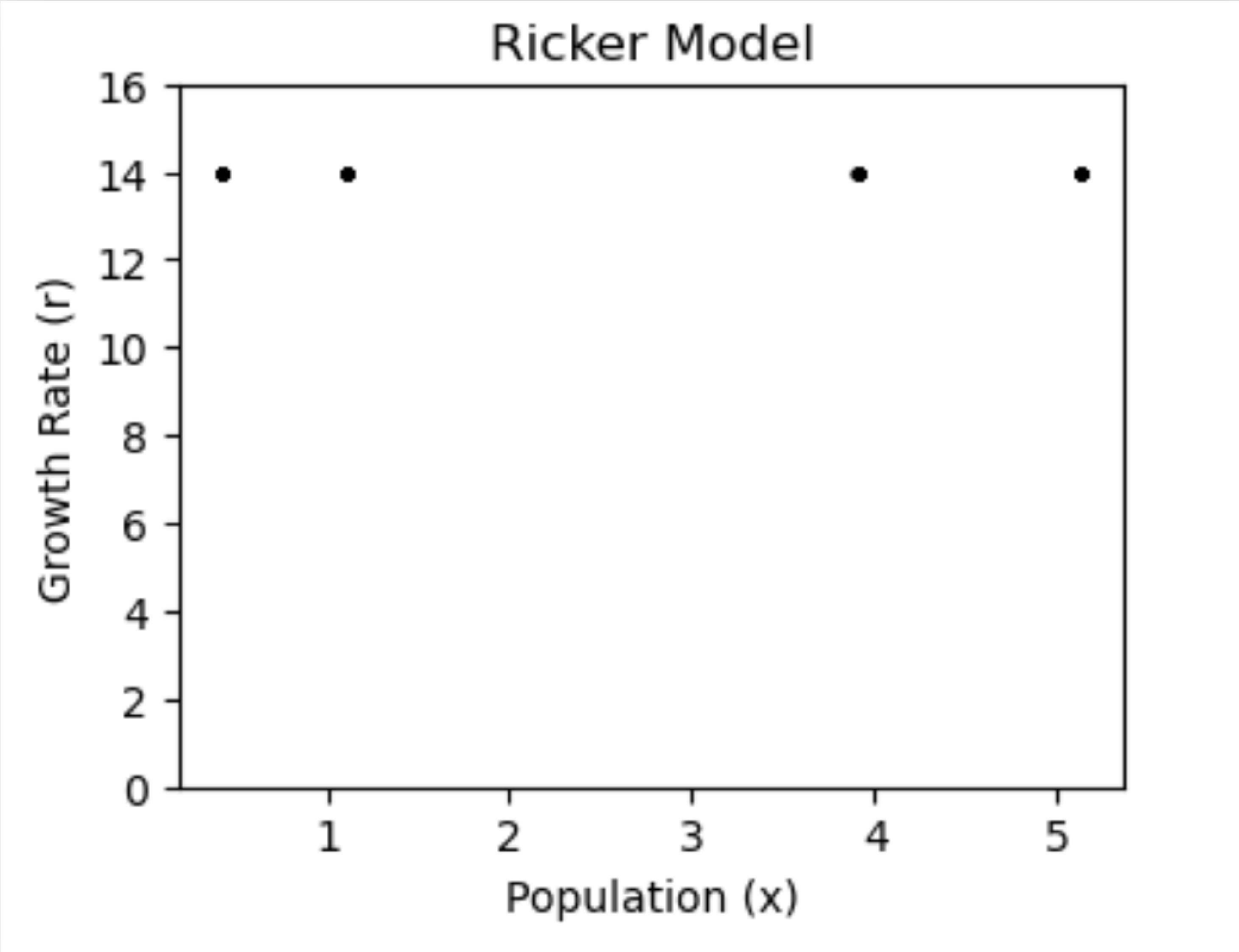
Ricker Model of Population Growth



Ricker Model of Population Growth



Ricker Model of Population Growth



Ricker Model of Population Growth

Thus, we will have a collection of attractor points for any fixed r

Plot the attractors for different values of r

Ricker Model of Population Growth

```
# Loop through values of r
r = 1.0
while r < r_lim:
    # Initial population x(0)
    x = 1

    # Iterate
    for n in range(500):
        x = r * x * np.exp(-x)
        if n > 50:  # Ignore transient behavior
            x_vals.append(x)
            r_vals.append(r)

    r = r + delta_r  # Increment r
```

Ricker Model of Population Growth

```
# Loop through values of r
r = 1.0
while r < r_lim:
    # Initial population x(0)
    x = 1

    # Iterate
    for n in range(500):
        x = r * x * np.exp(-x)
        if n > 50:  # Ignore transient behavior
            x_vals.append(x)
            r_vals.append(r)

    r = r + delta_r  # Increment r
```

