User's Manual

Matrix Analysis of Reticular Structures
Release 2.0

AMER 2.0

Alejandro Hernández–Martínez

Translated and reviewed by:
León F. Gay–Alanís & Adrián D. García–Soto

UNIVERSIDAD
DE GUANAJUATO
Campus Guanajuato
División de Ingenierías
Departamento de Ingeniería Civil

Table of Contents

Current Updates

Versions 1.2 and 1.3 of this software were not published due to the limited nature of updates in these versions, and because they were made for specific research purposes. Regarding version 2.0 updates, we have the following:

- In this version, it is possible to define a global work unit system that is used across all calculations and represent the default units. In addition, it is possible to define specific input units for each block, which are then automatically transformed to the global system. In addition, the user can define the units for result presentation.
- Equation system solving methods have been added. The user can now choose among LDL^T factorization, Cholesky (LL^T), LU, or LDU; in addition to the iteration methods Conjugate gradient and Gauss-Seidel. The default method is LDL^T.
- Capability to optimize memory size required for stiffness matrix storage. By default, the software stores a complete matrix, but it is possible to use a half-bandwidth storing option.
- Capability to calculate support reactions.
- The structure's degrees of freedom (DOF) are not determined in the *General Data block* (`DATOS GENERALES`) anymore. The software will calculate them from the user-defined member types.
- The software allows to release specific degrees of freedom in members.
- Capability for moving loads analysis.
- Capability for moving loads database reading and analysis.

Overview

The AMER 2.0 software is intended as a companion to the book "Basic Aspects of the Stiffness Method" (Hernández-Martínez 2015). The book´s objective is to support students' learning of the stiffness method through several examples and exercises of structures under static load conditions. The software additionally incorporates single or multiple moving loads analysis, as well as moving loads database reading and analysis.

The software was coded in Fortran95®, which requires the creation of an input data file containing both structural properties and load conditions for any structure to be analyzed. The input file is a text file, and it is read according to specific keywords indicating the software the data type to be read on each section. Therefore, all references to the text file are identified using `Courier New` font. Some keywords may appear too long, but the software requires only the identification of initial characters. For example, for `Materiales` (*materials*) it is only necessary to have the five first characters, **`Mater`**. The required initial characters of each keyword are identified in bold. Additionally, since many keywords are in Spanish, italic will be used to translate these keywords.

The software does NOT discriminate between uppercase and lowercase letters, and therefore the words **`Materi`**`ales`, **`MATER`**`iales`, **`MATER`**`IALES`, and **`MaTeR`**`iAlEs` are all the same. Special characters are ignored, so it is suggested to avoid them. It is possible to include user comments in the text file using the character "!", to improve file readability. All data after the character "!" will be ignored by the software.

Spaces or tabs are used to separate data, and any number of them can be included by the user to achieve adequate text file readability. This includes keywords used for assigning numerical values to variables, and thus all formats such as "**`A = `**`1`", "**`A =`**`1`", "**`A= `**`1`" and "**`A=`**`1`" are correctly read as **`A=`**`1`.

The input file may have some optional parameters. Optional parameters are presented in brackets `[ ]` to distinguish them from mandatory parameters. The data file created with optional parameters must NOT include brackets in the text, brackets are only used to indicate in this manual that parameters are optional.

There are two ways for running the software:

1. Double-clicking on the icon from Windows© explorer, and a window will appear as shown in Figure 1. It is necessary to indicate the input file name (including extension) and the software will create a results text file with the same name but with an extension `.RES`. It is highly recommended to configure your Windows© explorer to not hide the extensions name files, so that you can see them all.

Figure 1 AMER execution window

2. Using the operating system command prompt. It is necessary to enter the directory containing the software and input file in the command line, and then run the program using `C:\Users\…\>"Amer 2.0" [input filename] [results/output filename].` If the results filename is omitted, the program creates it with the same name of the input file but with extension `.RES`. If both filenames are omitted, the screen shown in Figure 1 will be displayed.

1.1 Execution example by mouse double-clicking

In the execution of the example, it is considered that data file of the structural system to be analyzed is previously created as specified in this manual and it has been saved in text format with `.DAT` extension; therefore, only the execution procedure is presented below. Also, it will be assumed that the executable program and the data file are in the directory `C:\AMER 2.0\`, as shown in Figure 2.

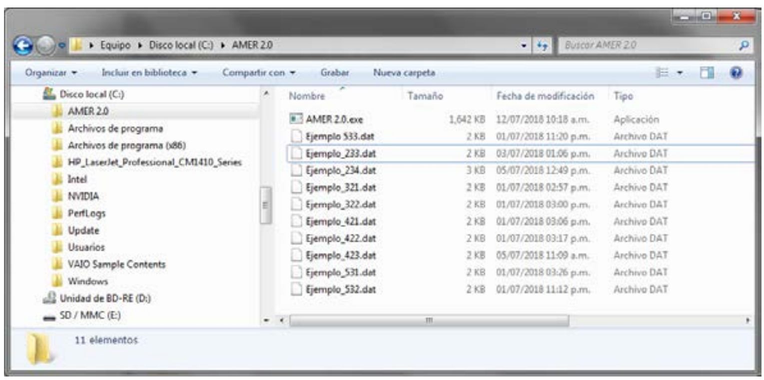


Figure 2 Data files and program directory

By double-clicking on the executable `AMER 2.0.exe` a screen like the one shown in Figure 1 is displayed. The program requests the name of the data file to be executed, so the full name must be entered (including the extension). If the file name contains spaces, the name must be entered in quotation marks (eg: `“Ejemplo 533.dat”`). Once the program has been executed, information about the process is displayed, like showed in Figure 3. To close the window, press the ENTER key (↵).

Figure 3 Information displayed after execution is finished

The results can be consulted in the working directory in file `Example_233.res`, which can be opened with software for text editing (see Figure 4).

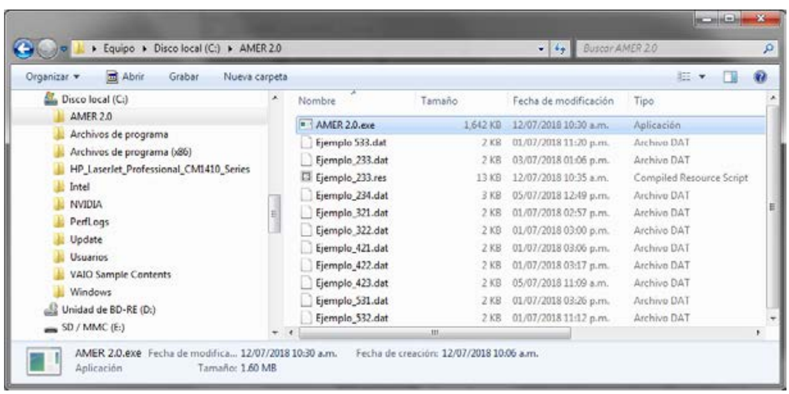


Figure 4 Working directory after program execution

1.2 Execution example using system command prompt

To open de command prompt, click on the start button and select “`All programs`” as shown in Figure 5. Subsequently, in the “`Accessories`” folder, the system command prompt is selected as showed in Figure 6.

Figure 5 Selecting Start/All programs

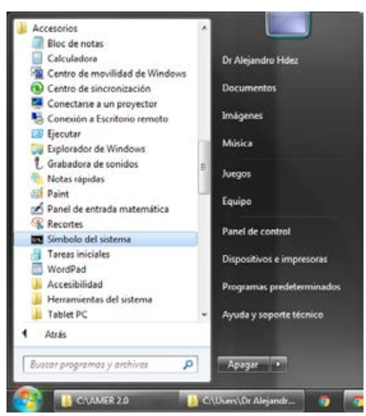


Figure 6 Selecting system command prompt

Clicking on the icon, displays the command line window like shown in Figure 7.

Figure 7 System command prompt

Now, the prompt must be placed in the working directory, which can be done by selecting the path in the Windows© explorer as shown in Figure 8, and copying it using the `Ctrl + C` key combination.

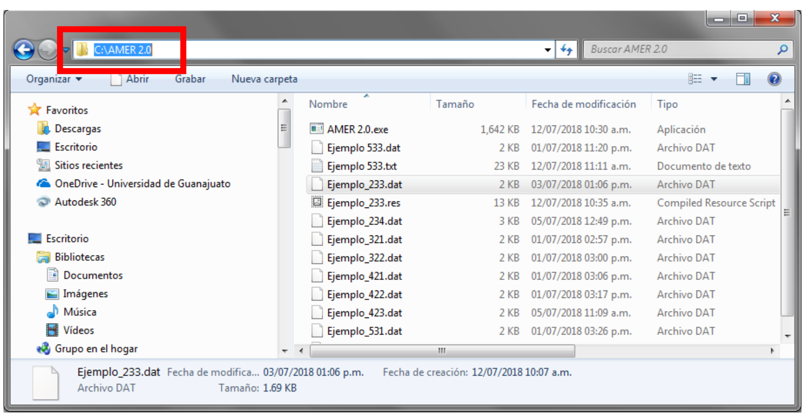


Figure 8 Selecting de path to work folder

Selecting the command line window, write the "`cd`" command adding a space, then click on the upper left corner of the window as shown in Figure 9, and from the displayed menu select `Edit` and then `Paste`, ending with an `ENTER` (↵), and the screen will be as show in Figure 10.

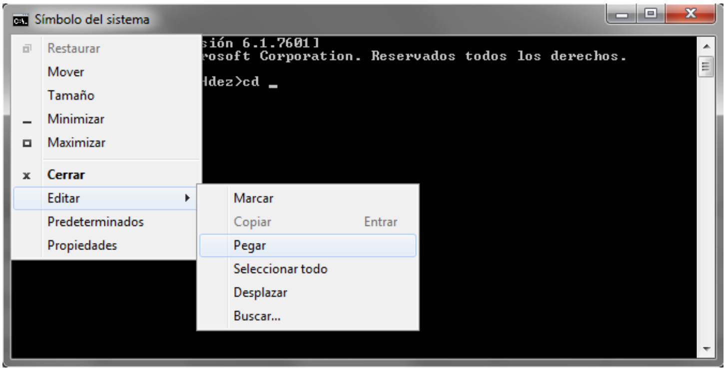


Figure 9 Defining the path in the command line prompt

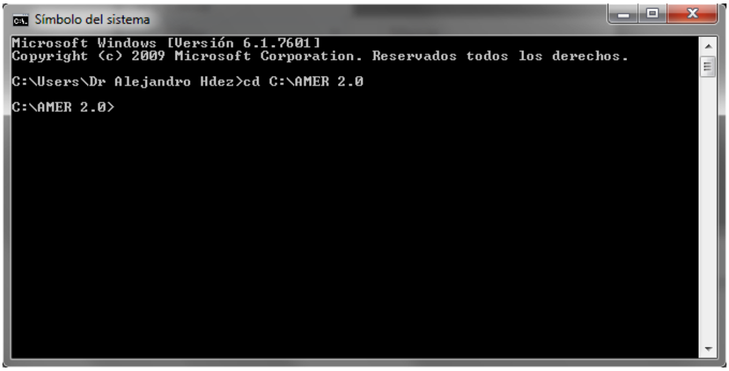


Figure 10 Command line prompt on the work folder

Alternatively, the selections of the working directory can be done through the following commands:

`cd \`	(moves the prompt to root directory)
`cd “AMER 2.0”`	(moves the prompt to desired directory)

This alternative is shown in Figure 11. Finally, as is shown in the same figure, the execution of the program can be done by typing the following command line and an `ENTER` (↵):

```
"AMER 2.0.exe" "Ejemplo 533.dat" "Ejemplo 533.txt"
```

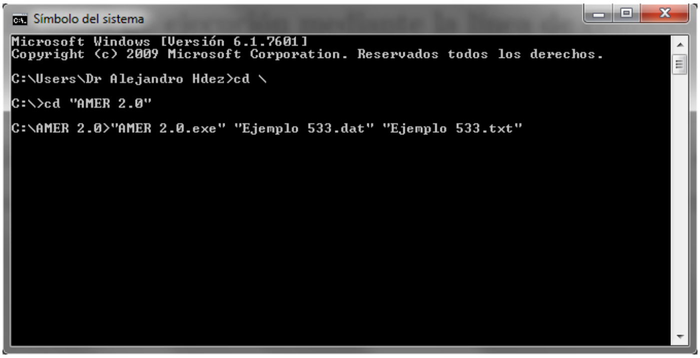


Figure 11 Selecting the working folder by commands

This leads the program to read the data file "`Ejemplo 533.dat`", which is written in quotes in this case, because of the space in its name. The results will be saved in a file named "`Ejemplo 533.txt`", having the name in quotations for the same reason. If the files names do not have spaces in between, the quotes can be omitted. At the end of the execution, a screen like Figure 12 is shown.

