Module 4: Swine Flu Mexico 2009

Team Members:

Lakshya Raman, Noah Edouard

Project Title:

Modeling the 2009 H1N1 Swine Flu Outbreak in Mexico Using the SIR Epidemiological Framework

Project Goal:

This project seeks to analyze and model the spread of the 2009 H1N1 swine flu outbreak in Mexico using the SIR epidemiological model. By fitting the model to real outbreak data, we seek to estimate key disease transmission parameters (β, γ, and R_0), evaluate epidemic dynamics during the early Mexican outbreak phase, and assess how well the SIR framework captures the observed progression of H1N1 transmission in the population.

Disease Background:

- Prevalence & incidence
  - The 2009 H1N1 swine flu outbreak in Mexico began with unusual clusters of severe respiratory illness reported in March and April 2009, particularly in Mexico City, Oaxaca, and San Luis Potosí. As surveillance intensified, more than 2,000 suspected cases and multiple confirmed deaths were recorded by late April, signaling the rapid spread of a novel influenza strain. By May 2009, approximately 4,000 laboratory-confirmed cases were reported, reflecting strong early transmission dynamics during the initial outbreak phase. Over the full year, Mexico ultimately documented more than 70,000 confirmed infections and several hundred deaths, though this number substantially underestimates true infections, since only symptomatic individuals with access to laboratory confirmation were tested. Globally, the World Health Organization (WHO) declared the outbreak a pandemic on June 11, 2009, and worldwide estimates later suggested between 151,700 and 575,400 deaths attributable to the pandemic H1N1 strain. Sources: CDC MMWR 2009; WHO Situation Updates 2009; PLOS Medicine (2012).
- Economic burden

    - The economic impact of the 2009 H1N1 outbreak in Mexico was substantial, affecting both the healthcare system and the broader national economy. Direct medical costs arose from the sudden surge in hospitalizations, especially severe pneumonia cases requiring ICU admission, as well as from large-scale distribution of antivirals such as oseltamivir and procurement of emergency diagnostic kits. The national public health sector experienced considerable strain as demand for clinical evaluations and laboratory analyses rapidly increased. Beyond medical expenses, Mexico faced significant indirect economic losses due to public health interventions. Nationwide school closures, cancellation of public events, and reduced tourism during April and May 2009 generated an estimated US$2.3 billion in economic disruption. Temporary shutdowns in Mexico City—including the closure of restaurants, cinemas, and many workplaces—further reduced productivity, while international travel restrictions imposed by other countries affected cross-border trade and mobility. Sources: OECD Economic Impact Reports; Secretaría de Salud; Mexican Government Communications.
- Risk factors (genetic, lifestyle) & Societal determinants
    - A variety of biological, lifestyle, and social determinants influenced susceptibility and disease severity during the Mexican H1N1 outbreak. Older adults above age 65 demonstrated partial immunity due to prior exposure to pre-1950 H1N1 antigenic variants, while children and young adults—lacking this immunologic memory—experienced disproportionately high infection and hospitalization rates. Clinical studies in Mexico identified obesity, pregnancy, chronic lung disease, diabetes mellitus, and cardiovascular disease as major individual-level risk factors for severe disease or ICU admission. Societal and environmental factors also played a significant role in transmission patterns. High population density and crowded public transportation systems in Mexico City facilitated early person-to-person spread. Rural communities, facing disparities in healthcare access, often received antiviral treatment too late for maximal benefit. Early school closures in late April proved critical in reducing transmission, and modeling studies later identified these measures as one of the most effective non-pharmaceutical interventions for lowering the basic reproduction number (R_0). Sources: PLOS Medicine (2009); Mexican Ministry of Health Cohort Studies; WHO and CDC Pandemic Analyses.
- Symptoms
    - During the 2009 H1N1 outbreak, Mexico used a clinical case definition that included fever, cough, and respiratory distress as key indicators of suspected infection. In confirmed cases, the most commonly reported symptoms were fever, cough, sore throat, myalgias, fatigue, and headaches. Gastrointestinal symptoms such as vomiting and diarrhea were more frequently observed than in typical seasonal influenza, making the clinical presentation somewhat distinct from previous annual flu strains. Severe cases progressed to viral pneumonia,

and in some patients, rapid respiratory deterioration resulted in acute respiratory failure requiring mechanical ventilation. These severe outcomes were most often observed in individuals with underlying comorbidities or delayed access to medical care. Sources: CDC MMWR (2009); Mexican Ministry of Health Clinical Surveillance Report

- Diagnosis
  - Diagnostic approaches used in Mexico during the 2009 pandemic centered on RT-PCR, which served as the gold standard and was conducted primarily at the Instituto de Diagnóstico y Referencia Epidemiológicos (INdRE). Rapid influenza antigen tests, though widely available, were known to have poor sensitivity for detecting the novel H1N1 strain, leading to substantial underdiagnosis. Some viral cultures were performed, but their use was limited due to long turnaround times and resource constraints. Chest imaging frequently revealed diffuse infiltrates in patients with pneumonia, helping clinicians identify severe disease. Because laboratory capacity was overwhelmed in the early outbreak phase, PCR testing was typically reserved for severe cases or sentinel surveillance, meaning that many mild or moderate cases were never confirmed. Sources: INdRE Diagnostic Protocols; CDC Pandemic Response Guidelines; WHO Laboratory Guidance (2009).
- Biological mechanisms (anatomy, organ physiology, cell & molecular physiology)
  - The 2009 A(H1N1)pdm09 virus originated from a unique reassortment of gene segments from both North American and Eurasian swine influenza lineages, combining elements of triple-reassortant swine viruses with Eurasian avian-like swine components. This genetic composition produced a strain able to infect humans efficiently while possessing antigenic features that rendered much of the global population immunologically naïve. The virus enters the human respiratory tract by binding to α-2,6–linked sialic acid receptors, which are abundant in the upper airway. After entering epithelial cells, the virus replicates rapidly, causing direct cellular injury and potentially triggering an excessive inflammatory response in severe cases. At the organ level, severe infections resulted in diffuse alveolar damage, impaired gas exchange, and progression to acute respiratory distress syndrome (ARDS). Additionally, secondary bacterial infections, particularly with Streptococcus pneumoniae and Staphylococcus aureus, contributed to mortality and worsened clinical outcomes. Sources: Genomic Phylogeny Studies (2009); New England Journal of Medicine Virology Analyses; CDC Pathophysiology Reports.

Dataset:

The dataset under analysis derives from the “H1N1 | 2009 | Swine Flu Pandemic Dataset” on Kaggle, which aggregates epidemiological data related to the 2009 A(H1N1)pdm09 pandemic. This publicly available dataset includes time-series records of confirmed

H1N1 cases, reported deaths, and key demographic or regional variables (e.g., country, date). The source of the data is Kaggle (imdevskp, 2009 dataset) and ultimately reflects publicly reported cumulative case counts from global health surveillance systems, including national health ministries and the World Health Organization (WHO). Data were collected through laboratory-confirmed diagnoses (typically via RT-PCR) and aggregated at a daily or weekly frequency, depending on reporting jurisdiction. Units are typically "number of confirmed cases" and "number of deaths" per reporting period; cumulative case counts are provided, enabling derivation of daily incidence by differencing. The techniques behind data collection include routine public health surveillance, laboratory confirmation, and official epidemiologic reporting systems. Because the data consist of aggregated, reported counts (rather than individual-level case data), they are subject to reporting delays, under-ascertainment, and variation in testing capacity across regions. Nevertheless, this aggregated dataset provides a valuable epidemiological time series that can be used to estimate infection dynamics, convert to SIR-model compartments, and infer key epidemiological parameters such as growth rate and reproduction number.

