International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 14, Number 2 (2021), pp. 100-104

An application in R software to solve engineering problems

Thomas Edison Guerrero Barbosa[1], Albert Miyer Suarez Castrillon[2] and Sir-Alexci Suarez Castrillon *[3]

[1, 3] Engineering Faculty, University Francisco of Paula Santander Ocaña, Colombia.
[2] Faculty of Engineering and Architecture. University of Pamplona, Colombia.

ORCIDs: 0000-0003-3690-256X (Thomas), 0000-0002-4071-2980 (Albert), 0000-0001-8010-0228 (Sir-Alexci)

Abstract

Structural analysis is a crucial preliminary stage in structural design. If it can be programmed, the modeller can save time and gain reliability. This paper shows an application made in the R software where the Matrix Displacement Method is programmed. We corroborate our results and check that they are the same as having done it without computational tools. This fact should be taken into account as this programming is reliable and saves work time.

Keywords: R software, Matrix Displacement Method, frame, forces, moments.

I. INTRODUCTION

R is an open-access programming language with a focus on statistical analysis. It has the advantage of being able to load several libraries or packages with calculation and graphing functionalities [1]. It is one of the most widely used programming languages in scientific research. It is also useful in machine learning, data mining, biomedical research, bioinformatics, and financial mathematics. R is part of the GNU system and is distributed under the GNU GPL license. It is available for Windows, Macintosh, Unix and GNU / Linux operating systems [2].

In civil engineering, structural analysis is fundamental since it is considered the activity before structural design. Therefore, performing correct structural analyses will allow achieving reliable structure designs [3]. In this article, we show structural analysis based on the Matrix Displacement Method [4], [5] for a simple frame programmed in R software. This structural analysis method is based on fundamental concepts from the elasticity theory [6], energy principles in structural mechanics [7], flexibility method [8] and matrix stiffness method [9]. Sometimes the method is modelled using Finite Elements [10]. One of the advantages of doing it this way is that it uses matrix analysis, which has two advantages: (*i*) analysis has a lower computational cost and (*ii*) easy implementation. It allows adjustment for other load conditions. We define the structural and load conditions showing how it can be programmed. We leave the complete code so that the reader can use it. We do not rely on any library; we use the tools by default when installed.

The rest of the paper is organised as follows. The case of study section details the frame to analyse. The results section describes the process that involves every step of the structural analysis. In the final section, we discuss the main findings and conclusions of this paper.

II. THE STUDY OF THE PROBLEM

The frame to be analyzed is shown in Figure 1:

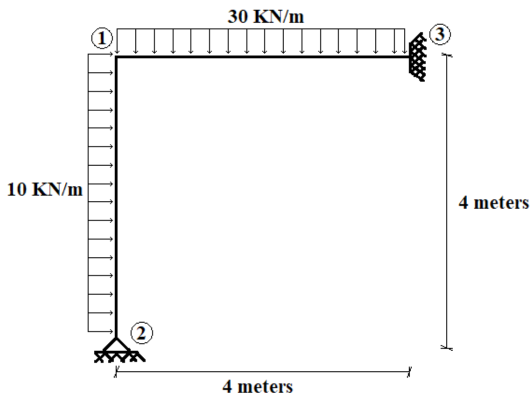

Fig. 1. Dimensions and loads of the frame elements

The structure is composed of a beam (b) and a column (c). The modulus of elasticity [11] (E) is $1.9*10^7$ KN/m^2. The length of each element is defined as "l". The beam and column's width (w) and height (h) are 0.30 mts - 0.35 mts and 0.30 mts - 0.40 mts, respectively.

III. RESULTS

The inertias (I) of each element (b and c) are estimated as follows:

$$I = \frac{(w*h)^3}{12} \qquad (1)$$

The table of coefficients of the stiffness matrix is shown in Table 1. Here, A represents the element area.

Table 1. Table of coefficients of the stiffness matrix

Element	$\frac{AE}{l}$	EI	$\frac{2EI}{l}$	$\frac{4EI}{l}$	$\frac{6EI}{l^2}$	$\frac{12EI}{l^3}$
b	498750	20368	10184	20368	7638	3819
c	570000	30400	15200	30400	11400	5700

The components of the stiffness matrix of each element are identified as “k”. The subscripts b and c identify each element. Besides, the stiffness matrix for the frame.