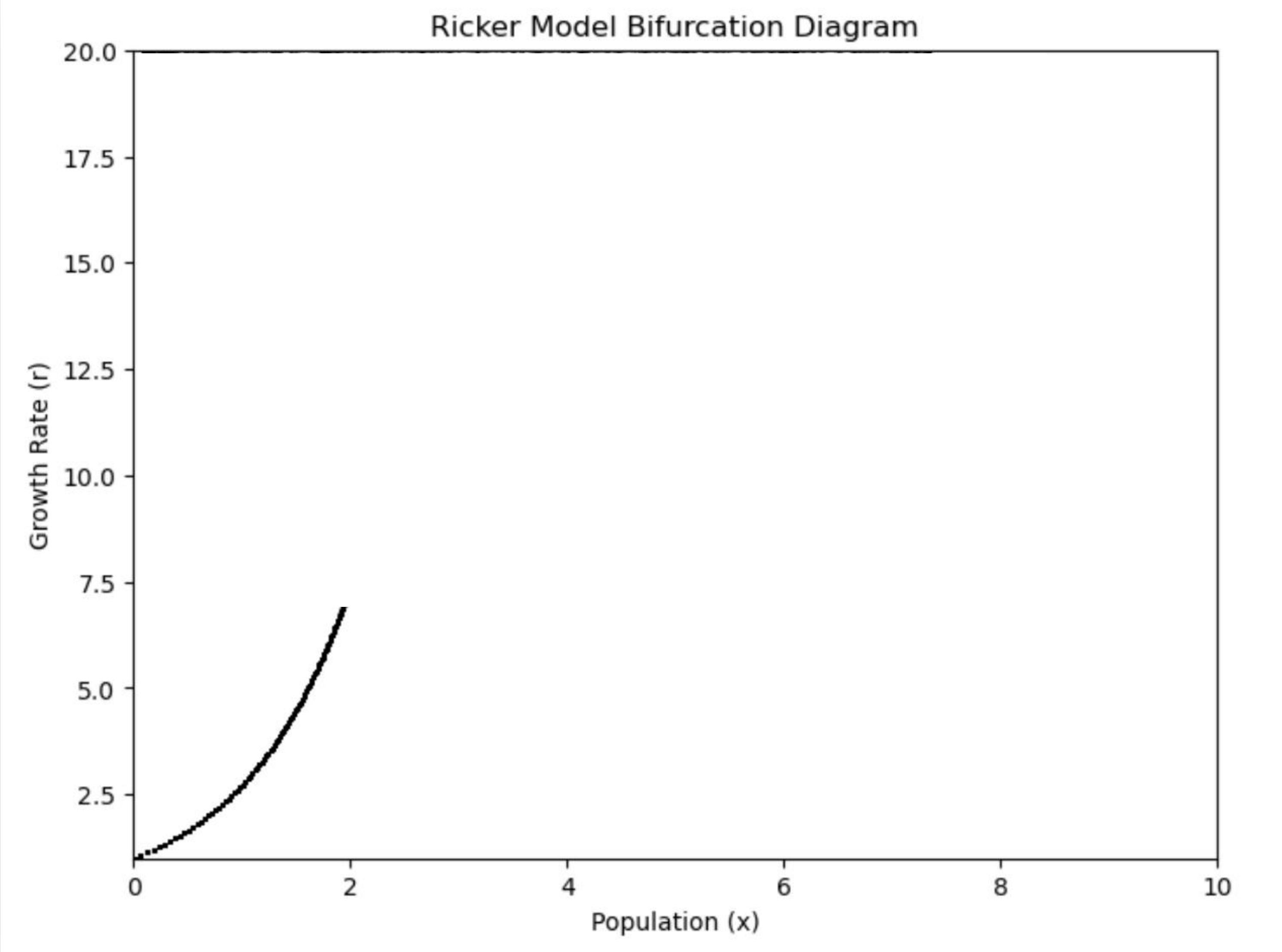
Ricker Model of Population Growth

```
# Loop through values of r
r = 1.0
while r < r_lim:
    # Initial population x(0)
    x = 1

    # Iterate
    for n in range(500):
        x = r * x * np.exp(-x)
        if n > 50:  # Ignore transient behavior
            x_vals.append(x)
            r_vals.append(r)

    r = r + delta_r  # Increment r
```

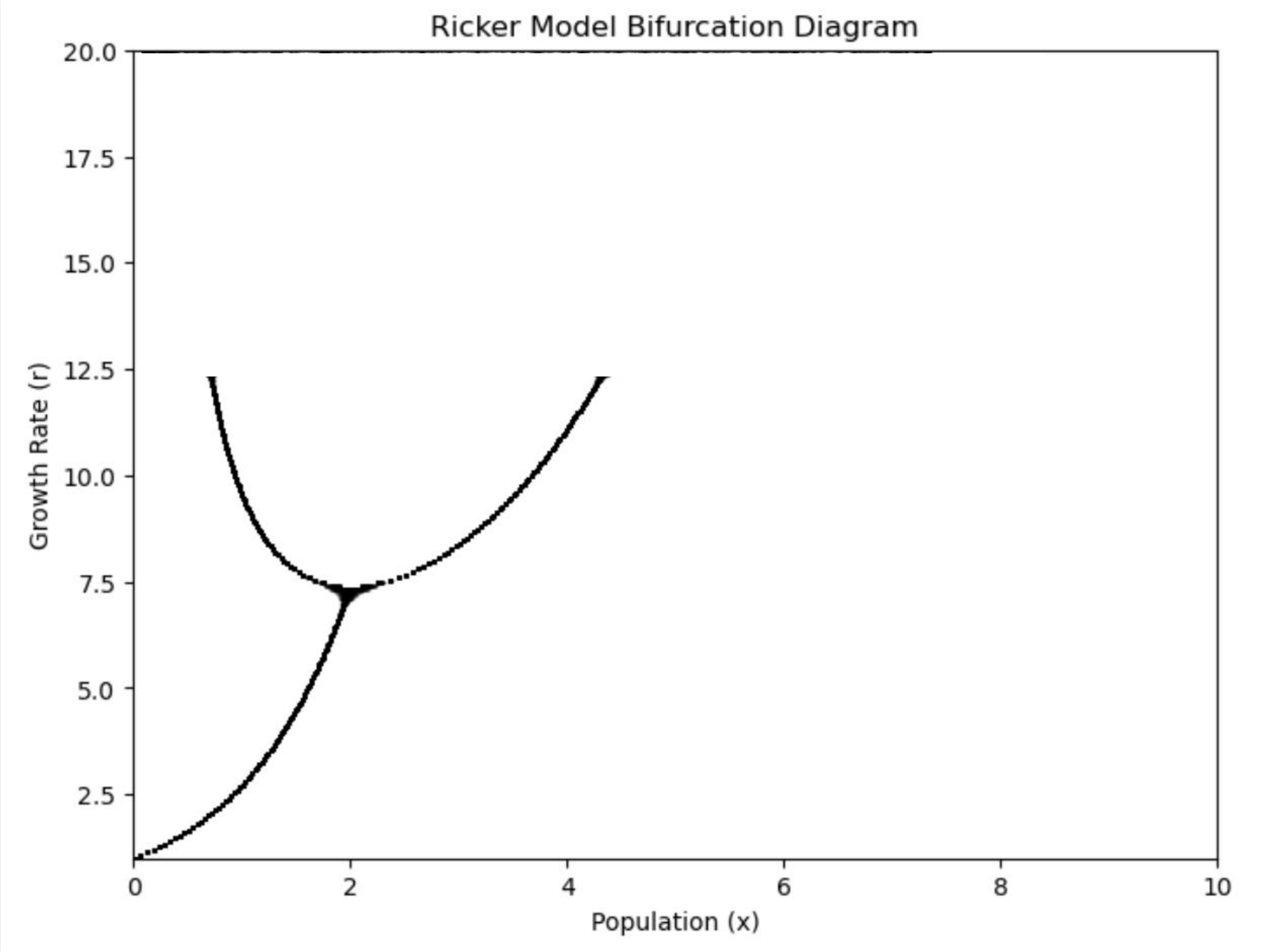
Ricker Model of Population Growth

```
# Loop through values of r
r = 1.0
while r < r_lim:
    # Initial population x(0)
    x = 1

    # Iterate
    for n in range(500):
        x = r * x * np.exp(-x)
        if n > 50:  # Ignore transient behavior
            x_vals.append(x)
            r_vals.append(r)

    r = r + delta_r  # Increment r
```