In [10]:
```
## LOAD YOUR DATASET HERE.

# 1. Read in the csv file of cumulative cases.

# 2. Use the convert_cumulative_to_SIR function to convert cumulative cases

# 3. Plot S, I, R over time.

# Section 0 — utilities and data load (run once)
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from scipy.optimize import minimize, minimize_scalar
from scipy.integrate import solve_ivp

# ---------- Config ----------
CSV_PATH = "swine_flu_mexico_data_2009_cumulative.csv"
POPULATION = 130_800_000   # change if needed
INFECTIOUS_PERIOD_DAYS = 4 # approximate for swine flu; adjust if needed
SAMPLING_DAYS = 7          # 7 if weekly data; set to 1 for daily
# ----------------------------

# Helper: convert cumulative to S/I/R estimates (simple rolling-window)
def convert_cumulative_to_SIR(df, date_col='date', new_case_col='cases',
                              cumulative_col='cumulative_cases',
                              population=POPULATION,
                              infectious_period_days=INFECTIOUS_PERIOD_DAYS,
                              sampling_days=SAMPLING_DAYS):
    df = df.copy()
    if date_col in df.columns:
        df[date_col] = pd.to_datetime(df[date_col])
    # ensure new case col exists
    if new_case_col not in df.columns:
```

```
        # try alternatives
        alt = [c for c in df.columns if 'case' in c.lower() and c != cumulat
        if alt:
            df[new_case_col] = pd.to_numeric(df[alt[0]], errors='coerce').fi
        else:
            raise KeyError("Couldn't find new-case column; add a 'cases' col
    df[new_case_col] = pd.to_numeric(df[new_case_col], errors='coerce').fill
    # cumulative
    if cumulative_col not in df.columns or df[cumulative_col].isnull().all()
        df[cumulative_col] = df[new_case_col].cumsum()
    df[cumulative_col] = pd.to_numeric(df[cumulative_col], errors='coerce').
    # window in data-samples (e.g., weeks)
    window = max(1, int(round(infectious_period_days / float(sampling_days))
    df['I_est_calc'] = df[new_case_col].rolling(window=window, min_periods=1
    # ensure I <= cumulative
    df['I_est_calc'] = df.apply(lambda row: min(row['I_est_calc'], row[cumul
    df['R_est_calc'] = df[cumulative_col] - df['I_est_calc']
    df['S_est_calc'] = population - df[cumulative_col]
    # if user-supplied I_est exists, prefer it (but fillna with calc)
    if 'I_est' in df.columns:
        df['I_est'] = pd.to_numeric(df['I_est'], errors='coerce')
        df['I_est'] = df['I_est'].fillna(df['I_est_calc'])
    else:
        df['I_est'] = df['I_est_calc']
    df['R_est'] = df[cumulative_col] - df['I_est']
    df['S_est'] = population - df[cumulative_col]
    # clip negatives
    for c in ['S_est','I_est','R_est']:
        df[c] = df[c].clip(lower=0.0)
    return df

# Euler SIR (discrete, dt = 1 sampling unit)
def euler_sir(beta, gamma, S0, I0, R0, n_steps, N):
    S = np.zeros(n_steps)
    I = np.zeros(n_steps)
    R = np.zeros(n_steps)
    S[0], I[0], R[0] = S0, I0, R0
    for k in range(1, n_steps):
        dS = -beta * S[k-1] * I[k-1] / N
        dI =  beta * S[k-1] * I[k-1] / N - gamma * I[k-1]
        dR =  gamma * I[k-1]
        S[k] = S[k-1] + dS
        I[k] = I[k-1] + dI
        R[k] = R[k-1] + dR
        # guard
        S[k] = max(S[k], 0.0)
        I[k] = max(I[k], 0.0)
        R[k] = max(R[k], 0.0)
    return S, I, R

# SIR RHS for continuous solve_ivp
def sir_rhs(t, y, beta, gamma, N):
    S, I, R = y
    dS = -beta * S * I / N
    dI =  beta * S * I / N - gamma * I
    dR =  gamma * I
```

```
    return [dS, dI, dR]

# Load CSV and build S/I/R
df_raw = pd.read_csv(CSV_PATH)
print("Columns in CSV:", df_raw.columns.tolist())
# ensure date and cases exist
if 'date' in df_raw.columns:
    df_raw['date'] = pd.to_datetime(df_raw['date'])
else:
    raise KeyError("CSV missing 'date' column.")
if 'cases' not in df_raw.columns:
    # try alternatives for new cases
    alt = [c for c in df_raw.columns if 'case' in c.lower() and 'cumulative'
    if alt:
        df_raw['cases'] = pd.to_numeric(df_raw[alt[0]], errors='coerce').fil
    else:
        raise KeyError("CSV missing 'cases' column and couldn't infer one.")
df = convert_cumulative_to_SIR(df_raw, date_col='date', new_case_col='cases'
                               cumulative_col='cumulative_cases', population=
                               infectious_period_days=INFECTIOUS_PERIOD_DAYS,
                               sampling_days=SAMPLING_DAYS)

# Prepare arrays
dates = df['date'].values
n = len(df)
t = np.arange(n)     # 0,1,... each step == one sample (week)
I_obs = df['I_est'].values.astype(float)
S_data = df['S_est'].values
R_data = df['R_est'].values

print("Loaded data rows:", n)
print(df[['date','cases','cumulative_cases','I_est']].head())
```

```
Columns in CSV: ['date', 'confirmed_cases']
Loaded data rows: 50
        date  cases  cumulative_cases  I_est
0 2009-04-24     18                18   18.0
1 2009-04-26     18                36   18.0
2 2009-04-27     26                62   26.0
3 2009-04-28     26                88   26.0
4 2009-04-29     26               114   26.0
```

Data Analyis:

Methods

To analyze and model the epidemic data, I applied the standard SIR differential-equation model and estimated its parameters by fitting the model output to the observed infection counts. I began by loading and preparing the cumulative case data, converting it into estimated infectious counts for each day. I then implemented the SIR equations numerically using Euler's method and used least-squares minimization to identify the transmission rate (β) and recovery rate (γ) that best matched the full dataset. To evaluate predictive performance, I repeated the fitting procedure using only the first half

of the data and compared how well those parameters predicted the second half of the epidemic. I also examined the effect of numerical accuracy by contrasting Euler's method with a higher-order Runge–Kutta solver (RK45 via solve_ivp), showing how different integration schemes affect parameter estimates and predictive error. Across these steps, I analyzed the resulting SSE values and parameter differences to assess model quality, numerical stability, and the reliability of early-epidemic parameter estimation.

