Precalc AI Lab

Understanding Periodic Functions Using Nascar Tracks

In this Lab you will write Python code and run it. You can write the code yourself or get an LLM to do it. Run the code in Google Colab. If it does not run correctly, rewrite until it works correctly.

Purpose of Lab

- Understand what it means for a function to be periodic.
- Approximate the period of a periodic function from its graph.
- Interpret the period of a periodic function in context.
- Find and interpret the minimum and maximum values of a periodic function.

Context for Lab

If you are watching a Nascar race at Daytona from the stands, how does a car's distance from you change as it races around the track? What if you are instead at Indianapolis or Talladega?

Part 1: Creating a Graph of Distance of a Nascar from a Spectator

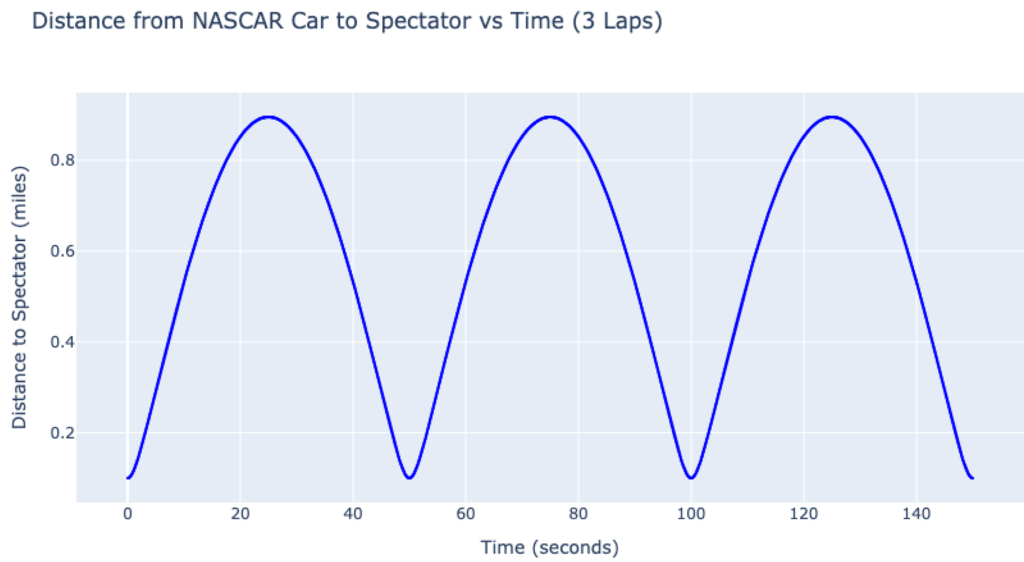
Use an LLM

Generate Python code to sketch an interactive graph for the distance between a spectator and a Nascar car driving around the Daytona racetrack for three laps as a function of time. We ask the LLM to ensure the following:

- The code is simple, easy to follow for a Precalculus student, and runs in Colab.
- Units of the y-axis is miles
- Units of the x-axis is seconds

- The spectator sits 0.1 miles away from the starting point
- The car, on average, takes 50 seconds to complete a lap
- The graph is interactive so you can trace over it and be given coordinates of points on the graph.
- The code uses the Plotly library for the interactive graph.

The graph we get (assuming Daytona is circular and the spectator is aligned with the start line) is the following with code given at the end of the lab. Your graph may look different with different assumptions about track shape and placement of the spectator.



Part 2: Periodic Functions and their properties

The distance function of the car from the spectator given in the figure is said to be periodic, because its values repeat on a regular interval or period.

- Since the function repeats every lap, the period is the time elapsed during a full lap, or 50 seconds.
- The minimum, 0.1 miles, is the distance from the spectator to the start line. The maximum, about 0.9 miles, is the furthest the car is away from the spectator.

Answer the following:

1. In 2020, Erik Jones attained the lap record completing a lap in roughly 44 seconds. Assuming he maintained this speed for three laps, what would the graph of the distance function look like? What would its period be? What is the minimum and maximum distance between the spectator and the car? You can use an LLM to plot the graph, but you might want to plot it on the same axes as your other graph to be able to compare.
2. While Nascar cars are regular cars that have been modified, Formula One cars are built only for racing and can maintain higher speeds. If a Formula One car were to race in Daytona, then based on its greater speed, the graph, which would also be periodic, would have a period of 40 seconds. Modify the code to give the distance to a Formula One car (you can change time on the second line). How does the graph compare to the first graph? Are the periods the same? Why or why not?

Part 3: Periodic Functions and their properties

Choose a different Nascar track (e.g. Talladega or Indianapolis) and either modify the code or use an LLM to generate new code for this track. The graph you create should have the same properties as the first graph, but make sure to adapt the lap time to something realistic for that track. You will need to specify a reasonable time for a Nascar to complete a lap (you can find average times online).

Your graph may be different to other people's depending on how you or the LLM create the code and depending on the track you choose. However, the fundamental properties will hold for any graph.

Use your graph to answer the following:

1. What is the furthest distance between the spectator and the car? What is the closest?
2. Find the period. What does the period represent in the context of the car?

