Matrix Structural Analysis

Group 3rd Report

DEPARTMENT OF CIVIL ENGINEERING

ACADEMIC YEAR: 2025-2026

GROUP 3rd

NAME	ROLL NUMBER
Shubham Suman	23CE30029
Ranvir Kumar	23CE10050
Mayukh Mondal	23CE30020
Rangoli Hanisree	23CE10056
Shashank Sahil	23CE10065

TABLE OF CONTENTS

Executive Summary

The **Forth Bridge**, located in Scotland, is a pioneering example of 19th-century engineering and a symbol of industrial innovation. Spanning the Firth of Forth near Edinburgh, this cantilever railway bridge serves as a vital connection between the northeast and south of the country, carrying rail traffic across one of the most challenging estuaries in the region. This report presents a detailed structural analysis of the Forth Bridge's steel cantilever framework under various loading conditions, including train loads and environmental forces such as wind and temperature effects. The study utilizes advanced structural analysis techniques to evaluate deflection, internal forces, and support reactions, highlighting the bridge's remarkable strength, stability, and endurance.

The design of the Forth Bridge represents a milestone in civil engineering, combining innovation, safety, and functionality to address the demands of modern transportation during its time. Its massive steel structure was designed to withstand both static and dynamic loads while maintaining stability under severe weather and fatigue conditions. Through analytical modeling, this study emphasizes how the cantilever design effectively distributes loads, ensuring minimal deflection and long-term performance.

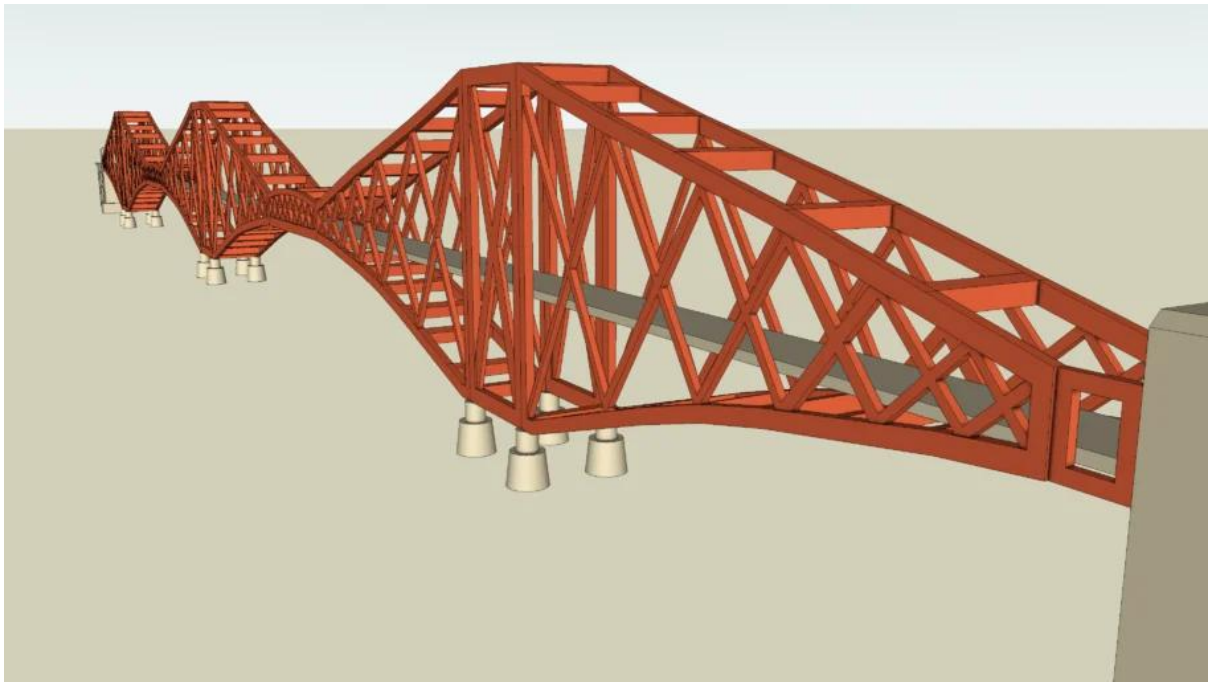
Structural analysis of iconic bridges like the Forth Bridge remains essential for preservation and safety. By continuously assessing stress distribution and deformation patterns, engineers can monitor the bridge's health and anticipate potential deterioration. This approach ensures not only the safety and reliability of railway operations but also contributes to the longevity of a globally recognized heritage structure maintaining its role as both a functional transport link and an enduring engineering masterpiece.

Background

The **Forth Bridge**, situated across the Firth of Forth in eastern Scotland, stands as one of the most iconic and historically significant railway bridges in the world. Completed in **1890**, it was the first major steel structure in Britain and, at the time of its construction, the **longest cantilever railway bridge** ever built. The bridge spans a total length of **2,528 meters**, with its three double cantilevers connected by two central suspended spans, carrying the double-track railway between **South Queensferry and North Queensferry**.

Designed by **Sir John Fowler** and **Sir Benjamin Baker**, the Forth Bridge was constructed to provide a safer and more direct rail route from Edinburgh to the north of Scotland, replacing unreliable ferry services. Its construction required over **53,000 tonnes of steel** and the labour of more than **4,000 workers**, marking a turning point in civil and structural engineering. The bridge's innovative cantilever design allowed it to endure heavy train loads and the harsh maritime climate of the Scottish coast.

Recognized as a **UNESCO World Heritage Site in 2015**, the Forth Bridge remains a functional and vital part of the UK's rail network. It is designed to withstand strong winds, temperature variations, and continuous train vibrations, reflecting exceptional durability and craftsmanship. Beyond its engineering excellence, the bridge has immense cultural and economic significance, symbolizing Scotland's industrial heritage and serving as a major transport link that continues to support regional connectivity and economic development more than a century after its completion.



Aeroelastic model of the bridge

Design & Engineering

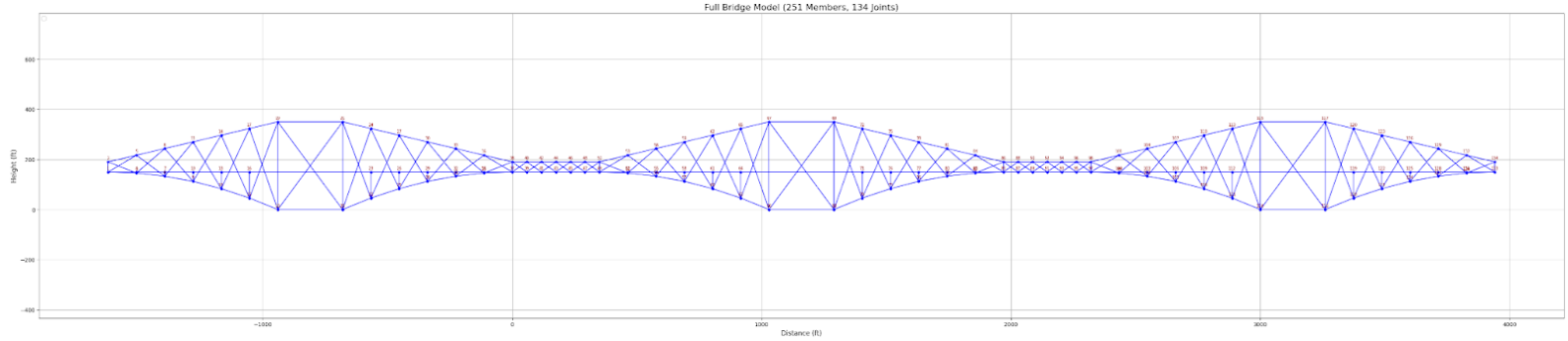
The **Forth Bridge** is a pioneering example of **cantilever steel bridge design**, representing one of the most ambitious engineering feats of the late 19th century. Its distinctive **double-cantilever structure**, connected by suspended spans, was designed to provide exceptional strength and stability across the **Firth of Forth's 2,528-meter span**. The bridge carries a **double-track railway**, supported by **three main towers** and a system of **massive cantilever arms** that balance the loads efficiently without the need for intermediate supports in the deep waters below.