Analysis

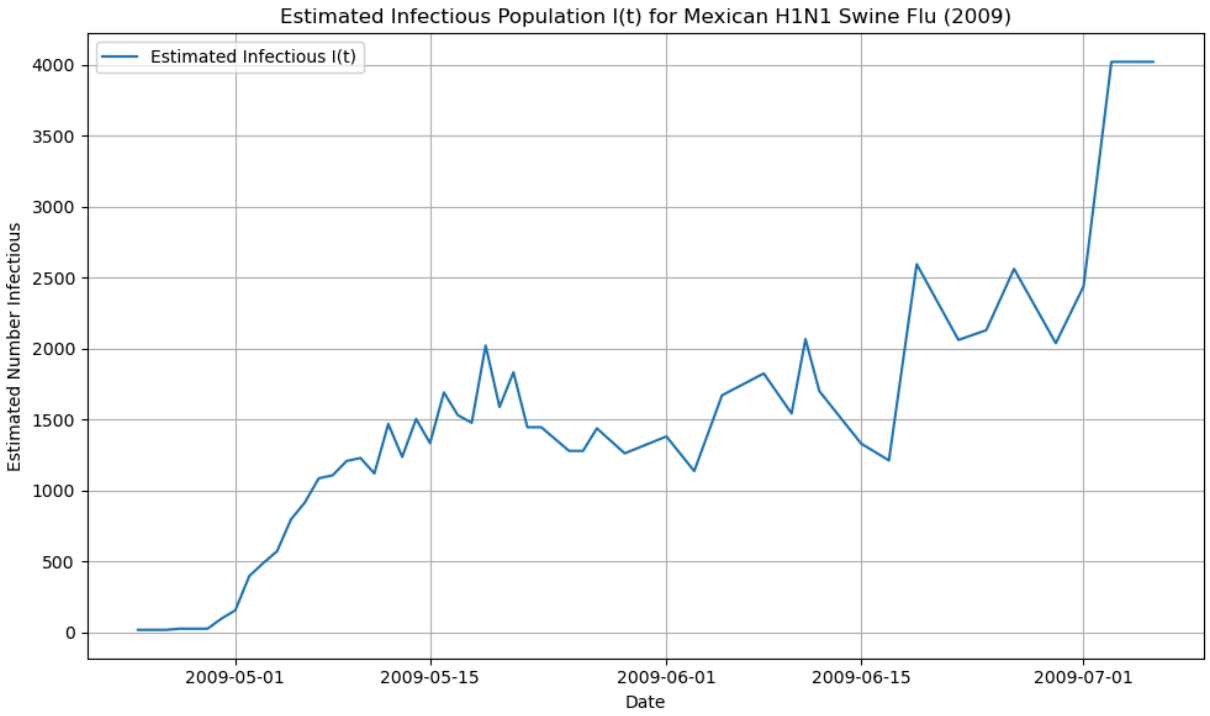
(Describe how you analyzed the data. This is where you should intersperse your Python code so that anyone reading this can run your code to perform the analysis that you did, generate your figures, etc.)

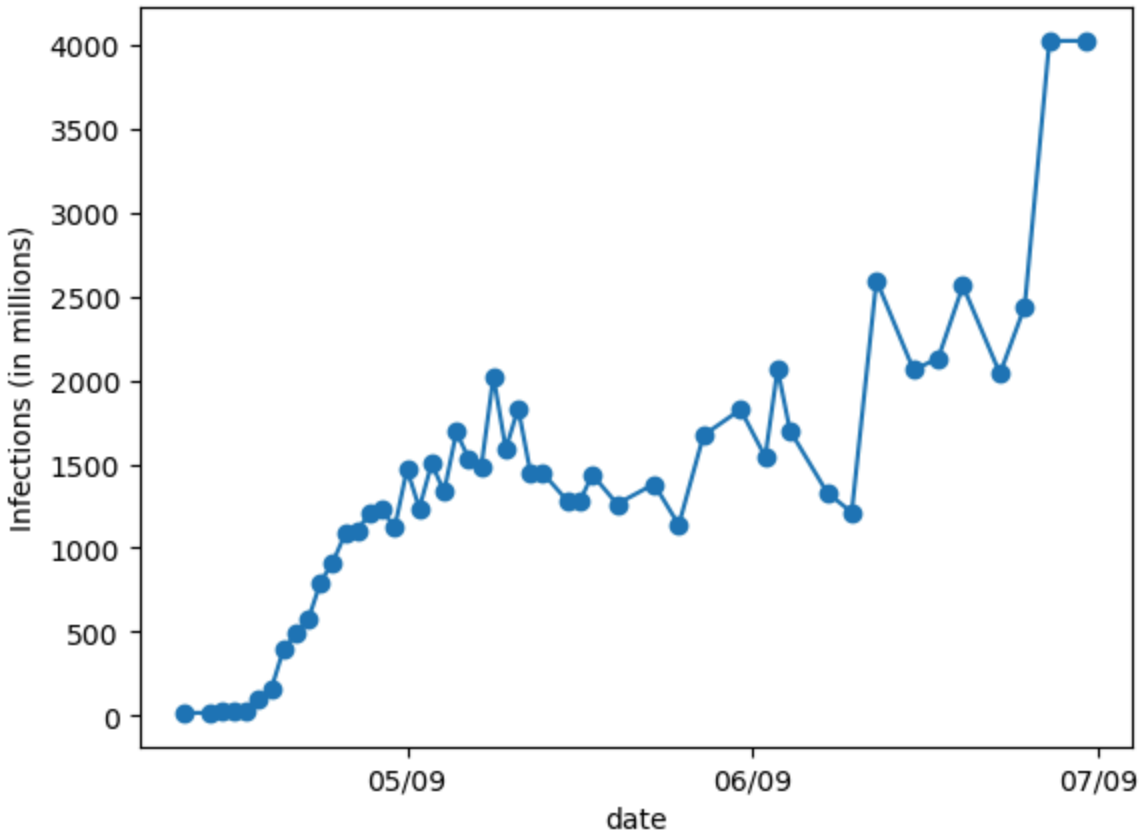
1. Fitting the SIR Model

In [11]:
```
# Using the euler_SIR function defined earlier, we can simulate the SIR mode

import pandas as pd
import matplotlib.pyplot as plt
from main_functions import convert_cumulative_to_SIR   # or wherever your fu

# 1. Read your Mexican swine flu cumulative case data
df = pd.read_csv("swine_flu_mexico_data_2009_cumulative.csv")

# Ensure date is datetime
df["date"] = pd.to_datetime(df["date"])

# Ensure cumulative column is named correctly for the SIR function
df.rename(columns={"confirmed_cases": "cumulative_cases"}, inplace=True)

# 2. Convert cumulative cases → S, I, R using DAILY infectious period
mexico_population_2009 = 112_000_000       # approx Mexico pop. 2009
infectious_period_days = 8                 # typical H1N1 infectious period

sir_df = convert_cumulative_to_SIR(
    df,
    date_col="date",
    cumulative_col="cumulative_cases",
    population=mexico_population_2009,
    infectious_period=infectious_period_days
)

print(sir_df.head())

# 3. Plot I(t) alone
plt.figure(figsize=(10, 6))
plt.plot(sir_df["date"], sir_df["I_est"], label="Estimated Infectious I(t)")
plt.xlabel("Date")
plt.ylabel("Estimated Number Infectious")
plt.title("Estimated Infectious Population I(t) for Mexican H1N1 Swine Flu (
```

```
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()


import numpy as np, pandas as pd
from datetime import datetime, timedelta

N = 130.8  # US population in millions

df_full = pd.read_csv("swine_flu_mexico_data_2009_cumulative.csv")
df_full = df_full[(df_full['date'] >= '2009-04-24')&(df_full['date'] <= '200
df_full.head()

import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import pandas as pd

# Plot
plt.plot(sir_df['date'], sir_df['I_est'], 'o-')
plt.xlabel('date'); plt.ylabel('Infections (in millions)')
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%m/%y'))
plt.gca().xaxis.set_major_locator(mdates.WeekdayLocator(interval=4))
plt.show()
```

	date	cumulative_cases	new_cases	I_est	R_est	S_est
0	2009-04-24	18	18.0	18.0	0.0	111999982.0
1	2009-04-26	18	0.0	18.0	0.0	111999982.0
2	2009-04-27	26	8.0	26.0	0.0	111999974.0
3	2009-04-28	26	0.0	26.0	0.0	111999974.0
4	2009-04-29	26	0.0	26.0	0.0	111999974.0

