EFFICIENT CULVERT LOAD RATING THROUGH DATA-DRIVEN SOLUTIONS

M VENTER[1]* and J BOOYSEN[1]**

[1]Zutari, Riverwalk Office Park, 41 Matroosberg Road, Ashlea Gardens, Pretoria 0081;
*Tel: 061 3014584; Email: madeleine.venter@zutari.com
**Email: jan.booysen@zutari.com

ABSTRACT

Culverts play a critical role in managing water flow beneath roads and railways, necessitating accurate load rating to ensure structural integrity and safety. Evaluating the structural capacity of culverts to support anticipated traffic load and other loads is imperative for maintaining the efficiency and safety of transportation operations.

Traditionally, the load rating process has been laborious and time-consuming, primarily due to the extensive data extracted from structural analysis programs such as Midas or Space Gass. In this paper, a case study involving 5 culverts is presented, highlighting the importance of automation in load rating assessments. The study proposes data-driven solutions that harness the power of programming languages such as Microsoft Visual Basic Application (VBA) and Python, coupled with specialized modules such as Pandas DataFrame. These tools facilitate efficient processing and in-depth analysis of culvert data, optimizing the load rating process.

The automation of load rating calculations through programming streamlines the assessment process, significantly reducing the time and effort required for accurate results. By embracing automation and leveraging advanced software, engineers can enhance their ability to swiftly and accurately evaluate culvert load ratings, ultimately enhancing infrastructure safety and operational efficiency in the transportation sector.

Keywords: Load-rating analysis, Automation, Computational Design, Software Engineering, Culverts.

1. INTRODUCTION

To understand the problem addressed in this paper, one needs to understand the concept of load-rating of structures first. The load-rating is essential in determining the maximum live load that can cross a bridge or culvert under serviceability conditions.

Rupp et al. (2022) propose a calculation to determine the Rating Factor (RF) for a structure, expressed as:

$$RF = \frac{(C - DL)}{LL}$$

In which RF is the rating factor, C is the load capacity of the structure, DL is the dead load effect, and LL is the live load effect. A rating factor less than 1 indicates insufficient capacity to support the live loads on the bridge/culvert.

The evaluation of the rating factor is necessary at each potential critical section (commonly slab midspan and corners for box culverts), considering various demand types and directions (positive and negative bending, shear, and torsion). Multiple load cases, including maximum and minimum live load envelope results, was assessed. Despite its conceptual simplicity, the load rating of a multi-barrel box culvert may involve numerous calculations; the lowest single load rating factor determines the overall structure's load rating.

Automating processes, such as load rating for structures, yields numerous advantages for both employees and clients. According to Brous et al. (2020), automation of processes provide numerous advantages for organisations. The primary benefit is the substantial enhancement of organizational efficiency. By employing software to replicate routine human tasks, automation accelerates processing time, heightens accuracy, and diminishes the likelihood of human error. A noteworthy outcome is the improved productivity resulting from the streamlining of time-consuming tasks. Automation with machines enables the swift handling of repetitive tasks that would otherwise consume significant time (Spring et al., 2022).

In addition to efficiency and productivity, financial gains constitute another significant advantage of automation. By eliminating wasted time, automation allows employees to redirect their efforts towards tasks demanding creative and technical thinking, translating into cost savings. The elements of client and employee satisfaction are pivotal for any business, and automation substantially enhances these factors. Tasks that typically require hours or days for completion can now be accomplished in minutes, offering substantial benefits to clients in terms of both time and cost savings. This efficiency boost can enhance a company's reputation, increasing the likelihood of continued collaboration.

The elimination of time-consuming and repetitive work liberates employees to focus on more intellectually challenging tasks. This shift not only enhances employee satisfaction but also contributes to increased workplace efficiency, heightened morale, and elevated overall productivity. The benefits of automation extend beyond these, and this paper delves into the automation of the load-rating process, showcasing its applicability to various engineering problems and tasks that would otherwise be tedious and time-consuming for both employees and clients.

1.1 Aim of paper

The aim of this paper is to provide an insight in the benefits that result from automating engineering processes and problems that prove time-consuming in an attempt to save time and cost and increase efficiency, productivity, reduced errors and employee and client satisfaction. By investigating a case-study considering load-rating of culverts, this research aims to address the issue of time-consuming manual load-rating methods and proposes a solution of using automated methods for time-efficient and cost-effective solutions.

1.1.1 Problem Statement

The load rating process of culverts plays a pivotal role in ensuring the safety and functionality of transportation infrastructure. As civil engineering projects continue to evolve and expand, the increasing demand for accurate and efficient load rating assessments becomes imperative. The traditional manual methods employed in the load rating process are not only time-consuming but also prone to human errors, limiting the ability to keep pace with the growing infrastructure demands. According to Mirzaei et al.

(2022), manual processing of datasets in infrastructure management is costly, time consuming and error-prone.

This paper addresses the need for a transformative shift in the load rating process of culverts by leveraging automation technologies. The current methods involve intricate calculations, extensive data analysis, and subjective judgment, leading to inconsistencies and delays. Furthermore, the complexity of varying culvert designs exacerbates the challenges in achieving standardized and precise load rating assessments.