Constructed primarily from **wrought iron and mild steel**, the bridge utilized over **53,000 tonnes of steel**, with each cantilever arm anchored by enormous masonry piers founded on bedrock. The structural form distributes compressive and tensile forces effectively one arm in tension and the other in compression allowing it to sustain heavy train loads while maintaining overall equilibrium. The design also incorporates **9 million rivets**, ensuring rigidity and long-term durability under repetitive stress cycles.

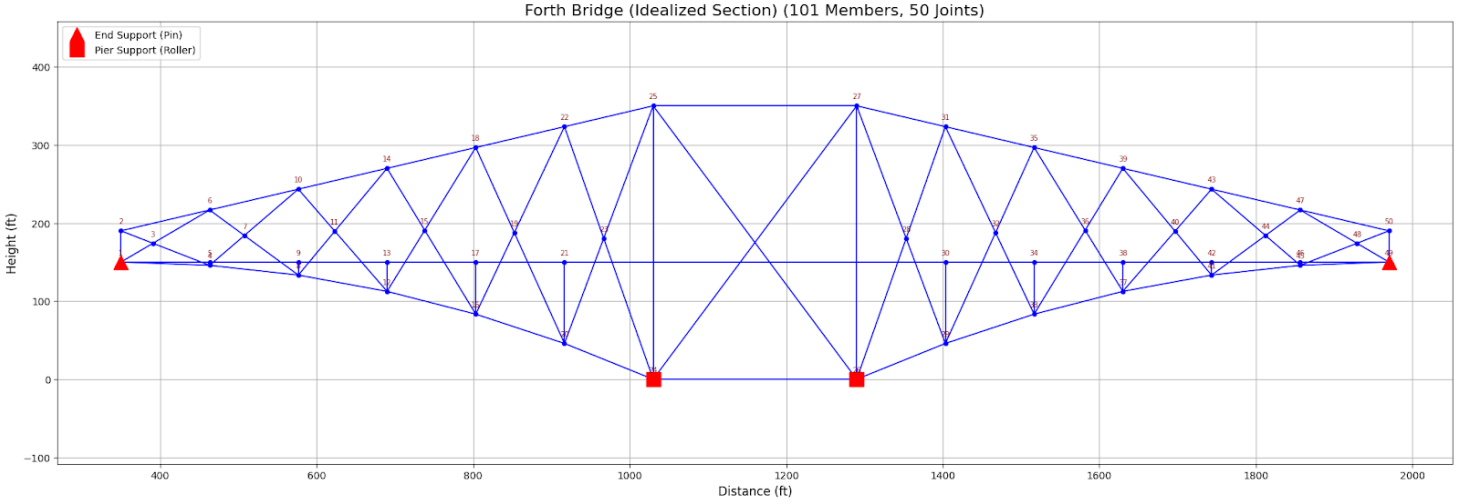
Given its exposed coastal location, the Forth Bridge was engineered to withstand **high wind pressures, temperature variations, and fatigue** from continuous rail traffic. It was one of the first major bridges to undergo **scientific stress analysis** during its design phase, ensuring that it could endure the region's harsh maritime climate. Modern conservation efforts have included **advanced corrosion protection and repainting techniques**, ensuring structural integrity well into the 21st century.

The bridge's design principles redundancy, balance, and robustness set new standards for bridge engineering worldwide. Its ability to support trains for over a century with minimal deflection and structural deterioration remains a testament to the foresight and precision of its designers, **Sir John Fowler** and **Sir Benjamin Baker**, and to the enduring legacy of Victorian engineering excellence.

Full Bridge Model



Idealized Section of the bridge



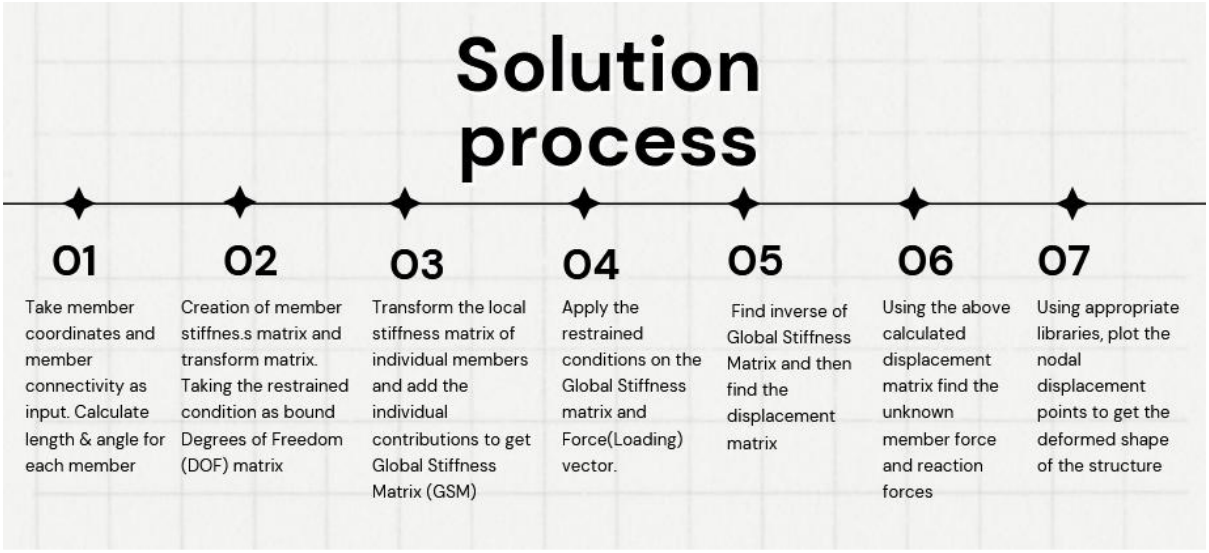
Problem Statement:

Perform a comprehensive structural analysis of a representative 2D frame/cantilever model of the Forth Bridge using matrix structural analysis techniques. The model will simulate static and dynamic behavior under combined load cases including: railway live loads (single-axle and train-consistent load patterns), dead loads, wind pressures (steady and gust components), thermal gradients (uniform and differential), and fatigue-inducing load cycles. Additional scenarios will examine extreme events such as abnormal impact loads, differential support settlement, and rare environmental combinations. The aim is to evaluate load distribution, member forces, support reactions, deflections, and vibration characteristics to confirm the bridge's capacity to maintain serviceability and structural integrity.

Solution Approach

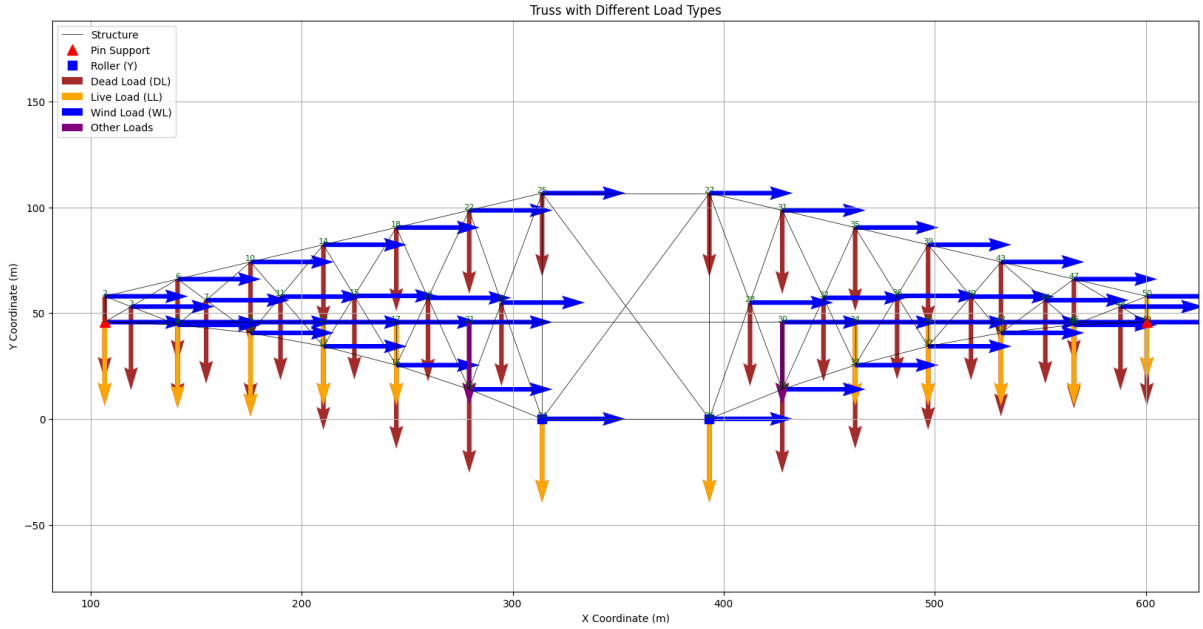
The structural analysis approach leverages the **Matrix Structural Analysis method** to break down the bridge into discrete elements, allowing for detailed assessment of load distribution, deflection, and internal forces. The main steps involved are:

1. **Structural Modeling:** The bridge is modeled as a 2D frame with steel members for simplicity. A minimum of 50 nodes ensures an accurate depiction of the deflected shape.
2. **Load Application:** Live loads (train weight) and wind loads are applied according to standards.
3. **Boundary Conditions:** Supports and constraints are defined based on actual bridge anchoring.
4. **Deflection and Reaction Analysis:** The FEM analysis provides deflection at key points, reaction forces at supports, and internal forces within members.
5. **Software Utilization:** Modern programming languages such as MATLAB software facilitate the complex calculations and plotting required for this study



Loads Acting on the Bridge

1. Wind load
2. Dead load
3. Live load



Code and Results

The analysis yielded several crucial insights into the structural integrity of the Forth Bridge:

● **Deflected Shape:** The deflection plots indicate that the bridge can withstand extreme loads with limited deformation, maintaining safety for passengers.By knowing the deflection,we can change the required steel properties, to keep the deflections well under the safe deflections as per the codes.
● **Reactions at Supports:** Reaction forces were highest at the arch bases, where most stress is concentrated, confirming design efficiency in load distribution.
● **Internal Forces:** Bending moments, shear forces, and axial forces were highest at the main arch and key joints, aligning with expected high-stress zones.

Code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#  FUNCTION: gauss_jordan(A, b)
#  Solves the linear system [A]{x} = {b} using Gauss-Jordan elimination

def gauss_jordan(A, b):
    """
    Solves the linear system Ax = b using Gauss-Jordan elimination
    with partial pivoting.
    """
    A = A.astype(float)
    b = b.astype(float)
    n = len(b)

    if b.ndim == 1:
        b = b.reshape(-1, 1)
    Ab = np.hstack([A, b])

    for i in range(n):
        # 1. Partial Pivoting
        pivot_row = i
        for k in range(i + 1, n):
            if abs(Ab[k, i]) > abs(Ab[pivot_row, i]):
                pivot_row = k
        Ab[[i, pivot_row]] = Ab[[pivot_row, i]]

        # 2. Handle Singular Matrix
        if abs(Ab[i, i]) < 1e-12:
            raise ValueError("Matrix is singular or near-singular.")

        # 3. Normalize the pivot row
        pivot = Ab[i, i]
        Ab[i, :] = Ab[i, :] / pivot

        # 4. Eliminate other rows
        for j in range(n):
            if i != j:
                factor = Ab[j, i]
                Ab[j, :] = Ab[j, :] - factor * Ab[i, :]

    x = Ab[:, -1]
    return x
```

```
#  FUNCTION: plot_truss(...)
#  Plots the original vs. deformed shape of the truss.

def plot_truss(orig_coords, def_coords, members, scale_factor, supports):
    """
    Plots the original and deformed truss shapes.
    """
    print(f"\n... Plotting deformed shape with scale factor: {scale_factor} ...")

    plt.figure(figsize=(20, 10))

    # Plot original structure (blue, solid)
    for idx, member in members.iterrows():
        j1 = member['joint 1'] - 1
        j2 = member['joint 2'] - 1
        x1, y1 = orig_coords[j1]
        x2, y2 = orig_coords[j2]
        label = 'Original' if idx == 0 else None
        plt.plot([x1, x2], [y1, y2], 'b-', label=label, linewidth=0.5)

    # Plot deformed structure (red, dashed)
    for idx, member in members.iterrows():
        j1 = member['joint 1'] - 1
        j2 = member['joint 2'] - 1
        x1_d, y1_d = def_coords[j1]
        x2_d, y2_d = def_coords[j2]
        label = f'Deformed (x{scale_factor})' if idx == 0 else None
        plt.plot([x1_d, x2_d], [y1_d, y2_d], 'r--', label=label, linewidth=1.0)

    # Plot support symbols on the original structure
    legend_labels = set()
    for joint_num, support_type in supports.items():
        joint_idx = joint_num - 1
        x, y = orig_coords[joint_idx]
```

```
        if support_type == 'pin':
            label = 'Pin Support' if 'pin' not in legend_labels else None
            plt.plot(x, y, 'r^', markersize=15, label=label)
            legend_labels.add('pin')
        elif support_type == 'roller_y':
            label = 'Roller Support' if 'roller_y' not in legend_labels else None
            plt.plot(x, y, 'rs', markersize=15, label=label)
            legend_labels.add('roller_y')
        elif support_type == 'roller_x':
            label = 'Roller Support' if 'roller_x' not in legend_labels else None
            plt.plot(x, y, 'rs', markersize=15, label=label)
            legend_labels.add('roller_x')

    # Plot joint numbers on original structure
    for i, (x, y) in enumerate(orig_coords):
        plt.text(x, y + 5, str(i + 1), fontsize=8, color='darkgreen', ha='center')

    plt.title('Truss Deformation')
    plt.xlabel('X Coordinate (m)')
    plt.ylabel('Y Coordinate (m)')
    plt.legend()
    plt.grid(True)
    plt.axis('equal')
    plt.show()

#  MAIN FUNCTION: solve_truss()
#  Defines, assembles, solves, and reports the 2D truss analysis.
```

```
#  MAIN FUNCTION: solve_truss()
#  Defines, assembles, solves, and reports the 2D truss analysis.

def solve_truss():
    """
    Main function to solve the 2D truss problem.
    """

    # --- 1a. Material and Section Properties ---
    # Define material (E = Young's Modulus) and member
    # size (A = Cross-Sectional Area).
    # Define variables in base units (Pascals and m^2).

    print("Initializing analysis...")
    E_GPa = 200.0       # GigaPascals (for steel)
    A_mm2 = 1000.0      # square mm

    E_Pa = E_GPa * 1e9  # Pascals (N/m^2)
    A_m2 = A_mm2 * 1e-6 # square m
    EA = E_Pa * A_m2    # N (Stiffness constant)
    print(f"  Material: E = {E_GPa} GPa")
    print(f"  Section:  A = {A_mm2} mm^2")

    # --- File and Plotting Inputs ---
    excel_file = 'Structure_data_idelized.xlsx'
    plot_scale_factor = 50.0 # Factor to make deformation visible


    print(f"  Reading geometry from '{excel_file}'...")
    try:
        df_joints = pd.read_excel(excel_file, sheet_name='joint')
        df_members = pd.read_excel(excel_file, sheet_name='member')
    except FileNotFoundError:
        print(f"*** CRITICAL ERROR: The file '{excel_file}' was not found. ***")
        return
    except Exception as e:
        print(f"*** CRITICAL ERROR reading Excel file: {e} ***")
        return
```

```
coords_ft = df_joints[['x coordinate', 'y coordinate']].values
coords = coords_ft * 0.3048  # Convert ft to m

num_joints = len(coords)
num_members = len(df_members)
num_dof = 2 * num_joints

print("\n--- [ System Geometry ] ---")
print(f"  Successfully read {num_joints} joints.")
print(f"  Successfully read {num_members} members.")
print(f"  Total Degrees of Freedom (DOFs): {num_dof}")


# A 'pin' support restrains X and Y motion. A 'roller_y' restrains only Y (vertical) motion.
supports = {
    1: 'pin',
    24: 'roller_y',
    26: 'roller_y',
    49: 'pin'
}

# Internally, the program needs to know which DOFs
# are "free" to move and which are "supported" (restrained).
# Loop through the 'supports' dict and map the 1-based
# joint numbers to 0-based DOF indices (e.g., Joint 1 -> DOFs 0, 1).
all_dofs = list(range(num_dof))
support_dofs = []

print("\n--- [ Support Conditions ] ---")
for joint_num, support_type in supports.items():
    if joint_num > num_joints:
        print(f"*** CRITICAL ERROR: Support joint {joint_num} does not exist. Max joint is {num_joints}. ***")
        return

    joint_idx = joint_num - 1
    dof_x = 2 * joint_idx
    dof_y = 2 * joint_idx + 1
```

```
    if support_type == 'pin':
        support_dofs.extend([dof_x, dof_y])
        print(f"  Joint {joint_num:<3} -> PIN     (Restrains DOFs {dof_x}, {dof_y})")
    elif support_type == 'roller_y':
        support_dofs.append(dof_y)
        print(f"  Joint {joint_num:<3} -> ROLLER  (Restrains DOF {dof_y})")
    elif support_type == 'roller_x':
        support_dofs.append(dof_x)
        print(f"  Joint {joint_num:<3} -> ROLLER  (Restrains DOF {dof_x})")

support_dofs = sorted(list(set(support_dofs)))
free_dofs = [dof for dof in all_dofs if dof not in support_dofs]

if len(free_dofs) == 0:
    print("*** CRITICAL ERROR: No free DOFs. The structure is fully restrained. ***")
    return


# In real life, we define all possible loads
# separately (Dead, Live, Wind, etc.) and then combine them
# using load factors (e.g., 1.2*DL + 1.6*LL).
# We create separate force vectors for each case,
# populate them, and then add them together to get 'F_global'.
```

```
print("\n--- [ Applied Loads ] ---")

# Loads
# --- Dead Load (DL) ---
# The self-weight of the structure.
# Here, we'll apply a simplified 5 kN downward force to ALL 50 joints.
F_dead = np.zeros(num_dof)
dl_force_per_joint = -5000.0 # N (downward)
for i in range(num_joints):
    F_dead[2*i + 1] = dl_force_per_joint
print(f"  Loaded Case 1: Dead Load (DL) -> {dl_force_per_joint} N at ALL {num_joints} joints.")

# --- Live Load (LL) ---
# The "traffic" (trains). This is applied
# to the deck joints (the bottom horizontal chord).
F_live = np.zeros(num_dof)
deck_joints = [1, 4, 8, 13, 17, 21, 26, 30, 34, 38, 42, 46, 50]
ll_force_per_joint = -20000.0 # N (downward)
for joint_num in deck_joints:
    dof_y = 2 * (joint_num - 1) + 1
    F_live[dof_y] = ll_force_per_joint
print(f"  Loaded Case 2: Live Load (LL) -> {ll_force_per_joint} N at {len(deck_joints)} deck joints.")

# --- Wind Load (WL) ---
# A horizontal force on all exposed joints.
# Here, we'll apply a 1.5 kN horizontal force to ALL 50 joints.
F_wind = np.zeros(num_dof)
wl_force_per_joint = 1500.0 # N (in positive X-dir)
for i in range(num_joints):
    F_wind[2*i] = wl_force_per_joint
print(f"  Loaded Case 3: Wind Load (WL) -> {wl_force_per_joint} N at ALL {num_joints} joints.")

# --- Other Loads (User given case) ---
# Engineering: Any other specific point loads.
F_other = np.zeros(num_dof)
F_other[2*(21-1)+1] = -10000.0 # 10 kN down at Joint 21
F_other[2*(30-1)+1] = -10000.0 # 10 kN down at Joint 30
print(f"  Loaded Case 4: Other (User) -> -10kN at J21, -10kN at J30.")
```

```
# --- Load Combination ---
# Create the final load vector, F_global.
# For real design load will be e.g., 1.2*DL + 1.6*LL + 0.5*WL
F_global = 1.2*F_dead + 1.6*F_live + 0.5*F_wind + F_other
print("  ... All load cases combined into F_global.")


# --- ASSEMBLE GLOBAL STIFFNESS MATRIX [K] ---

# We loop through every single member and calculate its individual stiffness (k_e).
# We then "stamp" this 4x4 member matrix into the big 100x100
# global stiffness matrix [K_global].

print("\nAssembling global stiffness matrix [K]...")
K_global = np.zeros((num_dof, num_dof))

for i in range(num_members):
    j1_num = df_members.iloc[i]['joint 1']
    j2_num = df_members.iloc[i]['joint 2']
    j1 = j1_num - 1
    j2 = j2_num - 1

    x1, y1 = coords[j1]
    x2, y2 = coords[j2]

    dx = x2 - x1
    dy = y2 - y1
    L = np.sqrt(dx**2 + dy**2)

    if L == 0:
        print(f"Warning: Member {i+1} ({j1_num}-{j2_num}) has zero length. Skipping.")
        continue

    # c = cos(theta), s = sin(theta)
    c = dx / L
    s = dy / L
```

```
    # Element stiffness matrix (k_e) in global coordinates
    k_e = (EA / L) * np.array([
        [ c*c,  c*s, -c*c, -c*s],
        [ c*s,  s*s, -c*s, -s*s],
        [-c*c, -c*s,  c*c,  c*s],
        [-c*s, -s*s,  c*s,  s*s]
    ])

    # Get the 4 DOFs associated with this member
    dof_indices = [2*j1, 2*j1+1, 2*j2, 2*j2+1]

    # "Stamp" the 4x4 k_e into the 100x100 K_global
    for row_local, row_global in enumerate(dof_indices):
        for col_local, col_global in enumerate(dof_indices):
            K_global[row_global, col_global] += k_e[row_local, col_local]

print("... Assembly complete.")

# We now have the main equation [K]{U} = {F}.

K_ff = K_global[np.ix_(free_dofs, free_dofs)]
F_f = F_global[free_dofs]

print("... Solving for unknown displacements {U_f}...")

try:
    U_f = gauss_jordan(K_ff, F_f)

    # ---Reconstruct Full Displacement Vector ---
    # Create the full displacement vector {U} by
    # "filling in the blanks." We put the calculated U_f values
    # in the 'free' slots and 0.0 in the 'supported' slots.
    U_global = np.zeros(num_dof)
    U_global[free_dofs] = U_f

    print("... Solution found. Starting post-processing.")
```

```
    # --- Nodal Deformations ---
    # Display the displacement (delta X, delta Y)
    # for every joint in the structure.
    # Reshape the 100x1 U_global vector into a 50x2
    # matrix and print it in a formatted loop.
    print("\n--- [ 1. Nodal Deformations (m) ] ---------------------------------")
    Displacements_m = U_global.reshape((num_joints, 2))
    print(" Joint |         dX (m)         |         dY (m)")
    print("-------+------------------------+------------------------")
    for i in range(num_joints):
        dx = Displacements_m[i, 0]
        dy = Displacements_m[i, 1]
        print(f" {i+1:<5} | {dx:22.6e} | {dy:22.6e}")

    # --- Support Reactions ---
    # Calculate the reaction forces at the supports.
    # The formula is {R} = [K_sf]{U_f} - {F_s_applied}
    K_sf = K_global[np.ix_(support_dofs, free_dofs)]
    F_s_from_deformation = K_sf @ U_f
    F_s_applied = F_global[support_dofs]

    R_s = F_s_from_deformation - F_s_applied

    print("\n--- [ 2. Support Reactions (N) ] ----------------------------------")
    Sum_Reactions_Fx = 0.0
    Sum_Reactions_Fy = 0.0

    for i, dof in enumerate(support_dofs):
        joint_num = (dof // 2) + 1
        direction = 'X' if (dof % 2) == 0 else 'Y'
        reaction_force = R_s[i]
        print(f" Joint {joint_num:<3} | {direction}-dir: {reaction_force:20.3f} N")

        if direction == 'X':
            Sum_Reactions_Fx += reaction_force
        else:
            Sum_Reactions_Fy += reaction_force
```