Figure 12 Execution ending by command line prompt

Once the execution is finished, you can close the window using the exit command, or simply by clicking on the upper right corner of the window. Finally, in the working directory you can find the result files, like showed in Figure 13.

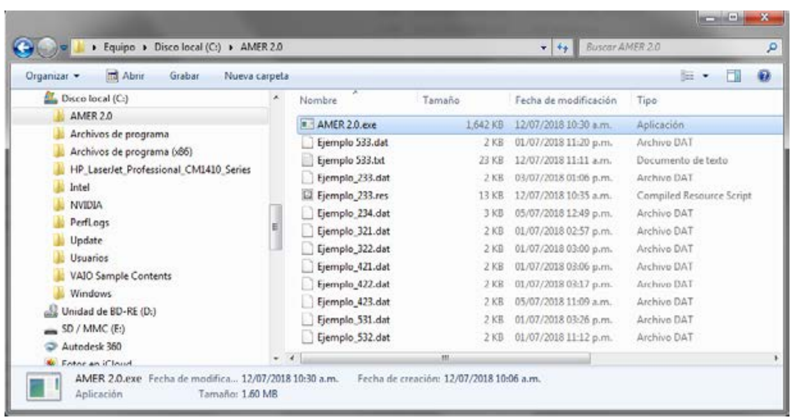


Figure 13 Files in execution directory

1 Basic Structure of the Input File

The AMER 2.0© software identifies data using blocks as shown in Figure 14. Each block is indicated by one or two keywords, and Figure 14 represents the basic structure throughout the input file.

⋮

// **Datos gener**ales de análisis

⋮

// **Mater**iales

⋮

// **Secci**ones

⋮

// **Nodos**

⋮

// **Barra**s

⋮

// **Apoyo**s

⋮

// [**Casos** de **Carga** estaticos]

⋮

// [**Camio**nes]

⋮

// [**Carri**les]

⋮

// [**Base**s de datos]

Figure 14 General data structure in the input file

Each data block start is identified using two diagonals "**//**" and one or more keywords, as seen on Figure 14. The data blocks can be arranged in any convenient order. Before the first data block, it is permitted to include lines with any relevant information such as project name, location, user name, etc., since all of them (before the first data block) will be considered comments by default. The required data blocks and their contents are:

- // `Datos generales de analisis`. (*General analysis data*). This block contains general information: units, required midterm results, solution algorithm, and the stiffness matrix storage mode.
- // `Materiales`. (*Materials*). Containing the mechanical properties of materials used in the project.
- // `Secciones`. (*Sections*). This block contains the geometric properties of the structural member sections.
- // `Nodos.` (*Joints*). Containing joint coordinates and other optional nodal characteristics
- // `Barras.` (*Members*). The information contained in this block is member incidences, member sections, material and member type.
- // `Apoyos.` (*Supports*) For the structure's support locations and conditions.
- // `[Casos de carga]`. (*Static loads*) Static load conditions acting on the structure.
- // `[Camiones]`. (*Trucks*). Containing information on the geometry and load values for truck moving loads.
- // `[Carriles].` (*Lanes*). This block is used to define where are moving loads applied.
- // `[Bases de datos]`. (*Databases*). This block is for the analysis of moving loads using vehicle records (e.g., weigh-in-motion databases).

2 General Analysis Data

2.1 Work Units

One of the most significant changes made in this software version is the analyst ability to choose the work units to be used, defining force and length units. Table 1 shows available units. By default, angular work units are radians and cannot be modified, only angular units can be selected for results report.

Table 1 Type of units available

Force		Length		Angle	
MN	Mega-newton	**m**	Meter	**rad**	Radians
kN	Kilo-newton	**cm**	Centimeter	**gon**	Centesimal degrees (gonio)
N	Newton	**mm**	Millimeter	**gra**	Sexagesimal degrees
kip	Kilo-pound	**in**	Inch		
lb	Pound	**ft**	Foot		
oz	Ounce	**yd**	Yard		
ton	Metric Ton				
kg	Kilogram				

The work unit definition can be done as shown on Figure 15 (where **Fuerza** stands for force and **Longitud** for length).

```
// Datos generales de análisis
   …
   [Unidades de fuerza=kg  Longitud=cm]
   …
```

Figure 15 Definition of work units

It is also possible to not define the work units, in such a case the user will have to be careful to use a consistent set of units throughout the analysis to avoid errors. If work units are not defined in the **Datos gener**`ales` (*General Data*) block, but they are defined in any other block, the specified data units will not be used. Therefore, any time it is important to use a specific set of work units, they will have to be defined in the general data block.

2.2 Result output options

There are several options for the intermediate results output, as shown in Figure 16 and described below.

```
// Datos generales de análisis
   …
   [Reporta datos de la estructura]
   [Reporta matrices de rigidez de barras]
   [Reporta matriz de rigidez de estructura]
   …
```

Figure 16 Options for output file

After the block title in Figure 16, each line is described as follows:

- `Reporta` **`datos`** `de la estructura`. (*Report Structure Data*) Using this command, the input data is written in the output file, allowing the user to verify that the read data analysis is right.
- `Reporta` **`matri`**`ces de rigidez de` **`barra`**`s`. (*Report member stiffness matrices*). Writes the member stiffness matrices in the output file.
- `Reporta` **`matri`**`z de rigidez de la` **`estru`**`ctura`. (*Report structure stiffness matrix*). Writes the structure stiffness matrix in the output file.

2.3 Equation system solution algorithm

There are several algorithms for solving an equation system like the one shown in Eq. (2.1). Those algorithms can be classified in two types: (a) direct solution methods, and (b) iterative methods.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n-1} & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,n-1} & a_{2,n} \\ a_{3,1} & a_{3,3} & a_{33} & \cdots & a_{3,n-1} & a_{3,n} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_{1,n-1} & a_{2,n-1} & a_{3,n-1} & \cdots & a_{n-1,n-1} & a_{n-1,n} \\ a_{1,n} & a_{2,n} & a_{3,n} & \cdots & a_{3,n-1} & a_{nn} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ b_n \end{Bmatrix} \tag{2.1}$$

The user can choose among six algorithms for solving the equation system, as shown in Figure 17. The first four are direct methods, (1) LDL^T factorization, (2) LL^T Cholesky factorization, (3) LU factorization, and (4) LDU factorization. The two first methods are for symmetric matrices, while the last two can work with unsymmetrical systems.

```
// Datos generales de analisis
   …
   [Solucion Factorización LDLT]
   [Solucion Factorización Cholesky]
   [Solucion Factorización LU]
   [Solucion Factorizacion LDU]
   [Solucion Gradiente Conjugado [Tolerancia=*] [itol=*] [itmax=*] ]
   [Solucion Gauss-Seidel [Tolerancia=*] [itol=*] [itmax=*] ]
   …
```

Figure 17 Definition of equation system solution algorithm

The last two solution algorithms are iterative, namely (5) Conjugate gradient method and (6) Gauss-Seidel method. For using iterative methods, the values of some additional parameters are required:

- **`Toler`**`ancia` (*Tolerance*). After each iteration the error is assessed, and the iterative process stops when the error value is under a defined tolerance threshold for convergence. If no tolerance value is defined, a default value of **`toler`**`ancia`$=1\times10^{-6}$ is used.
- **Error assessment criterion**. The parameter **`itol`** defines the error assessment criterion to be used among the eight options shown in Table 2. All eight options are valid for the Conjugate Gradient method, while for Gauss Seidel only options 1 – 4 are valid. The default option is number 3.

Table 2 Error evaluation creteria

`itol`	Error assessment criteria
1	$\text{error} = \left\|\{x\}^{(n)}\right\|_\infty - \left\|\{x\}^{(n-1)}\right\|_\infty$
2	$\text{error} = \left\|\{x\}^{(n)}\right\|_2 - \left\|\{x\}^{(n-1)}\right\|_2$
3	$\text{error} = \left\|[A]\{x\}-\{b\}\right\|_\infty$
4	$\text{error} = \left\|[A]\{x\}-\{b\}\right\|_2$
5	$\text{error} = \left\|\{r\}\right\|_\infty$
6	$\text{error} = \left\|\{r\}\right\|_2$
7	$\text{error} = \left\|\{\Delta x\}\right\|_\infty$
8	$\text{error} = \left\|\{\Delta x\}\right\|_2$

- **Maximum number of iterations. `itmax=*`** defines the maximum number of iterations, after which the analysis stops regardless of the reached accuracy. If no value is defined by the user, default are $2n$ for the Conjugate Gradient method and $20n^2$ for Gauss-Siedel, where n is the number of unknown variables in the system.

If no solution algorithm is defined by the user, the software will use LDL^T factorization by default.

2.3.1 LDL^T factorization algorithm

If the coefficient matrix shown in eq. (2.1) is symmetrical, it can be expressed as shown in eq. (2.2)

$$[A]=[L][D][L]^T \tag{2.2}$$

If the coefficient matrix has order 4, the eq. (2.2) can be developed as follows

$$\begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{21} & a_{22} & a_{32} & a_{42} \\ a_{31} & a_{32} & a_{33} & a_{43} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & l_{32} & 1 & \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} d_{11} & & & \\ & d_{22} & & \\ & & d_{33} & \\ & & & d_{44} \end{bmatrix} \begin{bmatrix} 1 & l_{21} & l_{31} & l_{41} \\ & 1 & l_{32} & l_{42} \\ & & 1 & l_{43} \\ & & & 1 \end{bmatrix} \tag{2.3}$$

The algorithm used for the factorization is

$$\left.\begin{aligned} d_{ii} &= a_{ii} - \sum_{k=1}^{i-1} l_{ik} d_{kk} l_{ik} \\ l_{ji} &= \left.\frac{a_{ji} - \sum_{k=1}^{i-1} l_{jk} d_{kk} l_{ik}}{d_{ii}} \right\} j = i+1,\, i+2, \cdots n \end{aligned}\right\} i = 1, 2, 3 \cdots n$$

Once the equation system is factorized, the solution can be found using:

$$\left. y_i = b_i - \sum_{k=1}^{i-1} l_{ik} y_k \right\} i = 1,\, 2,\, 3 \cdots n$$

$$\left. x_i = \frac{y_i}{b_i} - \sum_{k=i+1}^{n} l_{ki} x_k \right\} i = n,\, n-1,\, n-2, \cdots 1$$

2.3.2 Cholesky factorization algorithm

If the coefficient matrix in eq. (2.1) is symmetrical, it can be expressed as:

$$[A]=[L][L]^T \tag{2.4}$$

Which, if the matrix is order 4 means:

$$\begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} \\ a_{21} & a_{22} & a_{32} & a_{42} \\ a_{31} & a_{32} & a_{33} & a_{43} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ l_{31} & l_{32} & l_{33} & \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} & l_{41} \\ & l_{22} & l_{32} & l_{42} \\ & & l_{33} & l_{43} \\ & & & l_{44} \end{bmatrix} \tag{2.5}$$

