Facultad de Ingeniería – Udelar

Departamento de Diseño industrial – IIMPI

MODELADO DE SISTEMAS MECÁNICOS EMPLEANDO EL MÉTODO DE LOS ELEMENTO FINITOS

REFERENTE CLASE #3

LABORATORIO 1: INTRODUCCIÓN AL MÉTODO DE RIGIDEZ.

PROFESOR

Dr. Henry Figueredo Losada

Montevideo. Uruguay.
Noviembre 2019

TEMA II. FORMULACIÓN DE ELEMENTOS FINITOS 1D-ESTÁTICOS.

LABORATORIO 1. INTRODUCCIÓN AL MÉTODO DE RIGIDEZ.

Sumario.

Objetivos.

Utilizar la generación de un programa en Octave que realice los pasos principales en el análisis de un sistema estructural de resortes unidimensional mediante el método general para análisis de sistemas discretos MEF.

Bibliografía.

1. G.R.Liu "The Finite element Method- A practical course".
2. Amar Khennane "Introduction to Finite Element Analysis Using MATLAB and Abaqus"
3. Kwon and Bang "The Finite Element Method using MATLAB - Kwon and Bang".
4. A.J.M. Ferreira "MATLAB Codes for finite Element Analysis".

1. Introducción.

El uso de pc es una parte esencial para el modelado de sistemas mecánicos empleando elementos finitos. Para resolver problemas de ingeniería y para interpretar los resultados son necesarios crear algoritmos de programas de computadoras bien ideados, mantenidos y apoyados. Se dispone de muchos paquetes comerciales de elementos finitos que cumplen estos requisitos como se nombraron algunos en la primera clase introductorias. *Crear nuestros propios programas de computación con códigos fuentes disponibles, aumentaran nuestro propio proceso aprendizaje. Como uno de los objetivos principales del curso será inicial el montaje de una biblioteca propia de elementos para modelar problemas reales por intermedio de la discretización.*

Cada tema estará provisto para realizar programas de computación que corren en paralelo con las clases teóricas. El estudiante deberá hacer un esfuerzo para implementar en los programas los pasos dados en el aspecto teórico. Alentamos el uso de los paquetes comerciales para evaluar y

suplementar el proceso de aprendizaje. En la Bibliografía encontraran libros recomendados y códigos implementados para solución utilizando elementos finitos.

2. Definición de Software Octave v5.2.

Octave o GNU Octave es un programa y lenguaje de programación basado principalmente para realizar cálculos numéricos y está orientado fundamentalmente al análisis numérico. Es considerado el equivalente libre de MATLAB. Entre varias características que comparten, se puede destacar que ambos ofrecen un intérprete, permitiendo ejecutar ordenes en modo interactivo. La sintaxis es casi idéntica a la utilizada en MATLAB.

Detalles técnicos:

- Octave está escrito en C++ usando la biblioteca STL.
- Tiene un intérprete de su propio lenguaje (de sintaxis casi idéntica a Matlab), y permite una ejecución interactiva o por lotes.
- Su lenguaje puede ser extendido con funciones y procedimientos, por medio de módulos dinámicos.
- Utiliza otros programas GNU para ofrecer al usuario la posibilidad de crear gráficos para luego imprimirlos o guardarlos (Grace).
- Dentro del lenguaje también se comporta como una consola de órdenes (shell). Esto permite listar contenidos de directorios, por ejemplo.
- Además de correr en plataformas Unix también lo hace en Windows.
- Puede cargar archivos con funciones de Matlab (reconocibles por la extensión .m).
- Tiene ayuda en español.

Al ser su licencia Licencia pública general de GNU, puede ser compartido y utilizado libremente.

3. Organización General para el desarrollo de un Algoritmo de Trabajo.

El estilo de programación y sintaxis serán adaptados similar a la estructura organizativa de trabajo propuesta por el Ansys Workbench, Fig. 1.

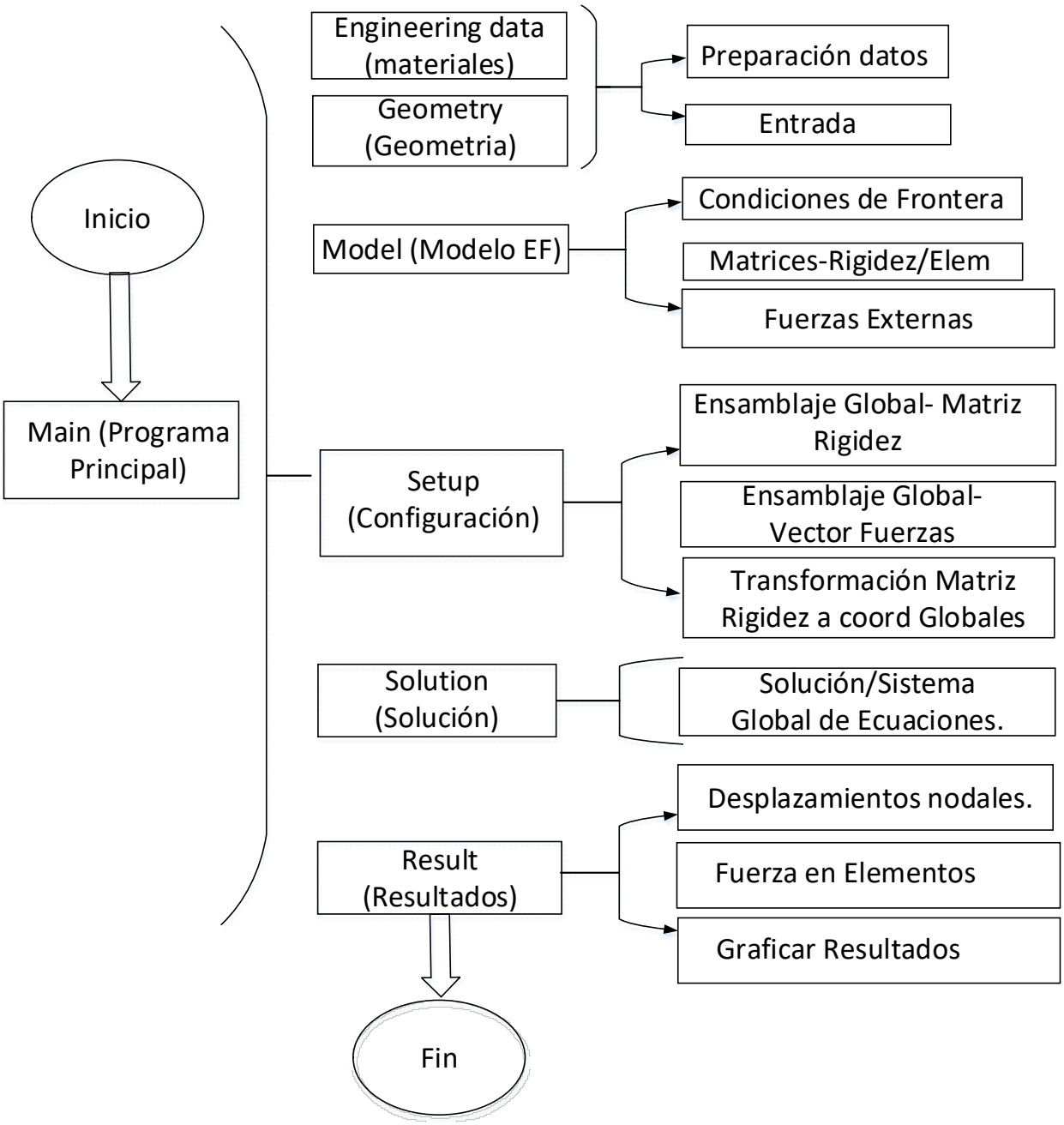
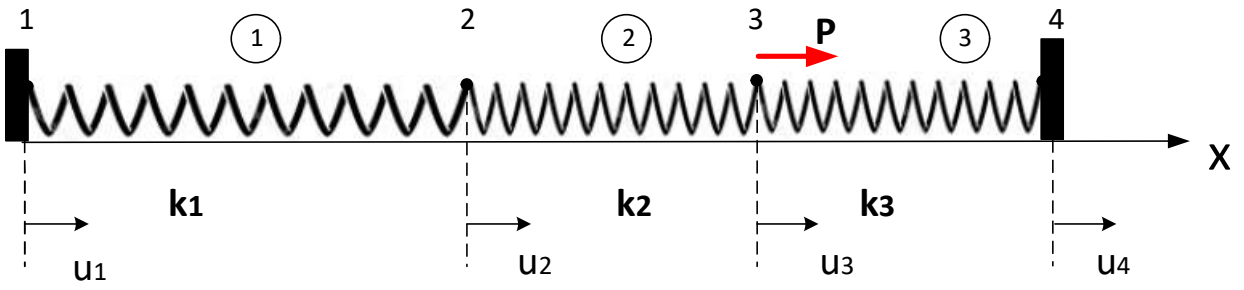


Figura 1. Esquema del programa para resolver estructuras formadas por elementos de resortes unidimensionales.

4. Solución computacional de MEF para un sistema mecánico con 3 elementos de tipo resortes (Algoritmo 1.0).

Ejemplo 1.1 : Dado el sistema resortes mostrado en la Fig. 1 , donde los nodos 1 y 4 están restringidos al desplazamiento , $u_1 = u_2 = 0$ y considerando las constantes elásticas de los resortes $k_1 = 100\ N/mm$; $k_2 = 200\ N/mm$; $k_3 = 100\ N/mm$ y $P = 500\ N$.



Obtenga:

1. La matriz de rigidez **[K]** y el vector de fuerzas **{F}** que representan el comportamiento del sistema.
2. Calcular los desplazamientos de los nodos 2 y 3
3. Calcular las fuerzas de reacción de los nodos 1 y 4.
4. Calcular la fuerza de tracción o compresión a la que está sometido el resorte 2.

Código del programa: Main_Resortes

En esta sección vamos a explicar la generación de un programa en Octave que realice las siguientes acciones, Fig.1:

Paso 1: Datos de entrada (Material + Geometría)

k: variables que contiene las constantes elásticas de cada elemento (resorte);

connNodos: Matriz de conectividad de nodos.

Variables

Nonodos: Numero de nodos totales de la estructura;

NoElem: Número de Elementos totales de la estructura (resortes);

NonodesElem: Número de nodos por elemento, en este caso 2;

Nodof: Número de grados de libertad por nodo, en este caso 1;

Nodof_t: Numero de grados de libertad totales de la estructura, *Nonodos*Nodof*;

Elemprops: Vector que contiene las constantes elásticas de cada elemento (resortes);

geom_CoorNodos: Coordenadas de los nodos, en este caso variables x.

Funciones: No hay funciones definidas solo scripts que constituyen el programa son:

Main_resortes.m: Principal Script de inicio que llama a cada parte de script del programa, Fig. 1.

resorte_data.m : Script para la introducción de los datos del sistema;

resort_model.m: Script que proporciona Condiciones de frontera, cargas;

resort_setup.m: Script que desarrolla el ensamblaje de la matriz de rigidez global, vector de fuerzas globales, transformación de coordenadas locales a globales.

resort_solution.m: Script con que se obtiene la solución del sistema global de ecuaciones, una vez obtenidos la solución de los desplazamientos, se obtienen las reacciones en los apoyos, las fuerzas que equilibran cada elemento y las fuerzas aplicadas en cada nodo.

resort_Resultados: Script utilizado para imprimir resultados en ficheros, graficar etc.

A continuación, se describe cada Script:

Fichero Main_resortes.m

El Script **Main_resortes.m** es el principal y va a llamar a cada parte de los Script que conforman el programa total por cada una de sus partes constituidas. Particularizando para el ejemplo queda como:

```
# Main programa principal
# Lineal Resortes
```

```
clc     # Limpia la ventana de comandos
clear   # Elimina todas las variables de la memoria del Octave

#<include truss_data>
# Incluir Propriedades del material y Geometría
resorte_data

#<include truss_model>
# Incluir Modelo (Condiciones de Frontera, cargas, Matrices de rigidez)
resorte_model

#<include truss_setup>

# Setup: Ensamblaje de la Matriz de rigidez Global
#        Ensamblaje del vector de fuerzas
#        Tranformacion de coordenadas locales a globales

resorte_setup

#<include truss_solution>
# Solución: Solución del sistema global de ecuaciones

resorte_solution

#<include truss_Resultados>

# Resultados: Desplazamiento nodales
#             Fuerza en los elementos
```

```
#              Grafico de resultados


resorte_Result


#Fin del Main_resortes.m
```

Comentario 1.1: Nótese que el Script Main_resortes es el que distribuye cuando es llamado cada parte de los Scripts del programa principal, de esta forma nos permite separar por partes cada modulo de Script para rastrear errores.

Fichero resorte_data.m

El Script **resorte_data** es el Script en que se introducen los datos de la matriz de conectividad, *connNodos*, constantes elásticas de los resortes (k) *Elemprops* y coordenadas de los nodos (x,Y) *geom_CoorNodos.*Particularizando para el ejemplo queda como:

```
# Resorte_data


# Entrada de datos

global Nonodos

Nonodos= 4;     # Número de nodos


global NoElem

NoElem = 3;     # Número de Elementos


global NonodesElem

NonodesElem = 2;     # Número de nodos por elemento


global Nodof

Nodof = 1;   # Número de grados de libertad por nodo
```

```
global Nodof_t=Nonodos*Nodof;  # Total Número de grados de libertad


# Propriedades Resortes k


global Emprops=zeros(1,NoElem);
k1=100;
k2=200;
k3=100;


Emprops=[k1,k2,k3]; # E and A de los elementos


# Coordenadas de los nodos X, Y
global geom_CoorNodos=zeros(Nonodos,1) # Inicializa Matriz [4x1] con valores zeros


geom_CoorNodos=[0 ; 10;  20 ; 30] ; # Coordenadas X


# Conectividad de Elementos
global connNodos=zeros(NoElem ,2);
connNodos=[1 2; 2 3; 3 4];
# Fin Resorte_data
```

Comentario 1.2: Nótese que **Emprops** es el vector que contiene las propiedades elásticas de los materiales en el modelo.

Fichero resorte_modelo.m

El Script **resorte_modelo** es el Script en que se introducen los datos como condiciones de Frontera ***nf***, y Cargas en los nodos ***load***. Particularizando para el ejemplo queda como:

```
# Archivo resorte_model


# Condiciones de Frontera, cargas


# Condiciones de Frontera (Asignamos 0 para restringir grado de libertad 1 para libre)
global nf
nf=ones(Nonodos,1); # Inicializa la matriz con 1


nf(1)=0;
nf(4)=0;


# Vector de cargas
global load
load=zeros(Nonodos,1);
load(3)=500;


# Fin Resorte_modelo
```

Fichero resorte_setup.m

El Script **resorte_setup** es el Script en que realiza el ensamblaje de la matriz de rigidez global ***KK*** e impone las condiciones de frontera global ***nf*** a la matriz de rigidez global y la transformación de coordenadas del sistema local al sistema global. Particularizando para el ejemplo queda como:

```
# Archivo resort_setup


# Setup: Ensamblaje de la Matriz de rigidez Global
#       Ensamblaje del vector de fuerzas
#       Tranformacion de coordenadas locales a globales
```

```
# Utilizando ConnNos calculamos las propiedades de la matriz local.


k=zeros(Nodof_t,Nodof_t,NoElem);


global KK
KK=zeros(Nodof_t,Nodof_t);


for i=1:NoElem
   k(connNodos(i,1),connNodos(i,1),i)=Elemprops(i);
   k(connNodos(i,2),connNodos(i,2),i)=Elemprops(i);
   k(connNodos(i,1),connNodos(i,2),i)=-Elemprops(i);
   k(connNodos(i,2),connNodos(i,1),i)=-Elemprops(i);
   KK=KK+k(:,:,i)
endfor


global KK1_temp
KK1_temp=KK;


# Imponiendo condiciones de Frontera a la matriz de rigidez


for i=1:Nonodos
   if (nf(i)==0)
     KK(:,i)=0;
     KK(i,:)=0;
     KK(i,i)=1;
   endif
 endfor
# Fin resort_setup
```

Comentario 1.3: Nótese que para este ejemplo no fue necesaria la transformación de coordenadas locales para globales porque son coincidentes los 2 sistemas de coordenadas. La utilización de un arreglo en la forma *KK (x,y,i)* donde *i=Número de elementos* almacena todas las matrices de rigidez de cada elemento en el modelo.

Fichero resorte_solution.m

El Script **resorte_solution** es el Script en que realiza la solución del sistema global de ecuaciones. Particularizando para el ejemplo queda como:

```
# Archivo resort_solution


# Solución: Solución del sistema global de ecuaciones


fsolDisp=KK\load;


global FF
FF=KK1_temp*fsolDisp-load;


# Fin resort_solution
```

Fichero resorte_result.m

El Script **resorte_result** es el Script en pos-procesador que realiza la función de visualización y organización de los datos de resultados del modelo, este es adaptado para mostrarlos las informaciones relevantes de los resultados. Particularizando para el ejemplo queda como:

```
# Archivo resorte_result


% Mejorar la salida...
file =fopen('ResultadosTruss.txt','w');  # Crear archivo .txt


%*****************************************************************
```

```
fprintf(file,'******* IMPRIMIENDO DATOS DEL MODELOS **********\n\n\n');


fprintf(file,'************************************************************\n');
fprintf(file,'Número de nodos: %g\n', Nonodos );
fprintf(file,'Número de Elementos: %g\n', NoElem);
fprintf(file,'Números de Nodos por Elementos: %g\n', NonodesElem );
fprintf(file,'Números de grados de Libertad por nodos: %g\n', Nodof);
fprintf(file,'Numeros de grados de Libertad por Elementos: %g\n\n\n', NonodesElem*Nodof);


fprintf(file,'************************************************************\n');
fprintf(file,'  Nodo   Coordenada X\n');
for i=1:Nonodos
  fprintf(file,' %g,          %g\n',i,geom_CoorNodos(i,1));
end
fprintf(file,'\n');
%
%
% Print element connectivity
%
fprintf(file,'************************************************************\n');
  fprintf(file,'Elementos     Nodo_1         Nodo_2 \n');
for i=1:NoElem
  fprintf(file,'%g,          %g,          %g\n',i,connNodos(i,1),connNodos(i,2));
end
fprintf(file,'\n');
%
% Print element property
%
```

```
fprintf(file,'************************************************************\n');
fprintf(file,'Elemento     k            \n');
for i=1:NoElem
  fprintf(file,'   %g,      %g,          \n',i,Elemprops(i));
end
fprintf(file,'\n');


%
% Print Nodal freedom
%
fprintf(file,'***********************************************************\n');
fprintf(file,'*******************Nodal DOF**********************\n');
fprintf(file,'Nodo       DOF_x  \n');
for i=1:Nodof_t
  fprintf(file,'%g,           %g\n',i,nf(i));
end
fprintf(file,'\n');
%
% Print Nodal loads
%
fprintf(file,'************************************************************\n');
fprintf(file,'*****************Cargas nodales Aplicadas*******************\n');
    fprintf(file,'Nodo       Load_X \n');
for i=1:Nodof_t
    fprintf(file,'%g,            %g,\n',i,load(i));
end


%*************** Result *************
```

```
% Extract nodal displacements
%
    fprintf(file,'_____\n');
    fprintf(file,'****_____Resultados _____****\n');
    fprintf(file,'\n');
    fprintf(file,' Nodo          Desplazamiento_X\n');
for i=1:Nonodos
   fprintf(file,'  %g,          %g,\n',i,fsolDisp(i));
end
fprintf(file,'\n');
%
% Print Nos Loads
%

fprintf(file,'Nodo        Load_X      \n');
for i=1:Nonodos
   fprintf(file,'%g,         %7.2f,\n',i,load(i));
end

fprintf(file,'\n');
%
% Print reaction Nos Loads
%

fprintf(file,'Nodo             Reacción Load_X      \n');
for i=1:Nonodos
   fprintf(file,'%g,                  %7.2f,     \n',i,FF(i));
end
```

```
fprintf(file,'\n');


fclose(file);


# Fin resorte_result
```

5. Utilización del Programa Salome-Meca. Primeros pasos.

Para el curso será utilizado el programa de Elementos Finitos con licencia gratuita Salome-Meca, para facilitar la visualización de resultados.

Ejercicios propuestos.

Se recomienda cuando se intente modelar los problemas sobre el código Octave implementado comenzar a testear los problemas resuelto en el practico expositivo y a continuación sus ejercicios resueltos estableciendo una comparación entre ambos y adicionalmente la utilización de un programa comerciales de elementos finitos.

Ejercicio Propuesto 1.1: El sistema mecánico de resortes mostrado en la Fig. 1.1. Obtenga la matriz de rigidez **[K]** y determine los desplazamientos desconocidos y fuerzas de reacción.

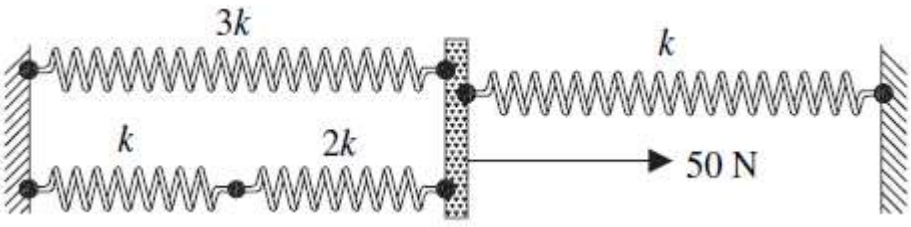


Figura 1.1 Sistema mecánico de resortes.

Ejercicio Propuesto 1.2: Para el sistema mecánico mostrado en la Fig. 1.5, determine el desplazamiento en los nodos 2 y 3 y la reacción en los nodos 1 y 4. Asuma que las barras verticales son rígidas en los nodos 2 y 3 y permanecen horizontales todo el tiempo. En el nodo 3 se aplica una fuerza de 1000 N. Los valores de las constantes elásticas de los resortes son $k_1 = 500\frac{N}{mm}$; $k_2 = k_3 = 300\frac{N}{mm}$; $k_4 = k_5 = 400\ N/mm$, respectivamente.

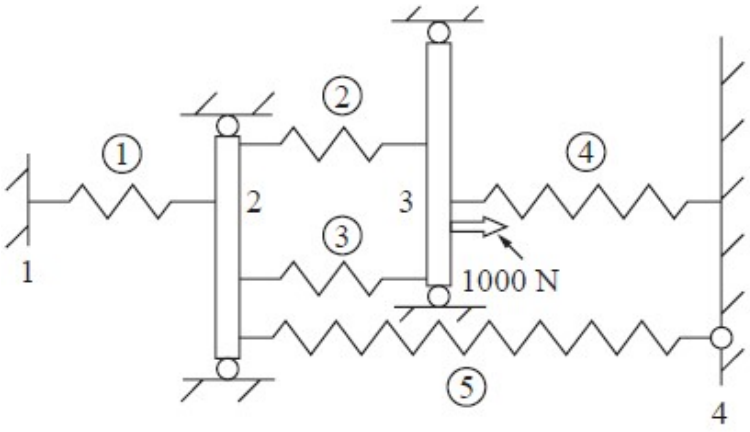


Figura 1.2 Sistema mecánico formado con elementos de tipo resortes.

Ejercicio Propuesto 1.3: El sistema mecánico de conectado mediante resortes mostrado en la Fig. 1.3 asumiendo que solo existe desplazamientos verticales para las masas y elementos rígidos en el sistema sin rotaciones. Obtenga la matriz de rigidez **[K]** y determine los desplazamientos desconocidos, y fuerzas de reacción.

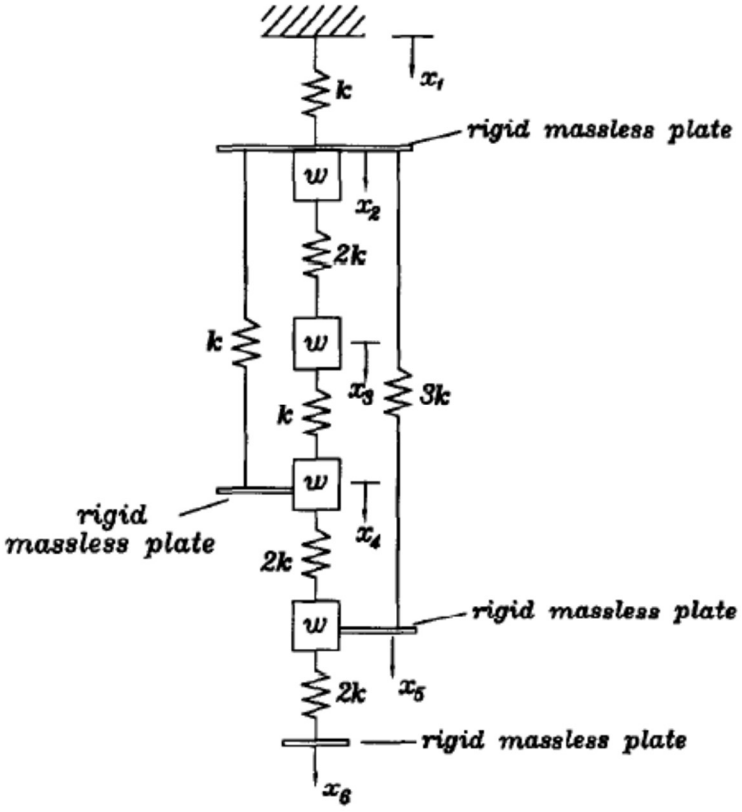


Figura 1.3. Sistema mecánico de resortes.