Volume-09 Issue 03, March-2025 ISSN: 2456-9348
Impact Factor: 8.232

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

EUROCODE-BASED ANALYSIS AND DESIGN OF BRIDGE SUBSTRUCTURES USING PYTHON PROGRAMMING

E.C. Nwadike,
I.C. Nwosu,
N.P. Ogbonna,
Lecturer, Civil Engineering Department, Federal Polytechnic Nekede, Owerri, Nigeria
nopsoftinc@yahoo.com

S.A. Nwose,
Civil Engineering Department, Delta State Polytechnic, Ogwashi Uku, Delta State, Nigeria

ABSTRACT
This research designed, analyzed, and detailed concrete bridge substructures in accordance with Eurocodes using both manual and computer methods. The design focused exclusively on the substructure—the lower part of a bridge—which supports the horizontal spans by carrying the weights of the superstructural members. In this work, a Python-based computer program was developed to design, analyze, and detail the bridge substructures. The Python program specifically addressed the design and analysis of abutments and pier caps using Courbon's method, incorporating advanced algorithms and visualization techniques.
The bridge spans considered ranged from 23.2 m to 40 m. The maximum design bending moments, maximum shear forces, and axial forces were 935.90 kNm, 366.69 kN, and 682.48 kN respectively. Manual analyses and designs were performed for bridge spans of 23.2 m, 25 m, 30 m, 35 m, and 40 m, and these results were then compared with those obtained from the Python program. The comparison showed that the outcomes from both the manual and Python-based analyses were essentially the same, with percentage differences of less than 1%.
This validation confirms that the developed Python program complies with Eurocodes BS EN 1992-1-1, BS EN 1991-1-4, and BS EN 1991-2, and can serve as a reliable and efficient tool for the analysis and design of bridge substructures. Consequently, these Eurocodes can be easily applied using this Python program to design bridge substructures with considerations for safety, serviceability, and economy. The Python-based tool enhances the performance of bridge structures by offering quick, accurate, and reliable analyses and designs.

Keywords:
Programming, codes, python, bridge, substructure, Eurocode, design

INTRODUCTION

According to [1], a bridge is a structure that enables passage over an obstacle without interfering with what lies below. Such obstacles may include rivers, roads, railways, canals, pipelines, or pedestrian paths, while the passage itself might accommodate roadways, rail lines, or pipelines. Bridges, which can span from only a few meters to several kilometers, rank among the largest constructions built by humans. Their design and material standards are exceptionally high because a bridge must support both its own weight and the dynamic loads of vehicles and pedestrians. Furthermore, these structures must endure a range of natural phenomena, such as temperature fluctuations, high winds, and seismic activity. Typically, bridges are built using frameworks of concrete, steel, or wood, with the roadway surface composed of asphalt or concrete. A conventional bridge is composed of two main parts: the superstructure and the substructure. The superstructure includes elements such as the deck, floor system, and primary trusses or girders that span horizontally and carry traffic loads. In contrast, the substructure—comprising towers, footings, abutments, columns, piers, and piles—supports these spans by lifting them over the ground surface [2]. The foundation type for a bridge is determined by several factors, including the intended bridge design, the depth of scour, the nature of the obstacle being crossed, and the characteristics of the underlying soil. Both well-type and pile-type foundations are viable options for bridge construction. In further related studies, [3] analysed composite bridge superstructures using a Modified Grillage

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

Method and found it to be a simpler approach that yielded results comparable to those obtained through the Finite Element Method or Orthotropic Plate Theory via the Finite Difference Method. [4] developed a PC program for designing slant bridges by enhancing the Semi Continuum Method to determine moments and shear forces in right bridges, which served as a basis for his software. [5] concentrated on load distribution along supports by employing the Courbon Method among other approaches, assessing both how live loads are shared among longitudinal supports and how bending moments are distributed in girder and slab bridges. Additionally, [6] used SAP2000 software to examine a single-span, two-path T-beam bridge, varying bridge lengths and deck slab depths to study their effects. [7] also contributed to the field by developing a VB.Net-based software package aimed at the analysis and design of concrete bridge substructures with simply supported spans, basing his calculations on the Class AA and Class A loading categories as per IRC standards. Under this research, computer software package for the analysis and design of concrete bridge substructures based on Euro codes will be developed using the Python programming language

CONCRETE BRIDGES AND THEIR STRUCTURAL IMPORTANCE

Concrete bridges have long been recognized for their durability, strength, and versatility. Their design and construction are governed by rigorous standards to ensure safe load transfer and longevity. As noted by [8] and [9], concrete remains one of the most widely used construction materials, particularly for bridge structures, due to its high compressive strength and favorable cost-to-performance ratio.

EUROCODE PROVISIONS FOR BRIDGE FOUNDATIONS

Eurocodes play a pivotal role in defining the design criteria for bridge foundations in Europe. Specifically, Eurocode 7 (EN 1997) outlines the geotechnical design requirements, while Eurocode 2 (EN 1992-1-1) addresses the design of concrete structures. In the context of bridge foundations, these codes provide detailed guidelines on load combinations, durability requirements, and safety factors. The provisions ensure that the foundation not only supports the bridge's substructure but also accommodates environmental loads such as temperature variations, wind, and seismic forces [15];[16].

BRIDGE SUBSTRUCTURE TYPES

Bridge substructures, comprising the lower part of the bridge that transfers loads from the superstructure to the foundation, are critical for overall stability. According to [2] the substructure includes elements such as abutments, pier caps, piers, columns, and piles. These components are designed to handle the vertical and horizontal forces imposed by the traffic loads and environmental effects. Research by [3] and subsequent studies have emphasized that an effective substructure design is essential to mitigate issues such as differential settlement and thermal stresses.

FOUNDATION TYPES FOR BRIDGE STRUCTURES

The choice of foundation type for concrete bridges is influenced by several factors, including soil conditions, load demands, and environmental exposures. Common foundation systems include shallow foundations, such as spread footings and mat foundations, as well as deep foundations like piles and caissons. The Eurocode provisions guide engineers in selecting the appropriate foundation type based on geotechnical investigations and structural analysis [16];[17] For instance, in areas with poor soil conditions or high load requirements, pile foundations are often recommended to ensure adequate load transfer and long-term stability.

INTEGRATION OF COMPUTATIONAL METHODS AND EUROCODE STANDARDS

Recent studies have highlighted the integration of computational tools with Eurocode standards to enhance the design and analysis of bridge substructures and foundations. Advanced software, often based on finite element analysis and other numerical methods, facilitates a more detailed assessment of load distributions, stress concentrations, and potential failure modes. This approach not only improves design accuracy but also expedites the optimization process for both substructure and foundation systems [5];[7]. By validating computational models against the prescribed Eurocode requirements, researchers and practitioners can ensure that the bridge designs meet both performance and safety criteria.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

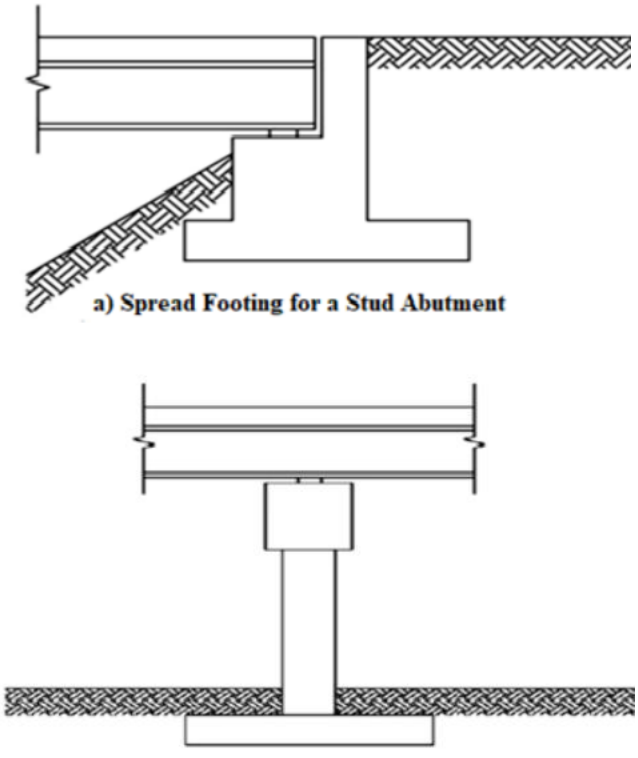

Figure 1: Spread Footing Applications [10]

PYTHON PROGRAMMING LANGUAGE

Python is a high-level, interpreted programming language designed for general-purpose use. Developed by Guido van Rossum and first released in 1991, Python is known for its emphasis on code readability and a syntax that enables developers to express ideas with fewer lines of code, utilizing significant whitespace for clarity. It facilitates both small-scale and large-scale programming with well-structured constructs.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional, and procedural programming. Additionally, it boasts a vast and comprehensive standard library that simplifies development.

Python interpreters are available across various operating systems. The reference implementation, Python, is open-source and follows a community-driven development model. Like many of its alternative implementations, CPython is managed by the non-profit Python Software Foundation.

Advantages of Python in Structural Engineering:

1. **Open Source and Community Support** - Python is freely available, and its active community continuously contributes to its development.
2. **Ease of Learning** - The language's simple syntax and readability make it accessible to engineers with limited programming experience.
3. **User-Friendly Data Structures** - Built-in data structures, such as lists and dictionaries, enable efficient handling of engineering data.
4. **Productivity and Speed** - Python facilitates rapid prototyping and execution, streamlining computational analyses in structural engineering.

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

METHODOLOGY

Example 1: Design of an Abutment
Under this section, we will design an abutment wall to resist lateral earth pressure from a backfill. *(Based on Eurocode 7 for geotechnical design and Eurocode 2 for concrete design.)*
Given/Assumed Data:

- Height of abutment wall (H): 6 m
- Backfill unit weight (γ\gammaγ): 18 kN/m³
- Internal friction angle (ϕ\phiϕ): 30°
- Abutment wall thickness: assumed 0.5 m
- Concrete strength fckf_{ck}fck: 30 MPa
- Partial factor for loads: 1.35 (as applicable)

Step 1: Lateral Earth Pressure Calculation
Using Rankine's theory, the lateral pressure at depth z is:

$$\sigma_h = \gamma z K_a \quad \text{where} \quad K_a = \tan^2\left(45^\circ - \frac{\phi}{2}\right)$$

For $\phi = 30^\circ$:

$$K_a = \tan^2(30^\circ) \approx 0.33$$

At mid-height ($z = 3\,\text{m}$):

$$\sigma_h = 18 \times 3 \times 0.33 \approx 18\,\text{kN/m}^2$$

Step 2: Design Check for Abutment

- The resultant lateral force is assumed to act at 1/3 of the wall height.
- Design moment and shear are calculated using statics.
- The wall is then checked for bending using Eurocode 2 provisions:

M_{Ed}=Resultant force×lever arm
$P = \frac{1}{2}\gamma H^2 K_a \approx \frac{1}{2} \times 18 \times 36 \times 0.33 \approx$

Suppose the resultant force per unit length is

$107\,\text{kN/m}$.
With lever arm $z = \frac{H}{3} \approx 2\,\text{m}$:

The abutment is designed for bending moment resistance using reinforcement provided as per Eurocode 2, ensuring the provided moment capacity M_{Rd} is at least M_{Ed}

Example 2: Design of a Pier Cap
Objective:
Design a pier cap to transfer loads from a pier to a bridge deck. *(Eurocode 2 is used for structural design.)*
Given/Assumed Data:

- Pier cap dimensions: 2.0 m (length) × 1.5 m (width)

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

- Total design load from the pier: 2000 kN
- Eccentricity assumed negligible for simplicity
- Concrete strength f_{ck}: 30 MPa, with reinforcement yield strength f_{yk} = 500 MPa

Step 1: Stress Distribution
Assuming uniform load:

$$\sigma = \frac{2000\,\mathrm{kN}}{2.0 \times 1.5} \approx 667\,\mathrm{kN/m^2}$$

This is within the allowable stress limits for 30 MPa concrete when including safety factors.
Step 2: Reinforcement Design
For a one-way slab action (assuming load transfer in one direction), calculate the required reinforcement ratio ρ from:

$$M_{Ed} = \alpha_1 f_{cd} bx(d - \frac{x}{2})$$

Assume effective depth d of 0.15 m and preliminary lever arm estimates.
Through iterative calculation (or using design charts per Eurocode 2), determine the area of reinforcement A_s required such that the moment capacity M_{Rd} meets or exceeds the applied moment.

Example 3: Design of a Pier (Column) in Compression
Objective:
Design a reinforced concrete pier to carry axial loads and bending moments. *(Eurocode 2 provisions apply.)*
Given/Assumed Data:

- Pier cross-sectional dimensions: 0.8 m × 0.8 m
- Design axial load N_{Ed}: 2000 kN
- Concrete strength f_{ck}: 40 MPa
- Reinforcement ratio initially assumed: 2%
- Effective height and slenderness effects are considered negligible for this example.

Step 1: Check Axial Capacity
The design axial capacity is:

$$N_{Rd} = \alpha_c f_{cd} A_c + A_s f_{yd}$$

Where:

$A_c = 0.64\,\mathrm{m}^2$

$f_{cd} = f_{ck}/\gamma_c$ (assuming $\gamma_c = 1.5$)

$A_s = 0.02 \times 0.64 \approx 0.0128\,\mathrm{m}^2$

$f_{yd} = f_{yk}/\gamma_s$ (assuming $\gamma_s = 1.15$ and $f_{yk} = 500\,\mathrm{MPa}$)

Calculate:

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

$$f_{cd} = \frac{40}{1.5} \approx 26.67\,\text{MPa}$$

$$f_{yd} = \frac{500}{1.15} \approx 435\,\text{MPa}$$

Then:

$$N_{Rd} = 26.67 \times 0.64 + 435 \times 0.0128 \quad \text{(in MN, converting units)}$$

$$N_{Rd} \approx 17.07 + 5.57 \approx 22.64\,\text{MN} \quad \text{(or 22640 kN)}$$

Since the applied load is 2000 kN, the pier has sufficient capacity with ample reserve. Detailed checks for slenderness and moment effects would follow similarly.

Example 4: Design of a Pile Foundation
Objective:
Determine the ultimate load capacity of a concrete pile based on Eurocode 7 provisions.
Given/Assumed Data:

- Pile diameter: 0.5 m
- Embedded length in soil: 10 m
- Soil resistance (frictional) q_s: 150 kN/m²
- End bearing capacity q_b: 1000 kN/m²
- Cross-sectional area of the pile A: $\pi \times (0.25)^2 \approx 0.196\ m^2$

Step 1: Calculate Side Friction Capacity

$$N_s = \pi d L q_s = \pi \times 0.5 \times 10 \times 150 \approx 2356\,\text{kN}$$

Step 2: Calculate End Bearing Capacity

$$N_b = A q_b = 0.196 \times 1000 \approx 196\,\text{kN}$$

Step 3: Total Ultimate Load Capacity

$$N_{ult} = N_s + N_b \approx 2356 + 196 \approx 2552\,\text{kN}$$

Design checks would incorporate partial safety factors as per Eurocode 7.

Python Program for the design of Retaining Walls

```
import tkinter as tk
from tkinter import ttk, messagebox
import math

# Function for Abutment Design Calculations
def calculate_abutment():
try:
H = float(entry_H.get())            # Height of abutment wall (m)
gamma = float(entry_gamma.get())      # Backfill unit weight (kN/m³)
phi_deg = float(entry_phi.get())      # Internal friction angle in degrees
except ValueError:
messagebox.showerror("Input Error", "Please enter valid numbers for Abutment parameters.")
return
```

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

```
# Calculate active earth pressure coefficient using Rankine's formula:
phi_rad = math.radians(phi_deg)
K_a = math.tan(math.radians(45) - phi_rad/2) ** 2

# Lateral earth pressure at mid-height (z = H/2)
z = H / 2
sigma_h = gamma * z * K_a  # in kN/m²

# Calculate resultant lateral force per unit length:
# For a triangular distribution, P = 0.5 * gamma * H² * K_a
P = 0.5 * gamma * H * H * K_a  # kN/m

# Assume lever arm is H/3 for moment calculation
lever_arm = H / 3
M_Ed = P * lever_arm  # kNm/m

result_abutment.set(f"Active Earth Pressure Coefficient, K_a = {K_a:.3f}\n"
f"Lateral Pressure at z=H/2: {sigma_h:.2f} kN/m²\n"
f"Resultant Lateral Force: {P:.2f} kN/m\n"
f"Design Moment (per unit length): {M_Ed:.2f} kNm/m")

# Function for Pier Cap Design Calculations
def calculate_pier_cap():
try:
L = float(entry_cap_length.get())     # Length of pier cap (m)
B = float(entry_cap_width.get())      # Width of pier cap (m)
load = float(entry_cap_load.get())    # Total design load from pier (kN)
except ValueError:
messagebox.showerror("Input Error", "Please enter valid numbers for Pier Cap parameters.")
return

# Calculate average stress in the cap:
area = L * B  # m²
stress = load / area  # kN/m²

result_pier_cap.set(f"Pier Cap Area: {area:.2f} m²\n"
f"Uniform Stress: {stress:.2f} kN/m²\n"
"Note: Further moment and reinforcement checks required per Eurocode 2.")

# Function for Pier (Column) Design Calculations (Axial Capacity Check)
def calculate_pier():
try:
width = float(entry_pier_width.get())   # Pier width (m)
depth = float(entry_pier_depth.get())    # Pier depth (m)
N_Ed = float(entry_pier_load.get())      # Design axial load (kN)
```

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

```
except ValueError:
messagebox.showerror("Input Error", "Please enter valid numbers for Pier parameters.")
return

A_c = width * depth  # Cross-sectional area in m²
f_ck = 40  # Concrete characteristic strength in MPa (assumed)
gamma_c = 1.5  # Partial safety factor for concrete
f_cd = f_ck / gamma_c  # Design compressive strength in MPa (N/mm²)
# Convert area from m² to mm²: 1 m² = 1e6 mm²
A_c_mm2 = A_c * 1e6

# Calculate concrete capacity (ignoring reinforcement contribution for simplicity)
N_Rd_conc = f_cd * A_c_mm2 / 1e3  # in kN, since f_cd in N/mm² and area in mm²

result_pier.set(f"Pier Cross-sectional Area: {A_c:.3f} m²\n"
f"Design Concrete Strength, f_cd: {f_cd:.2f} MPa\n"
f"Calculated Axial Capacity: {N_Rd_conc:.2f} kN\n"
f"Applied Load: {N_Ed:.2f} kN\n"
f"{'Adequate' if N_Rd_conc >= N_Ed else 'Inadequate'}")

# Function for Pile Foundation Design Calculations
def calculate_pile():
try:
d = float(entry_pile_diameter.get())  # Pile diameter (m)
L_pile = float(entry_pile_length.get())  # Embedded length (m)
q_s = float(entry_qs.get())          # Soil resistance (kN/m²)
q_b = float(entry_qb.get())          # End bearing capacity (kN/m²)
except ValueError:
messagebox.showerror("Input Error", "Please enter valid numbers for Pile parameters.")
return

# Pile cross-sectional area:
A_pile = math.pi * (d/2) ** 2  # m²

# Side friction capacity:
N_s = math.pi * d * L_pile * q_s  # kN

# End bearing capacity:
N_b = A_pile * q_b  # kN

N_ult = N_s + N_b  # Ultimate load capacity (kN)

result_pile.set(f"Pile Cross-sectional Area: {A_pile:.3f} m²\n"
f"Side Friction Capacity: {N_s:.2f} kN\n"
f"End Bearing Capacity: {N_b:.2f} kN\n"
f"Ultimate Load Capacity: {N_ult:.2f} kN")
```

IJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

```
# Create main window
root = tk.Tk()
root.title("Bridge Substructure Design Calculator")
root.geometry("700x600")

notebook = ttk.Notebook(root)
notebook.pack(expand=True, fill='both')

# ----------------- Tab 1: Abutment Design -----------------
tab1 = ttk.Frame(notebook)
notebook.add(tab1, text="Abutment Design")

frame_abutment = ttk.Frame(tab1, padding="10")
frame_abutment.pack(fill='both', expand=True)

ttk.Label(frame_abutment, text="Abutment Design Calculations", font=("Arial", 14, "bold")).grid(row=0,
column=0, columnspan=2, pady=5)

ttk.Label(frame_abutment, text="Height of Abutment, H (m):").grid(row=1, column=0, sticky='e')
entry_H = ttk.Entry(frame_abutment)
entry_H.grid(row=1, column=1)

ttk.Label(frame_abutment, text="Backfill Unit Weight, γ (kN/m³):").grid(row=2, column=0, sticky='e')
entry_gamma = ttk.Entry(frame_abutment)
entry_gamma.grid(row=2, column=1)

ttk.Label(frame_abutment, text="Internal Friction Angle, φ (°):").grid(row=3, column=0, sticky='e')
entry_phi = ttk.Entry(frame_abutment)
entry_phi.grid(row=3, column=1)

btn_abutment = ttk.Button(frame_abutment, text="Calculate Abutment", command=calculate_abutment)
btn_abutment.grid(row=4, column=0, columnspan=2, pady=10)

result_abutment = tk.StringVar()
ttk.Label(frame_abutment, textvariable=result_abutment, relief="sunken", padding=10).grid(row=5, column=0,
columnspan=2, sticky='nsew')

# ----------------- Tab 2: Pier Cap Design -----------------
tab2 = ttk.Frame(notebook)
notebook.add(tab2, text="Pier Cap Design")

frame_pier_cap = ttk.Frame(tab2, padding="10")
frame_pier_cap.pack(fill='both', expand=True)
```

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

```
ttk.Label(frame_pier_cap, text="Pier Cap Design Calculations", font=("Arial", 14, "bold")).grid(row=0, column=0, columnspan=2, pady=5)

ttk.Label(frame_pier_cap, text="Length of Pier Cap, L (m):").grid(row=1, column=0, sticky='e')
entry_cap_length = ttk.Entry(frame_pier_cap)
entry_cap_length.grid(row=1, column=1)

ttk.Label(frame_pier_cap, text="Width of Pier Cap, B (m):").grid(row=2, column=0, sticky='e')
entry_cap_width = ttk.Entry(frame_pier_cap)
entry_cap_width.grid(row=2, column=1)

ttk.Label(frame_pier_cap, text="Total Design Load, Load (kN):").grid(row=3, column=0, sticky='e')
entry_cap_load = ttk.Entry(frame_pier_cap)
entry_cap_load.grid(row=3, column=1)

btn_pier_cap = ttk.Button(frame_pier_cap, text="Calculate Pier Cap", command=calculate_pier_cap)
btn_pier_cap.grid(row=4, column=0, columnspan=2, pady=10)

result_pier_cap = tk.StringVar()
ttk.Label(frame_pier_cap, textvariable=result_pier_cap, relief="sunken", padding=10).grid(row=5, column=0, columnspan=2, sticky='nsew')

# ----------------- Tab 3: Pier (Column) Design -----------------
tab3 = ttk.Frame(notebook)
notebook.add(tab3, text="Pier Design")

frame_pier = ttk.Frame(tab3, padding="10")
frame_pier.pack(fill='both', expand=True)

ttk.Label(frame_pier, text="Pier (Column) Design Calculations", font=("Arial", 14, "bold")).grid(row=0, column=0, columnspan=2, pady=5)

ttk.Label(frame_pier, text="Pier Width (m):").grid(row=1, column=0, sticky='e')
entry_pier_width = ttk.Entry(frame_pier)
entry_pier_width.grid(row=1, column=1)

ttk.Label(frame_pier, text="Pier Depth (m):").grid(row=2, column=0, sticky='e')
entry_pier_depth = ttk.Entry(frame_pier)
entry_pier_depth.grid(row=2, column=1)

ttk.Label(frame_pier, text="Design Axial Load, N_Ed (kN):").grid(row=3, column=0, sticky='e')
entry_pier_load = ttk.Entry(frame_pier)
entry_pier_load.grid(row=3, column=1)

btn_pier = ttk.Button(frame_pier, text="Calculate Pier Capacity", command=calculate_pier)
btn_pier.grid(row=4, column=0, columnspan=2, pady=10)
```

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

```
result_pier = tk.StringVar()
ttk.Label(frame_pier, textvariable=result_pier, relief="sunken", padding=10).grid(row=5, column=0, columnspan=2, sticky='nsew')

# ----------------- Tab 4: Pile Foundation Design -----------------
tab4 = ttk.Frame(notebook)
notebook.add(tab4, text="Pile Design")

frame_pile = ttk.Frame(tab4, padding="10")
frame_pile.pack(fill='both', expand=True)

ttk.Label(frame_pile, text="Pile Foundation Design Calculations", font=("Arial", 14, "bold")).grid(row=0, column=0, columnspan=2, pady=5)

ttk.Label(frame_pile, text="Pile Diameter, d (m):").grid(row=1, column=0, sticky='e')
entry_pile_diameter = ttk.Entry(frame_pile)
entry_pile_diameter.grid(row=1, column=1)

ttk.Label(frame_pile, text="Embedded Length, L (m):").grid(row=2, column=0, sticky='e')
entry_pile_length = ttk.Entry(frame_pile)
entry_pile_length.grid(row=2, column=1)

ttk.Label(frame_pile, text="Soil Resistance, q_s (kN/m²):").grid(row=3, column=0, sticky='e')
entry_qs = ttk.Entry(frame_pile)
entry_qs.grid(row=3, column=1)

ttk.Label(frame_pile, text="End Bearing Capacity, q_b (kN/m²):").grid(row=4, column=0, sticky='e')
entry_qb = ttk.Entry(frame_pile)
entry_qb.grid(row=4, column=1)

btn_pile = ttk.Button(frame_pile, text="Calculate Pile Capacity", command=calculate_pile)
btn_pile.grid(row=5, column=0, columnspan=2, pady=10)

result_pile = tk.StringVar()
ttk.Label(frame_pile, textvariable=result_pile, relief="sunken", padding=10).grid(row=6, column=0, columnspan=2, sticky='nsew')

# Start the GUI event loop
root.mainloop()
```

RESULTS AND DISCUSSION

The input data and the results of the design of Bridge substructures such as Abutement, Pier Cap, Pier and Pile are shown in figure 2, 3,4 and 5 respectively

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

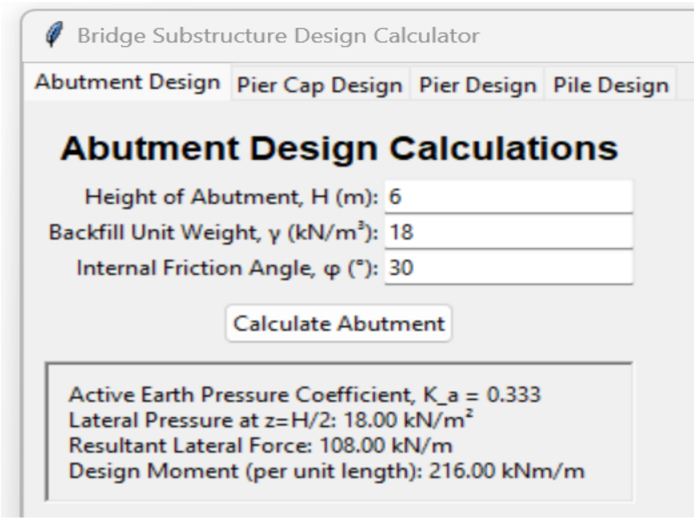

Figure 2 Results of Abutement Design

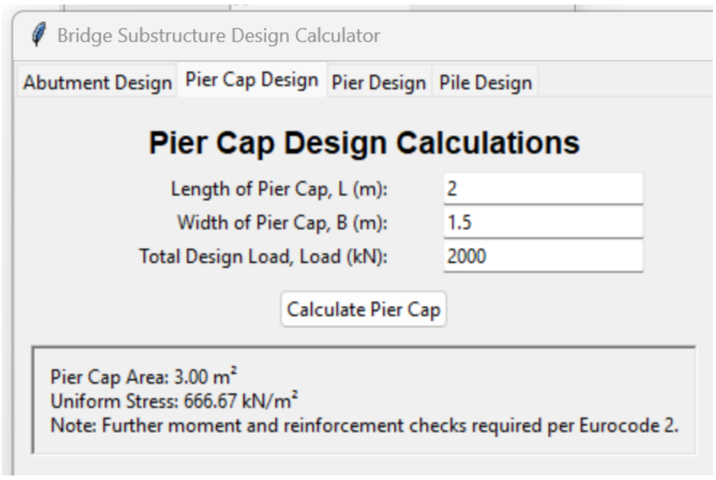

Figure 3 Results of Pier Cap Design

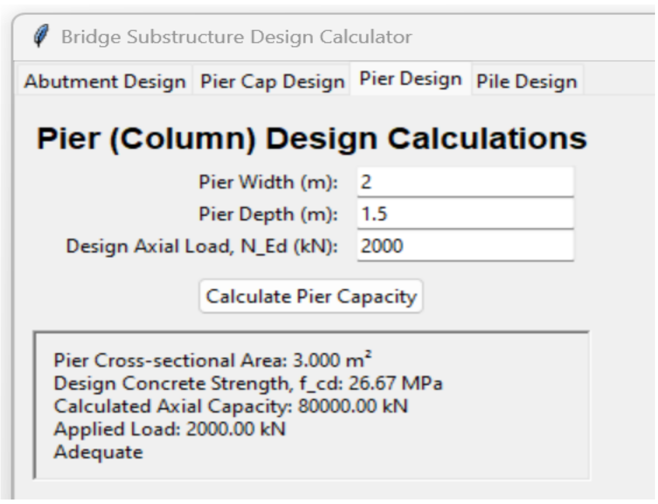

Figure 4 Results of Pier Design

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

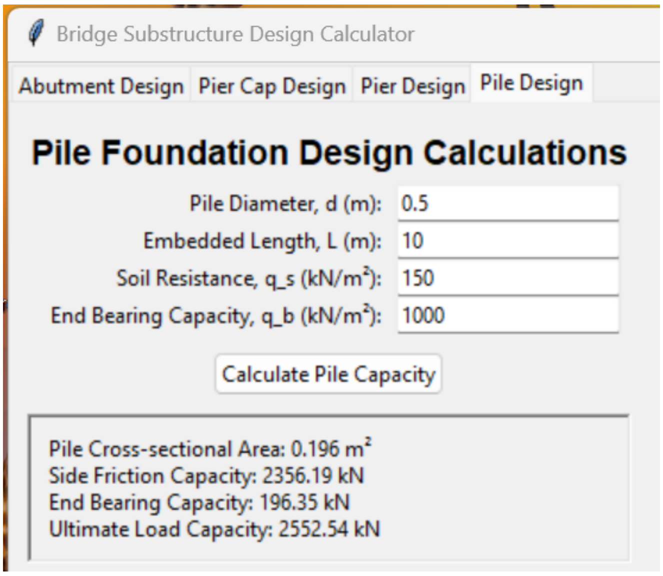

Figure 5 Results of Pile Design

CONCLUSION

The use of programming in the design of bridge substructures enhances accuracy, efficiency, and repeatability. This study provides a computational framework using Python, demonstrating its application in stability and safety assessment. These example calculations demonstrate simplified procedures for the analysis and design of bridge substructure components and foundations in accordance with Eurocodes. Each calculation provides a basic framework:

- Abutments are designed to resist lateral earth pressures.
- Pier caps are dimensioned to transfer loads from piers to the superstructure.
- Piers (or columns) are checked for axial and bending capacities.
- Pile foundations are evaluated based on side friction and end bearing resistances.

For actual projects, engineers must perform detailed analyses using accurate site data, full Eurocode formulas, and safety factors to ensure the reliability and performance of the bridge structure.

REFERENCES

[1] Prayerful, N.K., and Hanumant, B., (2015); Comparative Analysis of T-Beam Bridge by Rotational Method and Staad Pro. Ass. Prof, Civil engineering, Jay Want College of Engineering Sangli, India, B.E. Student. Civil Engineering, Jay Want College of Engineering Sangli, India.

[2] Soumya and Umadevi, 2015); Comparative Study of Courbon's Method and Finite Element Method of RC T-Beam and Deck Slab Bridge.

[3] Al-Sarraf, et al., (2005); Analysis of Composite Bridge Superstructures Using Modified GRillage Method.

[4] Mahadevan, A.K., (2005); To Design a Code for the Building of Skew Bridges.

[5] Kesarwani, S., et al., (2018); Analysis of T-Beam Along with Deck Slab by Courbon's Method.

[6] Manohar, et al., (2018); Finite Element Analysis of Slabs, Cross girders and Main girders in RC T-Beam Deck Slab Bridge.

[7] Patel, A.M. Analysis and design of bridge substructures using VB.NET, Department of Civil Engineering Indian Institute of Technology Roorkee, Roorkee-247 667 (India) June, 2008.

[8] Neville, A. M. (1995). *Properties of Concrete* (4th ed.). Longman.

[9] Mehta, P. K., & Monteiro, P. J. M. (2006). *Concrete: Microstructure, Properties, and Materials* (3rd ed.). McGraw-Hill.

[10] Federal Highway Administration (FHWA), Steel bridge Design Handbook, Substructure Design Publication No. FHWA-IF12-052 - Vol. 16 2012

iJETRM
International Journal of Engineering Technology Research & Management
Published By:
https://www.ijetrm.com/

[11] CEN 2004. Euro code 7: Geotechnical design - Part 1: General rules. EN 1997-1:2004 (E), November 2004, European Committee for Standardization: Brussels.
[12] CEN 2005. Euro code 8: Design of structures for earthquake resistance - Part 5: Foundations, retaining structures and geotechnical aspects.
[13] EN1998-5:2004 (E), May 2005, European Committee for Standardization: Brussels.
[14] CEN 2007. Euro code 7: Geotechnical design - Part 2: Ground investigation and testing. EN1997-2:2007 (E), March 2007, European Committee for Standardization: Brussels.
[15] BS EN 1992-1-1. (2000/2013). *Concrete – Specification, Performance, Production and Conformity*. British Standards Institution.
[16] BS EN 1997. (2004). *Geotechnical Design – General Rules*. British Standards Institution.
[17] IS 456:2000. (2000). *Code of Practice for Plain and Reinforced Concrete*. Bureau of Indian Standards.