And the factorization algorithm is:

$$\left.\begin{aligned} l_{ii} &= \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik} l_{ik}} \\ l_{ji} &= \left.\frac{a_{ji} - \sum_{k=1}^{i-1} l_{ik} l_{jk}}{l_{ii}} \right\} j = i+1, i+2, \cdots n \end{aligned}\right\} i = 1, 2, 3 \cdots n$$

Thus, the solution can be found applying the algorithm:

$$y_i = \left.\frac{b_i - \sum_{k=1}^{i-1} l_{ik} y_k}{l_{ii}} \right\} i = 1, 2, 3 \cdots n$$

$$x_i = \left.\frac{y_i - \sum_{k=i+1}^{n} l_{ki} \; x_k}{l_{ii}} \right\} i = n, n-1, n-2 \cdots 1$$

2.3.3 LU factorization algorithm

The coefficient matrix shown in eq. (2.1) may be expressed as:

$$[A]=[L][U] \tag{2.6}$$

Which for a matrix of order 4 means:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & l_{32} & 1 & \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ & u_{22} & u_{23} & u_{24} \\ & & u_{33} & u_{34} \\ & & & u_{44} \end{bmatrix} \tag{2.7}$$

Therefore, the factorization is made using the algorithm:

$$\left.\left.\begin{matrix} l_{ji} = a_{ji} / a_{ii} \\ u_{jk} = u_{jk} - u_{ik} l_{ji} \Big\} k = i+1, i+2, \cdots n \end{matrix}\right\} j = i+1, i+2, \cdots n \right\} i = 1, 2, 3, \cdots n$$

And the equation system solution is found by:

$$\left. y_i = b_i - \sum_{j=1}^{i-1} l_{ij} y_i \right\} i = 1, 2, 3, \cdots n$$

$$\left. x_i = \frac{y_i - \sum_{j=i+1}^{n} u_{ij} x_j}{u_{ii}} \right\} i = n, n-1, n-2, \cdots 1$$

2.3.4 LDU factorization algorithm

The coefficients matrix shown in eq. (2.1) may be expressed as

$$[A]=[L][D][U] \tag{2.8}$$

Expanding eq. (2.8) for an order 4 system results in

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & l_{32} & 1 & \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} d_{11} & & & \\ & d_{22} & & \\ & & d_{33} & \\ & & & d_{44} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ & 1 & u_{23} & u_{24} \\ & & 1 & u_{34} \\ & & & 1 \end{bmatrix}$$

The necessary algorithm in this case is

$$\left.\begin{aligned} d_{ii} &= a_{ii} - \sum_{k=1}^{i-1} l_{ik}\, d_{kk}\, u_{ki} \\ &\left.\begin{aligned} u_{ij} &= \frac{a_{ij} - \sum_{k=1}^{i-1} l_{ik}\, d_{kk}\, u_{kj}}{d_{kk}} \\ l_{ji} &= \frac{a_{ji} - \sum_{k=1}^{i-1} l_{jk}\, d_{kk}\, u_{ki}}{d_{kk}} \end{aligned}\right\} j = i+1,\, i+2,\, i+3 \cdots n \end{aligned}\right\} i = 1,\, 2,\, 3 \cdots n$$

Once the system has been factorized, the solution can be found with:

$$\left. y_i = \frac{b_i - \sum_{k=1}^{i-1} l_{ik}\, y_k}{d_{ii}} \right\} \quad i = 1, 2, 3 \cdots n$$

$$\left. x_i = y_i - \sum_{k=i+1}^{n} u_{ik}\, x_k \right\} \quad i = n, n-1, n-2 \cdots 1$$

2.3.5 Conjugate Gradient algorithm

The conjugate gradient algorithm used in the AMER 2.0 software is:

1. Calculate $\{r\}_1 = \{b\} - [A]\{x\}_1$
2. $\{p\}_1 = \{r\}_1$
3. For i=1, 2, … $i_{\max}$
   - 3.1. $\alpha = \dfrac{\{r\}_i \times \{r\}_i}{[A]\{p\}_i \times \{p\}_i}$
   - 3.2. $\{x\}_{i+1} = \{x\}_i + \alpha\{p\}_i$
   - 3.3. $\{r\}_{i+1} = \{r\}_i - \alpha[A]\{p\}_i$
   - 3.4. Evaluate convergence. Terminate if met.
   - 3.5. $\beta = \dfrac{\{r\}_{i+1} \times \{r\}_{i+1}}{\{r\}_i \times \{r\}_i}$
   - 3.6. $\{p\}_{i+1} = \{r\}_{i+1} + \beta\{p\}_i$

2.3.6 Gauss–Seidel Algorithm

The Gauss-Seidel algorithm is:

1. Take initial values from solution vector
2. For k=1, 2, 3 … $k_{\max}$
   2.1. Update the solution vector with:

$$\left. x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}}{a_{ii}} \right\} i = 1, 2, 3 \cdots n$$

   2.2 Evaluate convergence. Terminate if met.

2.4 Storing the structure's stiffness matrix

There are in the software two options for the storage of the structure's stiffness matrix, namely:

- `Almacenamiento completo`. (*Full Matrix storage*). This is the traditional way, storing the entire stiffness matrix.
- `Almacenamiento semiancho de banda.` (*Half-bandwidth storage*). Takes advantage of the stiffness matrix dispersion property by skipping null elements, optimizing storage space

Figure 18 shows how to select a desired storage option. It is suggested to define only one storage option; in case two options are defined, the latter will prevail. The default storage option is the full matrix one. As a result, if no storage option is set, the software may use more RAM memory space and take longer than using the second option.

```
// Datos generales de análisis
   …
   [Esquema de almacenamiento: Completo]
   [Esquema de almacenamiento: Semiacho de banda]
   …
```

Figure 18 Storage option definition

2.4.1 Bandwidth storage

A bandwidth storage scheme allows space optimization using the fact that, under the main diagonal in any column, all elements beyond p rows are zero; and also, in any row all elements beyond q columns are zero, as shown in eq. (2.9)

$$[A] = \begin{array}{c} \\ \\ p \\ \\ \\ \\ \end{array} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1,q+1} & & \\ a_{21} & a_{22} & & & \ddots & \\ \vdots & & \ddots & & & a_{n-q,n} \\ a_{p+1,1} & & & \ddots & & \vdots \\ & \ddots & & & a_{n-1,n-1} & a_{n-1,n} \\ & & a_{n,n-p} & \cdots & a_{n,n-1} & a_{nn} \end{bmatrix} \begin{array}{c} \\ \\ \\ q \\ \\ \\ \end{array} \tag{2.9}$$

where:

p = inferior half-bandwidth $a_{ij} = 0$ if $i > j + p$

q = superior half-bandwidth $a_{ij} = 0$ if $j > i + q$

Since not zero elements are located inside a band of known width, the matrix elements are stored in a matrix with n rows and $p + q + 1$ columns as shown in eq. (2.10)

In this way, the elements from the full matrix $A(i,j)$ are stored in the matrix $\hat{A}(\alpha, \beta)$ using:

$$\alpha = i$$
$$\beta = 1 + p + \left(j - i \right)$$

$$\left[\hat{A} \right] = \begin{array}{c} \left[\begin{array}{ccccccc} & & & a_{11} & a_{12} & \cdots & a_{1,q+1} \\ & & a_{21} & a_{22} & a_{23} & \cdots & a_{2,q+2} \\ & \iddots & & \vdots & & & \vdots \\ a_{p+1,1} & & & a_{p+1,p+1} & & & a_{p+1,q+p+1} \\ \vdots & & & \vdots & & & \vdots \\ \vdots & & & \vdots & & & a_{n-q,n} \\ \vdots & & & \vdots & & \iddots & \\ \vdots & & & a_{n-1,n-1} & a_{n-1,n} & & \\ a_{n,n-p} & \cdots & a_{n,n-1} & a_{nn} & & & \end{array} \right] \\ \underbrace{\hphantom{a_{n,n-p} \cdots a_{n,n-1}}}_{p} \qquad \hphantom{a_{nn}} \qquad \underbrace{\hphantom{a_{n-1,n} \cdots a_{2,q+2}}}_{q} \end{array} \tag{2.10}$$

The previously explained storage scheme is used with the algorithms LU and LDU, since in these cases the coefficient matrix is not necessarily symmetrical.

2.4.2 *Half-bandwifth storage*

Using the symmetry property of the stiffness matrix, it is only necessary to store the main diagonal and elements below it, as shown in eq. (2.11)

$$\left[\hat{A}\right] = \begin{bmatrix} & & & a_{11} \\ & & a_{21} & a_{22} \\ & \iddots & & \vdots \\ a_{p+1,1} & & & a_{p+1,p+1} \\ \vdots & & & \vdots \\ \vdots & & & a_{n-1,n-1} \\ \underbrace{a_{n,n-p} \quad \cdots \quad a_{n,n-1}}_{p} & & & a_{nn} \end{bmatrix} \tag{2.11}$$

This storage scheme is used with the factorization algorithms LDL^T, LL^T, as well as iterative methods (Conjugate gradient and Gauss-Seidel).

2.5 Joint renumbering

Regardless of joint numbering assigned by the user, the software internally numbers joints consecutively according to their appearance order in the data file (See section 5). This may not be the most efficient method in terms of the stiffness matrix storage, because it may increase the matrix dispersion. To reduce this problem, the algorithm of (Cutchil & McKee, 1969) may be used for optimizing joint numbering. To activate this option the user must use the line shown in Figure 19.

```
// Datos generales de análisis
   …
   [Renumera nodos]
   …
```

Figure 19 Joint renumbering command

In case the joint renumbering option is used, the only noticeable effect for the user will be the modification of the order in which joints are listed. Results and assigned joint numbers remain unchanged.

If the joint renumbering option is not activated, the data file order is preserved (see section 5). Evidently, this option is useful only if a half-bandwidth storage option is selected, because in a full matrix storage option this is almost irrelevant.

2.6 Sign convention for member results

There are two ways in which acting member forces are presented:

- `Convencion de signos de resultados: Ejes locales.` (*Convention of signs for results: local axes*). The forces positive signs are defined by the local reference coordinate system, as in Figure 21.
- `Convencion de signos de resultados: Fuerzas internas.` (*Convention of signs for results: Internal forces*). The forces positive signs are defined by their effects on members, as in Figure 22.

Figure 20 shows how to select any of the two systems. The local reference system is the default option.

```
// Datos generales de análisis
   …
   [Convención de signos en resultados: Ejes locales]
   [Convención de signos en resultados: Fuerzas internas]
   …
```

Figure 20 Definition of sign conventions for member forces

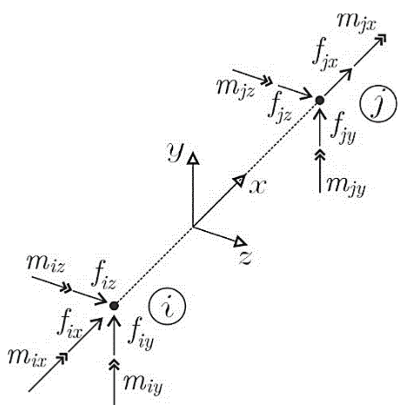


Figure 21 Positive directions of forces in local reference system

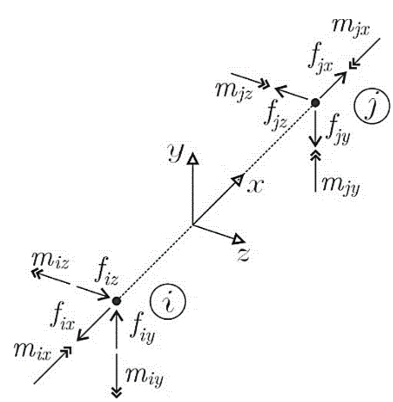


Figure 22 Positive directions of forces according to internal forces

3 Material Properties

Figure 23 shows the materials data block. As many materials as needed can be defined.

```
// Materiales  [Longitud=cm]   [Fuerza=kg]
   Material=Acero    E=2100000  [G=789473.68]  [W=7850.0]
   Material=Concreto E=221359.4362   [v=0.2]
   …
```

Figure 23 Material properties definition

As can be noted in Figure 23, units can be defined at the start of the block, length and force **Longitud=*, Fuerza=*** (substitute the * for any of the correspondent units shown in Table 1). If units are not defined in this data block, it is assumed data have the same units defined in the general data block.