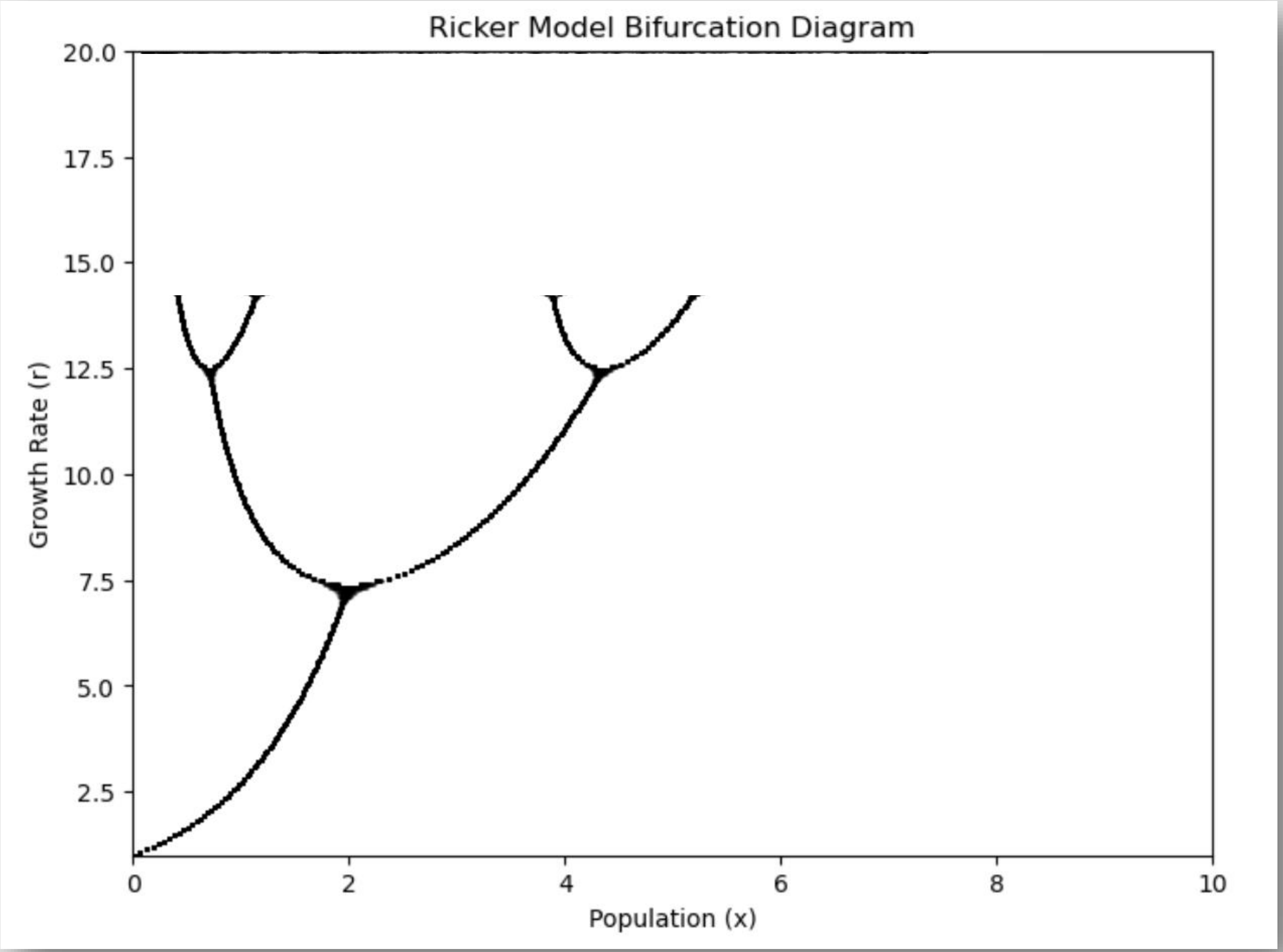
Ricker Model of Population Growth



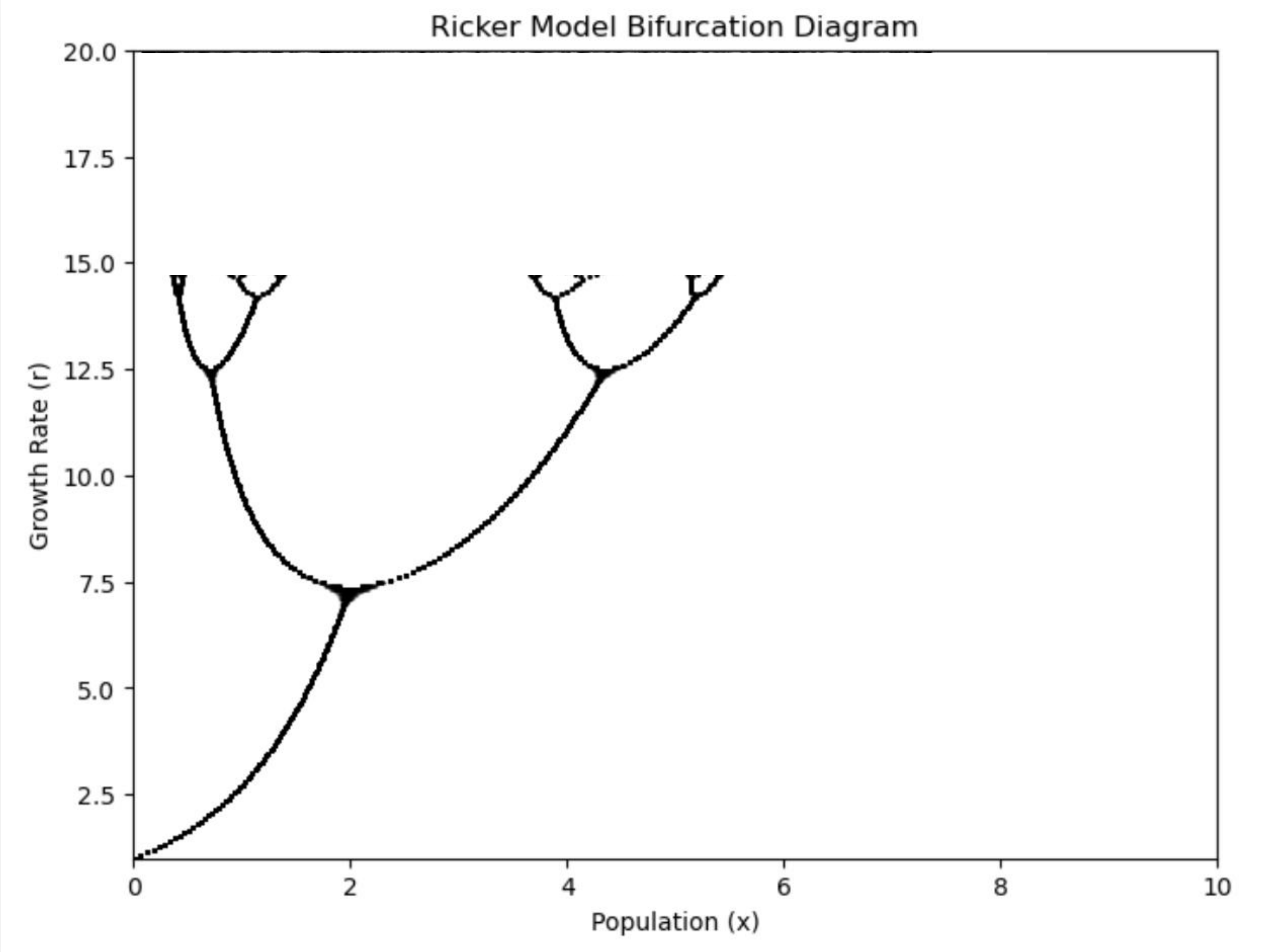
Ricker Model of Population Growth