Automation presents an opportunity to streamline the load rating process, significantly improving efficiency and accuracy. While the integration of machine learning in this research aims to develop a systematic and standardized approach for automating the load rating of culverts, it is important to acknowledge that this endeavor may necessitate further development, particularly in the realm of machine learning solutions. The implementation of automation not only accelerates the assessment process but also aims to enhance the reliability of results by minimizing the potential for human errors.

A similar study by Lawson et al. (2013) involved The Texas Department of Transportation (TxDOT) project 5-5849-01 to develop a Windows-based desktop application software called CULVLR: Culvert Load Rating, Version 1.0.0, aimed at automating load rating calculations for reinforced concrete box culverts. This project had major success in the United States, proving that the need for this type of software is high in South Africa.

The successful implementation of automated load rating processes for culverts will not only enhance the overall efficiency of infrastructure management but also contribute to the longevity and resilience of transportation systems. This research aims to provide a robust foundation for the development of a standardized, reliable, and scalable automated approach to culvert load rating, paving the way for advancements in the field of structural engineering and infrastructure management.

1.1.2 Scope of Paper

This paper aims to streamline culvert load rating by leveraging a structural analysis program called SpaceGass. This involves automating the creation of grillage frames, inputting material properties, and defining loading and boundary conditions using Visual Basic Application (VBA) within Microsoft Excel.

The scope of this paper also includes the data handling of the exported analysis results utilising Pandas DataFrame and Python. The Python package Pandas was used as it is particularly suitable for handling data due to its powerful and flexible data structures, like DataFrames, which make data manipulation and analysis straightforward and efficient. It provides a wide range of utilities for indexing, slicing, transforming, and aggregating data, which are invaluable for exploratory data work and postprocessing.

Although the structural capacity of these structures were determined according to AS5100 (2017) (Australian Standard for Bridge Design), the calculations and capacity evaluations do not form part of the scope of this paper.

1.2 Digital Load Rating Process

The load rating process of culverts involves first creating a digital model using structural engineering software such as SpaceGass or Midas. The as-built drawings of the structure are used to obtain the information of the structure, the dimensions, sizes of members, materials and boundary conditions. In this study, the software used to conduct the structural analysis was SpaceGass, a multi-purpose 3D analysis and design program for structural engineers widely used in Australia and New Zealand.

When creating a 3D digital model of a culvert or bridge, several steps are followed. Initially, the geometry of the structure is established by constructing a framework using nodes and elements. Following this, boundary conditions and restraints are applied to nodes or elements, which may include spring stiffnesses on external walls and beneath the bottom slab for culverts. Additionally, pin connections at wall-slab junctions, as indicated in the as-built drawings, might be included. Cross-sectional areas are then defined and assigned to relevant elements, and material properties are specified and assigned accordingly. Subsequently, primary loading such as self-weight, dead load, and live load are defined within the model. Finally, the structure undergoes analysis to assess its performance and behavior under various conditions.

After the structure is analysed, the results are exported in a chosen format, either a .csv or .txt file. As there are many nodes, the analysis results could be substantial. For a simple 1-cell box culvert for example, this process would be relatively effortless and straightforward. However, when a structure becomes complicated or there are multiple structures to be analysed, this process could become time-consuming. For example, Figure 1 represents the nodes and elements of a 7-cell culvert in SpaceGass. This specific model consists of 1712 nodes and 3248 elements.