For each material, the following information is required:

- **Material name**. A name for identification data must be assigned, this can be done using the command **MATERIAL=** followed by one word with no spaces, 30 characters max. length
- **Elasticity modulus**. Defined after **E=**
- **Shear Modulus**. Defined after **G=.** Or the value of the Poisson ratio **v=**.
- **Volumetric weight**. Defined after **W=** .

If the shear modulus or Poisson's ratio are included, the contribution of shear forces to the total displacements will be considered. Up to this program version, material's volumetric weight has no effect on the structural calculus.

If Poisson's ratio is defined, the shear modulus will be calculated according to eq. (3.1)

$$G = \frac{E}{2(1+\nu)} \tag{3.1}$$

4 Section Properties

The member section properties are defined in this data block. As shown in Figure 24, the line defining the block may include length units **[Longitud=*]** (substitute the * for any unit in Table 1).

```
// Secciones     [Longitud=cm]
Seccion=10cm2    Tipo=General  [A=12]  [Iy=123.4]  [Iz=12.3]  [J=12.3]  [Fy=1.2]  [Fz=1.2]
Seccion=20x45    Tipo=Rectangular  b=20  h=45          [A=*][Iy=*][Iz=*][J=*][Fy=*][Fz=*]
Seccion=50D      Tipo=Circular     D=50                [A=*][Iy=*][Iz=*][J=*][Fy=*][Fz=*]
Seccion=30R      Tipo=Circular     R=30                [A=*][Iy=*][Iz=*][J=*][Fy=*][Fz=*]
Seccion=W16x36   Tipo=I   d=23 bf=17.8 tw=0.75 tf=1.1  [A=*][Iy=*][Iz=*][J=*][Fy=*][Fz=*]
Seccion=H16x36   Tipo=H   d=23 bf=17.8 tw=0.75 tf=1.1  [A=*][Iy=*][Iz=*][J=*][Fy=*][Fz=*]
```

Figure 24 Section properties definition

For each section, the following definitions are to be used:

- **Section name**. Defined through the command **Seccion=** (*Section*) followed by one word with no spaces, max. 30 characters long.
- **Section type**. Defines the shape of the cross-section area, allowing the software to calculate its geometric properties. For this, use the **Tipo=** (*Type*) command followed by a section type showed in Table 3.
- **Section dimensions**. Section dimensions are defined accordingly with section type. Required dimensions for each section type are detailed in Table 3.
- **Geometric properties**. For the **General** type section, the user must define the values of: Cross-section Area (**A=**), z inertia (**Iz=**), y inertia (**Iy=**), torsion constant (**J=**), y shape factor (**Fy=**), and z shape factor (**Fz=**). If any variable is not defined its value will be considered zero. In any non-general shape type, any of these properties can be defined and the software will not calculate its value and will use the user-defined value.

In the following sections the different section type parameters are explained, as well as their local reference systems.

Table 3 Section types and dimension parameters

Section shape	**Tipo** (*Type*)	Dimension parameters
General	**Gene**ral or **G**	-
Rectangular	**Rect**angular or **R**	**b, h**
Circular	**Circ**ular or **C**	**R** or **D**
I	**I** or **IR** or **W**	**d, bf, tw, tf**
H	**H**	**d, bf, tw, tf**
Rectangular Hollow	**OR** or **PTR**	**b, h, t**

4.1 General type sections

General type sections **GENER**AL (or **G**) allows the user to define the geometric properties:

- Cross-section area, **A=***
- Inertia moment around y axis, **Iy=***
- Inertia moment around z axis, **Iz=***
- Torsional constant, **J=***
- Shape factor for shear on y axis **Fy=***
- Shape factor for shear on z axis **Fz=***

The section properties refer to the member's local axes, and the section is defined in the shaded areas on Figure 25..

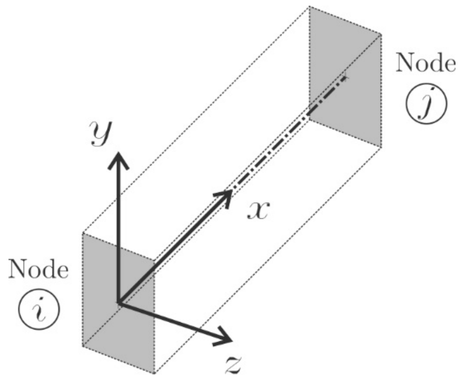


Figure 25 Local axes for a member

In a **General** section all parameters are optional, and any not defined value will be considered zero. This could lead to a singular stiffness matrix, making the analysis unsolvable.

4.2 Rectangular type section

If the section type is **Rectangular** (or **R**), the required parameters are section width **b=*** and section height **h=*** in concordance with local axes z and y as shown in Figure 26. The software will then calculate all geometric properties according to eqs. (4.1) to (4.5). The β constant in eq. (4.4) is obtained from Table 4.

$$A = bh \tag{4.1}$$

$$I_z = \frac{bh^3}{12} \tag{4.2}$$

$$I_y = \frac{b^3 h}{12} \tag{4.3}$$

$$J = \beta b^3 h \tag{4.4}$$

$$f_y = f_z = 1.2 \tag{4.5}$$

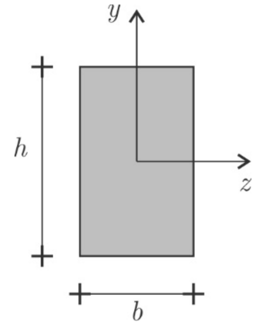


Figure 26 Rectangular section geometry

Table 4 β values for torsion constant in rectangular sections

h/b	β
1.0	0.141
1.5	0.196
2.0	0.229
2.5	0.249
3.0	0.263
4.0	0.281
5.0	0.291
6.0	0.299
10.0	0.312
∞	0.333

Optionally, the user may define the value of every parameter, as in a general type section. Any parameter which has a user-assigned value will not be calculated by the software.

4.3 Circular type section

If the section type is **CIRCULAR** (or **C**), the section diameter must be specified using **D=***, or the radius **R=***, as shown in Figure 27. All properties are then calculated with eqs. (4.6) to (4.9).

$$A = \pi r^2 \tag{4.6}$$

$$I_y = I_z = \frac{\pi r^4}{4} \tag{4.7}$$

$$J = \frac{\pi r^4}{2} \tag{4.8}$$

$$f_y = f_z = \frac{10}{9} \approx 1.111111111 \tag{4.9}$$

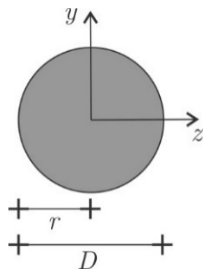


Figure 27 Circular section geometry

As in any other section type, parameter values can be defined by the user instead of calculated.

4.4 I section type

For an **I**, (**IR**, or **W**) section, the required parameters are shown in Figure 28. These are total height (**d=***), flange width (**bf=***), web thickness (**tw=***) and flange thickness (**tf=***). All other section parameters are calculated using eqs. (4.10) to (4.15)

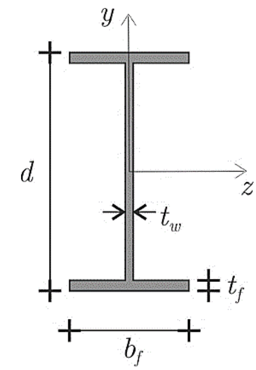


Figure 28 I section type geometry

$$A = d\,t_w + 2\left(b_f - t_w\right)t_f \tag{4.10}$$

$$I_z = \frac{t_w d^3}{12} + 2\left[\frac{\left(b_f - t_w\right)t_f^3}{12} + \left(b_f - t_w\right)t_f\left(\frac{d - t_f}{2}\right)^2\right] \tag{4.11}$$

$$I_y = \frac{d\,t_w^3}{12} + 2\left[\frac{t_f\left(b_f - t_w\right)^3}{12}\right] \tag{4.12}$$

$$J = 2b_f t_f^3 + \left(d - t_f\right)t_w^3 \tag{4.13}$$

$$f_y \approx \frac{A}{d\,t_w} \tag{4.14}$$

$$f_z \approx \frac{1.2A}{2b\,t_f} \tag{4.15}$$

4.5 H section type

An H section type is basically an I section type rotated 90°, as shown in Figure 29. As a result, for the calculation of inertia moments eqs. (4.11) and (4.12) are exchanged, as well as for shape factors in eqs. (4.14) and (4.15).

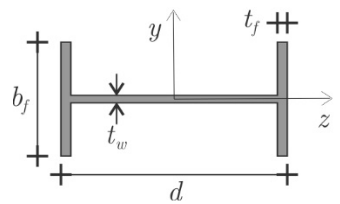


Figure 29 H section type geometry

4.6 Hollow rectangular section

When section type is defined as rectangular hollow **PTR**, the parameters required are total width (**b=***), total height (**h=***), and thickness (**t=***), as shown in Figure 30.

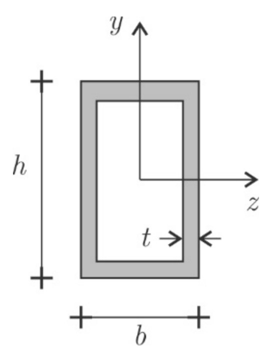


Figure 30 Rectangular hollow circle

Section properties are calculated using eqs. (4.16) to (4.21)

$$A = bh - (b-2t)(h-2t) \tag{4.16}$$

$$I_z = \frac{bh^3}{12} - \frac{(b-2t)(h-2t)^3}{12} \tag{4.17}$$

$$I_y = \frac{b^3 h}{12} - \frac{(b-2t)^3 (h-2t)}{12} \tag{4.18}$$

$$J = 2\frac{\left[(b-t)(h-t)\right]^2 t}{(b-t)+(h-t)} \tag{4.19}$$

$$f_y = \frac{A}{2ht} \tag{4.20}$$

$$f_z = \frac{A}{2bt} \tag{4.21}$$

5 JOINT COORDINATES

The format for entering joint coordinates is shown in Figure 31. In the data block title, the units can be chosen after **Longitud=*** as defined in Table 1. The joint number is **n** (integers numbers only), followed by the coordinates in a cartesian three-dimensional space. The joint numbers do not have to start from 1 or follow a sequential order.

```
// Nodos [Longitud=m]
   …
   n=100  x=0  y=0   z=0
   n=110  x=0  y=10  z=1.5
   …
```

Figure 31 Joint coordinates data format

Once the node identification is assigned, the global coordinates of each node are entered in the X direction (**x=***), Y direction (**y=***) and Z direction (**z=***). For two-dimensional structures, the values of the non-existent axis (for example Z) can be omitted and they are taken as zeros by default.

6 Member Properties and Incidences

The parameters for defining a structural member must be as shown in Figure 32:

- **Member identifier**. As in the case of joints, member numbers are entered through **n=** in integers numbers. No specific order or sequence must be followed.
- **Initial joint**. Indicated by **i=** number of initial joint.
- **Final joint.** Indicated by **j=** number of final joint.
- **Material**. Using **Mate**rial**=** followed by the name of a material as defined in the material data block.
- **Section**. Assigned using **Secc**ion**=** followed by the name of a defined section type.
- **Member type**. This parameter defines the degrees of freedom for analysis. Member types are shown in Table 5. The parameter is entered after **Tipo=** .
- **Releasing degrees of freedom**. Since it is possible that member types do not adequately model the degrees of freedom required by the user, specific DOF can be released using the command **Libe**ra, listing later the DOF to free, as shown in Figure 32.

```
// Barras
n=10  i=20  j=30  Material=Acero  Seccion=20x45  Tipo=Viga
[Libera][dxi][dyi][dzi][gxi][gyi][gzi][dxj][dyj][dzj][gxj][gyj][gzj]
…
```

Figure 32 Member data

Table 5 Member types and associated DOF

Member type	Type definition	Associated DOF
Truss	**Armadu**ra or **A**	dxi, dxj
Beam	**Viga** or **V**	dyi, gzi, dyj, gzj
Grid	**Reticu**la or **R**	dyi, gxi, gzi, dyj, gxj, gzj
2D Frame	**Marco2** or **M2**	dxi, dyi, gzi, dxj, dyj, gzj
3D Frame	**Marco** or **Marco3** or **M3**	dxi, dyi, dzi, gxi, gyi, gzi, dxj, dyj, dzj, gxj, gyj, gzj

When releasing specific DOF, the stiffness matrix is updated as follows:

$$[k'] = [k] - \{k_c\} k_l^{-1} \{k_r\} \tag{4.22}$$

where:

$[k']$ = Member stiffness matrix with released DOF.

$[k]$ = Original stiffness matrix (without released DOF).

$\{k_c\}$ = Column vector from the original stiffness matrix, corresponding to the DOF to be released.

k_l^{-1} = Inverse of the element in the diagonal of the original stiffness matrix corresponding to the DOF to be released.

$\{k_r\}$ = Row vector from the original stiffness matrix corresponding to the DOF to be released.

As a result, the perfect embedding forces of the member are modified with:

$$\{FEP'\} = \{FEP\} - [k]\frac{FEP}{k_l} \tag{4.23}$$

where:

$\{FEP'\}$ = Modified embedded forces vector

$\{FEP\}$ = Original, perfectly double-embedded member force vector.

$[k]$ = Original stiffness matrix for an embedded member

FEP = Force vector value corresponding to the DOF original double-embedded member.

k_l = Element in the diagonal of the original matrix corresponding to the DOF to be freed.

7 Supports

The initial line in the supports data block may contain the units to be used, as shown in Figure 33. To define supports, the required parameters are:

- **Joint identification**. Defined using **`n=`** follow by the identifier of the joint to be a support.
- **Full restriction**. If the support has a full restriction to any DOF, this must be indicated with **`dx`**, **`dy`** or **`dz`** for linear displacements on the global axes *X*, *Y* and *Z* respectively; and for angular displacements use **`gx`**, **`gy`** or **`gz`**.
- **Partial restriction**. If the joint has a partial restriction, use the value of the stiffness (spring constant) using **`KDx=*`**, **`KDy=*`**, **`KDz=*`** for linear displacements in the global axes directions *X*, *Y* and *Z* respectively; and with **`KGx=*`**, **`KGy=*`** or **`KGz=*`** for angular displacements. For linear stiffnesses use units of force/length, and for angular stiffnesses use units of force-length/angle.

Figure 33 illustrates support definitions. Full or partial restrictions not defined are considered unrestricted.

```
// Apoyos  [Fuerza=ton]  [Longitud=cm]
…
n=* [dx] [dy] [dz] [gx] [gy] [gz] [kdx=*] [kdy=*] [kdz=*] [kgx=*] [kgy=*] [kgz=*]
…
```

Figure 33 Support data definition

8 Static Loading Cases

Definition of loading cases allow the solution of static load cases acting on the structure. Also, intermediate results for each loading case can be reported to an output file. Figure 34 shows all options available in the loading cases definition, which are defined below.

```
// Casos de carga estatica
   Caso=Carga_Muerta
     [Reporta vector de fuerzas]
     [Reporta vector de desplazamientos]
     [Reporta desplazamientos nodales]
     [Reporta calculo de fuerzas en barras]
     [Reporta reacciones en apoyos]
     [Reporta Resultados de barras = 1 2 5]
     [Unidades de resultados [Longitud=*] [Fuerza=*] [Angulo=*]]
     [Cargas en los nodos [Longitud=*] [Fuerza=*] ]
        ⋮
     [Cargas en las barras [Longitud=*] [Fuerza=*] ]
        ⋮
   Caso=Carga_Viva
     [Reporta vector de fuerzas]
     [Reporta vector de desplazamientos]
     [Reporta desplazamientos nodales]
     [Reporta calculo de fuerzas en barras]
     [Reporta reacciones en apoyos]
     [Reporta Resultados de barras = 1 2 5]
     [Unidades de resultados [Longitud=*] [Fuerza=*] ]
     [Resultados de barras = 5-8]
     [Cargas en los nodos [Longitud=*] [Fuerza=*] ]
        ⋮
     [Cargas en las barras [Longitud=*] [Fuerza=*] ]

        ⋮
```

Figure 34 Definition of static loading cases

To define a static loading case, the following information is required:

- **Caso=Loading_case_name**. The loading case name must be a single word, with no spaces, 30 characters max. length.
- Reporta **vecto**r de **fuerz**as (*Report forces vector*). With this command the software writes the load case force vector in the output file.
- Reporta **vecto**r de **despl**azamientos (*Report displacements vector*). With this command the software writes the load case displacement vector in the output file.
- Reporta **despl**azamientos nodales (*Report node displacements*). With this command the software includes all de nodal displacements in the output file, ordered in columns by displacement type.
- Reporta calculo de **fuerz**as en **barra**s (*Report member forces calculation*). Including this command, the products of members stiffness matrices by joint displacements, for perfect embedding forces (FEP), and the final forces in the global system are reported in the output file for the load case.
- Reporta **reacc**iones en apoyos (*Report support reactions*). With this command the software includes the support reactions for the load case in the output file.
- **Unida**des de **resul**tados (*Result Units*). Defines units for presenting loading cases results. Units indicated in the general data block are used by default if no new units are specified here.
- **Resul**tados de las **barra**s (*Member Results*). Results of all members are reported by default. If not all member results are required, there are two ways of requesting a list: 1) making a list of all the member names to be presented, as shown in (**Caso=**Carga_muerta in Figure 34). 2) indicating a range of member numbers to be listed (initial–final) as shown in (**Caso=**Carga_Viva, Figure 34). Both ways can be used at the same time.

- **`Carga`**`s en los `**`nodo`**`s` (*Joint Loads*)**.** Defines the start of the joint loading data block. The units may be defined; if omitted, the units defined at the general data block will be used. Section 8.1 details how to enter joint loads.
- **`Carga`**`s en las `**`barra`**`s` (*Member Loads*). Defines the start of the member loads block and may have a separate unit system. Accepted load types are concentrated, uniformly distributed, or triangularly distributed.

8.1 Joint loads

Figure 35 shows how to enter joint loads. The loads can be concentrated forces in the global coordinate system X, Y and Z. First, load node must be defined using **`n=*`**, then the commands **`Fx=*`**, **`Fy=*`** y **`Fz=*`** are used to enter forces, or moments with **`Mx=*`**, **`My=*`** and **`Mz=*`**. As in previous cases, if no force value is defined, it will be considered zero.

```
Caso=Nombre_caso
…
Cargas en nodos  [Fuerza=*] [Longitud=*]
   n=* [Fx=*] [Fy=*] [Fz=*] [Mx=*] [My=*] [Mz=*]
…
```

Figure 35 Joint loads data input

8.2 Member Loads

8.2.1 Concentrated Loads

Figure 36 shows the definition of concentrated loads on members. First, the member identification number (**`n=*`**) must be entered, then the applied load according to local coordinate axes (**`Fy=*`**), and finally the distance from the member's initial joint (**`a=*`**). Figure 37 can be used as reference to define data on members.

```
Caso=Caso_carga
…
Cargas en barras [Longitud=*] [Fuerza=*]
  n=10  Fy=-10  a=1.5
  n=10  Fy=-12  a=3.5
…
```

Figure 36 Input parameter for concentrated member loads on local y axis

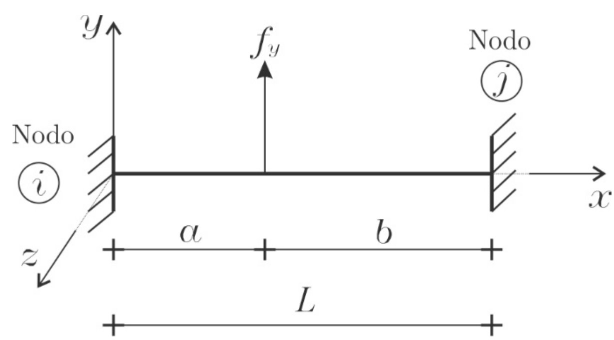


Figure 37 Parameters for concentrated member load on local y axis

The example showed in Figure 36 implies that two loads are applied to the member number 10, the fist load of 10 is applied at 1.5 of the initial node, the second load of 12 is applied at 3.5 of the same initial node. So, it is considered that two loads are applied at the same member. The negative sign of forces indicate that they act in the opposite direction to the local axis y.

To properly consider these forces on the analysis, member type must include y linear displacement (**Dy**) and angular displacement around z (**Gz**), in local axes. In other words, these loads CANNOT be applied in a member defined as **Tipo=Arma**`dura` (*Type=Truss*).

Forces at the extreme nodes are computed according eqs. (4.24) to (4.25)

$$m_{iz} = f_y \frac{a\,b^2}{L^2} \tag{4.24}$$

$$m_{jz} = -f_y \frac{a^2 b}{L^2} \tag{4.25}$$

$$f_{iy} = \frac{f_y\, b + m_{iz} + m_{jz}}{L} \tag{4.26}$$

$$f_{jy} = \frac{f_y a - m_{iz} - m_{jz}}{L} \tag{4.27}$$

In an analogous way, concentrated loads can be applied to member with respect to local axis z, as indicated in Figure 38.

```
Caso=Caso_carga
…
Cargas en barras [Longitud=m] [Fuerza=ton]
  n=20  Fz=-20  a=2
  n=21  Fz=-2   a=3
…
```

Figure 38 Input parameter for concentrated member loads on local z axis

Parameter input data are referred to Figure 39. When concentrated loads are applied on member respect to z local axis, the member must have active DOF for linear displacement on z axis (**Dz**) and angular rotation respect to y axis (**Gy**).

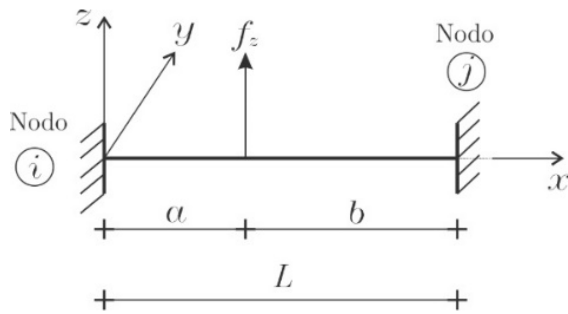


Figure 39 Parameters for concentrated member load on local z axis

8.2.2 Uniformly distributed loads

For the definition of uniformly distributed loads, use the parameters shown in Figure 40. The load definition includes member identification (**n=***) where the force will be applied, value for the distributed load (**Wy=***, referred to y local axis), distance from initial node (**a=***), and distance from final node (**b=***). The positive direction of the applied force is according to local axes, as illustrated in Figure 41. If multiple loads are defined on the same member, loads are added, as well as their effects on the member and the structure. Figure 41 can be used as reference to define input data of uniformly distributed loads acting on y local axis. Input data for uniform distributed loads are based on scheme showed in Figure 41.

```
Caso=Caso_carga
…
Cargas en barras [Longitud=*] [Fuerza=*]
  n=15  Wy=-3  a=0.0  b=1.5
  n=15  Wy=-5  a=1.5  b=0.0
…
```

Figure 40 Definition of uniformly distributed loads on members

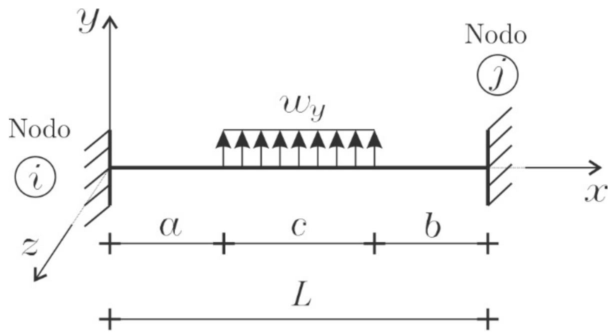


Figure 41 Definition of parameters for uniformly distributed loads on local y axis

Forces at extreme nodes are computed according to eqs.

$$m_{iz} = \frac{w_y}{L^2}\left[\frac{L^2}{2}\left((a+c)^2 - a^2\right) - \frac{2L}{3}\left((a+c)^3 - a^3\right) + \frac{1}{4}\left((a+c)^4 - a^4\right)\right] \tag{4.28}$$

$$m_{jz} = -\frac{w_y}{L^2}\left[\frac{L}{3}\left((a+c)^3 - a^3\right) - \frac{1}{4}\left((a+c)^4 - a^4\right)\right] \tag{4.29}$$

$$f_{iy} = \frac{w_y c\left(b + \frac{c}{2}\right) + m_{iz} + m_{jz}}{L} \tag{4.30}$$

$$f_{jy} = \frac{w_y c\left(a + \frac{c}{2}\right) - m_{iz} - m_{jz}}{L} \tag{4.31}$$

Distributed loads respect to z local axis can be applied to the members, replacing **Wy** by **Wz** and using Figure 42 as reference.

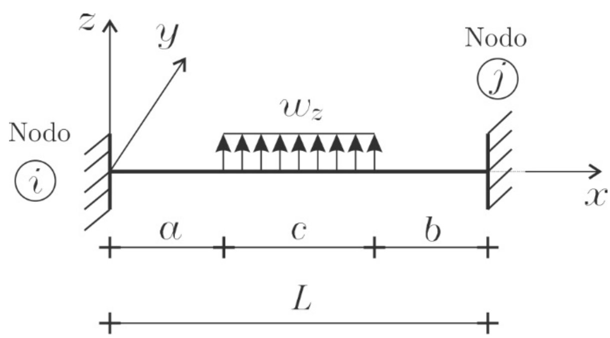


Figure 42 Definition of parameters for uniformly distributed loads on local z axis

To apply distributed loads on z axis, the member must have active de DOF of linear displacement on z local axis (**Dz**) and rotation around y local axis y (**Gy**).

8.2.3 Triangular loads

Figure 43 shows the input data that must be entered to define ascendant triangular loads on members in direction of y local axis. Load definition must include member identification (**n=***), maximum value load (**Tay=***), distance from initial node to the beginning of the load (**a=***), and distance from the end load to final node (**b=***).

```
Caso=Caso_carga
…
Cargas en barras [Longitud=*] [Fuerza=*]
   n=20  Tay=-10  a=0.0  b=2.5
   n=20  Tdy=-10  a=2.5  b=0.0
…
```

Figure 43 Input dada for triangular loads

Input data for triangular ascending loads are based on scheme showed in Figure 44.

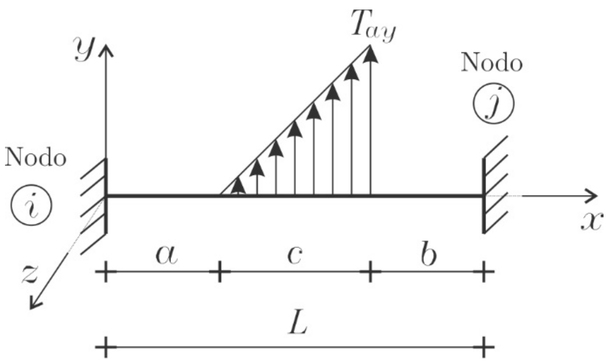


Figure 44 Parameters for ascending triangular load on y local direction

To define descending triangular load data on members acting in direction of y local axis, only **Tdy** data type must be specified, as showed in the second line load data in Figure 43. For triangular descending load, data are based on scheme showed in Figure 45.

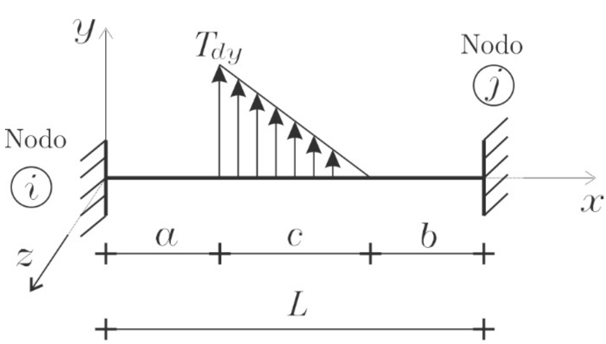


Figure 45 Parameters for descending triangular loads on y direction local axis

The forces at end nodes for triangular ascending loads are computed according to eqs. (4.32) to (4.35).

$$m_{iz} = \frac{T_{ay}c}{60L^2}\left[10a^2\left(3b+2c\right)+c^2\left(10a+5b+2c\right)+20abc\right] \tag{4.32}$$

$$m_{jz} = -\frac{T_{ay}\ c}{60L^2}\left[10b^2\left(3a+c\right)+c^2\left(15a+10b+3c\right)+40abc\right] \tag{4.33}$$

$$f_{iy} = \frac{\left(\frac{T_{ay}c}{2}\right)\left(a+\frac{c}{3}\right)+m_{iz}+m_{jz}}{L} \tag{4.34}$$

$$f_{jy} = \frac{\left(\frac{T_{ay}c}{2}\right)\left(\frac{b+2c}{3}\right)-m_{iz}-m_{jz}}{L} \tag{4.35}$$

Similarly, triangular loads can be applied with respect to z local axis direction on members, changing **`Tay`** or **`Tdy`** by **`Taz`** or **`Tdz`** respectively, using as reference Figure 46 for ascending triangular load, and Figure 47 for descending triangular load.

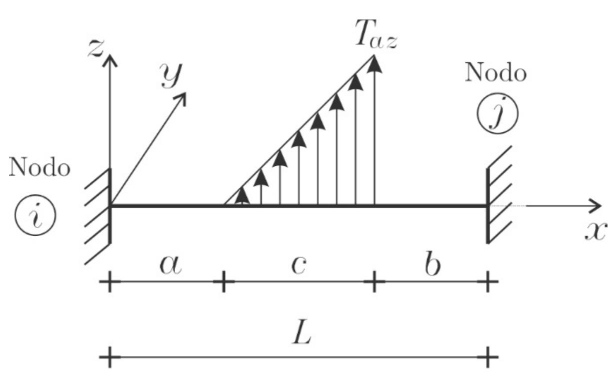


Figure 46 Parameters for ascending triangular load on z local axis direction

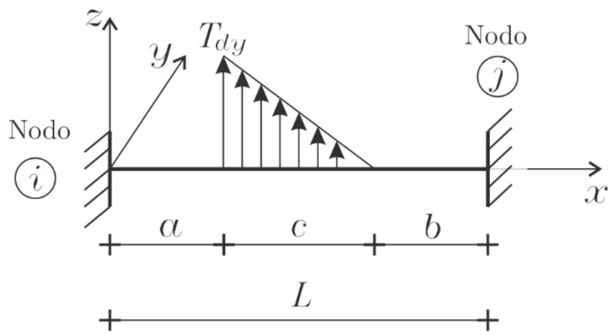


Figure 47 Parameters for descending triangular load on z local axis direction

9 Trucks for Moving Load Analysis

```
// Camiones
   Camion=Nombre_camion  [Longitud=*] [Fuerza=*]
      X=*   Fy=*
      X=*   Fy=*
       ⋮
   Camion=T3-S3  Longitud=m  Fuerza=ton
      X= 0.00   Fy=-6.50
      X= 3.50   Fy=-9.75
      X= 4.70   Fy=-9.75
      X= 8.95   Fy=-7.50
      X=10.15   Fy=-7.50
      X=11.35   Fy=-7.50

   Camion=T3-S2-R4  Longitud=m  Fuerza=ton
      X= 0.00   Fy=-5.3
      X= 3.50   Fy=-8.4
      X= 4.70   Fy=-8.4
      X= 8.95   Fy=-8.4
      X=13.35   Fy=-8.4
      X=14.55   Fy=-8.4
      X=18.80   Fy=-8.4
      X=20.00   Fy=-8.4
…
```

Figure 48 Truck data input

To use live moving loads (e.g. truck loads), the user must define the layout for such loads. Each truck will be defined as follows:

- **Truck name**. After the command **Camion=** assign a truck name, which is a single word, no spaces, 30 characters max length. It is optional in the same line to include the units to be used.

- **Truck loading layout**. Enter number pairs indicating distance (from truck first axle) (**`x=*`**), and load value (**`Fy=*`**). The loads are assumed to be applied in the *y* member local axis (see Section 8.2.1).

For example, Figure 49 shows the data input for truck named T3-S3 and Figure 50 shows the layout for truck named T3–S2-R4. Both trucks data input is defined in Figure 48.

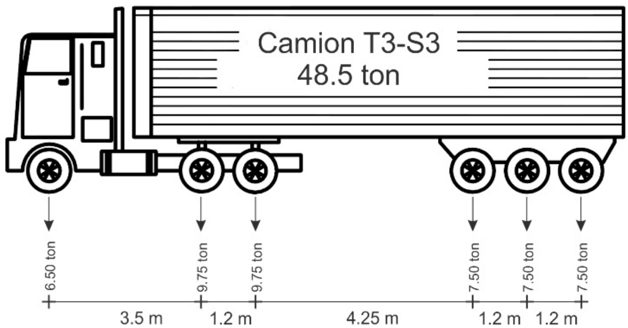


Figure 49 T3-S3 truck load layout

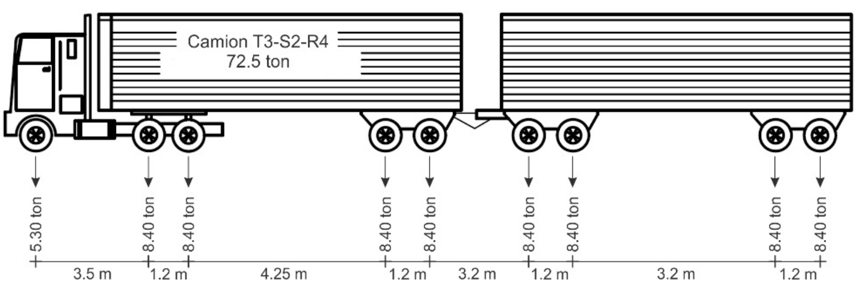


Figure 50 T3-S2-R4 truck load layout

Once the trucks layouts are defined, where and how those loads are applied must be defined. This is explained in next section.

10 Lanes for Moving Load Analysis

```
// Carriles
  Carril=Nombre_carril_1
     Barras que definen el carril = 1 2 3 4-10
     [Unidades de resultados [Fuerza=*] [Longitud=*] ]
     [Barras para resultados = 8 10 14-20]}

     Camion=Nombre_camion   [Longitud=*]
        [Direccion=Avanza  Incremento=*]
        [Direccion=Regresa Incremento=*]

     Camion=T3-S3   Longitud=cm
        Direccion=Avanza   Incremento=1
        Direccion=Regresa  Incremento=5

     Camion=T3-S2-R4   Longitud=cm
        Direccion=Avanza   Incremento=2

  Carril=Nombre_Carril_2
     …
```

Figure 51 Definition of moving live loads lanes

The lane definition for moving loads indicates the software which structural elements will be loaded and how truck loads must be applied to the structure. The loading path is defined through structural members, as shown in Figure 51, and explained below.

- **`Carril=Nombre_carril_1`** (*Lane=Lane_name_1*). This assign an identification name to the lane; one word with no spaces and 30 characters max length.
- **`Barras que definen el carril=*`** (*Lane members*). A list of member number on which the truck loads will be applied. The list number sequence represents the truck advance. It is recommended that the truck advancement coincides with the local members coordinate system. The numbers in the list can be separated by spaces, (i.e.: `1 2 3`, Figure 51) or indicating ranges with a hyphen

(i.e.: **4-10**, Figure 51). In the example shown in Figure 51 the lane is defined by members 1 to 10.

- **Unida**des de **resul**tados (*Result units*). Used to define the units in which results of moving loads analysis will be displayed. Since it is optional, if units are not defined the software will use those units defined in the general data block (see section 2.1).
- **Barra**s de **resul**tados (*Result members*). Indicates for which members results will be reported. A member list must be defined, with the same considerations as for defining a lane. For example, in Figure 51, members selected for displaying results are 8, 10, and 14 to 20. If not indicated, results for all members will be shown.
- **Camio**n=***** (*Truck*). Defines the truck to be applied as loading on the lane. The truck should be defined in the truck data block (section 9). Length units may be indicated as well for the truck step (increment for running the truck over the members). Afterwards, the user must indicate (a) direction for truck circulation **Direc**ción=**Avanz**a/**Regre**sa (*Direction=Forward/Backward*), and (b) the step at which the truck will be moving **Incre**mento=***** (*Step=**). For example, in Figure 51, it is defined that truck T3-S3 must move forward at a step of 1 cm (as shown in Figure 52), and backward 5 cm per step (as shown in Figure 53). The T3-S2-R4 truck moves only in 2 cm steps forward.

The resulting force envelopes are reported for each defined loading lane. In case force envelopes are required for each individual truck, separated lanes must be defined and apply individual trucks on each one.

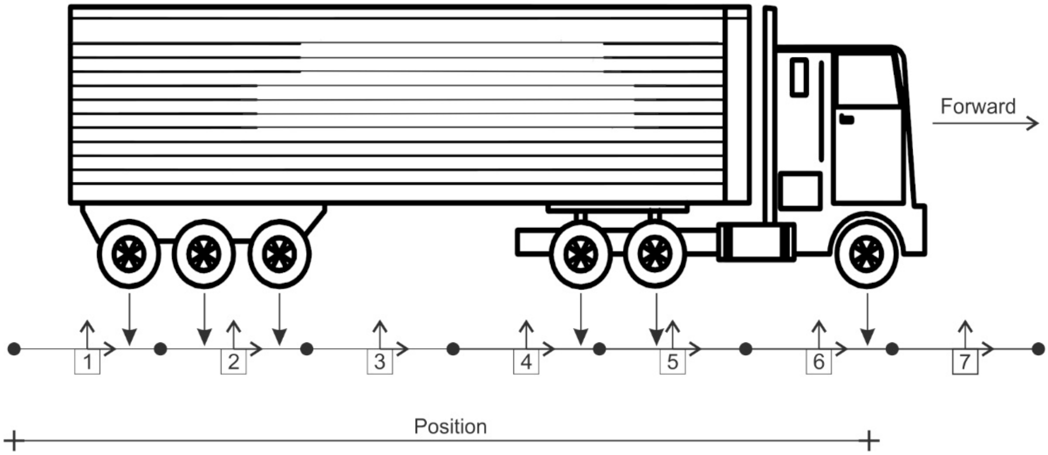


Figure 52 Truck moving forward

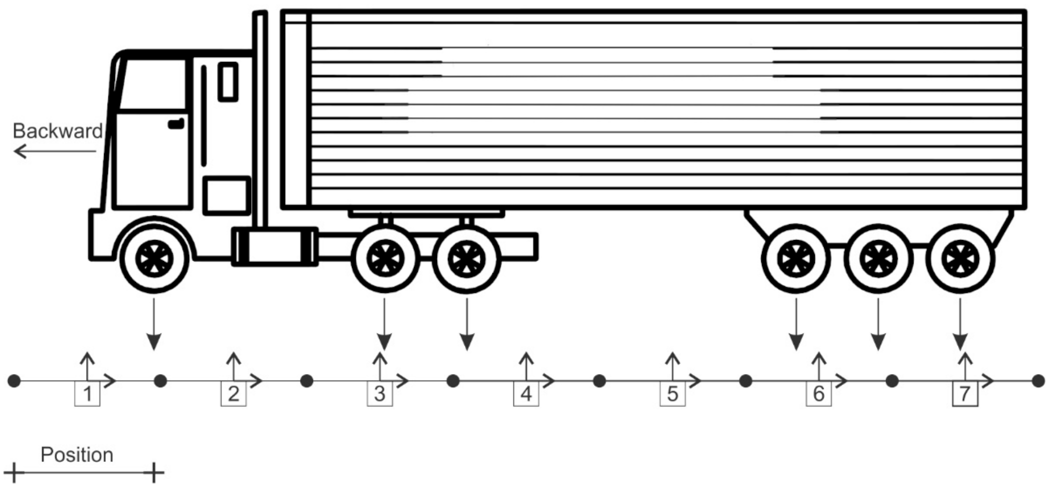


Figure 53 Truck moving backward

11 Moving Loads Analysis Using a Database

```
// Bases de datos
   Barras que definen el carril = 201-217
   [Unidades de resultados [Fuerza=*] [Longitud=*] ]
   [Barras de resultados = 302-309 400-409
   [Calcular envolventes de fuerzas]
   [Reporta reacciones en apoyos]

   Base=C:\...\BD1.csv   [Fuerza=*]  [Longitud=*]
      Direccion=Avanza   Incremento=*
      Dirección=Regresa  Incremento=*

   Base=C:\...\BD2.csv   [Fuerza=*]  [Longitud=*]
      Direccion=Regresa   Incremento=*
   …
```

Figure 54 Data input for moving load analysis using WIM database

To perform live loads analysis when truck data is stored in a database, the following parameters must be defined:

- `Barras que definen el carril` (*Lane members*). Member list on which the loads will be applied (same list structure as in section 10). For the example shown in Figure 54 loads are applied on members 201 to 217.
- `Unidades de resultados` (*Result units*). An optional line indicating the units to be used for results output. If omitted, result units will be those indicated at the general data block (section 2.1)
- `Barras de resultados` (*Result members*). Defines for which members results will be reported. Use the same structure as indicated in section 10. By default, all member results are shown.
- `Calcular envolventes de fuerzas` (*Calculate force envelopes*). To calculate envelopes for all forces, including those in the database and for each individual truck in database. By default, only the maximums for resulting forces

are obtained. Also, extreme forces (maximum and minimum) for each member and each vehicle is reported.

- `Reporta reacciones en apoyos` (*Report support reactions*). Report the extreme values (maximum and minimum) of forces at structure supports.
- **Database file**. Database name, using **`Base=file_name`**. A complete file access path is required if the database is not in the same directory as the software. In the same line indicate units **`Fuerza=*`** (*Force=**), **`Longitud=*`** (*Length=**); and afterwards one or two lines containing moving direction **`Direccion=Avanza/Regresa`** (*Direction=forward/backward*), and step moving increment **`Incremento=*`** (*Step=**).

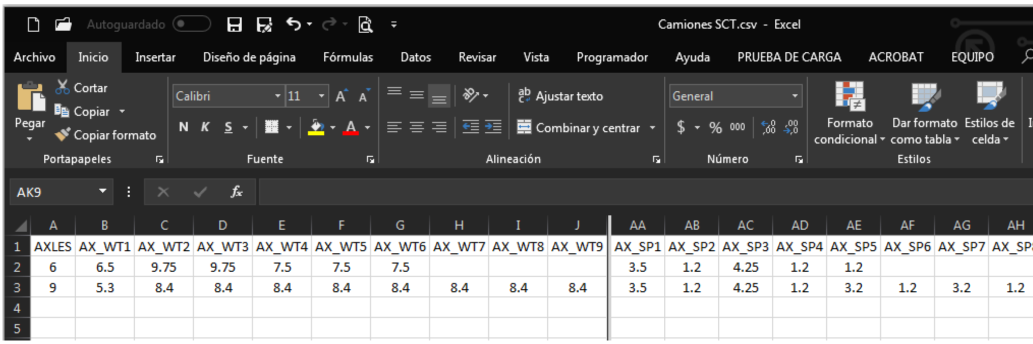
11.1 Database format

The truck database must be in a .CSV (comma-separated values) file that can be opened using Microsoft Excel© software, as shown in Figure 55, containing data on a maximum of 50 columns:

- First row is for headings. From row 2 on, there are data on weights and axe distances.
- **Column 1 (A)**. Number of vehicle axles
- **Columns 2–26 (B-Z)**. Weight per axle, ordered from axle 1 to 25. Nonexistent axes will have EMPTY cells. A zero value is assumed to be mistaken and the vehicle will not be considered,
- **Columns 27–50 (AA-AX)**. Distances between axles, 1–2, 2–3, 3–4, … 23–34, 24–25. Nonexistent axe distances must be in empty cells.

All the units including weights and axle distances are indicated to the software as shown in Figure 54. Data showed in Figure 55 corresponds to T3-S3 truck (Figure 49) and T3-S2-R4 truck (Figure 50).

The results will be presented in several files with `.RES` extension. One file for each database for member results (`DBfilename – barras.res`) and another one for each database if support reactions are required (`DBfilename – reacciones.res`) where the respective results are detailed for each member and vehicle defined in database. Additionally, in general output file, extreme force values will be reported for all databases, for members and reactions, if that is required.



	A	B	C	D	E	F	G	H	I	J	AA	AB	AC	AD	AE	AF	AG	AH
1	AXLES	AX_WT1	AX_WT2	AX_WT3	AX_WT4	AX_WT5	AX_WT6	AX_WT7	AX_WT8	AX_WT9	AX_SP1	AX_SP2	AX_SP3	AX_SP4	AX_SP5	AX_SP6	AX_SP7	AX_SP8
2	6	6.5	9.75	9.75	7.5	7.5	7.5				3.5	1.2	4.25	1.2	1.2			
3	9	5.3	8.4	8.4	8.4	8.4	8.4	8.4	8.4	8.4	3.5	1.2	4.25	1.2	3.2	1.2	3.2	1.2
4																		
5																		

Figure 55 Database file viewed through Microsoft Excel©

Appendix A Example of Data Input File for 2D Truss Analysis

This appendix shows the data input file for a plane truss corresponding to example 2.3.4 showed in the e-book "Aspectos Básicos del Método de Rigideces" (*Basic Aspects of Stiffness Method*) (Hernández-Martínez, 2015) It is noted that the referred book is currently only available in Spanish; nevertheless, the English reader could use the book (freely available at http://www.di.ugto.mx/GEMEC/) and infer the examples from the figures and numbers

```
Example 2.3.4 from book: Aspectos básicos del método de rigideces
                         Plane truss analysis

Units:         Force  = ton (metric ton)
               Length = cm

// Materiales  Fuerza=kg      Longitud=cm                  !Materials
       Material=Acero E=2100000

// Secciones   Longitud=cm                                 !Sections
       Seccion=5_cm2  Tipo=General   A=5
       Seccion=10_cm2 Tipo=General   A=10
       Seccion=20_cm2 Tipo=General   A=20

// Datos generales de análisis                             !General data
       Unidades de fuerza = ton      ! Work units
       Unidades de longitud = cm
       Escribe datos de la estructura
       Escribe matrices de barras
       Escribe matriz de estructura
       Solucion por Factorización LDLT
       Esquema de almacenamiento en Semi-ancho de banda

// Nodos       Longitud=m                                  ! Joints
       n=1     x=0     y=0
       n=2     x=3     y=0
       n=3     x=3     y=2
       n=4     x=6     y=0
       n=5     x=6     y=4
       n=6     x=9     y=0
       n=7     x=9     y=2
       n=8     x=12    y=0
```

```
// Barras                                                        ! Members
       n=1  i=1  j=2  Material=Acero  Seccion= 5_cm2  Tipo=Armadura
       n=2  i=1  j=3  Material=Acero  Seccion=20_cm2  Tipo=Armadura
       n=3  i=2  j=3  Material=Acero  Seccion=10_cm2  Tipo=Armadura
       n=4  i=2  j=4  Material=Acero  Seccion= 5_cm2  Tipo=Armadura
       n=5  i=3  j=4  Material=Acero  Seccion=10_cm2  Tipo=Armadura
       n=6  i=3  j=5  Material=Acero  Seccion=20_cm2  Tipo=Armadura
       n=7  i=4  j=5  Material=Acero  Seccion=10_cm2  Tipo=Armadura
       n=8  i=4  j=6  Material=Acero  Seccion=5_cm2   Tipo=Armadura
       n=9  i=4  j=7  Material=Acero  Seccion=10_cm2  Tipo=Armadura
       n=10 i=5  j=7  Material=Acero  Seccion=20_cm2  Tipo=Armadura
       n=11 i=6  j=7  Material=Acero  Seccion=10_cm2  Tipo=Armadura
       n=12 i=6  j=8  Material=Acero  Seccion=5_cm2   Tipo=Armadura
       n=13 i=7  j=8  Material=Acero  Seccion=20_cm2  Tipo=Armadura

// Apoyos                                                        ! Supports
       n=1    Dx      Dy
       n=8            Dy

// Casos de carga                                                ! Static load case
       Caso = Unico
              Reporta vector de fuerzas
              Reporta vector de desplazamientos
              Reporta desplazamientos
              Reporta calculo de fuerzas en barras
              Reporta reacciones

              Cargas en nodos
                     n=1  Fy=-2  ¡Load for DOF restricted (support)
                     n=3  Fy=-4
                     n=5  Fy=-4
                     n=7  Fy=-4
                     n=8  Fy=-2  !Load for DOF restricted (support)
```

Appendix B Example of Data Input File for Beam Analysis

This appendix shows the data input file for beam analysis shown in Example 4.2.3 of the e-book "Aspectos Básicos del Método de Rigideces" (*Basic Aspect of Stiffness Method*) (Hernández-Martínez, 2015). It is noted that the referred book is currently only available in Spanish; nevertheless, the English reader could use the book (freely available at http://www.di.ugto.mx/GEMEC/) and infer the examples from the figures and numbers.

```
Example 4.2.3 from book: Aspectos básicos del método de rigideces


// Datos generales de análisis                             ! General data
       Unidades de longitud=cm  fuerza=ton
       Escribe datos de estructura
       Escribe matrices de barras
       Escribe matriz de estructura
       Solucion Factorizacion Cholesky
       Esquema de almacenamiento en semi-ancho de banda

// Materiales                                              ! Materials
       Material=Conc_fc250   E=238.751963

// Secciones   Longitud=cm                                 ! Sections
       Seccion=25x60  Tipo=Rectangular      b=25    h=60

// Nodos       Longitud=m                                  ! Joints
       n=1     x=0
       n=2     x=3
       n=3     x=7
       n=4     x=9

// Barras                                                  ! Members
       n=1  i=1  j=2  Material=Conc_fc250  Seccion=25x60  Tipo=viga
       n=2  i=2  j=3  Material=Conc_fc250  Seccion=25x60  Tipo=viga
       n=3  i=3  j=4  Material=Conc_fc250  Seccion=25x60  Tipo=viga

// Apoyos                                                  ! Supports
       n=1 Dy Gz
       n=2 Dy
       n=3 Dy
```

```
// Casos de carga                                                    ! Static load case
       Caso = Unico
              Reporta vector de fuerzas
              Reporta vector de desplazamientos
              Reporta desplazamientos
              Reporta calculo de fuerzas en barras
              Reporta reacciones en apoyos

              Cargas en los nodos   Fuerza=ton     Longitud=m
                     n=2    Mz=-7
                     n=4    Fy=-10

              Cargas en barras      Fuerza=ton     Longitud=m
                     n=1    Wy = -2
                     n=2    Wy = -3
```

Appendix C Example of Data Input File for 2D Frame Analysis

This appendix shows the data input file corresponding to Example 5.3.3 of the e-book: "Aspectos Básicos del Método de Rigideces" (*Basic Aspects of Stiffness Method*) (Hernández-Martínez, 2015). It is noted that the referred book is currently only available in Spanish; nevertheless, the English reader could use the book (freely available at http://www.di.ugto.mx/GEMEC/) and infer the examples from the figures and numbers.

```
Example 5.3.2 of book: Aspectos básicos del método de rigideces
                       Example 3 – Plane frames

// Datos generales de análisis                                ! General data
       Unidades fuerza = ton
       Unidades longitud = cm
       Escribe datos de estructura
       Escribe matrices de barras
       Escribe matriz de estructura
       Solución Gauss-Seidel
       Esquema de almacenamiento completo

// Materiales  Fuerza=kg      Longitud=cm                     ! Materials
       Material=Conc_fc200   E=197989.898732

// Secciones   Longitud=cm                                    ! Sections
       Seccion=Columna Tipo=Rectangular     b=55    h=55
       Seccion=Trabe   Tipo=Rectangular     b=25    h=40

// Nodos       Longitud=m                                     ! Joints
       n=1     x=0     y=0
       n=2     x=6     y=0
       n=3     x=0     y=4
       n=4     x=6     y=4
       n=5     x=0     y=8
       n=6     x=6     y=8

// Barras                                                     ! Members
       n=1  i=1  j=3  Material=Conc_fc200  Seccion=Columna  Tipo=M2D
       n=2  i=2  j=4  Material=Conc_fc200  Seccion=Columna  Tipo=M2D
       n=3  i=3  j=5  Material=Conc_fc200  seccion=Columna  Tipo=M2D
       n=4  i=4  j=6  Material=conc_fc200  Seccion=Columna  Tipo=M2D
       n=5  i=3  j=4  Material=Conc_fc200  Seccion=Trabe    Tipo=M2D
       n=6  i=5  j=6  Material=Conc_fc200  Seccion=Trabe    Tipo=M2D
```

```
// Apoyos                                                        ! Supports
       n=1    Dx     Dy     Gz
       n=2    Dx     Dy     Gz

// Casos de carga estáticos                                      ! Static load cases
       Caso = Unico
              Cargas en los nodos   Fuerza=ton                       ! Loads on joints
                     n=4    Fx=2
                     n=6    Fx=3

              Cargas en barras      Fuerza=ton     Longitud=m  ! Loads on members
                     n=1    Wy=-1
                     n=3    Wy=-1
                     n=5    Wy=-2
                     n=6    Wy=-3

              Reporta vector de fuerzas                          ! Optional results
              Reporta vector de desplazamientos
              Reporta desplazamientos nodales
              Reporta calculo de fuerzas en barras
              Reporta Reacciones en apoyos
```

Appendix D EXAMPLE OF DATA INPUT FILE FOR GRID ANALYSIS

This appendix shows the data input file for Example 4.13 of the book "Análisis Estructural con Matrices" (*Structural Analysis with Matrices*) (Rojas-Rojas & Padilla-Punzo, 2009). Unfortunately, this reference is not freely available; however, figures for this example will be included in a future version of this manual.

```
Example 4.13 - Análisis Estructural con Matrices

// Datos generales de análisis ! General data
       Unidades de Fuerza   = ton                            ! Work units
       Unidades de Longitud = cm
       Escribe datos de estructura
       Escribe matrices de barras
       Escribe matriz de estructura

// Materiales   Fuerza=ton     Longitud=cm                   ! Materials
       Material=Unico          E=221          G=88

// Secciones    Longitud=cm                                  ! Sections
       Seccion=25x50  Tipo=General    Iz=260416.666667  J=178906.25
       Seccion=20x50  Tipo=General    Iz=208333.333333  J=98000

// Nodos        Longitud=m                                   ! Joints
       n=1     x=0.0   z=0.0
       n=2     x=0.0   z=3.0
       n=3     x=4.0   z=3.0
       n=4     x=4.0   z=0.0

// Barras                                                    ! Members
       n=1  i=1  j=2  Material=Unico  Seccion=25x50  Tipo=Reticula
       n=2  i=2  j=3  Material=único  Seccion=20x50  Tipo=Reticula
       n=3  i=4  j=3  Material=Unico  Seccion=25x50  Tipo=Reticula

// Apoyos                                                    ! Supports
       n=1 Dy Gx Gz
       n=4 Dy Gx Gz

// Casos de carga
       Caso=Unico
              Reporta vector de fuerzas                      ! Optional results
              Reporta vector de desplazamientos
              Reporta desplazamientos nodales
              Reporta reacciones en apoyos
              Reporta calculo de fuerzas en barras

              Cargas en las barras    Fuerza=ton     Longitud=m  ! Loads on bars
                     n=1              Fy=-5          a=1.5   b=1.5
                     n=3              Fy=-5          a=1.5   b=1.5
                     n=2              Wy=-2          a=0     b=0
```

Appendix E Example of Data Input File for 3D Frame Analysis

This appendix shows the data input file for example 4.14 of book “Análisis Estructural con Matrices” (*Structural Analysis with Matrices*) (Rojas-Rojas & Padilla-Punzo, 2009). Unfortunately, this reference is not freely available; however, figures for this example will be included in a future version of this manual.

```
Example 4.14 - Análisis Estructural con Matrices
               Marco 3D

// Datos generales de análisis       ! General data
       Unidades de Fuerza   = ton    ! Work units
       Unidades de Longitud = cm
       Escribe datos de estructura   ! Intermedia results
       Escribe matrices de barras
       Escribe matriz de estructura

// Materiales  Fuerza=ton     Longitud=cm                         ! Materials
       Material=Unico         E=158   G=47

// Secciones   Longitud=cm                          ! Sections
       Seccion=25x50  Tipo=General  A=1250  Iz=260417  Iy=65104  J=178906
       Sección=30x60  Tipo=General  A=1800  Iz=540000  Iy=135000 J=370980
       Seccion=30x30  Tipo=General   A=900  Iz=67500   Iy=67500  J=114210

// Nodos       Longitud=m                                         ! Joints
       n=1    x=-5   y=4    z=0
       n=2    x=0    y=4    z=0
       n=3    x=0    y=4    z=6
       n=4    x=0    y=0    z=0

// Barras                                                         ! Members
       n=1  i=1  j=2  Material=Unico  Seccion=25x50  Tipo=M3D
       n=2  i=3  j=2  Material=unico  Seccion=30x60  Tipo=M3D
       n=3  i=4  j=2  Material=Unico  Seccion=30x30   Tipo=M3D

// Apoyos                                                         ! Supports
       n=1 dx dy dz gx gy gz
       n=3 dx dy dz gx gy gz
       n=4 dx dy dz gx gy gz

// Casos de carga                                          ! Static load case
       Caso = Unico
              Reporta vector de fuerzas
              Reporta vector de desplazamientos
```

```
Reporta desplazamientos nodales
Reporta reacciones en apoyos
Reporta calculo de fuerzas en barras

Cargas en barras      Fuerza=ton     Longitud=m ! Loads on members
       n=1    Wy=-6
       n=2    Fy=-15  a=3
```

References

Cutchil, E., & McKee, J. (1969). Reducing de bandwith of sparse symmetric matrices. *Proceedings of ACM National Conference* (pp. 157-172). New York, NY: Asociation for Computing Machinery.

Hernández-Martínez, A. (2015). *Aspectos Básicos del Método de Rigideces.* Guanajuato: Universidad de Guanajuato.

Rojas-Rojas, R. M., & Padilla-Punzo, H. M. (2009). *Análisis Estructural con Matrices.* Mexico City: LIMUSA.