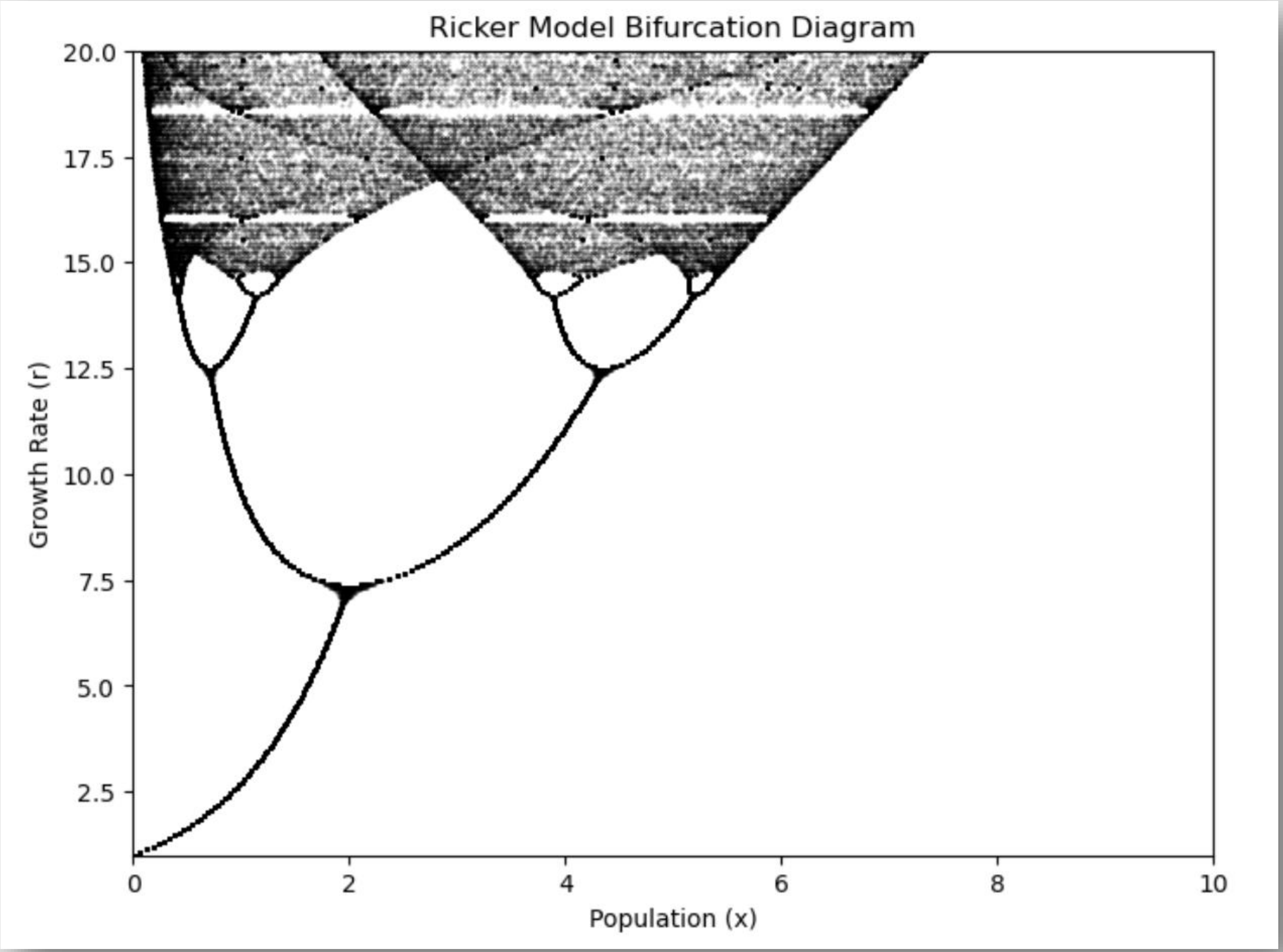
Ricker Model of Population Growth



Ricker Model of Population Growth



Ricker Model of Population Growth



What do we learn from these examples?