```
        # --- Equilibrium Check ---
        # The sum of all external forces
        # (Applied Loads + Support Reactions) must equal zero.
        Sum_Applied_Fx = np.sum(F_global[0::2])
        Sum_Applied_Fy = np.sum(F_global[1::2])

        print("\n--- [ 3. Equilibrium Check ] ------------------------------------")
        print(f"  Sum Applied Fx:    {Sum_Applied_Fx:18.3f} N")
        print(f"  Sum Reaction Fx:   {Sum_Reactions_Fx:18.3f} N")
        print(f"  ------------------------------------------")
        print(f"  NET FORCE X:       {Sum_Applied_Fx + Sum_Reactions_Fx:18.3f} N   (Should be ~0)")
        print("\n")
        print(f"  Sum Applied Fy:    {Sum_Applied_Fy:18.3f} N")
        print(f"  Sum Reaction Fy:   {Sum_Reactions_Fy:18.3f} N")
        print(f"  ------------------------------------------")
        print(f"  NET FORCE Y:       {Sum_Applied_Fy + Sum_Reactions_Fy:18.3f} N   (Should be ~0)")

        # --- Internal Member Forces ---
        # Loop through all members, get their start/end
        # joint displacements {u_global}, and use the formula
        # Force = (EA/L) * [-c, -s, c, s] * {u_global}
        print("\n--- [ 4. Internal Member Forces (N) ] --------------------------")
        for i in range(num_members):
            j1_num = df_members.iloc[i]['joint 1']
            j2_num = df_members.iloc[i]['joint 2']
            j1 = j1_num - 1
            j2 = j2_num - 1

            x1, y1 = coords[j1]
            x2, y2 = coords[j2]

            dx = x2 - x1
            dy = y2 - y1
            L = np.sqrt(dx**2 + dy**2)
```

```
            if L == 0: continue

            c = dx / L
            s = dy / L
            u_vec_global = U_global[[2*j1, 2*j1+1, 2*j2, 2*j2+1]]
            T = np.array([-c, -s, c, s])
            delta_L = T @ u_vec_global
            force = (EA / L) * delta_L

            status = "Tension" if force > 0 else "Compression"
            print(f" Member {i+1:<4} ({j1_num:<3}-{j2_num:<3}) | {abs(force):18.3f} N  ({status})")

        # --- Plot Deformed Shape ---
        deformed_coords = coords + (Displacements_m * plot_scale_factor)
        plot_truss(coords, deformed_coords, df_members, plot_scale_factor, supports)

    except ValueError as e:
        print(f"\n*** AN ERROR OCCURRED DURING SOLVING: ***")
        print(e)
        print("This often means the structure is unstable (a mechanism).")
        print("Please check your 'supports' dictionary to ensure the")
        print("model is statically determinate (not free to move).")
    except Exception as e:
        print(f"\n*** AN UNEXPECTED ERROR OCCURRED: {e} ***")

# --- Run the main function ---
if __name__ == "__main__":
    solve_truss()
```

OUTPUT

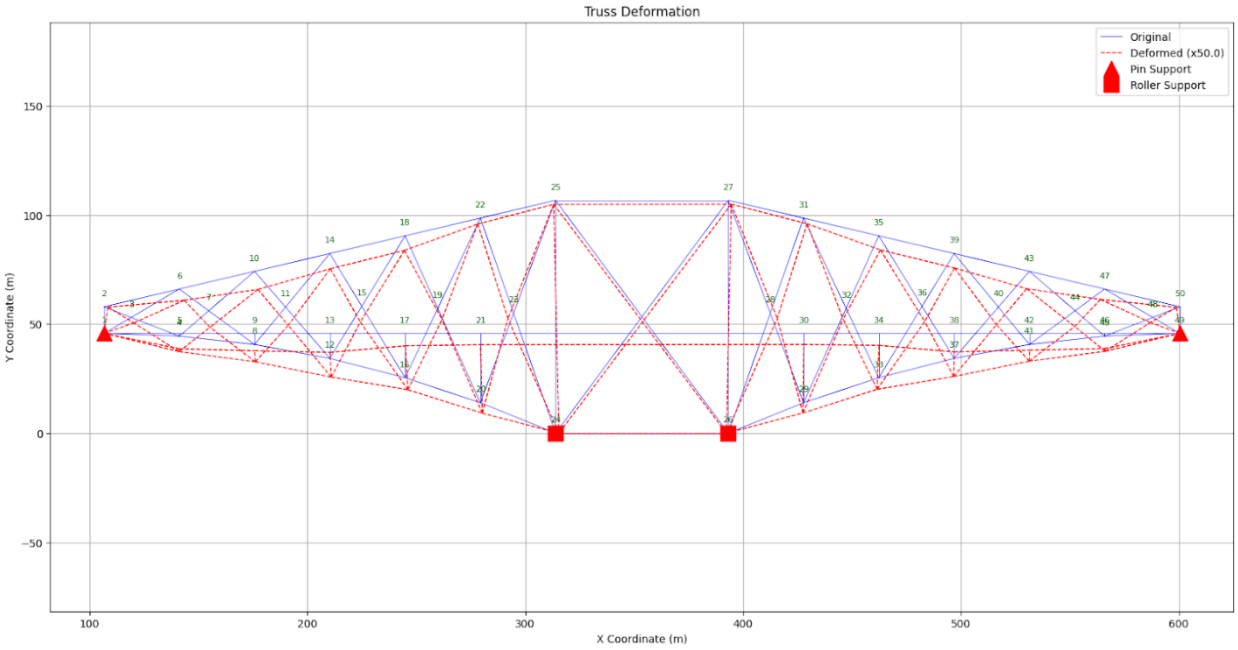
```
Initializing analysis...
  Material: E = 200.0 GPa
  Section:  A = 1000.0 mm^2
  Reading geometry from 'Structure_data_idelized.xlsx'...

--- [ System Geometry ] ---
  Successfully read 50 joints.
  Successfully read 101 members.
  Total Degrees of Freedom (DOFs): 100

--- [ Support Conditions ] ---
  Joint 1   -> PIN     (Restrains DOFs 0, 1)
  Joint 24  -> ROLLER  (Restrains DOF 47)
  Joint 26  -> ROLLER  (Restrains DOF 51)
  Joint 49  -> PIN     (Restrains DOFs 96, 97)

--- [ Applied Loads ] ---
  Loaded Case 1: Dead Load (DL) -> -5000.0 N at ALL 50 joints.
  Loaded Case 2: Live Load (LL) -> -20000.0 N at 13 deck joints.
  Loaded Case 3: Wind Load (WL) -> 1500.0 N at ALL 50 joints.
  Loaded Case 4: Other (User) -> -10kN at J21, -10kN at J30.
  ... All load cases combined into F_global.

Assembling global stiffness matrix [K]...
... Assembly complete.
... Solving for unknown displacements {U_f}...
... Solution found. Starting post-processing.


--- [ 1. Nodal Deformations (m) ] ---------------------------------
 Joint |         dX (m)         |         dY (m)
-------+------------------------+------------------------
 1     |          0.000000e+00 |          0.000000e+00
 2     |          3.592716e-02 |         -3.363461e-03
 3     |          2.357968e-02 |         -5.055009e-02
 4     |         -1.876768e-03 |         -1.410183e-01
 5     |          6.477000e-04 |         -1.410564e-01
 6     |          4.455778e-02 |         -1.030336e-01
 7     |          1.097484e-02 |         -1.512593e-01
 8     |          7.433531e-03 |         -1.589020e-01
 9     |          1.165860e-03 |         -1.590544e-01
 10    |          3.101789e-02 |         -1.663048e-01
 11    |          2.568319e-02 |         -1.670498e-01
 12    |          1.287181e-02 |         -1.716795e-01
 13    |          1.554480e-03 |         -1.738512e-01
 14    |          4.868530e-03 |         -1.390808e-01
 15    |         -1.414299e-02 |         -1.441519e-01
 16    |          2.625187e-02 |         -1.099085e-01
 17    |          1.813560e-03 |         -1.137693e-01
 18    |         -6.322473e-03 |         -1.335777e-01
 19    |          3.837366e-02 |         -1.021669e-01
 20    |          1.755767e-02 |         -9.587524e-02
 21    |          1.943100e-03 |         -1.034952e-01
 22    |         -2.381462e-02 |         -5.469057e-02
 23    |         -5.248898e-02 |         -4.947554e-02
 24    |          2.987859e-02 |          0.000000e+00
 25    |         -1.762872e-02 |         -3.627282e-02
 26    |         -1.678362e-02 |          0.000000e+00
 27    |          2.681463e-02 |         -3.533417e-02
 28    |          6.392242e-02 |         -4.798607e-02
 29    |         -5.504554e-03 |         -9.386533e-02
 30    |          1.943100e-03 |         -1.014853e-01
 31    |          3.276771e-02 |         -5.215036e-02
 32    |         -2.798976e-02 |         -9.882384e-02
 33    |         -1.561545e-02 |         -1.059994e-01
 34    |          1.813560e-03 |         -1.098602e-01
 35    |          1.487554e-02 |         -1.293188e-01
 36    |          2.403816e-02 |         -1.390383e-01
 37    |         -3.997067e-03 |         -1.669360e-01
 38    |          1.554480e-03 |         -1.691077e-01
 39    |          3.185375e-03 |         -1.325297e-01
 40    |         -1.835619e-02 |         -1.605886e-01
 41    |         -9.577071e-04 |         -1.526224e-01
 42    |          1.165860e-03 |         -1.535876e-01
 43    |         -2.377990e-02 |         -1.596713e-01
 44    |         -3.924398e-03 |         -1.447312e-01
 45    |          5.374409e-03 |         -1.384227e-01
 46    |          6.477000e-04 |         -1.386640e-01
 47    |         -3.801819e-02 |         -9.528984e-02
 48    |         -2.065544e-02 |         -4.685961e-02
 49    |          0.000000e+00 |          0.000000e+00
 50    |         -3.099464e-02 |         -5.217324e-03
```

```
--- [ 2. Support Reactions (N) ] ---------------------------------
 Joint 1   | X-dir:            39624.051 N
 Joint 1   | Y-dir:           131075.244 N
 Joint 24  | Y-dir:           222830.608 N
 Joint 26  | Y-dir:           248818.518 N
 Joint 49  | X-dir:           -77124.051 N
 Joint 49  | Y-dir:           133275.631 N

--- [ 3. Equilibrium Check ] -------------------------------------
  Sum Applied Fx:             37500.000 N
  Sum Reaction Fx:           -37500.000 N
  -------------------------------------------
  NET FORCE X:                    0.000 N   (Should be ~0)


  Sum Applied Fy:           -736000.000 N
  Sum Reaction Fy:           736000.000 N
  -------------------------------------------
  NET FORCE Y:                    0.000 N   (Should be ~0)
```

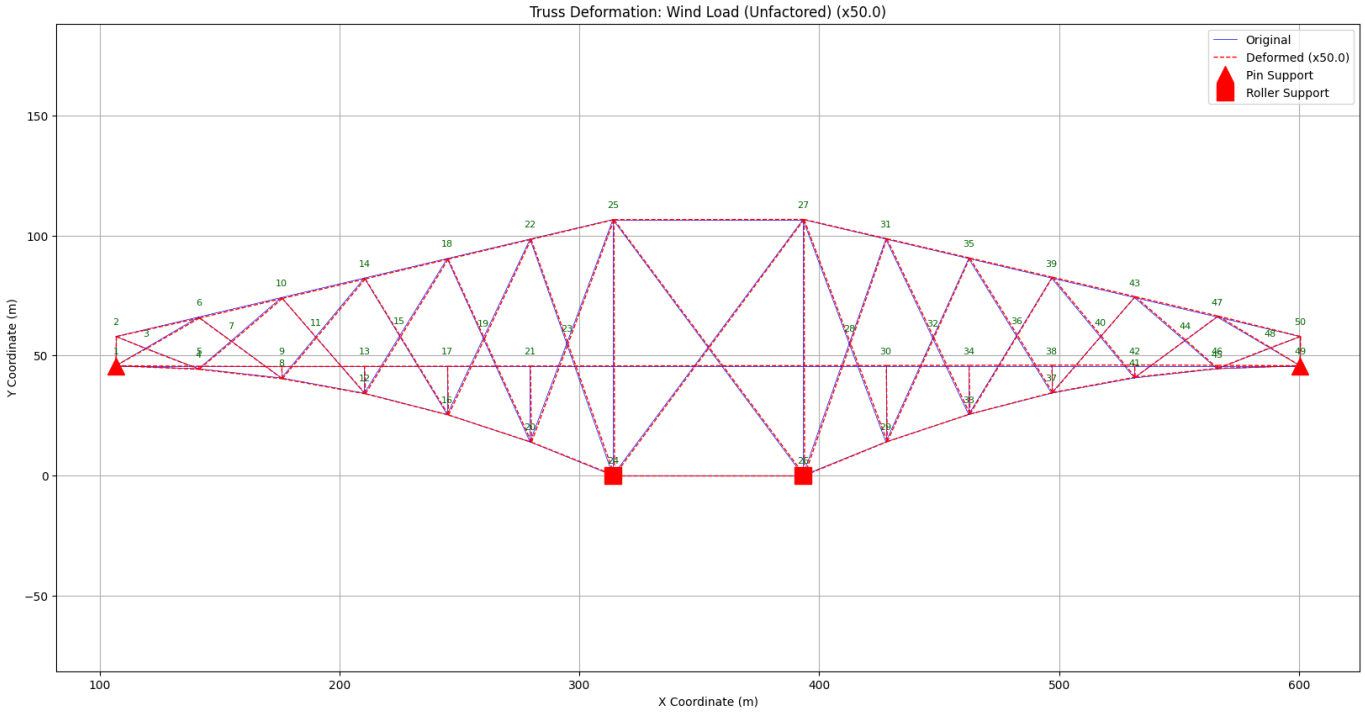
```
--- [ 4. Internal Member Forces (N) ] --------------------------
 Member 1    (2  -6  ) |          81308.885 N  (Compression)
 Member 2    (6  -10 ) |         155951.701 N  (Compression)
 Member 3    (10 -14 ) |         108313.779 N  (Compression)
 Member 4    (14 -18 ) |          54290.290 N  (Compression)
 Member 5    (18 -22 ) |           5867.438 N  (Tension)
 Member 6    (22 -25 ) |          57709.835 N  (Tension)
 Member 7    (25 -27 ) |         112162.711 N  (Tension)
 Member 8    (27 -31 ) |          54365.285 N  (Tension)
 Member 9    (31 -35 ) |           1454.325 N  (Tension)
 Member 10   (35 -39 ) |          59987.454 N  (Compression)
 Member 11   (39 -43 ) |         112896.313 N  (Compression)
 Member 12   (43 -47 ) |         161216.249 N  (Compression)
 Member 13   (47 -50 ) |          77736.456 N  (Compression)
 Member 14   (1  -4  ) |          19124.966 N  (Tension)
 Member 15   (4  -8  ) |          64538.933 N  (Tension)
 Member 16   (8  -12 ) |          43611.315 N  (Tension)
 Member 17   (12 -16 ) |          13666.968 N  (Compression)
 Member 18   (16 -20 ) |          69600.572 N  (Compression)
 Member 19   (20 -24 ) |         131623.613 N  (Compression)
 Member 20   (24 -26 ) |         117762.490 N  (Compression)
 Member 21   (26 -29 ) |         132763.198 N  (Compression)
 Member 22   (29 -33 ) |          73714.264 N  (Compression)
 Member 23   (33 -37 ) |          22066.824 N  (Compression)
 Member 24   (37 -41 ) |          31757.634 N  (Tension)
 Member 25   (41 -45 ) |          45179.092 N  (Tension)
 Member 26   (45 -49 ) |           1649.801 N  (Compression)
 Member 27   (25 -26 ) |          43062.576 N  (Compression)
 Member 28   (27 -24 ) |          45436.748 N  (Compression)
 Member 29   (2  -1  ) |          55174.886 N  (Compression)
 Member 30   (25 -24 ) |          68003.042 N  (Compression)
 Member 31   (27 -26 ) |          66243.297 N  (Compression)
 Member 32   (50 -49 ) |          85586.030 N  (Compression)
 Member 33   (2  -3  ) |          84140.266 N  (Tension)
 Member 34   (6  -3  ) |          66593.990 N  (Compression)
 Member 35   (1  -3  ) |          73365.350 N  (Compression)
 Member 36   (4  -3  ) |          77071.325 N  (Tension)
 Member 37   (6  -7  ) |          18008.988 N  (Tension)
 Member 38   (10 -7  ) |          38832.403 N  (Tension)
 Member 39   (4  -7  ) |          34327.906 N  (Tension)
 Member 40   (8  -7  ) |          12839.571 N  (Tension)
 Member 41   (10 -11 ) |          27008.437 N  (Compression)
 Member 42   (14 -11 ) |          51738.801 N  (Tension)
 Member 43   (8  -11 ) |          48331.138 N  (Tension)
 Member 44   (12 -11 ) |          31476.166 N  (Compression)
 Member 45   (14 -15 ) |          39135.866 N  (Compression)
 Member 46   (18 -15 ) |          69119.933 N  (Tension)
 Member 47   (12 -15 ) |          66341.481 N  (Tension)
 Member 48   (16 -15 ) |          43386.874 N  (Compression)
 Member 49   (18 -19 ) |          56060.826 N  (Compression)
 Member 50   (22 -19 ) |          71607.652 N  (Tension)
 Member 51   (16 -19 ) |          69260.796 N  (Tension)
 Member 52   (20 -19 ) |          60316.142 N  (Compression)
 Member 53   (22 -23 ) |          62389.280 N  (Compression)
 Member 54   (25 -23 ) |          88951.546 N  (Tension)
 Member 55   (20 -23 ) |          86954.683 N  (Tension)
 Member 56   (24 -23 ) |          66764.377 N  (Compression)
 Member 57   (27 -28 ) |          89924.947 N  (Tension)
 Member 58   (31 -28 ) |          61648.170 N  (Compression)
```