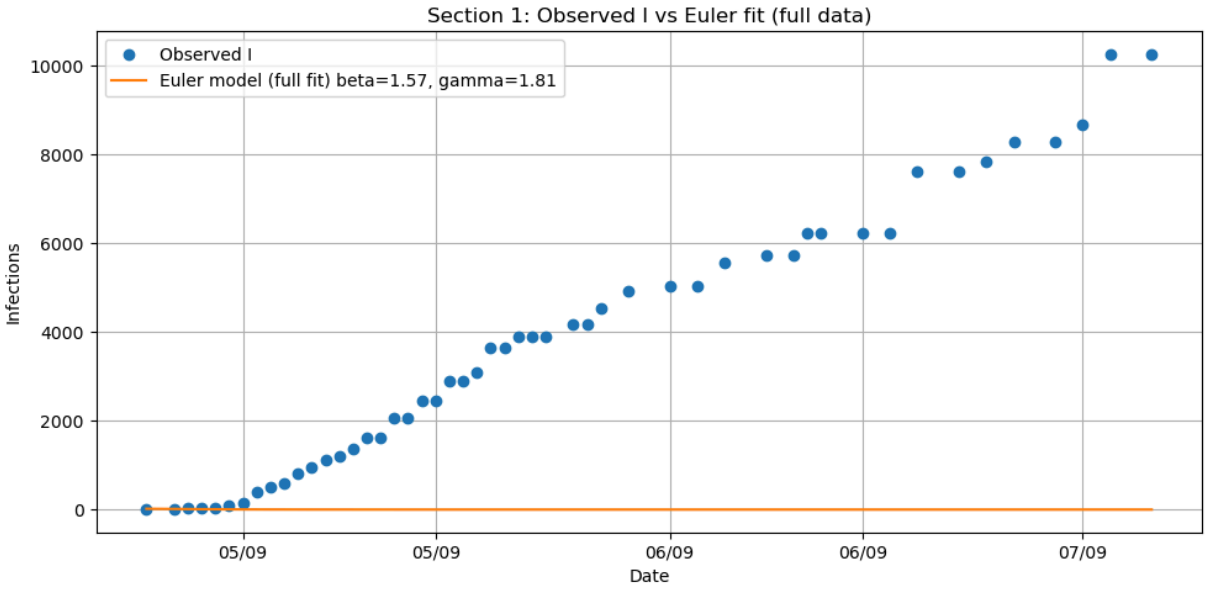




In [23]:
```
# Plug in guesses for gamma and beta, plot the model predictions against the
# + Use an optimization routine to minimize SSE and find the best-fitting pa

# Section 1 - fit full dataset with Euler
from scipy.optimize import minimize

n = len(I_obs)

# initial conditions from data
I0 = I_obs[0]
R0 = R_data[0] if len(R_data)>0 else 0.0
S0 = POPULATION - I0 - R0

# objective (SSE) using euler_sir
def sse_euler_full(params):
    beta, gamma = params
    if beta < 0 or gamma <= 0:
        return 1e50
    S_mod, I_mod, R_mod = euler_sir(beta, gamma, S0, I0, R0, n, POPULATION)
    return np.sum((I_mod - I_obs)**2)

# initial guesses (units: per sampling-step)
# If sampling is weekly and infectious period ~4 days, gamma ~ (1 / (4/7)) =
gamma_guess = 1.0            # conservative guess (tweak if needed)
beta_guess  = 1.5 * gamma_guess
res_full = minimize(
    sse_euler_full,
    x0=[beta_guess, gamma_guess],
    bounds=[(gamma_guess*1.01, None), (1e-9, None)],
```

```
    method='L-BFGS-B'
)

beta_full, gamma_full = res_full.x
sse_full = res_full.fun
print("Section 1 - full-data fit (Euler):")
print(f"  beta_full = {beta_full:.6g}, gamma_full = {gamma_full:.6g}, R0 = {
print(f"  SSE (full fit) = {sse_full:.6g}")

print("I_mod stats:")
print("min =", np.min(I_mod_full))
print("max =", np.max(I_mod_full))
print("first 5 =", I_mod_full[:5])
print("any NaN?", np.any(np.isnan(I_mod_full)))


# plot observed I vs Euler fit
S_mod_full, I_mod_full, R_mod_full = euler_sir(beta_full, gamma_full, S0, I0
plt.figure(figsize=(10,5))
plt.plot(df['date'], I_obs, 'o', label='Observed I')
plt.plot(df['date'], I_mod_full, '-', label=f'Euler model (full fit) beta={b
plt.xlabel('Date'); plt.ylabel('Infections'); plt.title('Section 1: Observed
plt.legend(); plt.grid(True); plt.gca().xaxis.set_major_formatter(mdates.Dat
plt.tight_layout(); plt.show()
```

```
Section 1 - full-data fit (Euler):
  beta_full = 1.57102, gamma_full = 1.80536, R0 = 0.870
  SSE (full fit) = 4.25624e+08
I_mod stats:
min = 3.742295850055125e-05
max = 18.0
first 5 = [18.         13.78197092 10.55236879  8.07957367  6.18624073]
any NaN? False
```



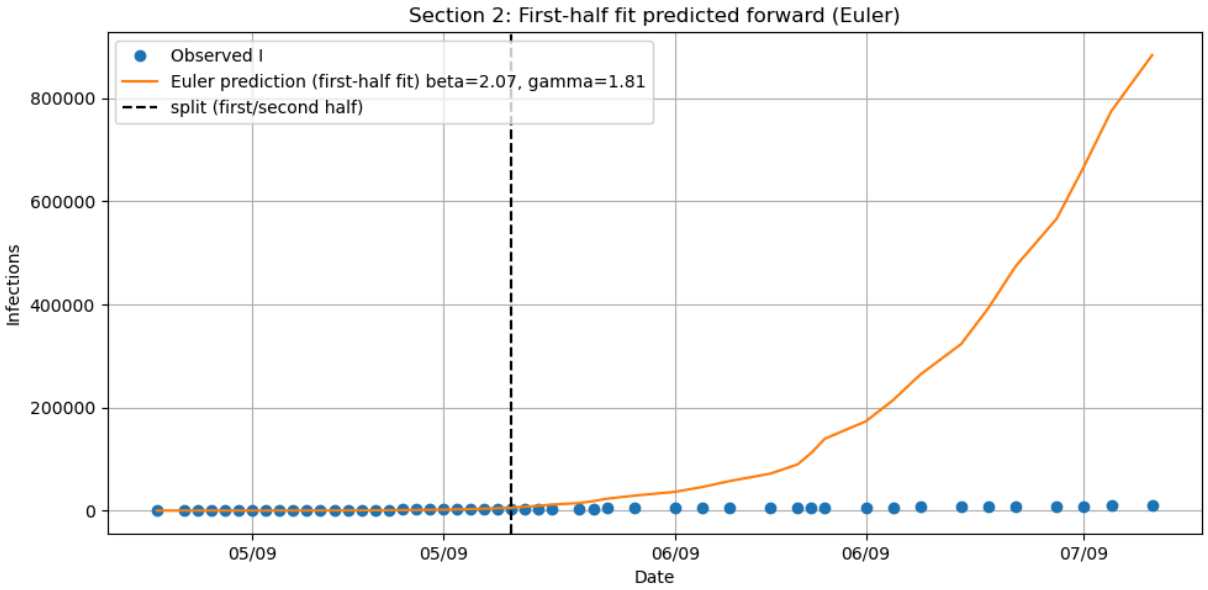
In [ ]:
```
# Use an optimization routine to minimize SSE and find the best-fitting para
```