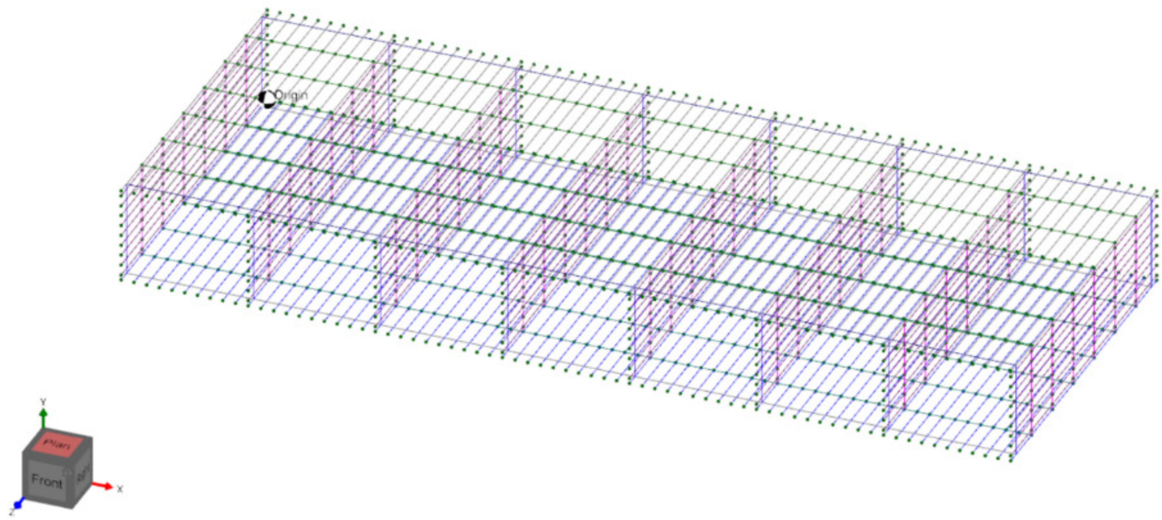

Figure 1: Example of grillage model of 7 cell culvert

2. METHODOLOGY

2.1 Incorporating Data-Driven Solutions

The time-consuming nature of ‘building’ grillage models of culverts in structural analysis programs were previously described. Due to the relatively straightforward structure of box culverts, it is reasonable to assume that their shape and size can be defined using specific parameters. Therefore, the automation of this process is justified. A few parameters that describe a culvert include the following:

- Number of cells.
- Cell size (height and width).

- Culvert length.
- Slab thicknesses (top and bottom).
- Wall thicknesses.
- Material properties.
- Boundary conditions.
- Loading.

The automation of this procedure begins with developing a programming solution to write the position of nodes and placing elements between these nodes in an x, y, z space in a spreadsheet format that can be imported into SpaceGass. The grillage is formed by the number of cells and number of frames (specified by the user). Each frame consists of the number of cells and the cell width and height is divided into 10 members. An example of a frame can be seen in Figure 4. The user specifies the number of frames, which are then positioned along the length of the culvert, with transverse members added between them.

The above process is automated using Visual Basic Application (VBA) within excel. The main parameters of the culvert are populated and by pressing a button to run a macro, the information for the nodes and elements are generated. A snippet from the code used to create nodes can be seen in Figure 2, where nested loops are used to loop through the number of frames, creating nodes and subsequently inserting them into a worksheet in Microsoft Excel. A snippet of the code used to create elements can be seen in Figure 3, where a similar process is followed, however some constants are also added to the worksheet.

```vb
'Top slab nodes
nodeNumber = 1001
For f = 1 To nFrames

'First (Left) External Cell
For j = LBound(myArray1) To UBound(myArray1)
    With wsNodes
        .Cells(rowCounter + 1, 1).value = nodeNumber + (f - 1) * 1000
        .Cells(rowCounter + 1, 2).value = myArray1(j) + ((f - 1) * sFrames * Tan(2 * Pi * skewAngle / 380))
        '.Cells(rowCounter + 1, 2).Value = extCellWidth * ((j - 1) / 10) + ((f - 1) * sFrames * Tan(2 * Pi * skewAngle / 380))
        .Cells(rowCounter + 1, 3).value = cellHeight
        .Cells(rowCounter + 1, 4).value = (f - 1) * sFrames
        rowCounter = rowCounter + 1
        nodeNumber = nodeNumber + 1
    End With
Next j
currX = currX + extCellWidth
```

Figure 2: Code used in producing nodes

```vb
'Bottom slab members
    For f = 1 To nFrames
    For i = 1 To NoCells
        For j = 1 To 10
        With wsMembers
            .Cells(rowCounter + 1, 1).value = 1200 + ((f - 1) * 1000) + ((i - 1) * 10) + j
            .Cells(rowCounter + 1, 2).value = 0
            .Cells(rowCounter + 1, 3).value = 0
            .Cells(rowCounter + 1, 4).value = "n/a"
            .Cells(rowCounter + 1, 5).value = "Normal"
            .Cells(rowCounter + 1, 6).value = 1200 + ((f - 1) * 1000) + ((i - 1) * 10) + j
            .Cells(rowCounter + 1, 7).value = 1200 + ((f - 1) * 1000) + ((i - 1) * 10) + j + 1
            .Cells(rowCounter + 1, 8).value = 1
            .Cells(rowCounter + 1, 9).value = 1
            .Cells(rowCounter + 1, 10).value = "FFFFFF"
            .Cells(rowCounter + 1, 11).value = "FFFFFF"
            .Cells(rowCounter + 1, 12).value = 0
            .Cells(rowCounter + 1, 13).value = 0
            .Cells(rowCounter + 1, 14).value = 0
            .Cells(rowCounter + 1, 15).value = 0
            .Cells(rowCounter + 1, 16).value = 0
            rowCounter = rowCounter + 1
        End With
    Next j
    Next i
    Next f
```