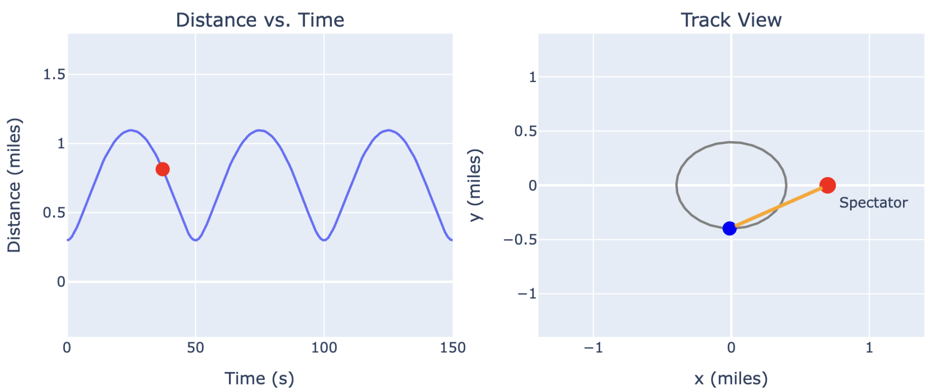
Use an LLM

Find an approximate time it would take a Formula One car to complete a lap of this track and modify your code to generate a graph of a Formula One car completing three laps. Make sure to keep the same scale as the first graph so you can compare.

3. Are the periods of the Formula One graph and the Nascar graph the same? Why? What about the vertical distance in the graph between the minimum and maximum values?
4. Now compare the graphs of your new track with both car types to the Daytona graphs with both car types. Which ones have the same period? Why? What about the minimum and maximum values? Explain, in context of the tracks and the cars.

Further discussion

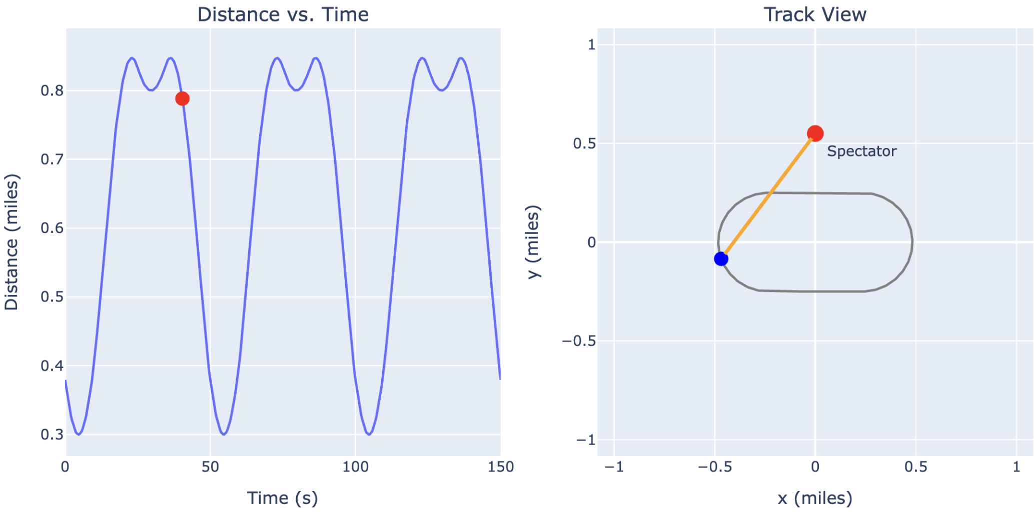
1. **Make the math visible.** Ask the LLM to walk you in all detail through the math it used to create the graphs. Make sure you tell it you are a precalculus student and want to learn, so that it does not use any fancy math. Then, ask it to show you where that math is used in the code. Provide an explanation in your own words of how the python code figures out the numbers for each quadratic form to display them.
2. **Interactive track view from above.** Ask the LLM to make interactive side-by-side plots in which one plot is the distance between the car and spectator, and the other plot is a track view from the top that displays the position of the car and the spectator and a line segment for the distance between them. It should look like something like this:



3. **Most tracks are not circular.** How can the graphs look like when the tracks are not circular?
   a) Get the LLM to create distance time graphs for the following cases:
      i. An oval-shaped track composed of two straight line segments and two semicircles at the ends.

ii. A real F1 track (ask the LLM to look up the data for you)

b) For (i), you should get a plot that looks as follows. Can you explain the "double dents" at the top?



Sample Python Code

```
import numpy as np
import plotly.graph_objects as go


# --- Parameters ---
lap_length_miles = 2.5 # One lap around Daytona is 2.5 miles
lap_time_seconds = 50 # One lap takes 50 seconds
num_laps = 3 # Total number of laps
spectator_offset_miles = 0.1 # Spectator is 0.1 miles behind the start


# --- Derived values ---
R = lap_length_miles / (2 * np.pi) # Radius of the circular track
omega = 2 * np.pi / lap_time_seconds # Angular speed in radians/sec
total_time = lap_time_seconds * num_laps


# --- Time samples ---
t = np.linspace(0, total_time, 2000) # time from 0 to 150 sec


# --- Angle of car over time ---
theta = omega * t

# --- Car position on the circle ---
car_x = R * np.cos(theta)
car_y = R * np.sin(theta)


# --- Spectator location (0.1 miles behind start line, on x-axis) ---
spectator_x = R + spectator_offset_miles
spectator_y = 0


# --- Distance from spectator to car ---
distance = np.sqrt((car_x - spectator_x)**2 + (car_y - spectator_y)**2)


# --- Create interactive plot ---
fig = go.Figure()

fig.add_trace(go.Scatter(
  x=t,
  y=distance,
  mode='lines+markers',
  marker=dict(size=2, color='blue'),
  line=dict(width=2),
  name='Distance to Spectator',
  hovertemplate='Time: %{x:.2f} s<br>Distance: %{y:.3f} mi'
))

fig.update_layout(
  title='Distance from NASCAR Car to Spectator vs Time (3 Laps)',
  xaxis_title='Time (seconds)',
  yaxis_title='Distance to Spectator (miles)',
  hovermode='x unified',
  width=900,
  height=500
)

fig.show()
```