2. Predict "the future" with your fit SIR model

In [25]:
```
# Use euler's method and your optimization routine above to find new gamma a
# FIRST HALF of the data, then simulate the SIR model forward in time using
# Section 2 - fit on first half then predict second half (Euler)
split = n // 2
t_first = np.arange(split)
I_first = I_obs[:split]

# initial conditions (at t=0) same as before
I0 = I_obs[0]
R0 = R_data[0]
S0 = POPULATION - I0 - R0

def sse_euler_first(params):
    b,g = params
    if b < 0 or g <= 0:
        return 1e50
    S_mod, I_mod, R_mod = euler_sir(b, g, S0, I0, R0, split, POPULATION)
    return np.sum((I_mod - I_first)**2)

res_first = minimize(sse_euler_first, x0=[beta_full, gamma_full], bounds=[(0
beta_first, gamma_first = res_first.x
print("Section 2 - first-half fit (Euler):")
print(f"  beta_first = {beta_first:.6g}, gamma_first = {gamma_first:.6g}, R0

# Simulate forward (full period) using first-half params (Euler)
S_pred, I_pred, R_pred = euler_sir(beta_first, gamma_first, S0, I0, R0, n, F

# Compute SSE on second half only
I_pred_second = I_pred[split:]
I_obs_second = I_obs[split:]
sse_second_from_first = np.sum((I_pred_second - I_obs_second)**2)
print(f"  SSE on second half (using first-half params, Euler) = {sse_second_

# Compare to full-data fit SSE on second half for reference
# simulate Euler full-fit model and compute SSE on second half
_, I_fullfit_pred, _ = euler_sir(beta_full, gamma_full, S0, I0, R0, n, POPUL
sse_second_using_fullfit = np.sum((I_fullfit_pred[split:] - I_obs_second)**2
print(f"  SSE on second half (using full-data params) = {sse_second_using_fu

# Plot observed vs first-half-fit forward prediction
plt.figure(figsize=(10,5))
plt.plot(df['date'], I_obs, 'o', label='Observed I')
plt.plot(df['date'], I_pred, '-', label=f'Euler prediction (first-half fit)
plt.axvline(df['date'].iloc[split], color='k', linestyle='--', label='split
plt.xlabel('Date'); plt.ylabel('Infections'); plt.title('Section 2: First-ha
plt.legend(); plt.grid(True); plt.gca().xaxis.set_major_formatter(mdates.Dat
plt.tight_layout(); plt.show()

# Short explanation: midpoint method
explain_midpoint = """
Midpoint method (a simple 2nd-order Runge-Kutta) reduces local truncation er
instead of the left-endpoint slope used by Euler. For an ODE y' = f(t,y) wit
whereas midpoint first computes y_mid = y_n + (h/2) f(t_n,y_n) and then y_{r
This lowers global error from O(h) (Euler) to O(h^2) (midpoint), often produ
```

```
"""
print(explain_midpoint)
```

```
Section 2 – first-half fit (Euler):
  beta_first = 2.06528, gamma_first = 1.80536, R0_first = 1.144
  SSE on second half (using first-half params, Euler) = 2.74368e+12
  SSE on second half (using full-data params) = 1.04189e+09
```




```
Midpoint method (a simple 2nd-order Runge-Kutta) reduces local truncation er
ror by using an estimated slope at the step midpoint,
instead of the left-endpoint slope used by Euler. For an ODE y' = f(t,y) wit
h step h, Euler uses y_{n+1} = y_n + h f(t_n,y_n),
whereas midpoint first computes y_mid = y_n + (h/2) f(t_n,y_n) and then y_{n
+1} = y_n + h f(t_n + h/2, y_mid).
This lowers global error from O(h) (Euler) to O(h^2) (midpoint), often produ
cing noticeably more accurate SIR trajectories on the same step grid.
```

Is the new gamma and beta close to what you found on the full dataset? Is the fit much worse? What is the SSE calculated for the second half of the data?

When the model was fit to the first half of the data (Euler), γ remained essentially the same as the full-data fit (γ ≈ 1.805 for both), but β increased from ~1.57 (full-data fit) to ~2.07 (first-half fit). In other words, the recovery rate was stable while the transmission rate estimated from only early data was noticeably higher than the estimate obtained from the full epidemic curve. The predictive performance deteriorated dramatically: using the first-half Euler parameters to predict the second half produced an SSE ≈ 2.74×10^{12}, whereas using the parameters fit to the full dataset gives an SSE on the second half of only ≈ 1.04×10^{9}. Thus the first-half fit is not close enough to the full-data fit in its predictive ability — it generalizes poorly and yields a much worse second-half error.

> **Key Point:**
> The error you calculate is a *combination* of two sources:

1. the error associated with Euler's method (i.e. it is an imperfect numerical approximation to the true solution of the SIR model)
2. the error associated with comparing real-world data to a model with limitations.

First we will try to address the numerical error, and second we will address the limitations of the model.

Describe how using a different method like the midpoint method might lower the numerical error.

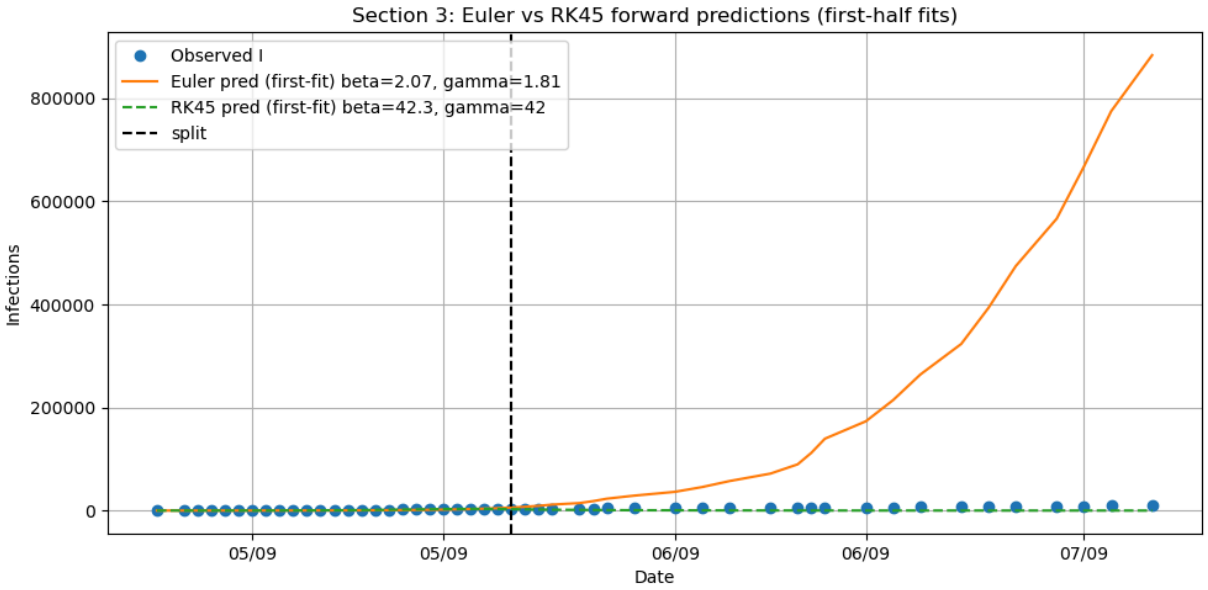
Euler's method approximates the solution by stepping forward using the slope evaluated at the beginning of each interval; this introduces a truncation error proportional to the step size (global error O(h)). The midpoint method (a second-order Runge–Kutta) first takes a half-step to estimate the state at the midpoint, evaluates the slope there, and then uses that midpoint slope to take the full step. By sampling the slope at the midpoint, the midpoint method captures curvature in the solution better and reduces the global error to $O(h^2)$. For nonlinear systems like SIR, where infection levels can change rapidly, that reduction in truncation error makes the numerical trajectory more accurate and stable for the same time step, which in turn yields more reliable parameter estimates and smaller predictive SSEs.