Figure 3: Code used in producing elements

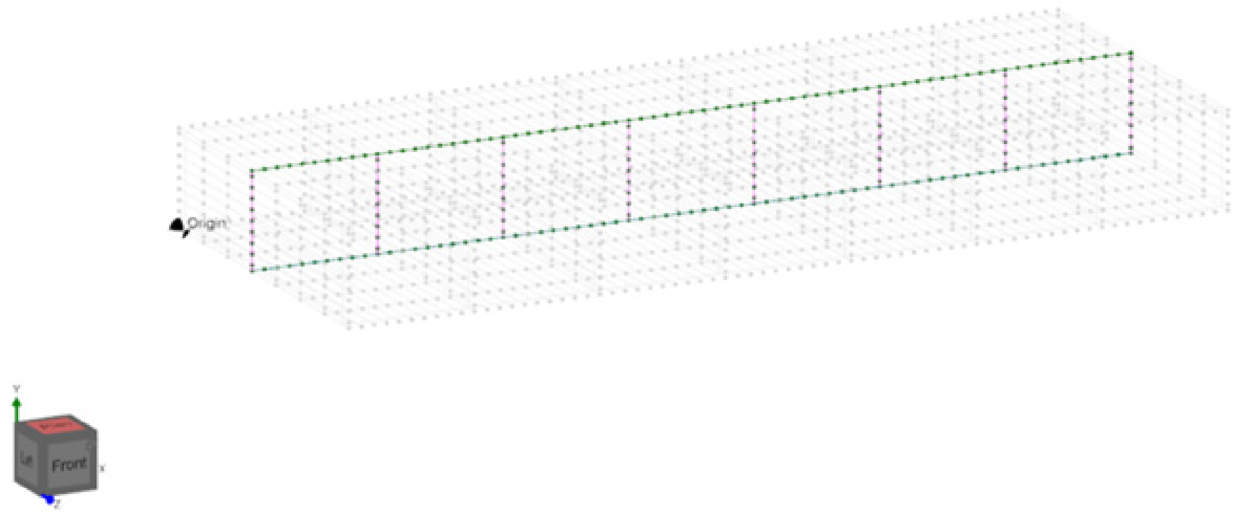

Figure 4: Frame example in SpaceGass

In generating the grillage model, it is critical is to name the elements in the automation process. The naming convention is critical because it ensures that specific elements can be easily identified and accessed during post-processing, such as when analyzing critical data points like maximum bending moments and shear forces in various structural components. This systematic approach facilitates efficient and accurate analysis, essential for structural assessment and optimization. Continuity and adherence to guidelines and specifications are essential to ensure conformity. The naming convention used for the elements in this project can be seen in Figure 5. In a similar way, the nodes are also numbered using a specific method, in order for post-processing to be simplified.

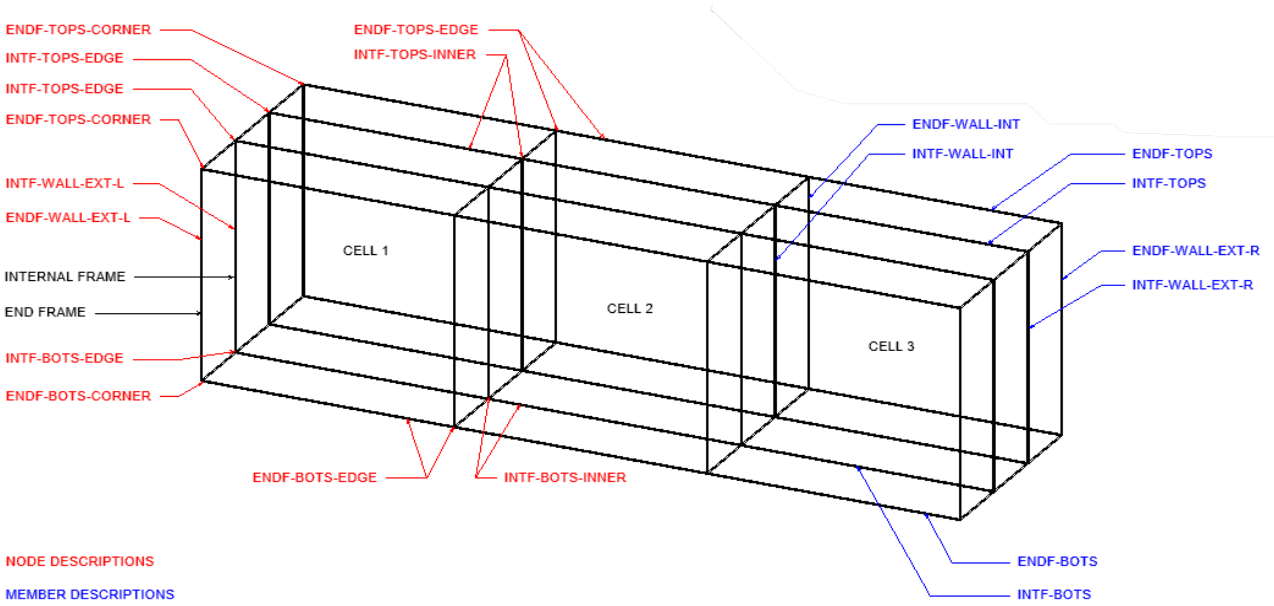

Figure 5: Naming of culvert members

After the grillage model is built, the cross-sectional geometry of the members are generated and imported using the SpaceGass library. The cross-sectional geometry of the culvert members are also generated through an automated process in VBA and Microsoft Excel using the same parameters as specified above. The result of importing the cross-sectional geometry can be seen in Figure 6. It is important to import geometry for transverse members, to provide stiffness between the longitudinal members.