A **changing system** can be mathematically modeled in many ways, the most common of which are:

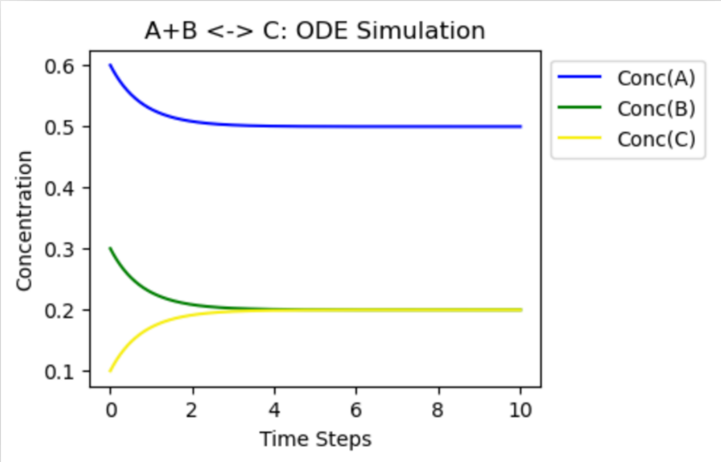
1. **Fully continuous** (differential equations)
2. **Fully discrete**
3. **Mixed discrete and continuous** (Ricker model)

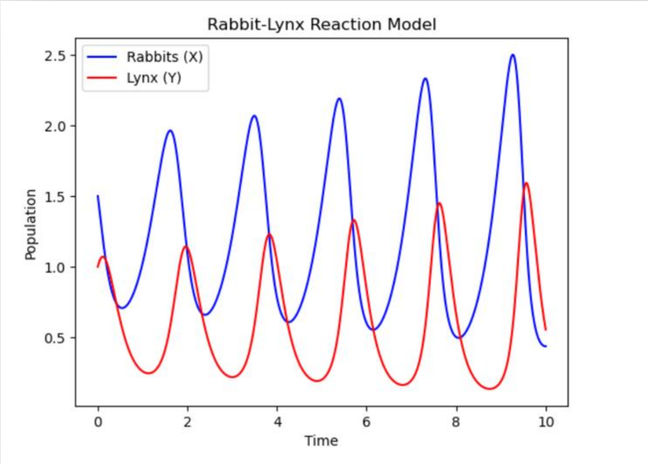
What do we learn from these examples?

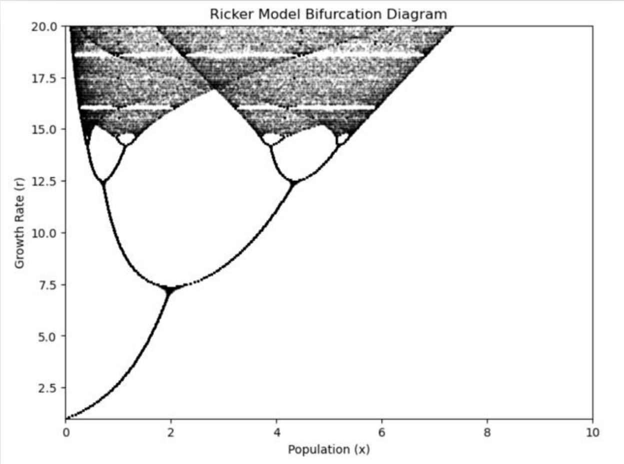
It is very hard to predict the behavior of dynamical systems just by staring at the governing equation

- For some parameters, a system might behave well
- For others, we might get wild oscillations or chaos

What do we learn from these examples?







$$A(t+s) = A(t) + s\,(K_c\,C(t) - K_{ab}\,A(t)\,B(t))$$

$$B(t+s) = B(t) + s\left(K_c\,C(t) - K_{ab}\,A(t)\,B(t)\right)$$

$$C(t+s) = C(t) + s\,(K_{ab}\,A(t)\,B(t) - K_c\,C(t))$$

$$X(t+s) = X(t) + s\,(k_1\,X(t) - k_2\,X(t)\,Y(t))$$

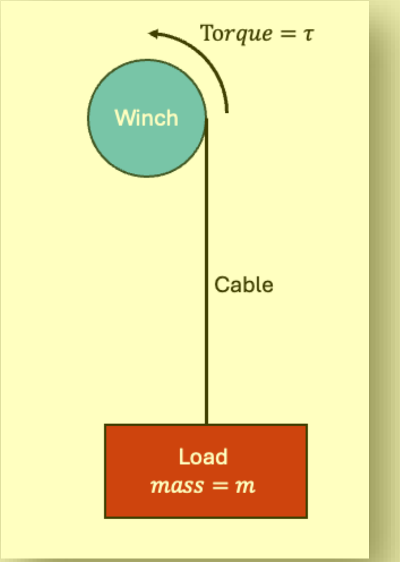
$$Y(t+s) = Y(t) + s\,(k_2\,X(t)Y(t) - k_3\,Y(t))$$

$$\mathrm{x}(n+1) = r\,x(n)\,e^{-x(n)}$$

Simulating Continuous Systems

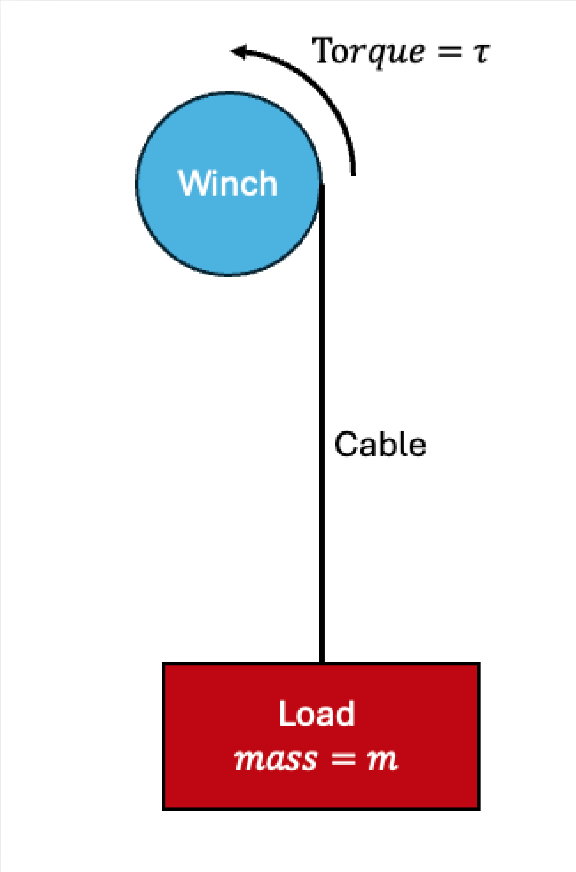
Ricker Model of Population Growth

Rotational Motion



Rotational Motion

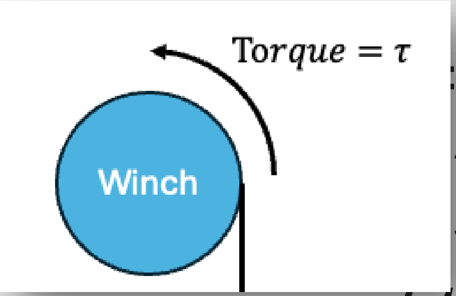
Consider a weight attached to a winch:



Rotational Motion

Define the following variables for the wheel:

$R =$ radius of wheel



= torque

) = angular displacement at time t

$\omega(t) =$ angular velocity

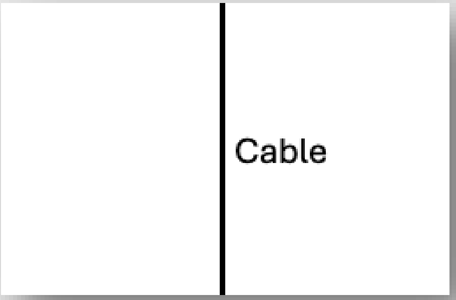
$\mu(t) =$ angular acceleration

$m_w =$ mass of wheel

Rotational Motion

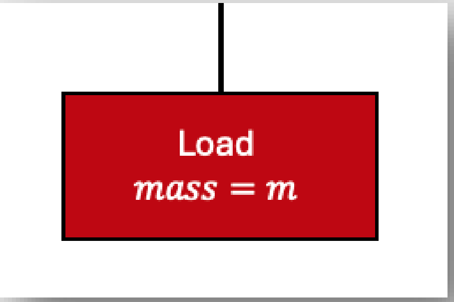
Define the following variables for the cable:

$$T = \text{tension in the cable}$$



Rotational Motion

Define the following variables for the wheel:



m = mass of load

$y(t)$ = vertical displacement of load

$v(t)$ = vertical velocity of load

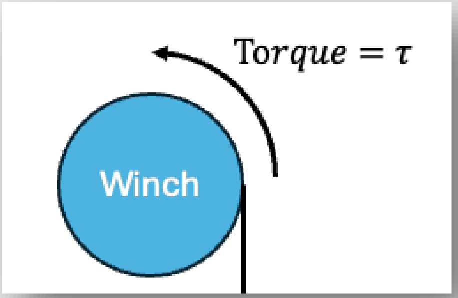
$a(t)$ = vertical acceleration of load

Apply Newton's Laws to Each Component

Let's start with the wheel:

1. The wheel has torque generated from the winch's motor: τ
2. The tension in the cable pulls the other way, and creates a reverse torque of $T\ R$
3. The difference is what determines the angular motion of the wheel:

$$m_w\, R^2\, \mu = \tau - TR$$

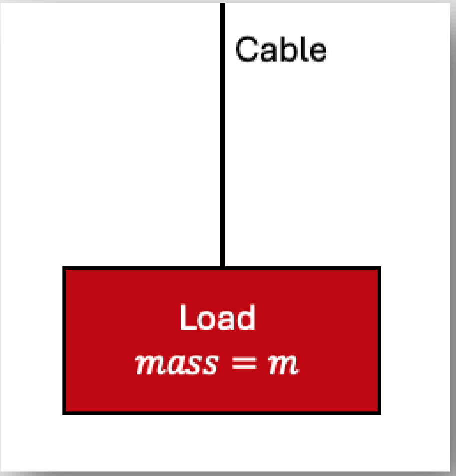


Apply Newton’s Laws to Each Component

Now the load:

1. The upward force is the cable's tension: T
2. The downward force is the weight $m\ g$
3. The difference explains the motion:

$$m\ a = T - m\ g$$

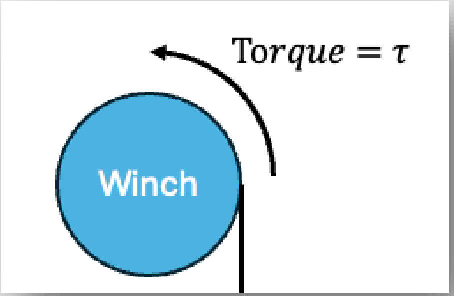


What are the differential equations?

First, the ones relating displacement, velocity, and acceleration for the wheel:

$$\theta'^{(t)} = \omega(t)$$

$$\omega'(t) = \mu(t)$$

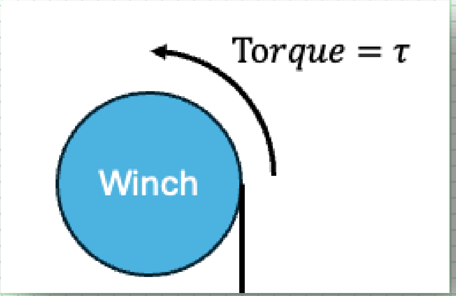


What are the differential equations?

The change in angular displacement over time

$$\theta'(t) = \omega(t)$$

$$\omega'(t) = \mu(t)$$

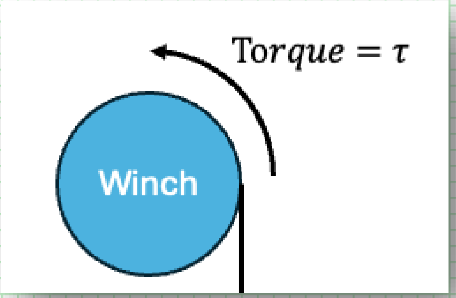


What are the differential equations?

Angular velocity

$$\theta'(t) = \omega(t)$$

$$\omega'(t) = \mu(t)$$

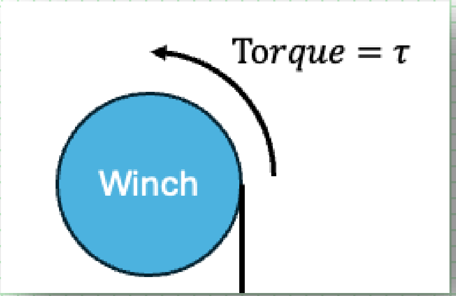


What are the differential equations?

$$\theta'^{(t)} = \omega(t)$$

The change in angular velocity over time

$$\omega'(t) = \mu(t)$$

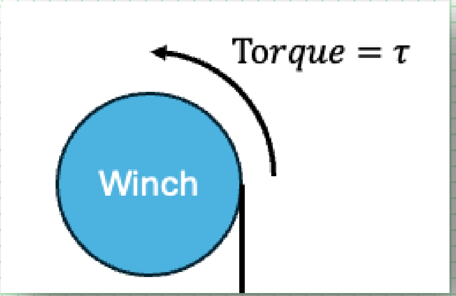


What are the differential equations?

$$\theta'^{(t)} = \omega(t)$$

$$\omega'(t) = \mu(t)$$

Angular acceleration

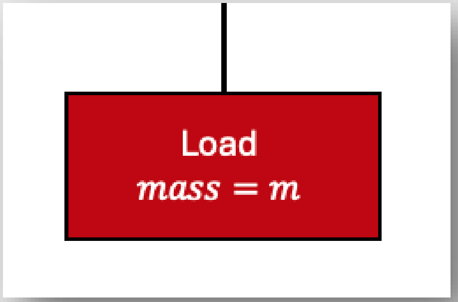


What are the differential equations?