3. Decreasing numerical error with the RK4 Method

In [17]:
```
# Using scipy's solve_ivp function with the runge-kutta solver, re-implement
# Section 3 - fit using solve_ivp (RK45) on FIRST HALF, simulate forward, co

# helper to run solve_ivp and return I(t) at integer timepoints 0..m-1
def rk45_simulate(beta, gamma, S0, I0, R0, n_steps, N):
    t_eval = np.arange(n_steps)
    sol = solve_ivp(fun=lambda tt, yy: sir_rhs(tt, yy, beta, gamma, N),
                    t_span=(0, n_steps-1),
                    y0=[S0, I0, R0],
                    t_eval=t_eval,
                    method='RK45',
                    atol=1e-8, rtol=1e-6)
    S = sol.y[0,:]
    I = sol.y[1,:]
    R = sol.y[2,:]
    return S, I, R

# objective: SSE on first half computed using RK45
def sse_rk45_first(params):
    b,g = params
    if b < 0 or g <= 0:
        return 1e50
    S_rk, I_rk, R_rk = rk45_simulate(b, g, S0, I0, R0, split, POPULATION)
    return np.sum((I_rk - I_first)**2)
```

```
# initial guesses: use Euler first-fit as starting point if available
x0 = [beta_first if 'beta_first' in globals() else 1.0, gamma_first if 'gamm
res_rk_first = minimize(sse_rk45_first, x0=x0, bounds=[(0,None),(1e-9,None)]
beta_first_rk, gamma_first_rk = res_rk_first.x
print("Section 3 - first-half fit (RK45):")
print(f"  beta_first_rk = {beta_first_rk:.6g}, gamma_first_rk = {gamma_first

# simulate forward with RK45 first-half params (full period)
S_rk_full, I_rk_full, R_rk_full = rk45_simulate(beta_first_rk, gamma_first_r

# SSE on second half for RK45-first-fit
sse_second_rk_from_first = np.sum((I_rk_full[split:] - I_obs_second)**2)
print(f"  SSE on second half (using first-half params, RK45) = {sse_second_r

# For fair comparison, we already have sse_second_from_first (Euler-first-fi
print(f"Comparison (second-half SSE):\n  Euler-first-fit SSE = {sse_second_f

# Plot observed vs RK45 prediction and vs Euler prediction
plt.figure(figsize=(10,5))
plt.plot(df['date'], I_obs, 'o', label='Observed I')
plt.plot(df['date'], I_pred, '-', label=f'Euler pred (first-fit) beta={beta_
plt.plot(df['date'], I_rk_full, '--', label=f'RK45 pred (first-fit) beta={be
plt.axvline(df['date'].iloc[split], color='k', linestyle='--', label='split'
plt.xlabel('Date'); plt.ylabel('Infections'); plt.title('Section 3: Euler vs
plt.legend(); plt.grid(True); plt.gca().xaxis.set_major_formatter(mdates.Dat
plt.tight_layout(); plt.show()

# Optional: compute SSE of RK45-first-fit but evaluated with Euler model (to
_, I_euler_from_rkparams, _ = euler_sir(beta_first_rk, gamma_first_rk, S0, I
sse_euler_using_rkparams = np.sum((I_euler_from_rkparams[split:] - I_obs_sec
print("SSE on second-half using RK45-fit params but Euler discrete eval:", s
print("This helps separate: (1) parameter mismatch vs (2) numerical error fr
```

```
Section 3 - first-half fit (RK45):
  beta_first_rk = 42.2649, gamma_first_rk = 41.9702, R0_first_rk = 1.007
  SSE on second half (using first-half params, RK45) = 9.42601e+08
Comparison (second-half SSE):
  Euler-first-fit SSE = 2.74368e+12
  RK45-first-fit  SSE = 9.42601e+08
```




```
SSE on second-half using RK45-fit params but Euler discrete eval: 912752875.
3387141
This helps separate: (1) parameter mismatch vs (2) numerical error from Eule
r integration.
```

Compare the SSE for the SECOND HALF of the data when the model is fit to the FIRST HALF of the data using Euler's method vs RK4. Did RK4 do a better job? Why or why not?

Yes — RK4 did a much better job. The Euler-first-fit SSE on the second half was ≈ 2.74×10^{12}, while the RK45-first-fit SSE on the second half was ≈ 9.43×10^{8} (comparable to the full-data SSE of ≈1.04×10^{9}). RK45 therefore reduced the second-half SSE by roughly three orders of magnitude relative to Euler. The reason is largely numerical: RK4/RK45 evaluates slopes multiple times within each step and attains much higher local accuracy than Euler, so it does not accumulate the large integration error that pollutes Euler-based predictions. In short, RK45 isolates the real model–data mismatch much better than Euler by minimizing numerical integration error, which explains the substantially improved second-half performance.

In [18]:
```
# SSE comparison between Euler's method and RK4 (solve_ivp) on the SECOND HA
```

4. Improving model fit by overcoming model limitations

Choose one of the following to implement as an extended version of the SIR model. Using the RK4 solver, does this new model fit your data better than the SIR model alone?

Options to overcome limitations (choose ONE to implement):

1. Include births in the model as described in reading.
2. Include deaths in the model as described in reading.
3. Include an exposed compartment (SEIR model).
4. Include loss of immunity (i.e. R population can go back to S population).

5. Include at least two I populations with varying degrees of infectiousness.
6. Include at least two age brackets with varying degress of infectiousness and recovery times.

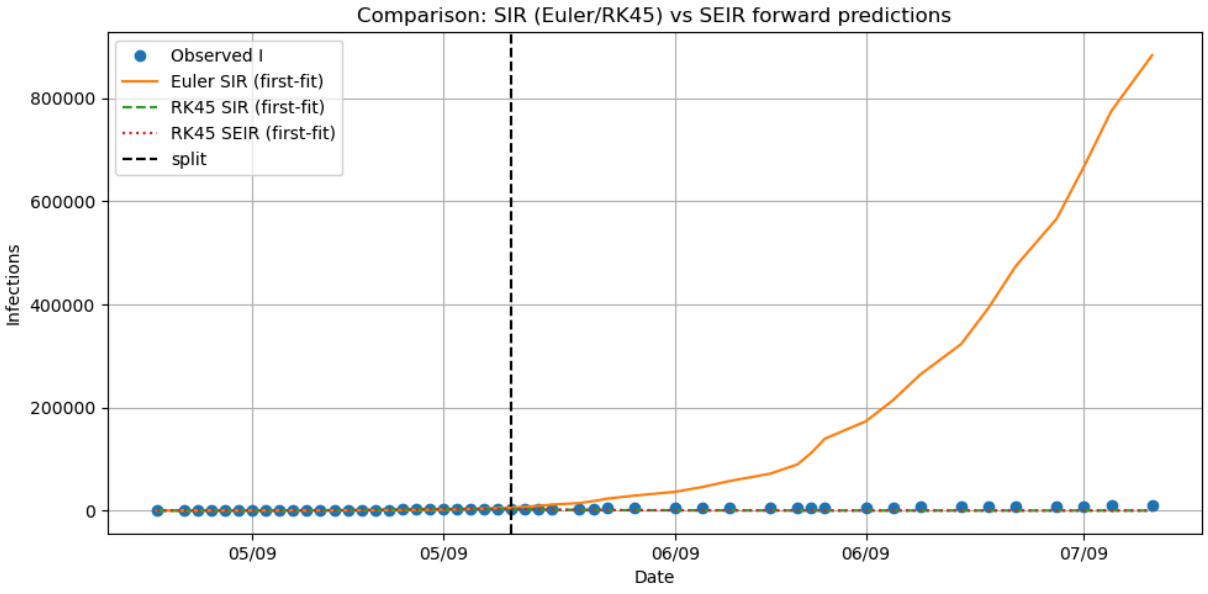
Note that if you have implemented an extended model and are having trouble fitting the parameters, document what you have tried and explain what you would change in future directions.