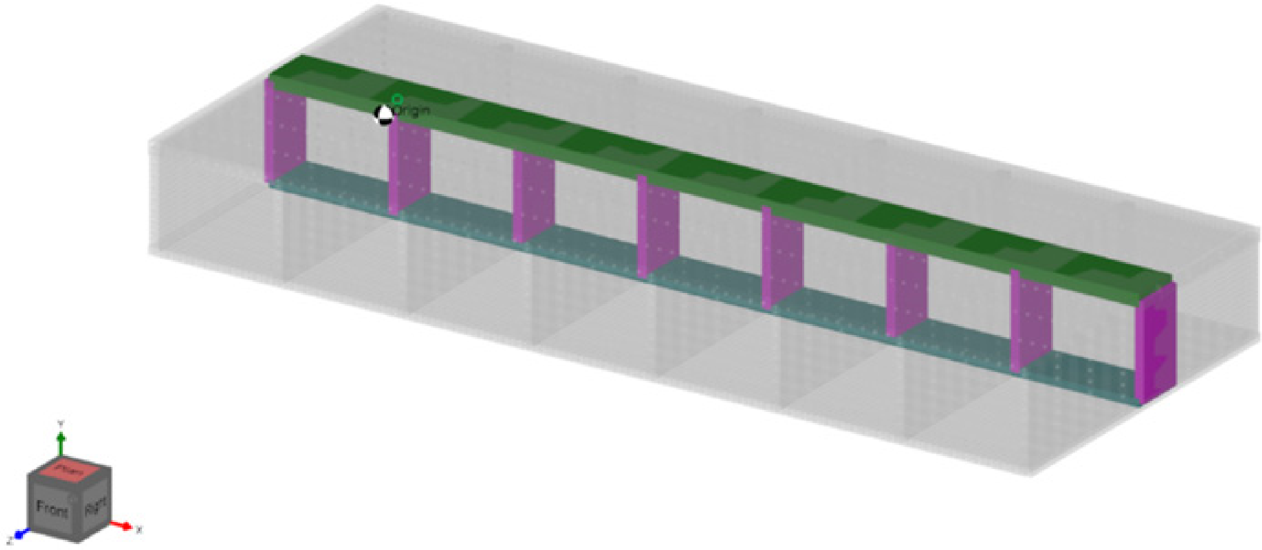

Figure 6: Cross-sectional geometry for members

The next step of the process involves generating loads to apply to members of the grillage. This is also done by an automated process. By providing material parameters such as reinforced concrete density and soil density as well as parameters such as fill height and surcharge, loads are generated as line loads on longitudinal members. By inputting the loading parameters as depicted below in Table 1, in conjunction with the previously defined geometry, the loading is automatically generated upon activating a designated button, thereby applying it accurately to the specified elements illustrated in Figure 5.

Table 1: Loading information

Asphalt thickness (mm)	0
Asphalt/Road Width (m)	6.8
Asphalt density (kg/m³)	2200
Depth of fill (m)	0
Fill Density (kg/m³)	2500
Coefficient of pressure (ko)	0.3
Concrete Density (kg/m³)	2600

The nature of this project involved analysing the structure for 21 different types of vehicles with between 2-3 load cases for each vehicle, according to AS5100:

- One vehicle positioned 600 mm from the edge of the barrier to the centreline of the tyre/dual tyre, the second vehicle placed in the next lane 600 mm from the centreline of the road to the centreline of the tyre/dual tyre.
- One vehicle positioned 600 mm from the centreline of the road to the centreline of the tyre/dual tyre with the other vehicle in the next lane also 600 mm from the centreline of the road to the centreline of the tyre/dual tyre.
- For straddling vehicles, one case was analysed where one vehicle was placed over adjacent lanes, with a 0-1000 mm offset from the centreline of the lanes to the centreline of the vehicle.

In total, each culvert has to be analysed for 43 moving loadcases. The live loads are generated by simply specifying lane geometry and vehicle properties as seen in Table 2. By creating lanes and wheel loads, moving vehicle loads are generated in Excel and imported into SpaceGass. An example of a moving load can be seen in Figure 8. For post-processing purposes, the vehicle positioning for each load case had to be drafted to submit to the client. VBA was once again utilised to draw each of these vehicle positionings in the form of an Excel graph using the available data (culvert geometry and moving load parameters). In aggregate, a total of 43 images could be rendered for a culvert within a matter of seconds using the automated process. This seemingly straightforward yet labor-intensive endeavor could potentially require hours or even days to accomplish using Computer-Aided Design (CAD) software. An excerpt of the code employed for this purpose is provided in Figure 9, while the outcome of generating one of the load cases is depicted in Figure 10.