The same for the load:

$$y'(t) = v(t)$$

$$v'(t) = a(t)$$

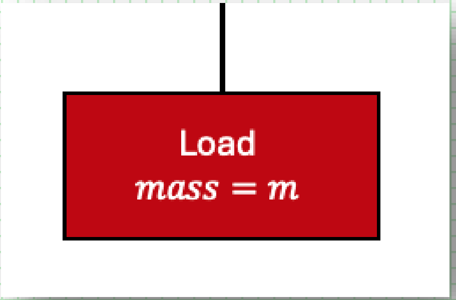


What are the differential equations?

The change in height over time

$$y'^{(t)} = v(t)$$

$$v'(t) = a(t)$$

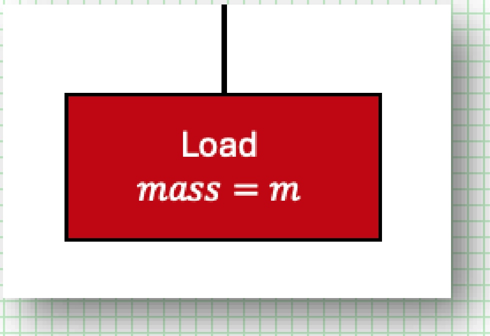


What are the differential equations?

$$y'^{(t)} = v(t)$$

The velocity of the load

$$v'(t) = a(t)$$

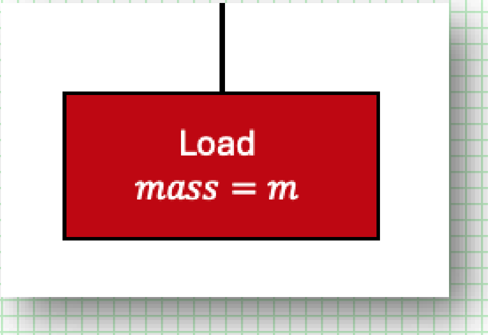


What are the differential equations?

$$y'^{(t)} = v(t)$$

$$v'(t) = a(t)$$

The change in velocity over time

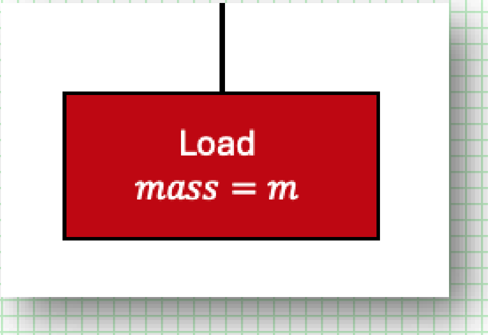


What are the differential equations?

$$y'^{(t)} = v(t)$$

$$v'(t) = a(t)$$

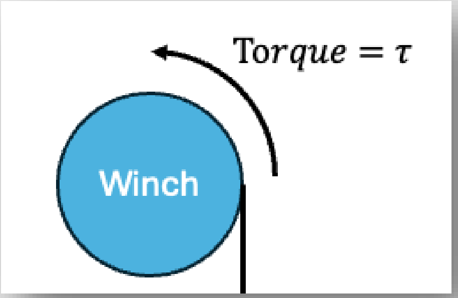
The acceleration of the load



Relate the Equations Through Tension

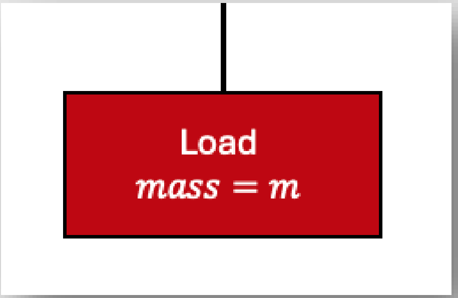
From the wheel:

$$T = \frac{\tau - m_w\, R^2\, \mu}{R}$$



From the load:

$$T = m\, R\, \mu - m\, g$$



Relate the Equations Through Tension

Equating the two and solving for μ:

$$\mu(t) = \frac{\tau - m\,g\,R}{m\,R^2 + m_w\,R^2}$$

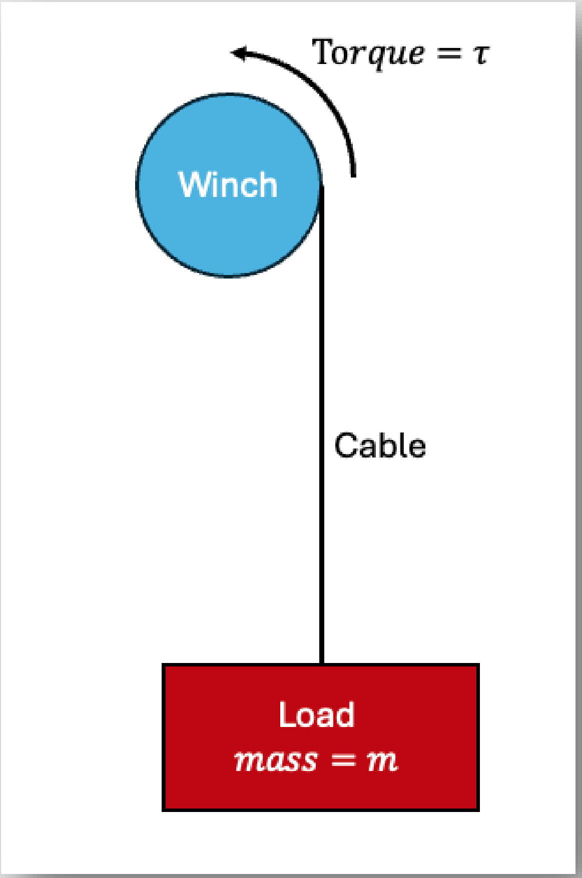
Equations That Describe the System

$$\mu(t) = \frac{\tau - m\,g\,R}{m\,R^2 + m_w\,R^2}$$

$$\theta'(t) = \omega(t)$$

$$\omega'(t) = \mu(t)$$

$$y(t) = R\,\theta(t)$$



Equations That Describe the System

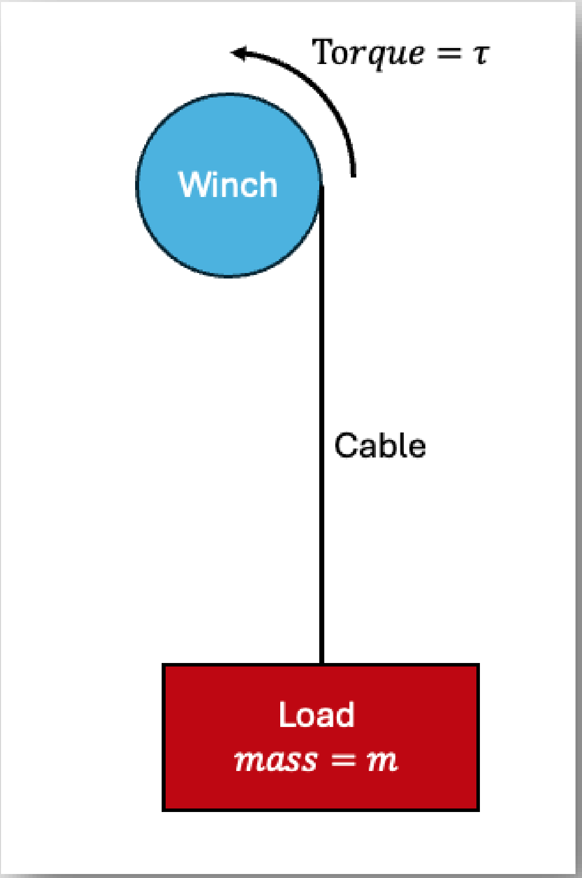
$$\mu(t) = \frac{\tau - m\,g\,R}{m\,R^2 + m_w\,R^2}$$

$$\theta'(t) = \omega(t)$$

$$\omega'(t) = \mu(t)$$

The change in height of the load

$$y(t) = R\,\theta(t)$$



Equations That Describe the System

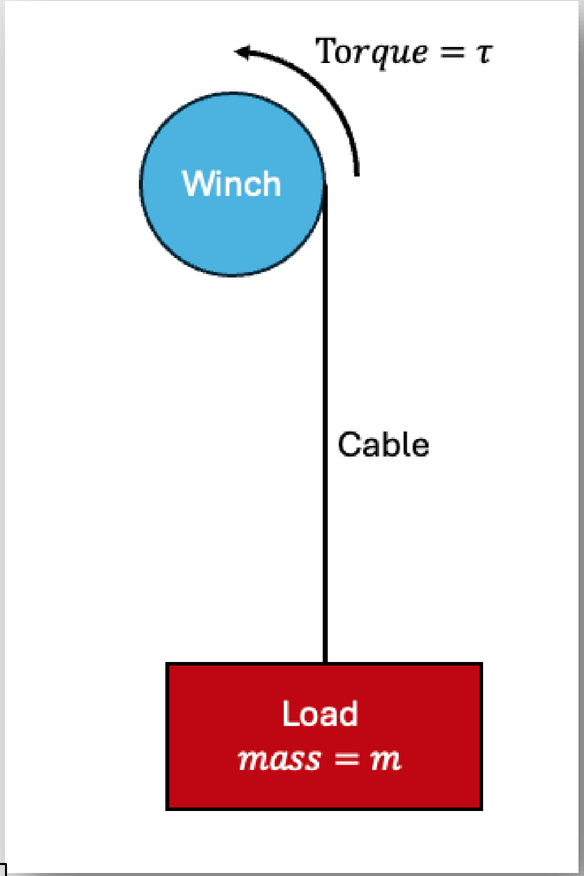
$$\mu(t) = \frac{\tau - m\,g\,R}{m\,R^2 + m_w\,R^2}$$

$$\theta'(t) = \omega(t)$$

$$\omega'(t) = \mu(t)$$

$$y(t) = \underbrace{R\,\theta(t)}$$

Radius of the winch times angle of displacement



Computing Variables in a Simulation

```
def step():
    # Advance the simulation by one time step
    t = t + delta_t

    # Calculate the angular acceleration based tension
    angular_acceleration = (torque - m * g * R) / m * R**2 + mw * R**2

    # Calculate the angular velocity
    angular_velocity = angular_velocity + angular_acceleration * delta_t

    # Convert angular velocity to linear distance
    delta_angle = angular_velocity * delta_t
    y = y + delta_angle * R
```

Computing Variables in a Simulation

```
def step():
    # Advance the simulation by one time step
    t = t + delta_t

    # Calculate the angular acceleration based tension
    angular_acceleration = (torque - m * g * R) / m * R**2 + mw * R**2

    # Calculate the angular velocity
    angular_velocity = angular_velocity + angular_acceleration * delta_t

    # Convert angular velocity to linear distance
    delta_angle = angular_velocity * delta_t
    y = y + delta_angle * R
```

Computing Variables in a Simulation

```
def step():
    # Advance the simulation by one time step
    t = t + delta_t

    # Calculate the angular acceleration based tension
    angular_acceleration = (torque - m * g * R) / m * R**2 + mw * R**2

    # Calculate the angular velocity
    angular_velocity = angular_velocity + angular_acceleration * delta_t

    # Convert angular velocity to linear distance
    delta_angle = angular_velocity * delta_t
    y = y + delta_angle * R
```

Computing Variables in a Simulation

```
def step():
    # Advance the simulation by one time step
    t = t + delta_t

    # Calculate the angular acceleration based tension
    angular_acceleration = (torque - m * g * R) / m * R**2 + mw * R**2

    # Calculate the angular velocity
    angular_velocity = angular_velocity + angular_acceleration * delta_t

    # Convert angular velocity to linear distance
    delta_angle = angular_velocity * delta_t
    y = y + delta_angle * R
```

Computing Variables in a Simulation

```
def step():
    # Advance the simulation by one time step
    t = t + delta_t

    # Calculate the angular acceleration based tension
    angular_acceleration = (torque - m * g * R) / m * R**2 + mw * R**2

    # Calculate the angular velocity
    angular_velocity = angular_velocity + angular_acceleration * delta_t

    # Convert angular velocity to linear distance
    delta_angle = angular_velocity * delta_t
    y = y + delta_angle * R
```