$$k_b = \begin{bmatrix} 498750 & 0 & 0 & -498750 & 0 & 0 \\ 0 & 3819 & 7638 & 0 & -3819 & 7638 \\ 0 & 7638 & 20368 & 0 & -7638 & 10184 \\ -498750 & 0 & 0 & 498750 & 0 & 0 \\ 0 & -3819 & -7638 & 0 & 3819 & -7638 \\ 0 & 7638 & 10184 & 0 & -7638 & 20368 \end{bmatrix}$$

$$k_c = \begin{bmatrix} 5700 & 0 & -11400 & -5700 & 0 & -11400 \\ 0 & 570000 & 0 & 0 & -570000 & 0 \\ -11400 & 0 & 30400 & 11400 & 0 & 15200 \\ -5700 & 0 & 11400 & 5700 & 0 & 11400 \\ 0 & -570000 & 0 & 0 & 570000 & 0 \\ -11400 & 0 & 15200 & 11400 & 0 & 30400 \end{bmatrix}$$

$$k = \begin{bmatrix} 504450 & 0 & 11400 & 11400 \\ 0 & 573819 & 7638 & 0 \\ 11400 & 7638 & 50768 & 15200 \\ 11400 & 0 & 15200 & 30400 \end{bmatrix}$$

The vector of forces and moments (identified as F^F) is estimated from a straightforward summation analysis of forces and moments equal to zero, as shown in Figure 2. The vector is shown as follows:

$$\begin{matrix} F_x \\ F_y \\ M_b \\ M_c \end{matrix} = \begin{bmatrix} -20 \\ 60 \\ 26.67 \\ 13.33 \end{bmatrix}$$

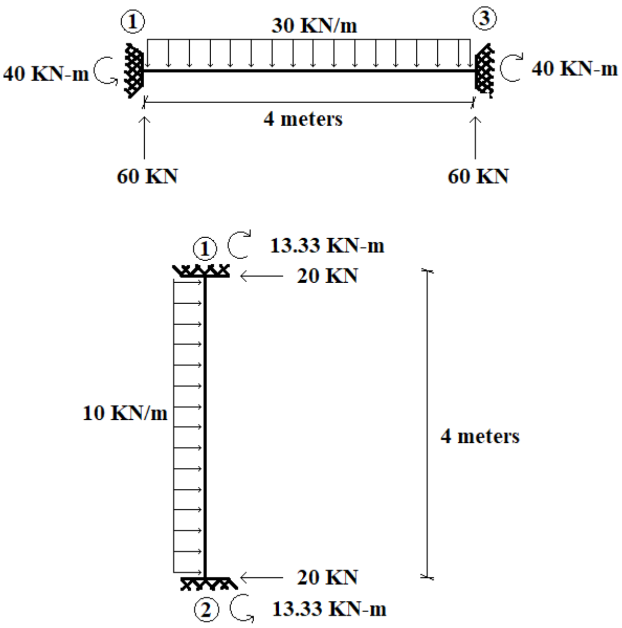

Fig. 2. Internal forces and moments

The deformations (δ) are determined as follows:

$$-F^F = k * \delta \qquad (2)$$

When the δ are estimated, the internal forces (IF) for each element (b and c) are calculated following this expression:

$$IF_b = (k_b * \delta_b) + F_b^F \qquad (3)$$

$$IF_c = (k_c * \delta_c) + F_c^F \qquad (4)$$

The vectors of the IF_b and IF_c are:

$$IF_b = \begin{bmatrix} 27.50 \\ 56.16 \\ 30.02 \\ -27.50 \\ 63.84 \\ -45.37 \end{bmatrix} \quad IF_c = \begin{bmatrix} -12.50 \\ 56.16 \\ 0.00 \\ -27.50 \\ -56.16 \\ -30.02 \end{bmatrix}$$

In Figure 3, the forces and moments calculated from the structural analysis using our code in R software can be noted. Note that the frame is in equilibrium; this is, all the forces and moments add zero.

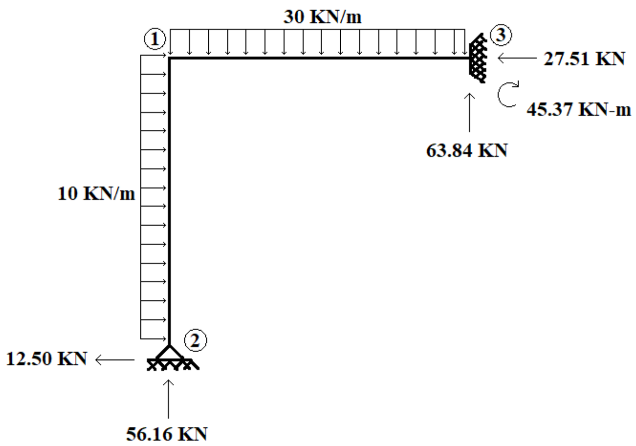

Fig. 3. Forces and moments estimated from the structural analysis

Finally, the code programed in R software is shown as follows:

Analysis of the exercise matrix of a flat portal

Material and geometry characteristics

```
# Modulus of elasticity (KN/m2)
E <- 1.9e7
# Geometry of the elements (m)
l13 <- 4
l21 <- 4
Anchov <- 0.30
Altov <- 0.35
Anchoc <- 0.30
```

```
Altoc <- 0.40
# Área de los elementos (m)
Av <- Anchov*Altov
Ac <- Anchoc*Altoc
# Inercias para los elementos
Iv <- (Anchov*(Altov)^3)/12
Ic <- (Anchoc*(Altoc)^3)/12

# Table of stiffness matrix coefficients #
elemento13 <- c(Av*E/l13,E*Iv,2*E*Iv/l13,4*E*Iv/l13,6*E*Iv/l13^2,12*E*Iv/l13^3)
elemento21 <- c(Ac*E/l21,E*Ic,2*E*Ic/l21,4*E*Ic/l21,6*E*Ic/l21^2,12*E*Ic/l21^3)
tabla_coef <- data.frame (cbind(elemento13,elemento21))
tabla_coef <- data.frame (rbind(elemento13,elemento21))
nombres_tabla_coef <- c("AE/L", "EI", "2EI/L", "4EI/L", "6EI/L^2", "12EI/L^3")
colnames(tabla_coef)<-nombres_tabla_coef
```

Rigidity matrix of each element referred to the axis system

```
# (element13)
k13 <- matrix(0,6,6)
k13[1,1] <- Av*E/l13 # alternatively it can also be u1[1,1] <- tabla_coef[1,1]
k13[1,4] <- -Av*E/l13
k13[2,2] <- 12*E*Iv/l13^3
k13[2,3] <- 6*E*Iv/l13^2
k13[2,5] <- -12*E*Iv/l13^3
k13[2,6] <- 6*E*Iv/l13^2
k13[3,2] <- 6*E*Iv/l13^2
k13[3,3] <- 4*E*Iv/l13
k13[3,5] <- -6*E*Iv/l13^2
k13[3,6] <- 2*E*Iv/l13
k13[4,1] <- -Av*E/l13
k13[4,4] <- Av*E/l13
k13[5,2] <- -12*E*Iv/l13^3
k13[5,3] <- -6*E*Iv/l13^2
k13[5,5] <- 12*E*Iv/l13^3
k13[5,6] <- -6*E*Iv/l13^2
k13[6,2] <- 6*E*Iv/l13^2
k13[6,3] <- 2*E*Iv/l13
k13[6,5] <- -6*E*Iv/l13^2
k13[6,6] <- 4*E*Iv/l13
# (element12)
k21 <- matrix(0,6,6)
k21[1,1] <- 12*E*Ic/l21^3
k21[1,3] <- -6*E*Ic/l21^2
k21[1,4] <- -12*E*Ic/l21^3
k21[1,6] <- -6*E*Ic/l21^2
k21[2,2] <- Ac*E/l21
k21[2,5] <- -Ac*E/l21
k21[3,1] <- -6*E*Ic/l21^2
k21[3,3] <- 4*E*Ic/l21
k21[3,4] <- 6*E*Ic/l21^2
k21[3,6] <- 2*E*Ic/l21
k21[4,1] <- -12*E*Ic/l21^3
k21[4,3] <- 6*E*Ic/l21^2
k21[4,4] <- 12*E*Ic/l21^3
k21[4,6] <- 6*E*Ic/l21^2
k21[5,2] <- -Ac*E/l21
k21[5,5] <- Ac*E/l21
k21[6,1] <- -6*E*Ic/l21^2
k21[6,3] <- 2*E*Ic/l21
k21[6,4] <- 6*E*Ic/l21^2
k21[6,6] <- 4*E*Ic/l21

# Global stiffness matrix #
k_global <- matrix(0,4,4)
k_global[1,1] <- k13[1,1]+k21[1,1]
k_global[2,2] <- k13[2,2]+k21[2,2]
k_global[3,3] <- k13[3,3]+k21[3,3]
k_global[4,4] <- k21[6,6]
k_global[1,2] <- k13[1,2]+k21[1,2]
k_global[1,3] <- k13[1,3]+k21[4,6]
k_global[1,4] <- k21[4,3]
k_global[2,3] <- k13[2,3]+k21[5,6]
k_global[2,4] <- k21[3,5]
k_global[3,4] <- k21[3,6]
k_global[2,1] <- k13[1,2]+k21[1,2]
k_global[3,1] <- k13[1,3]+k21[4,6]
```

```
k_global[4,1] <- k21[4,3]
k_global[3,2] <- k13[2,3]+k21[5,6]
k_global[4,2] <- k21[3,5]
k_global[4,3] <- k21[3,6]

# Reactions and moments #

# Element 13 y 21
FF <- matrix(0,4,1)
FF[1,1] <- -20
FF[2,1] <- 60
FF[3,1] <- 26.67
FF[4,1] <- 13.33
FF1 <- FF*-1
# Externas
Fext <- matrix(0,4,1)
Fext[1,1] <- 0
Fext[2,1] <- 0
Fext[3,1] <- 0
Fext[4,1] <- 0

# Equations for the deformations #
# The inverse of the global matrix is determined
inv_k_global <- solve(k_global)
# The system of equations is solved
deformaciones <- solve(k_global,FF1)
# element deformations 13
delta13 <- matrix(0,6,1)
delta13[1,1] <- deformaciones[1,1]
delta13[2,1] <- deformaciones[2,1]
delta13[3,1] <- deformaciones[3,1]
delta13[4,1] <- 0
delta13[5,1] <- 0
delta13[6,1] <- 0
# element deformations 12
delta21 <- matrix(0,6,1)
delta21[1,1] <- 0
delta21[2,1] <- 0
delta21[3,1] <- deformaciones[4,1]
delta21[4,1] <- deformaciones[1,1]
delta21[5,1] <- deformaciones[2,1]
delta21[6,1] <- deformaciones[3,1]

# Calculation of internal forces #
############################
# External forces element 13
FF13 <- matrix(0,6,1)
FF13[1,1] <- 0
FF13[2,1] <- 60
FF13[3,1] <- 40
FF13[4,1] <- 0
FF13[5,1] <- 60
FF13[6,1] <- -40
# External forces element 13
FI13 <- (k13%*%delta13)+(FF13)
# External forces element 13
FF21 <- matrix(0,6,1)
FF21[1,1] <- -20
FF21[2,1] <- 0
FF21[3,1] <- 13.33
FF21[4,1] <- -20
FF21[5,1] <- 0
FF21[6,1] <- -13.33
# External forces element 13
FI21 <- (k21%*%delta21)+(FF21)
```