Table 2: Moving load parameters

Parameter	**Value**	**Unit**
Number of cells	7	
Number of lanes	2	m
Lane width	3.4	m
Total bridge width	7.3	m
Kerb + Shoulder Width	0.25	m
Middleman width	0	m
Offset in-lane	0.6	m
Culvert type	In-situ	
Depth of fill	0	m
Skew	0	°
Straddling offset	1	m
Kerb Width	0.25	m
Shoulder Width	0	m
Load spacing	0.2	m

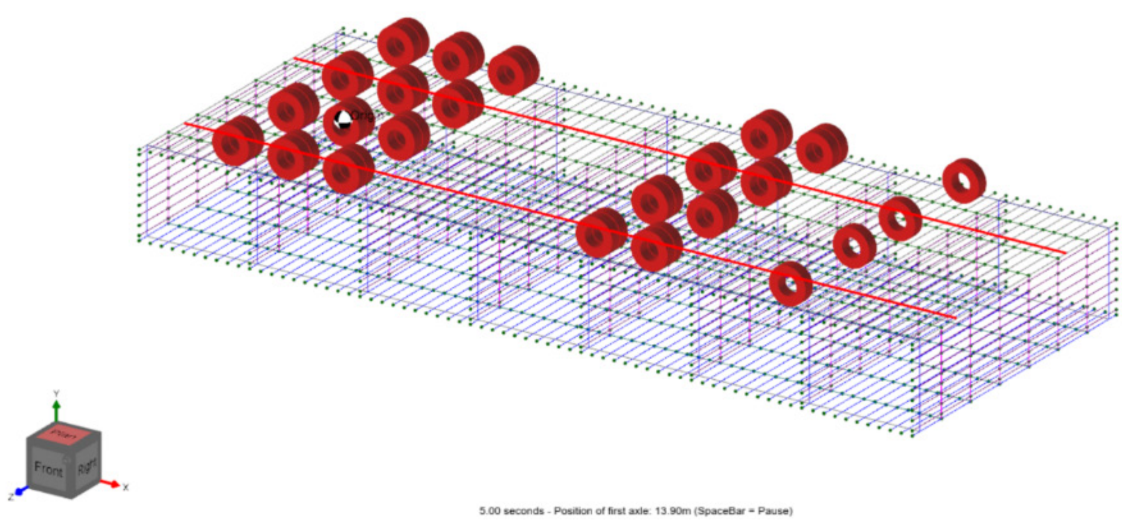

Figure 8: Moving load generation in SpaceGass (6-axle all terrain crane & 3-axle dolly)

```
Sub DrawCulvert(chart As chart, CulvertLength As Double, CulvertHeight As Double, TopSlabThk As Double, BotSlabThk As Double, FillHeight As Double, HeadWallHeight As Double,
BarrierLeftDim As Double, BarrierRightDim As Double, LaneWidth As Double, KerbShoulder As Double, Optional LaneDescr As String, Optional ShoulderDescr As String)

    Dim DimSpacing As Double
    Dim Offset As Double
    DimSpacing = 0.7

 '- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 'BASIC CULVERT GEOMETRY
 ' Draw a box
    'Draw the culvert outline
    'DrawBoxchart, Left, Top, Width, Height
    'DrawLine chart, startX, startY, endX, endY, DashStyle

    DrawLine chart, 0, 0, CulvertLength, 0.025, msoLineSolid
    DrawLine chart, 0 + HeadWallWidth, CulvertHeight, CulvertLength - HeadWallWidth, CulvertHeight, msoLineSolid
    DrawLine chart, 0, 0, 0, CulvertHeight, msoLineSolid
    DrawLine chart, CulvertLength, 0, CulvertLength, CulvertHeight, msoLineSolid

    'Draw hidden lines
    DrawLine chart, 0, BotSlabThk, CulvertLength, BotSlabThk, msoLineDash
    DrawLine chart, 0, CulvertHeight - TopSlabThk, CulvertLength, CulvertHeight - TopSlabThk, msoLineDash

    'Draw headwalls Left and Right
    DrawLine chart, 0, CulvertHeight + HeadWallHeight, HeadWallWidth, CulvertHeight + HeadWallHeight, msoLineSolid
    DrawLine chart, 0, CulvertHeight, 0, CulvertHeight + HeadWallHeight, msoLineSolid
    DrawLine chart, HeadWallWidth, CulvertHeight, HeadWallWidth, CulvertHeight + HeadWallHeight, msoLineSolid

    DrawLine chart, CulvertLength, CulvertHeight + HeadWallHeight, CulvertLength - HeadWallWidth, CulvertHeight + HeadWallHeight, msoLineSolid
    DrawLine chart, CulvertLength, CulvertHeight, CulvertLength, CulvertHeight + HeadWallHeight, msoLineSolid
    DrawLine chart, CulvertLength - HeadWallWidth, CulvertHeight, CulvertLength - HeadWallWidth, CulvertHeight + HeadWallHeight, msoLineSolid
```