```
Member 59   (26 -28 ) |         63825.180 N  (Compression)
Member 60   (29 -28 ) |         85714.419 N  (Tension)
Member 61   (31 -32 ) |         71104.861 N  (Tension)
Member 62   (35 -32 ) |         55617.591 N  (Compression)
Member 63   (29 -32 ) |         58091.613 N  (Compression)
Member 64   (33 -32 ) |         66962.825 N  (Tension)
Member 65   (35 -36 ) |         68990.984 N  (Tension)
Member 66   (39 -36 ) |         37202.945 N  (Compression)
Member 67   (33 -36 ) |         40018.547 N  (Compression)
Member 68   (37 -36 ) |         64771.853 N  (Tension)
Member 69   (39 -40 ) |         49260.709 N  (Tension)
Member 70   (43 -40 ) |         26344.727 N  (Compression)
Member 71   (37 -40 ) |         29641.998 N  (Compression)
Member 72   (41 -40 ) |         44703.737 N  (Tension)
Member 73   (43 -44 ) |         38302.699 N  (Tension)
Member 74   (47 -44 ) |         21683.719 N  (Tension)
Member 75   (41 -44 ) |         17518.155 N  (Tension)
Member 76   (45 -44 ) |         32886.361 N  (Tension)
Member 77   (47 -48 ) |         74879.442 N  (Compression)
Member 78   (50 -48 ) |         82017.948 N  (Tension)
Member 79   (45 -48 ) |         75917.355 N  (Tension)
Member 80   (49 -48 ) |         82344.293 N  (Compression)
Member 81   (4  -5  ) |          6000.000 N  (Compression)
Member 82   (8  -9  ) |          6000.000 N  (Compression)
Member 83   (12 -13 ) |         38000.000 N  (Compression)
Member 84   (16 -17 ) |         38000.000 N  (Compression)
Member 85   (20 -21 ) |         48000.000 N  (Compression)
Member 86   (29 -30 ) |         48000.000 N  (Compression)
Member 87   (33 -34 ) |         38000.000 N  (Compression)
Member 88   (37 -38 ) |         38000.000 N  (Compression)
Member 89   (41 -42 ) |         38000.000 N  (Compression)
Member 90   (45 -46 ) |         38000.000 N  (Compression)
Member 91   (1  -5  ) |          3750.000 N  (Tension)
Member 92   (5  -9  ) |          3000.000 N  (Tension)
Member 93   (9  -13 ) |          2250.000 N  (Tension)
Member 94   (13 -17 ) |          1500.000 N  (Tension)
Member 95   (17 -21 ) |           750.000 N  (Tension)
Member 96   (21 -30 ) |             0.000 N  (Compression)
Member 97   (30 -34 ) |           750.000 N  (Compression)
Member 98   (34 -38 ) |          1500.000 N  (Compression)
Member 99   (38 -42 ) |          2250.000 N  (Compression)
Member 100  (42 -46 ) |          3000.000 N  (Compression)
Member 101  (46 -49 ) |          3750.000 N  (Compression)

... Plotting deformed shape with scale factor: 50.0 ...
```

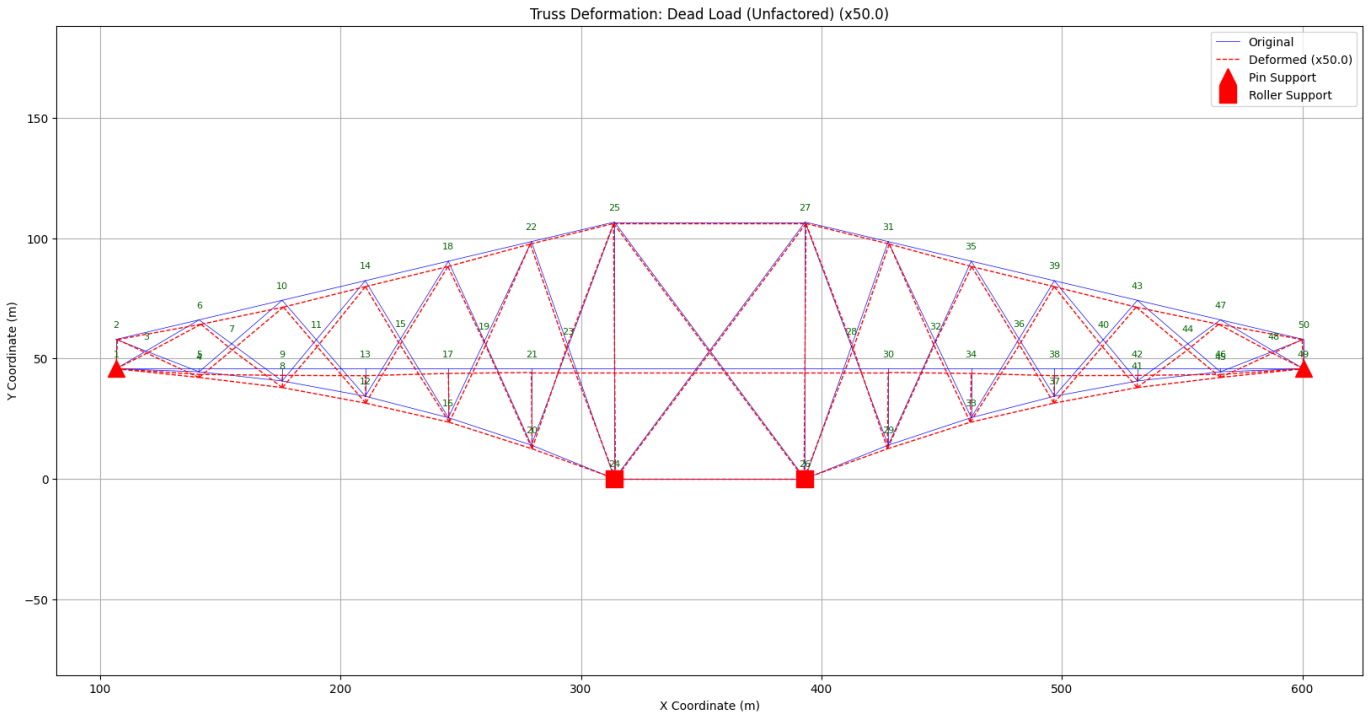
Deformed Shape Due to All the Loads Combined



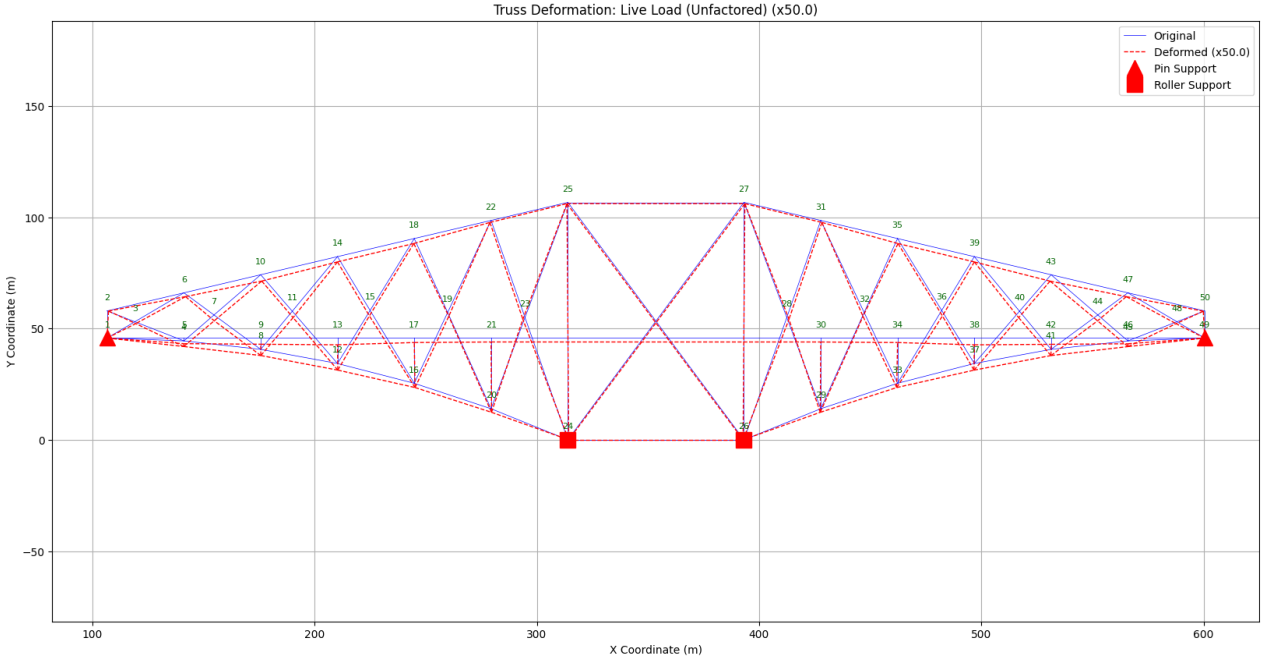
Deformed Shape Due to Wind Loads



Deformed Shape Due to Dead Load



Deformed Shape Due to Live Load



Conclusion

The **Forth Bridge** stands as a timeless symbol of engineering excellence, representing a perfect balance of strength, durability, and innovation. Designed in the late 19th century and still serving as a critical railway link today, the bridge exemplifies how structural ingenuity can meet functional demands while withstanding environmental and operational stresses. Its cantilever design, pioneering use of steel, and carefully calculated load distribution continue to inspire modern bridge engineering practices across the world.

The analysis of the Forth Bridge highlights the effectiveness of its design in managing various load conditions dead loads, live train loads, wind forces, and temperature variations while maintaining remarkable stability and minimal deflection. The structure's resilience over more than a century of service is a testament to the foresight of its designers and the precision of its construction techniques.

Beyond its structural significance, the bridge holds immense cultural and historical value as a **UNESCO World Heritage Site**, symbolizing Scotland's industrial heritage and its contribution to global civil engineering. The continued preservation and monitoring of the bridge underscore the importance of combining traditional engineering wisdom with modern analytical tools to ensure safety, functionality, and longevity.

In essence, the Forth Bridge is not just a functional transport link but a living monument to human ingenuity demonstrating that with sound design, robust materials, and visionary engineering, structures can endure the test of time and remain both safe and iconic for generations to come.

AI tools were not used to generate the report.

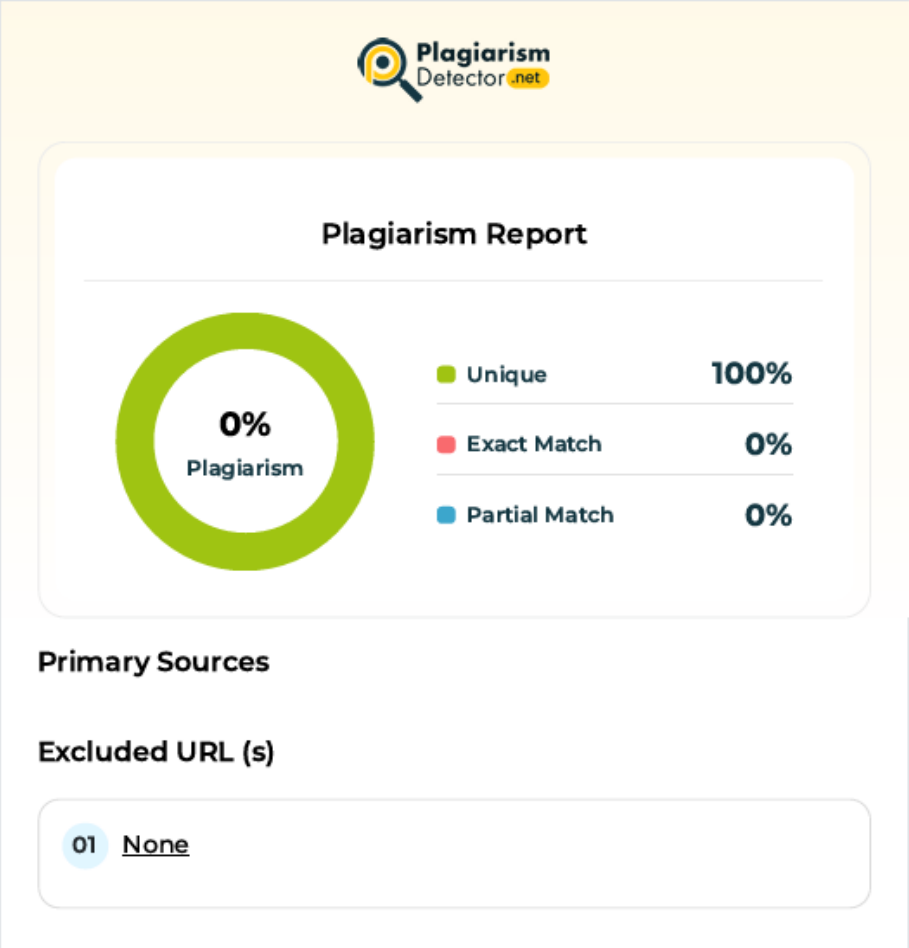
References

https://www.historicenvironment.scot/advice-and-support/listing-scheduling-and-designations/world-heritage-sites/forth-bridge/

https://www.theforthbridges.org/about-the-forth-bridges/forth-bridge/forth-bridge-facts-figures/

https://en.wikisource.org/wiki/The_Forth_Bridge/General_Description_of_the_Structure

Plagiarism Check



Thank You