IV. CONCLUSIONS

R software is used to program the Matrix Shift Method. This fact involves advantages to doing it without any technological tool since it can be adapted to other frames, with different geometry and different loading conditions. Besides, the versatility and proper use of matrices in R software greatly facilitate the programming of this type of engineering problem. We were able to corroborate that our estimates are correct because they coincide with those made on paper; however, not having programmed it involves a more significant expenditure of time and the possibility of calculation errors. A future research line may be oriented to adapt this same code to more complex forms of frames with two or more levels.

REFERENCES

[1] R. C. Team, *R: A language and environment for statistical computing*. Vienna, Austria, 2013.

[2] H. Aizaki, T. Nakatani, y K. Sato, *Stated preference methods using R*. Boca Raton: CRC Press, Taylor & Francis Group, 2015.

[3] B. S. Taranath, *Structural analysis and design of tall buildings: steel and composite construction*. Boca Raton: CRC/Taylor & Francis, 2012.

[4] P. Singh, *Matrix structural analysis*. Dorrance Publishing Co., 2020.

[5] J. S. Przemieniecki, *Theory of matrix structural analysis*. New York: McGraw-Hill, 1975.

[6] L. D. Landau, E. M. Lifshitz, J. B. Sykes, W. H. Reid, y E. H. Dill, «Theory of Elasticity: Vol. 7 of Course of Theoretical Physics», *Phys. Today*, vol. 13, n.º 7, p. 44, ene. 2009, doi: 10.1063/1.3057037.

[7] T. M. Charlton, *Energy principles in theory of structures*. London; New York: Oxford University Press, 1973.

[8] C. A. Felippa y K. C. Park, «A direct flexibility method», *Comput. Methods Appl. Mech. Eng.*, vol. 149, n.º 1, pp. 319-337, oct. 1997, doi: 10.1016/S0045-7825(97)00048-0.

[9] A. El-Nady, H. M. Negm, y K. A. Hassan, «Analysis of general shells of revolution using Chebyshev series», *J. Eng. Appl. Sci.*, vol. 52, pp. 797-814, ago. 2005.

[10] R. H. MacNeal, *Finite Elements: Their Design and Performance*. New York, 1993.

[11] L. J. Zukor, «Nature and properties of engineering materials. Z. D. Jastrzebski. Wiiey, New York-London, 1959, xvii + 571 pp. $11.00», *J. Appl. Polym. Sci.*, vol. 4, n.º 12, pp. 372-373, 1960, doi: https://doi.org/10.1002/app.1960.070041224.