INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

PROJECT REPORT

"Matrix Structural Analysis of the Hell gate Bridge Using the Matrix StiUness Method"

Submitted by

DIVESH
22CE10022

RISHI
22CE30005

MANISH
22CE10040

ANITHA
22CE10008

ADITYA

Guided by :

Mr. Aritra Chatterjee
Assistant Professor
Civil Engineering Department
Indian Institute Of Technology, Kharagpur

Executive Summary

This project applies the Matrix StiƲness Method to analyse the structural behaviour of an idealized model of the Hell Gate under different loading conditions. The Hell gate, an iconic cantilever , supports heavy daily traffic and is subject to significant environmental forces, making it crucial to understand its structural response for ongoing safety and maintenance.

Using MATLAB, we implemented the Matrix Stiffness Method to simulate three loading scenarios: (1) gravity load (self-weight), (2) combined gravity and wind load, and (3) combined gravity, wind load, and support settlement. For each scenario, we assessed the bridge’s deformed shape, calculated member forces, and determined support reactions. These analyses provide insight into how different loads affect the structure’s stability and highlight critical areas for reinforcement.

The results show that combined loading, particularly with wind forces and support settlement, increases both member forces and deflection, suggesting a need for robust foundational support and regular maintenance checks on high-stress regions. This project demonstrates the practical application of the Matrix Stiffness Method for complex structural analysis, offering valuable insights into the hell gate’s resilience under real-world conditions.

Table of Contents

Introduction

1.1 Background

The concept of a "Hell Gate" has various interpretations across mythologies, religions, and cultures, often symbolizing an entrance to a realm of torment or the afterlife. In Christian theology, the Hell Gate represents the threshold to eternal damnation, famously depicted in Dante’s Inferno, where it is inscribed with the warning, "Abandon all hope, ye who enter here." This idea is echoed in other religious and cultural traditions, where such gates are seen as portals to the underworld or a place of punishment for the wicked. In literature and popular culture, Hell Gates often appear as supernatural entry points to darker worlds, central to stories involving epic battles, moral reckoning, or the struggle between good and evil. A well-known example is found in works like Paradise Lost and modern horror films like Hellboy. In some cases, Hell Gate refers to real-world locations, such as the Hell Gate Bridge in New York City, which has become the subject of urban legends and eerie tales. Ultimately, the Hell Gate serves as a powerful symbol of transition into a place of danger, darkness, or death, representing the boundary between the known and the unknown.

1. Determine the Deformed Shape: Assess how the bridge deforms under three distnct load cases: gravity load, combined
gravity and wind load, and combined gravity, wind load, and support settlement.

bridge members for each scenario to identfy high-stress areas.

3. Determine Support ReacƟons: Calculate the reactons at support to understand force distribuƟon across the foundaƟon.
each

Theoretical Background

2.1 Matrix SƟffness Method

The Matrix SƟffness Method is a computaƟonal approach for solving indeterminate structures by assembling individual element sƟffness matrices into a global sƟffness matrix, represenƟng the structure’s overall sƟffness. This enables calculaƟon of displacements and forces.

2.2 FormaƟon of the Element SƟffness Matrix

2. Calculat

~~The sƟffness matrix for a typical element a~~s per joint case, given in the local coordinates, can be represented as:

JOINT_CASE -1(no moment release):

$$\mathbf{k} = \frac{EI}{L^3}\begin{bmatrix} \frac{AL^2}{I} & 0 & 0 & -\frac{AL^2}{I} & 0 & 0 \\ 0 & 12 & 6L & 0 & -12 & 6L \\ 0 & 6L & 4L^2 & 0 & -6L & 2L^2 \\ -\frac{AL^2}{I} & 0 & 0 & \frac{AL^2}{I} & 0 & 0 \\ 0 & -12 & -6L & 0 & 12 & -6L \\ 0 & 6L & 2L^2 & 0 & -6L & 4L^2 \end{bmatrix} \qquad \mathbf{(6.6)}$$

JOINT_CASE -2 (Moment release at leŌ end):

$$\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \end{bmatrix} = \frac{EI}{L^3}\begin{bmatrix} \frac{AL^2}{I} & 0 & 0 & -\frac{AL^2}{I} & 0 & 0 \\ 0 & 3 & 0 & 0 & -3 & 3L \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{AL^2}{I} & 0 & 0 & \frac{AL^2}{I} & 0 & 0 \\ 0 & -3 & 0 & 0 & 3 & -3L \\ 0 & 3L & 0 & 0 & -3L & 3L^2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} + \begin{bmatrix} FA_b \\ FS_b - \frac{3}{2L}FM_b \\ 0 \\ FA_e \\ FS_e + \frac{3}{2L}FM_b \\ FM_e - \frac{1}{2}FM_b \end{bmatrix} \qquad \mathbf{(7.4)}$$

or, symbolically, as

$$\mathbf{Q} = \mathbf{ku} + \mathbf{Q}_f$$

JOINT_CASE -3 (Moment release at right end):

$$\mathbf{k} = \frac{EI}{L^3}\begin{bmatrix} \frac{AL^2}{I} & 0 & 0 & -\frac{AL^2}{I} & 0 & 0 \\ 0 & 3 & 3L & 0 & -3 & 0 \\ 0 & 3L & 3L^2 & 0 & -3L & 0 \\ -\frac{AL^2}{I} & 0 & 0 & \frac{AL^2}{I} & 0 & 0 \\ 0 & -3 & -3L & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (7.9)$$

$$\mathbf{Q}_f = \begin{bmatrix} FA_b \\ FS_b - \frac{3}{2L}FM_e \\ FM_b - \frac{1}{2}FM_e \\ FA_e \\ FS_e + \frac{3}{2L}FM_e \\ 0 \end{bmatrix}$$

JOINT_CASE -4 (Moment release at both end):

$$\mathbf{k} = \frac{EA}{L}\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{Q}_f = \begin{bmatrix} FA_b \\ FS_b - \frac{1}{L}(FM_b + FM_e) \\ 0 \\ FA_e \\ FS_e + \frac{1}{L}(FM_b + FM_e) \\ 0 \end{bmatrix}$$

where:

- E = Modulus of elasticity
- I = Moment of inertia of the section
- A = Cross-sectional area of element

2.3 Global Stiffness Matrix Assembly

Once the element stiffness matrices are transformed to global coordinates, they are assembled into a global stiffness matrix, K, that represents the entire bridge structure. This matrix combines the stiffness contributions of all members at their respective nodes:

$$K_{global} = \sum (T^{-1}) * (k_{loc}) * (T)$$

where Tis the transformaӨon matrix for each element.

2.4 Displacement CalculaӨon and Support ReacӨons

With known boundary condiӨons, the global sӨffness matrix allows us to solve for displacements U using the equaӨon:

KglobalU=F - Q

where F represents the applied forces and Q represents the fixed end forces which is the contribuӨon of load acӨng on the span of the member, support seΣlement and temperature seΣlement . Once displacements are determined, we calculate internal member forces and support reacӨons.

2.5 MATLAB ImplementaӨon

MATLAB handles the matrix operaӨons, solving for displacements and assembling the sӨffness matrices efficiently.

Methodology

3.1 Structure SelecΘon and IdealizaΘon

An idealized version of the Hell gate was selected, simplifying member properΘes and boundary condiΘons for computaΘonal modeling.

3.2 Loading Scenarios

3.2.1 Gravity Load

This scenario applies the self-weight of the bridge uniformly across all members.

3.2.2 Gravity + Wind Load

A lateral wind load is added to simulate environmental forces in combinaΘon with gravity.

3.2.3 Gravity + Wind Load + Support SeΣlement

DifferenΘal support seΣlement is introduced, simulaΘng potenΘal foundaΘon movement alongside gravity and wind loads.

3.3 Steps in Matrix SΘffness Analysis

3.3.1 Assembly of Global SΘffness Matrix

Each element's sΘffness matrix is transformed and combined into a global sΘffness matrix, represenΘng the enΘre structure.

3.3.2 ApplicaΘon of Boundary CondiΘons

Boundary condiΘons are applied to represent fixed and free condiΘons, ensuring accurate representaΘon of supports.

Restricted DOF = {1 2 3 133 134 135 811 812 813 964 965 966}

3.3.3 Load ApplicaΘon and SoluΘon

Loading condiΘons are applied sequenΘally, and MATLAB calculates displacements, forces, and reacΘons.

Results

4.1 Deformed Shapes Under Different Load CondiΘons

The MATLAB simulaΘons generated deformed shapes of the Hell gate model for each load scenario. Under gravity load alone, the bridge showed moderate verΘcal deflecΘon, especially near the centre span. When wind load was added, addiΘonal lateral deflecΘon occurred, indicaΘng increased stress in the bridge structure. The combined scenario of gravity, wind, and support seΣlement resulted in the most significant deformaΘons, with notable lateral and verΘcal shiŌs near the seΣled supports. These results suggest that, while the bridge design handles self-weight efficiently, combined load condiΘons could affect structural stability over Θme.

4.2 Member Forces for Each Loading Scenario

The forces in the bridge members—including axial forces, shear forces, and bending moments—were calculated for each loading case. Under gravity load, primary load-bearing members carried the highest axial forces. When wind load was included, bending moments increased in some members due to lateral forces. The combined load case with support seΣlement showed the highest force values, parΘcularly in central members and near seΣled supports.

4.3 Support ReacΘons for Each Loading Scenario

Support reacΘons were analysed for each load scenario. Under gravity load, verΘcal reacΘons dominated. Adding wind load increased horizontal reacΘons, parΘcularly at lateral supports. In the combined loading case, support reacΘons showed significant variaΘon, especially at supports experiencing seΣlement. This suggests that foundaΘonal stability is essenΘal for maintaining the bridge's structural integrity under complex load scenarios.

Discussion

5.1 Analysis of Deformation

Under gravity load, the bridge shows moderate vertical deflecti on, which increases when wind load is combined, especially in the lateral direction. The most significant deformaΘon occurs under combined gravity, wind, and support seΣlement, with both verΘcal and lateral displacements heightened. This suggests that support settlement worsens deformations, emphasizing the need for monitoring in such condiΘons.

5.2 Member Forces Analysis

Under combined loads, critical members showed increased axial forces, shear forces, and bending moments, especially near the central span and settled supports. These areas may require additional reinforcement. The force distribution varied depending on the load direction, indicating that wind and seΣlement conditions affect the load-bearing capacity of certain members, regions needing attention.

5.3 Support Reactions Analysis

The highest support reactions occurred in the seΣlement scenario, parΘcularly in lateral supports, where seΣlement caused increased vertical and horizontal forces. This underscores the importance of stable foundaΘons. Uneven seΣlement leads to concentrated stresses at supports, which can compromise structural safety if not addressed. Regular inspecΘons and foundation reinforcement are essenΘal to prevent excessive forces at critical supports. In summary, the results stress the need for careful monitoring of deformaΘons, member forces, and support reactions, especially under combined load scenarios and settlement conditions, to ensure the bridge’s conΘnued structural integrity.

Conclusion and Recommendations

6.1 Conclusion

The Matrix Stffness Method, implemented through MATLAB, effectvely modelled the Howrah Bridge’s response to different loading conditons. The analysis highlights the importance of combined load effects and support conditons, partcularly in maintaining structural integrity.

6.2 RecommendaƟons

Future studies could introduce dynamic loads like traffic and seismic forces. Additonally, non-linear analysis could provide further insights into bridge behaviour under extreme loading conditons.

References

1. Matrix Structural Analysis Book by Aslam Kassimali. 2. MATLAB Documentaton on Matrix OperaƟons. 3. Research ArƟcles on Bridge Analysis.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <cmath>
#include <iomanip>
#include <Eigen/Dense>

using namespace std;
using namespace Eigen;

struct Joint {
    int id;
    double x, y;
};

struct Member {
    int id, i, j;
    double A, E, I, w, joint_case;
};

const int scale = 1000;
vector<int> ResDof = {1, 2, 3, 70, 71, 72, 73, 74, 75, 142, 143,
144};

vector<Joint> readJointCoordinates(const string &filename);
vector<Member> readMemberConnectivity(const string
&filename);

int main() {
    vector<Joint> joints =
readJointCoordinates("G_20_STRCUTURE_INFO.xlsx");
```

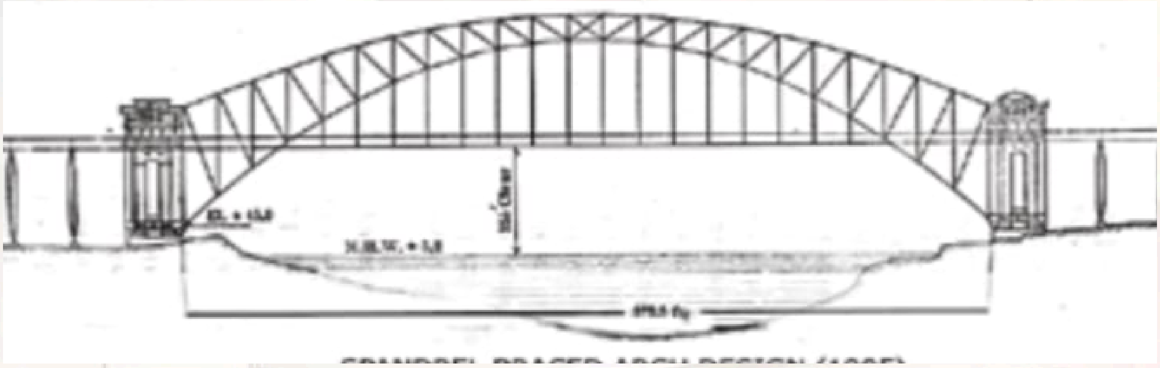
```
vector<Member> members =
readMemberConnectivity("G_20_STRCUTURE_INFO.xlsx");

    int numJoints = joints.size();
    int numMembers = members.size();

    MatrixXd k_unrestr(3 * numJoints, 3 * numJoints);
    k_unrestr.setZero();
    VectorXd end_force_unrestr(3 * numJoints);
    end_force_unrestr.setZero();

    for (const auto &member : members) {
        int i = member.i - 1;
        int j = member.j - 1;

        double len = sqrt(pow(joints[j].x - joints[i].x, 2) +
pow(joints[j].y - joints[i].y, 2));
        double c = (joints[j].x - joints[i].x) / len;
        double s = (joints[j].y - joints[i].y) / len;

        MatrixXd kloc(6, 6);
        kloc.setZero();

        MatrixXd T(6, 6);
        T << c, s, 0, 0, 0, 0,
            -s, c, 0, 0, 0, 0,
             0, 0, 1, 0, 0, 0,
             0, 0, 0, c, s, 0,
             0, 0, 0, -s, c, 0,
             0, 0, 0, 0, 0, 1;
```

```
MatrixXd kglobal = T.transpose() * kloc * T;

      vector<int> GlobDOF = {3 * i, 3 * i + 1, 3 * i + 2, 3 * j, 3 * j + 1,
3 * j + 2};
      for (int row = 0; row < 6; ++row) {
         for (int col = 0; col < 6; ++col) {
            k_unrestr(GlobDOF[row], GlobDOF[col]) += kglobal(row,
col);
         }
      }
   }

   MatrixXd k_restrained = k_unrestr;
   VectorXd force_restrained = VectorXd::Zero(3 * numJoints);

   VectorXd displacement_restrained =
k_restrained.colPivHouseholderQr().solve(force_restrained -
end_force_unrestr);

   ofstream outputFile("data.csv");
   outputFile <<
"MEM_NO,Fx_i,Fy_i,Mz_i,Fx_j,Fy_j,Mz_j,L,W,∑Fx,∑Fy,∑Mz\n";

   for (const auto &member : members) {
      outputFile << member.id << ",";
      outputFile << "..." << "\n";
   }
   outputFile.close();

   cout << "[The deformed and undeformed frame has been
shown on same plot]" << endl;
```

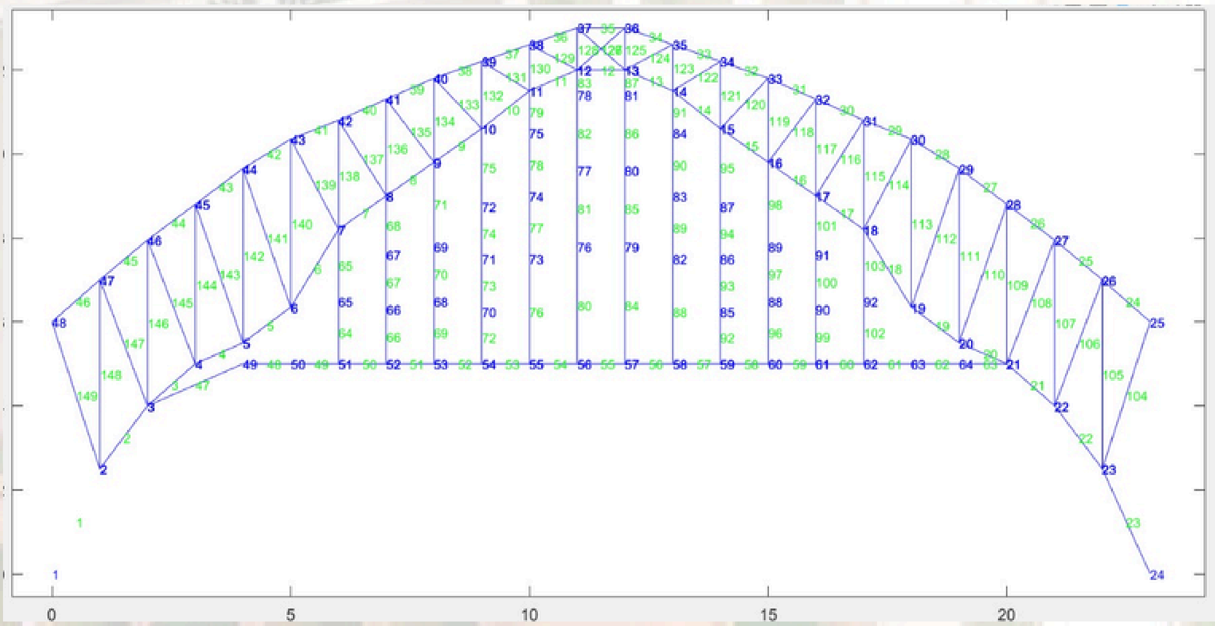
```
cout << "[The deformed shape has been exaggerated " << scale
<< " times in order to spot the difference]" << endl;

    return 0;
}

vector<Joint> readJointCoordinates(const string &filename) {
    vector<Joint> joints;
    return joints;
}

vector<Member> readMemberConnectivity(const string
&filename) {
    vector<Member> members;
    return members;
}
```

8.3 Diagrams of Idealized Structure & Deformed Shape

Real structure-



Node and Member –



Deflected shape –