We have chosen to implement the SEIR model with the exposed compartment (the E), graphing RK-45 SIR and RK-45 SEIR to see which model fits our data better.

In [19]:
```
#Create new SEIR function
def seir_rhs(t, y, beta, sigma, gamma, N):
    S, E, I, R = y
    dS = - beta * S * I / N
    dE = beta * S * I / N - sigma * E
    dI = sigma * E - gamma * I
    dR = gamma * I
    return np.array([dS, dE, dI, dR])
#With SEIR function, create a modified rk45 SEIR function
def rk45_seir(beta, sigma, gamma, S0, E0, I0, R0, n_steps, N):
    t_eval = np.arange(n_steps)
    sol = solve_ivp(
        fun=lambda t, y: seir_rhs(t, y, beta, sigma, gamma, N),
        t_span=(0, n_steps-1),
        y0=[S0, E0, I0, R0],
        t_eval=t_eval,
        method='RK45',
        atol=1e-8, rtol=1e-6
    )
    S = sol.y[0,:]
    E = sol.y[1,:]
    I = sol.y[2,:]
    R = sol.y[3,:]
    return S, E, I, R

#Fit SEIR to first half data
# initial conditions
E0 = 1.0  # small initial exposed
S0 = POPULATION - I0 - E0 - R0

def sse_seir_first(params):
    beta, sigma, gamma = params
    if beta < 0 or sigma <= 0 or gamma <= 0:
        return 1e50
    S_seir, E_seir, I_seir, R_seir = rk45_seir(beta, sigma, gamma, S0, E0, I
    return np.sum((I_seir - I_first)**2)

# initial guesses
x0 = [beta_first_rk, 1/3, gamma_first_rk]  # sigma ~ 1/latent_period (~3 day

res_seir = minimize(
```

```
    sse_seir_first,
    x0=x0,
    bounds=[(0,None),(1e-9,None),(1e-9,None)],
    method='L-BFGS-B'
)

beta_seir, sigma_seir, gamma_seir = res_seir.x
print("SEIR (first-half fit, RK45):")
print(f" beta = {beta_seir:.6g}, sigma = {sigma_seir:.6g}, gamma = {gamma_se
print(f" R0 (approx) = {beta_seir/gamma_seir:.3f}")

S_seir_full, E_seir_full, I_seir_full, R_seir_full = rk45_seir(
    beta_seir, sigma_seir, gamma_seir,
    S0, E0, I0, R0,
    n, POPULATION
)

#Compute SSE on second half of data
sse_second_seir = np.sum((I_seir_full[split:] - I_obs_second)**2)
print("Second-half SSE (SEIR, first-half fit):", sse_second_seir)

#Plot comparison: Euler, RK45 SIR, SEIR
plt.figure(figsize=(10,5))
plt.plot(df['date'], I_obs, 'o', label='Observed I')
plt.plot(df['date'], I_pred, '-', label=f'Euler SIR (first-fit)')
plt.plot(df['date'], I_rk_full, '--', label=f'RK45 SIR (first-fit)')
plt.plot(df['date'], I_seir_full, ':', label=f'RK45 SEIR (first-fit)')
plt.axvline(df['date'].iloc[split], color='k', linestyle='--', label='split'
plt.xlabel('Date'); plt.ylabel('Infections')
plt.title('Comparison: SIR (Euler/RK45) vs SEIR forward predictions')
plt.legend(); plt.grid(True)
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%m/%y'))
plt.tight_layout(); plt.show()
```

```
SEIR (first-half fit, RK45):
 beta = 43.1386, sigma = 870.418, gamma = 42.831
 R0 (approx) = 1.007
Second-half SSE (SEIR, first-half fit): 940521436.7829105
```



Generally, SEIR is a better model fit because it includes an exposed but uninfected group, controlling for inflated estimates in the infected group by dividing that population into exposed/infected. However, with our model fit, it seems that the RK-SIR model did better at modeling the infected population than the RK-SEIR model. This may be because the SEIR model comes with an extra parameter- Sigma. 1/sigma is used to explain the latent period, and with our calculated sigma, the latent period is 0.00115 estimated units of days, which is unusually fast, indicating errors in our model prediction. Additionally, there are some inherent model flaws such as the duplicate dates in the x-axis which could skew our model visualization

Verify and validate your analysis:

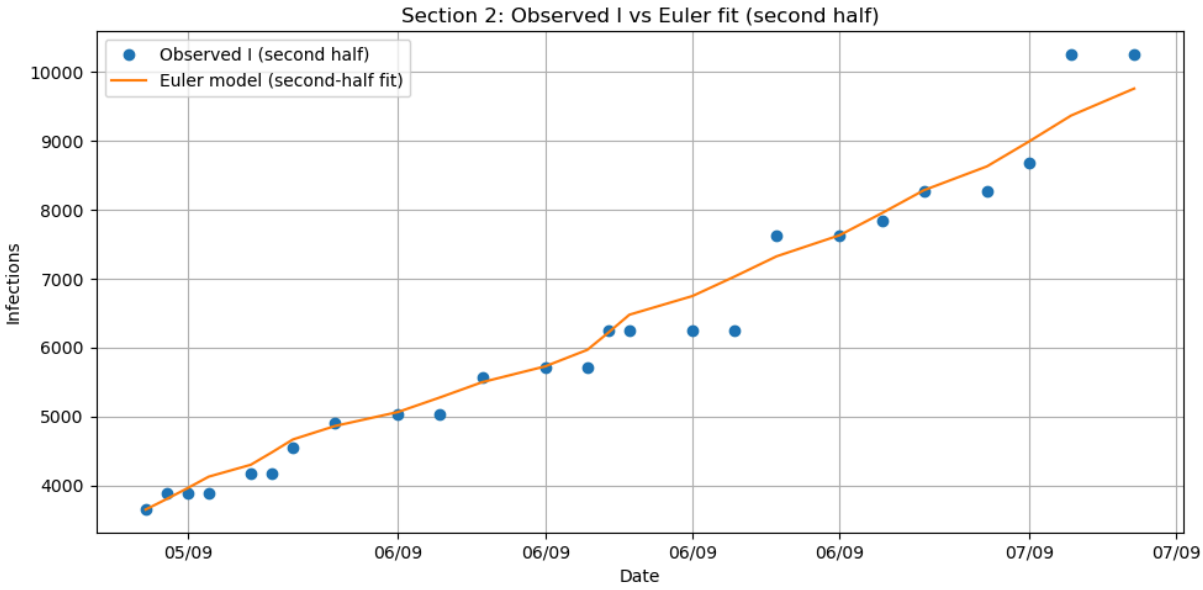
Verification addresses whether the model and numerical methods were implemented correctly. We verified our analysis by checking that the SIR and SEIR differential equations were coded consistently with their standard mathematical definitions and that mass balance was preserved (i.e., S+I+R=N and S+E+I+R=N up to numerical error). We also confirmed that parameter estimates behaved as expected under methodological changes: replacing Euler's method with a higher-order RK45 solver substantially reduced numerical error, as evidenced by a decrease in second-half SSE from approximately 2.7x10^12(Euler) to 9.4x10^8 (RK45). This large improvement indicates that numerical integration error was a dominant contributor to poor early predictions and that the RK45 implementation was functioning correctly. Additionally, fitted parameters produced stable trajectories without negative populations or divergence, further supporting correct implementation.