Figure 9: Snippet of code for generating vehicle positioning images

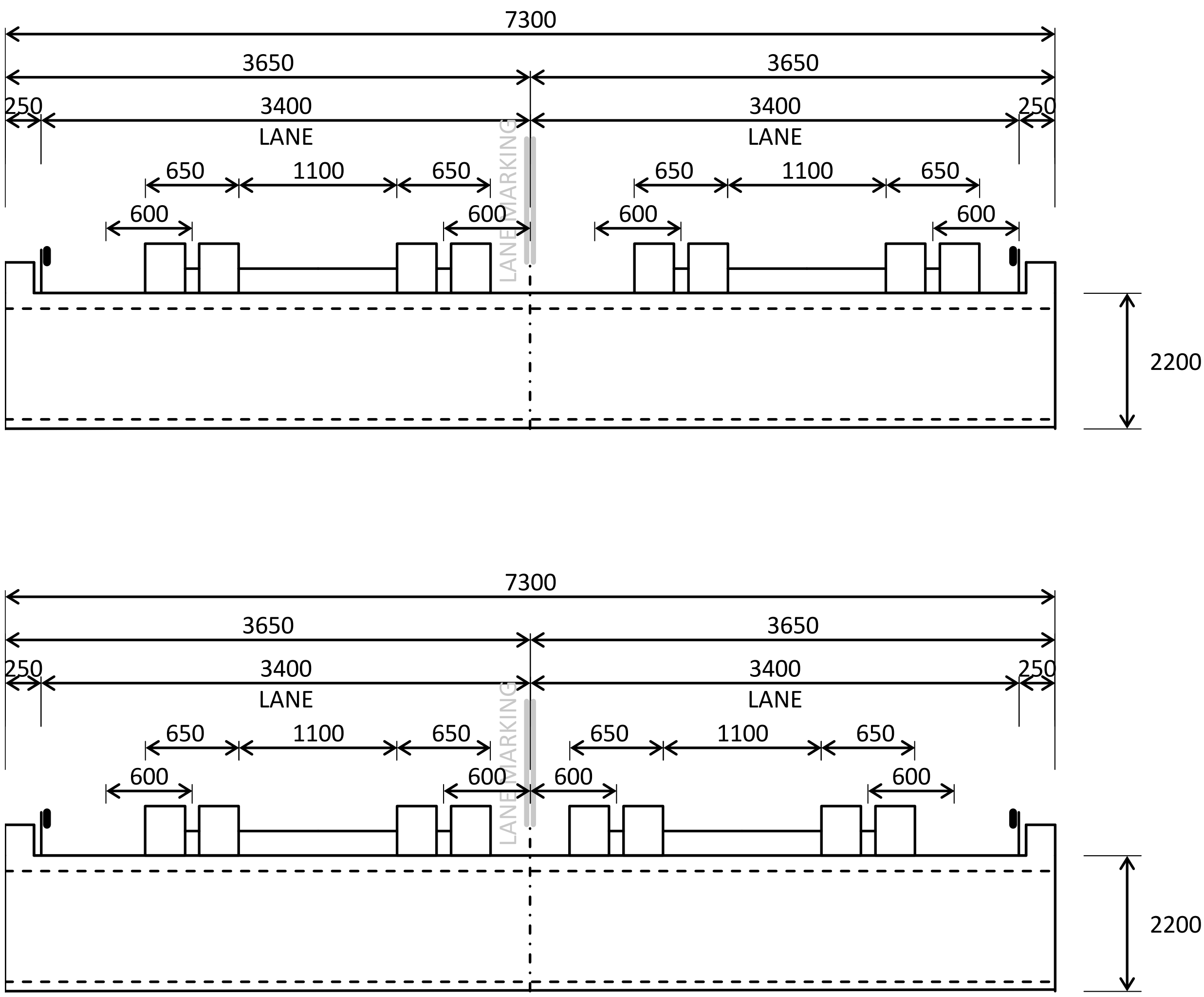

Figure 10: Culvert with vehicle positioning as generated by VBA

3. DATA PROCESSING AND ANALYSIS OF RESULTS

3.1 Data Processing and Analyzing Results

After the model has been set up and analysed, the results are extracted into a .txt or .csv file. For the culvert used as an example in Section 2.1 and 2.2, the .txt file exported normally contains a minimum of 690 000 lines of data. The extracted data from the program encompasses crucial information such as the maximum bending moment, shear, and axial force, along with their respective members as well as the loadcase associated with the maximum action. To obtain this information, a script needs to run through every line in the post-processing file, which could take a considerable amount of time and processing power. A python script was used for this purpose and a library called Pandas was utilised. Pandas is a powerful open-source data manipulation and analysis library for Python.

Python Pandas DataFrame software is widely praised for its versatility and efficiency in handling data manipulation and analysis tasks. Its intuitive syntax and powerful functionality make it a go-to tool for data scientists and analysts. Pandas' DataFrame offers a convenient structure for organizing and analyzing data, allowing for seamless integration with other Python libraries and tools. However, it's important to acknowledge its limitations. For instance, Pandas may struggle with handling extremely large datasets due to memory constraints, requiring users to employ alternative solutions such as distributed computing frameworks. Additionally, while Pandas excels in handling structured data, it may encounter challenges with unstructured or semi-structured data formats. Despite these limitations, Python Pandas DataFrame remains a valuable asset in the data analysis toolkit, offering a robust solution for many common data manipulation tasks. Python Pandas DataFrame software is widely validated and utilized within the data science and research communities. It has undergone rigorous testing and validation processes by both its developers and the broader community of users.

A DataFrame is a two-dimensional, tabular data structure with labeled axes (rows and columns). It is similar to a spreadsheet, and it allows you to store and manipulate structured data easily.

A script was developed, with a snippet shown in Figure 11, to search for the maximum value for each action (bending Moment, shear and axial). The developed script goes through the dataset, picking out specific rows that have the maximum values in certain columns, and organizes this information into two separate output tables. The final output is a summary of these maximum values along with the corresponding load cases and members.

The conclusion of this is a well structured table with maximum actions, member numbers as well as the load case associated with the action. This is very powerful in the post-processing of any structure as the post-processing is known to be a tedious and time-consuming task.

```python
# read through the input dataframe loadcase by loadcase and member for member for each maximum column value
for loadcase in loadcase_list:
    for member in member_list:
        # create a variable which represent the dataframe location of the maximum value for a column of data
        a = df.loc[(loadcase, member, slice(None)), 'Moments'].idxmax()
        # add the row data located at the found location to the output dataframe
        output = pd.concat([output, df.loc[a, :]], axis=1)
        # print the row data located at the found location to the output dataframe
        print(df.loc[a, :])


    for member in member_list:
        # create a variable which represent the dataframe location of the maximum value for a column of data
        b = df.loc[(loadcase, member, slice(None)), 'c3'].idxmax()
        # add the row data located at the found location to the output dataframe
        output1 = pd.concat([output1, df.loc[b, :]], axis=1)
        # print the row data located at the found location to the output dataframe
        print(df.loc[b, :])
```

Figure 11: Snippet of code for post-processing

4. CONCLUSIONS

This paper underscores the pivotal role of culverts in ensuring the smooth flow of water beneath roads and railways, emphasizing the critical need for accurate load rating to uphold structural integrity and safety standards. The conventional load rating process, characterized by labour-intensive efforts and prolonged durations, is brought into focus. Through the lens of a case study, this paper sheds light on the transformative potential of automation in load rating assessments.

The proposed data-driven solutions, harnessing the capabilities of programming languages such as VBA and Python, along with specialized modules like Pandas DataFrame, offer a paradigm shift in the evaluation of culvert structural capacity. The automation of load rating calculations not only expedites the assessment process but also substantially diminishes the time and effort invested in obtaining precise results. Manually completing a load-rating analysis of one culvert could take an engineer anywhere between 1-2 weeks, where-as the proposed solution would take less than one day. The major advantage of the proposed solution is thus the time and cost-saving advantage for both the client and consultant however, numerous other advantages are achieved such as minimisation of errors and structured digital outputs. By embracing this technological evolution and incorporating advanced software tools, engineers can elevate their ability to swiftly and accurately evaluate culvert load ratings, thereby fortifying infrastructure safety and operational efficiency within the transportation sector.

In essence, the integration of automation not only addresses the challenges associated with traditional load rating methodologies but also paves the way for a more efficient and effective approach to culvert assessment. This advancement holds the promise of enhancing overall transportation infrastructure safety, ensuring its resilience to vehicular weights and other loads, and contributing significantly to the optimization of operational processes in the broader context of transportation engineering. As we embrace automation, we stand at the forefront of a transformative era, where precision, speed, and safety converge to shape the future of culvert load rating assessments.

Future research endeavors aim to integrate the developed automation into software platforms, consolidating all processes within a single program. Recommendations include simplifying the process by utilizing the same coding software and employing Application

Programming Interface (API) within SpaceGass or similar structural analysis programs to reduce the need for program switching.

5. ACKNOWLEDGEMENTS

I would like to thank the following people for their contribution to this study and the success thereof:

- My co-author and mentor, Jan Booysen, for his continued support, guidance and active contribution to this study.
- My company, Zutari, and client, Aurecon, for allowing me to think outside the box and produce and explore ways to solve problems and invent solutions for future engineers and the engineers of today.

6. REFERENCES

Brous, P, Janssen, M & Herder, P. 2020. The dual effects of the Internet of Things (IoT): A systematic review of the benefits and risks of IoT adoption by organizations. *International Journal of Information Management*, 51:101952.

Lawson, WD, Wood, TA, Morse, SM, Jayawickrama, PW & Newhouse, CD. 2013. Software development to implement the TxDOT culvert rating guide (No. FHWA/TX-13-5-5849-01-1). Texas. Dept. of Transportation. Research and Technology Implementation Office.

Mirzaei, K, Arashpour, M, Asadi, E, Masoumi, H, Bai, Y & Behnood, A. 2022. 3D point cloud data processing with machine learning for construction and infrastructure applications: A comprehensive review. *Advanced Engineering Informatics*, 51:101501.

Spring, M, Faulconbridge, J & Sarwar, A. 2022. How information technology automates and augments processes: Insights from Artificial-Intelligence-based systems in professional service operations. *Journal of Operations Management*, 68(6-7):592-618.

Standards Australia, 2017. Australian Bridge Design Standard, AS5100.

Rupp, S, Tatar, J, Wagner, D & Shenton, H. 2022. Comparison of results of different approaches to load rating bridges and culverts with missing information. In *Bridge Safety, Maintenance, Management, Life-Cycle, Resilience and Sustainability* (pp. 537-541). CRC Press.