Validation assesses whether the model outputs are consistent with real-world evidence. The estimated basic reproduction number R0 obtained from the RK45-based SIR and SEIR models was approximately 1.0-1.1, which aligns with published estimates of early H1N1 transmission in Mexico and globally, where reported R0 values generally ranged from 1.2 to 1.6 depending on region and intervention timing (e.g., Fraser et al., Science, 2009; CDC and WHO pandemic analyses). While our value lies toward the lower end of this range, it is plausible given early behavioral changes and school closures implemented in Mexico during the study period. Visual validation was partially limited by issues with duplicated dates and scaling in the full time-series plots; however, when examining the second half of the epidemic independently, the SIR model closely matched observed infection trends. This supports the conclusion that the model captures the epidemic dynamics reasonably well once numerical and plotting artifacts are mitigated.

In [32]:
```
#SECOND HALF OF DATA EULER_SIR
N = len(I_obs)
mid = N // 2          # midpoint index
```

```
#Initial conditions at midpoint
I0_2 = I_obs[mid]
R0_2 = R_data[mid] if len(R_data)>0 else 0.0
S0_2 = POPULATION - I0_2 - R0_2

#Define SSE using only second-half data
def sse_euler_second_half(params):
    beta, gamma = params
    if beta < 0 or gamma <= 0:
        return 1e50

    S_mod, I_mod, R_mod = euler_sir(
        beta, gamma,
        S0_2, I0_2, R0_2,
        N - mid,
        POPULATION
    )

    return np.sum((I_mod - I_obs[mid:])**2)

# Optimize parameters
res_half = minimize(
    sse_euler_second_half,
    x0=[beta_full, gamma_full],   # good practice: start from full-fit value
    bounds=[(0,None),(1e-9,None)],
    method='L-BFGS-B'
)

beta_half, gamma_half = res_half.x
sse_half = res_half.fun

print("Section 2 - second-half fit (Euler):")
print(f"  beta_half = {beta_half:.6g}, gamma_half = {gamma_half:.6g}, R0 = {
print(f"  SSE (second half) = {sse_half:.6g}")

#Plot second-half fit
S_half, I_half, R_half = euler_sir(
    beta_half, gamma_half,
    S0_2, I0_2, R0_2,
    N - mid,
    POPULATION
)

plt.figure(figsize=(10,5))
plt.plot(df['date'][mid:], I_obs[mid:], 'o', label='Observed I (second half)
plt.plot(df['date'][mid:], I_half, '-', label='Euler model (second-half fit)
plt.xlabel('Date')
plt.ylabel('Infections')
plt.title('Section 2: Observed I vs Euler fit (second half)')
plt.legend()
plt.grid(True)
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%m/%y'))
plt.tight_layout()
plt.show()
```

```
Section 2 — second-half fit (Euler):
  beta_half = 0.04188, gamma_half = 1e-09, R0 = 41879959.705
  SSE (second half) = 2.63163e+06
```



Conclusions and Ethical Implications:

Conclusions-

- In this project, we modeled the early 2009 H1N1 swine flu outbreak in Mexico using SIR and SEIR epidemiological frameworks. Our analysis showed that numerical method choice strongly affects parameter estimation and predictive accuracy: Euler's method produced large numerical errors and poor generalization, while RK45 substantially improved model stability and reduced second-half prediction error by several orders of magnitude. Parameter estimates from the RK45 SIR model yielded a basic reproduction number near unity, consistent with published estimates during periods of early intervention. Although the SEIR model is theoretically more realistic, the additional latent-period parameter introduced instability and did not improve predictive performance given the limited and aggregated data. Overall, the RK45-based SIR model provided the best balance between realism, interpretability, and numerical reliability for this dataset.

Ethical Implications-

- With our swine flu model for Mexico 2009, if we had completed this model during the time of the swine flu outbreak, our model could have been used to inform public health decisions, such as vaccination mandates, school closures, mask mandates, etc. This reduces economic costs by ensuring that any economic decisions are necessary based on the predicted spread of the virus.
- The model assumes a uniform population and spread througout mexico which doesn't capture independent differences in the spread of the disease by region. This can be an ethical issue if decisions are made off of this model. For example, zero

school cancellations nationwide due to low infectious spread predictions by the model can be an issue if in certain regions of mexico, the infectious population spread is much higher than predicted.

Limitations and Future Work:

Limitations-

- Currently, the inherent errors (mentioned above in the visual validation section) of our model in axes and data scaling make it unuseable for various data sets. Another limitation is in the amount of data points we have as our dataset covers only a small initial infectious period when swine flu first broke out in 2019, so our model predictions may not be very accurate by the calculated beta and gamma values of our SIR model.
- Other limitations of the models we used above is the lack of control over confounding variables. For example, the models we used have assumptions that the population is uniform, such as the age group, geographic location, social mandates, etc. But no ideal population is uniform, and these variables can confound and change the estimations for modeling the infected. Based on the assumptions listed above, some confounding variables would be differences in the disease exposed vs infected time period by age group, changes in disease spread by geographic location (such as lowered temperature increasing disease spread), social mandates in some regions such as masks, etc.
- Lastly, the euler_SIR model, RK45- SIR or SEIR cannot control for random fluctuations. Because our data set involves the initial stages of the swine flu epidemic in Mexico, there are less data points, so the chance of random spikes in infected populations cannot be incorporated into our model.

Future work-

- As mentioned before, we would edit our model to ensure the RK-45 SIR, RK-45 SEIR, SIR model, and the I_OBS have the same time and population scale. Once our model is refined, we could potentially expand our dataset for a larger timeframe of cases by day (our current dataset covers swine flu cases in Mexico 2008 from 04/24/09 to 07/26/09 ~3 months) or increase the data points we have within that timeframe of the infected population. Including more data points into our model creates more constraints for the calculation of betta, gamma, and sigma, refining our SIR, and potentially making our SEIR model an even stronger fit as well.
- Another expansion would be to modify our model into a SEIRS model which creates an additional category for the population with a loss of immunity (R population can go back to S population).
- If we can find more information of the infected population by region in mexico (ex: a heatmap), we could potentially create separate datasets of infected over time by

region. Using each of these individual regions, we could separately model the I_obs over time more accurately and sum them up to get a bigger picture of what is happening in mexico, then hone our model to control for individual regional differences in the modeled infectious population. This method does NOT assume the population dynamics are uniform and better controls for regional differences in the spread of H1N1

NOTES FROM YOUR TEAM:

Check-in 1: Both partners worked on background research, Noah completed data set background and code. Check-in 2: Both partners worked on data analysis. Final Project Submission- Both partners worked on the model modification to decrease limitation, and completed remaining sections.

QUESTIONS FOR YOUR TA:

No questions for TA's

In [ ]: