

# EPI

# GUÍA DE USUARIO

**Versión 2.0**

Mayo 1999

Juan Carlos García Orden

GRUPO DE MECÁNICA COMPUTACIONAL

DEPTO. DE MECÁNICA DE MEDIOS CONTINUOS Y TEORÍA DE ESTRUCTURAS  
ESCUELA TÉCNICA SUP. DE INGENIEROS DE CAMINOS, CANALES Y PUERTOS  
UNIVERSIDAD POLITÉCNICA DE MADRID.

# Índice

1. Características generales	1
2. Estructura orientada a objetos	2
3. Ejecutando EPI	3
4. Resumen de instrucciones	3
5. Ejemplo de aplicación. Problema de los tres cuerpos.	20

EPI es un programa capaz de analizar la dinámica de sistemas multicuerpo flexibles. La codificación se ha realizado en C++ empleando los conceptos básicos de programación orientada a objetos.

## 1. Características generales

- Es capaz de analizar la dinámica no lineal de una gran variedad de sistemas multicuerpo compuestos de sólidos rígidos y deformables. Todos ellos pueden estar acoplados mediante distintos tipos de uniones (rótulas esféricas, cilíndricas, prismáticas, etc), y pueden existir además muelles y amortiguadores de comportamiento no lineal.
- Es posible introducir fuerzas externas dependientes del tiempo actuando en cualquier punto del sistema. La dependencia temporal puede definirse de forma analítica o mediante una historia en una serie de instantes de tiempo.
- Es posible introducir un número arbitrario de paredes rígidas, que pueden interactuar con cualquier punto del sistema a lo largo del movimiento.
- Tiene disponible una amplia variedad de integradores: (Runge-Kutta de orden 4 explícito, diferencias centrales, Adams-Bashforth/Adams-Moulton de órdenes 4 y 5 con esquema explícito P(EC)E) e implícito con esquema P(EC)<sup>n</sup>E, familia  $\beta$ -Newmark general (incluye la regla trapecoidal), método Hilber, Hughes, Taylor (HHT), métodos BDF hasta orden 4, regla del punto medio implícita y algoritmo energía-momento.
- La definición básica del sistema se realiza mediante las coordenadas de un cierto número de puntos referidas a un sistema inercial. Una vez que el número de puntos y sus posiciones y velocidades iniciales han sido especificadas, el usuario define los puntos que pertenecen a cada cuerpo, rígido o deformable. El usuario también define las diferentes restricciones (de una librería básica) para modelizar las distintas uniones entre los cuerpos.
- La interfase gráfica se realiza mediante el programa GEOMVIEW, que permite manipular los objetos gráficos que representan el sistema a medida que se efectúa el cálculo (giros, traslaciones, cambios de escala, cambios de iluminación, etc).

## 2. Estructura orientada a objetos

Dentro de la estructura del programa, la mayor parte de los objetos (restricciones, elementos tanto rígidos como deformables, etc.) se definen como clases que derivan de una clase genérica en la que se definen una serie de funciones comunes. Estas funciones tienen significados diferentes en el contexto de cada clase derivada. Algunas de estas funciones comunes son:

- Cálculo y ensamblaje de la matriz de masas  $\mathbf{M}$  (`calc_M`, `ens_M`). A las restricciones no se les asigna masa generalmente. No obstante, al ser esta función común a todos los elementos, la modelización de uniones con masa es directa.
- Cálculo y ensamblaje del vector de fuerzas  $\mathbf{f}$  (`calc_Fq`, `ens_Fq`); para una restricción,  $\mathbf{f} \equiv \mathbf{f}_\Phi$ ; para un elemento elástico  $\mathbf{f} \equiv \mathbf{f}_{\text{int}}$ ; para un muelle, define la fuerza que actúa en los dos puntos extremos, etc. Es interesante observar que los cuerpos rígidos definen un vector de fuerzas a través de las restricciones que expresan la constancia de las distancias entre los puntos que los definen.
- Cálculo y ensamblaje de la matriz de rigidez  $\mathbf{K}$  (`calc_K`, `ens_K`). Es necesaria en todos los casos para definir correctamente la matriz tangente consistente que interviene en el método de integración empleado, siempre que éste sea de tipo implícito. Si el método es explícito, el cálculo de la matriz de rigidez no es necesario.

En el caso de las restricciones, la matriz de rigidez tiene su razón de ser en el empleo del método de penalización. En el caso de los cuerpos deformables discretizados, está fundamentada en los términos material-geométrico tradicionales. Es interesante observar que, al emplear el método de penalización, los cuerpos rígidos definen una matriz de rigidez a través de las restricciones internas de distancia constante.

Las clases derivadas más importantes son:

- Cuerpos rígidos: `massp`, `rigidbar`, `rigidbody2d`, `rigidbody` para partículas, barras unidimensionales rígidas, cuerpos rígidos bidimensionales y tridimensionales respectivamente.
- Elementos finitos (deformables): `elem1D`, `triang`, `brick` para elementos deformables unidimensionales, bidimensionales y tridimensionales respectivamente.

- Restricciones: `consd`, `tie`, `line3`, `angle`, `plane4` para la librería básica de restricciones, a partir de las cuales se pueden definir las uniones prácticas más representativas.

### 3. Ejecutando EPI

- La definición del modelo y las órdenes de cálculo se definen en un fichero de entrada, que debe llamarse `input.mbs`, y que se compone de líneas con instrucciones.
- Cada instrucción está encabezada por un comando clave, al que siguen argumentos alfanuméricos. Los comandos clave deben escribirse de forma literal, sin abreviaturas. Los argumentos numéricos se suministran en formato libre (el número de espacios entre los diferentes argumentos es arbitrario).
- Para activar el interfase gráfico con GEOMVIEW, un fichero llamado `.geomview` debe estar presente en el directorio de trabajo. El contenido de este fichero (una sola línea) debe ser:

```
(emodule-define "EPI./epi")
```

- Ejecutar GEOMVIEW. Una vez arrancado, el nombre "EPI" debe aparecer en primer lugar en la ventana etiquetada como "External Modules". Seleccionando "EPI" con el ratón, el cálculo comienza.

### 4. Resumen de instrucciones

#### ABAMEX45

Integrador temporal predictor-corrector Adams-Bashfort/Adams-Moulton de órdenes 4 y 5 respectivamente, con la secuencia P(EC)E (método explícito).

#### ABAMIM45

Integrador temporal predictor-corrector Adams-Bashfort/Adams-Moulton de órdenes 4 y 5 respectivamente, con la secuencia P(EC)<sup>n</sup>E (método implícito).

#### ANGLE3 *i j k*

Restricción que expresa que el ángulo formado por las direcciones *ij* e *ik* es constante.

*i*: Primer punto (entero)  
*j*: Segundo punto (entero)  
*k*: Tercer punto (entero)

**ANGLE4** *i j k l*

Restricción que expresa que el ángulo formado por las direcciones *ij* e *kl* es constante.

*i*: Primer punto (entero)  
*j*: Segundo punto (entero)  
*k*: Tercer punto (entero)  
*l*: Cuarto punto (entero)

**ANGLEHIS** *i j k his*

Restricción que expresa que el ángulo formado por las direcciones *ij* e *ik* viene dado por la historia de ángulos *his*.

*i*: Primer punto (entero)  
*j*: Segundo punto (entero)  
*k*: Tercer punto (entero)  
*his*: Historia de ángulos (entero)

\* Observación: La historia *his* debe definirse previamente con su correspondiente instrucción **HISTORY**.

**ANGLELAW** *i j k law*

Restricción que expresa que el ángulo formado por las direcciones *ij* e *ik* viene dado por la ley temporal analítica *law*.

*i*: Primer punto (entero)  
*j*: Segundo punto (entero)  
*k*: Tercer punto (entero)  
*law*: Ley analítica (entero)

\* Observación: La ley analítica *law* debe definirse previamente con su correspondiente instrucción **LAW** .

**AUGMENTED**

Emplea la formulación de lagrangiano aumentado. Este método no se encuentra disponible para todos los elementos en esta versión.

**BDF1**

Integrador multipaso lineal tipo BDF de 1 paso. Coincide con el método de Euler implícito.

**BDF2**

Integrador multipaso lineal tipo BDF de 2 pasos.

**BDF3**

Integrador multipaso lineal tipo BDF de 3 pasos.

**BDF4**

Integrador multipaso lineal tipo BDF de 4 pasos.

**BRICK *i j k l m n o p mat***

Define un hexaedro de ocho nodos elástico delimitado por los puntos  $i, j, k, l, m, n, o, p$  de un material  $mat$  con un modelo hiperelástico de Saint Venant-Kirchhoff. Admite grandes desplazamientos y deformaciones.

$i$ : Primer punto (entero)

$j$ : Segundo punto (entero)

$k$ : Tercer punto (entero)

$l$ : Cuarto punto (entero)

$m$ : Quinto punto (entero)

$n$ : Sexto punto (entero)

$o$ : Séptimo punto (entero)

$p$ : Octavo punto (entero)

$mat$ : Número identificador del material (entero)

**CENTRAL**

Método de integración temporal de diferencias centrales.

**COLORPLASTIC *smax smin***

Muestra diferentes colores en los elementos deformables (de momento, solo unidimensionales) según diferentes niveles de deformación plástica.

$smax$ : Deformación plástica máxima (real)

$smin$ : Deformación plástica mínima (real)

Azul color: Deformación plástica negativa.

Rojo color: Deformación plástica positiva.

**COLORSTRAIN *smax smin***

Muestra diferentes colores en los elementos deformables según diferentes niveles de tensión (componente X)

$smax$ : Deformación máxima (real)

$smin$ : Deformación mínima (real)

Azul color: Deformación negativa.

Rojo color: Deformación positiva.

**COLORSTRESSX *stmax stmin***

Muestra diferentes colores en los elementos deformables según diferentes niveles de tensión (componente X)

$stmax$ : Tensión máxima (real)

*stmin*: Tensión mínima (real)  
Azul color: Compresión.  
Rojo color: Tracción.

COLORSTRESSY *stmax stmin*

Muestra diferentes colores en los elementos deformables según diferentes niveles de tensión (componente Y).

*stmax*: Tensión máxima (real)  
*stmin*: Tensión mínima (real)  
Azul color: Compresión.  
Rojo color: Tracción.

COLORSTRESSZ *stmax stmin*

Muestra diferentes colores en los elementos deformables según diferentes niveles de tensión (componente Z)

*stmax*: Tensión máxima (real)  
*stmin*: Tensión mínima (real)  
Azul color: Compresión.  
Rojo color: Tracción.

CONSDIS *i j*

Restricción que expresa que la distancia entre los puntos *i* y *j* permanece constante.

*i*: Primer punto (entero)  
*j*: Segundo punto (entero)

CONSDIS1 *i x y z*

Restricción que expresa que la distancia entre el punto *i* y el definido por la coordenadas (*x,y,z*) permanece constante.

*i*: Primer punto (entero)  
*x,y,z*: Coordenadas del segundo punto (reales)

CONTACT *i wall penalty*

Expresa la posibilidad de contacto entre el punto *i* y la pared *wall* con un parámetro de penalización *penalty*.

*i*: Punto (entero)  
*wall*: Pared (entero)  
*penalty*: Parámetro de penalización (real)

DAMPER *i j c*

Define un amortiguador entre los puntos *i* y *j* con un coeficiente de amortiguamiento constante *c*.



*i*: Primer punto (entero)  
*j*: Segundo punto (entero)  
*c*: Coeficiente de amortiguamiento (real).

DEFBAR *i j nel mat mass*

Define una barra unidimensional elastoplástica entre los puntos *i* y *j*, discretizada en *nel* elementos unidimensionales (tipo “truss”), con un material *mat* y con masa *mass*.

*i*: Primer punto (entero)  
*j*: Segundo punto (entero)  
*nel*: Número de elementos (entero)  
*mat*: Material (entero)  
*mass*: Masa total (real)

ELEM1D *i j mat mass*

Define un elemento unidimensional (tipo “truss”) elastoplástico del punto *i* al *j*.

*i*: Primer punto (entero)  
*j*: Segundo punto (entero)  
*mat*: Material (entero)  
*mass*: Masa total (real)

ELEM1D1 *i x y z mat mass*

Define un elemento unidimensional (tipo “truss”) elastoplástico del punto *i* al punto definido por las coordenadas (*x,y,z*).

*i*: Primer punto (entero)  
*x,y,z*: Coordenadas del segundo punto (reales)  
*mat*: Material (entero)  
*mass*: Masa total (real)

EQUATION *i coordi a j coordj b*

Define una restricción lineal entre las coordenadas de los puntos *i* y *j* de la forma:

$$a(\text{coordi})_i + b(\text{coordj})_j = 0$$

*i* Primer punto (entero).  
*coordi* Coordenada del punto *i* afectada por la restricción (carácter).  
Opciones válidas son X,Y,Z.  
*a* Coeficiente que afecta a la coordenada del primer punto *i* (real).  
*j* Segundo punto (entero).  
*coordj* Coordenada del punto *j* afectada por la restricción (carácter).  
Opciones válidas son X,Y,Z.

*b* Coeficiente que afecta a la coordenada del segundo punto *j* (real).

\* Observación: En esta versión no está disponible la formulación de esta restricción con el algoritmo energía-momento.

**FEXTT** *i law1 law2 law3*

Fuerza aplicada en el punto *i* definida por las leyes analíticas *law1*, *law2*, *law3* para sus componentes (*x, y, z*) respectivamente.

*i*: Punto (entero)

*law1*: Ley analítica para la componente X (entero)

*law2*: Ley analítica para la componente Y (entero)

*law3*: Ley analítica para la componente Z (entero)

\* Observación: Las leyes analíticas *law1*, *law2*, *law3* deben definirse previamente con sus correspondientes instrucciones **LAW** .

**FHIS** *i coord his*

Fuerza aplicada en el punto *i* según la dirección *coord* según la historia temporal definida en *his*.

*i*: Punto (entero)

*coord*: Dirección absoluta de aplicación de la fuerza (carácter). Posibles valores son X, Y o Z

*his*: Historia temporal (entero)

\* Observación: La historia *his* debe definirse previamente con su correspondiente instrucción **HISTORY**.

**FIX** *i tin tfin*

Restricción que fija la posición del punto *i* entre los instantes *tin* y *tfin*.

*i*: Número identificador del punto (entero)

*tin*: Tiempo inicial (real) *tfin*: Tiempo final (real)

**GRAVITY** *g*

*g*: Aceleración de la gravedad (real), actuando siempre en la dirección Z en sentido negativo.

**HHT** *alphah*

Integrador temporal Hilber, Hughes y Taylor (HHT)

*alphah*: parámetro  $\alpha_h$  del integrador (real). ( $-1/3 \leq \text{alphah} \leq 0$ ).

La regla trapezoidal se obtiene para  $\alpha_h = 0,0$ .

**HISTORY** *i name*

Define la historia temporal *i* que se encuentra en el archivo *name*

*i*: Historia temporal (entero)

*name*: Nombre del fichero con la historia temporal (cadena de caracteres)

**LAW** *i a0 a1 a2 b0 b1 b2 ti tfin*

Ley temporal analítica con el formato:

$$L(t) = \frac{a0 + a1t + a2t^2}{b0 + b1t + b2t^2}$$

*i*: Ley analítica (entero)

*a0, a1, a2, b0, b1, b2* (reales) coeficientes de la expresión anterior.

*ti, tfin*: Instantes inicial y final respectivamente entre los que está definida la ley. Si *tfin* = 0, se sobreentiende que la ley esta definida a cualquier instante  $t > ti$ .

\* Observación: Por defecto, la ley 0 se reserva a  $L(t) = 0.$ , es decir:  $a0 = a1 = a2 = 0.$ ;  $b0 = 1.$ ;  $b1 = b2 = 0.$

**LAWR** *i a0 a1 a2 a3 b0 b1 b2 b3*

Ley analítica en distancia (*r*) con el formato:

$$L(r) = \frac{a0 + a1r + a2r^2 + a3r^3}{b0 + b1r + b2r^2 + b3r^3}$$

*i*: Ley analítica (entero)

*a0, a1, a2, a3, b0, b1, b2, b3* (reales) coeficientes de la expresión anterior.

\* Observación: Por defecto, la ley 0 se reserva a  $L(r) = 0.$ , es decir:  $a0 = a1 = a2 = a3 = 0.$ ;  $b0 = 1.$ ;  $b1 = b2 = b3 = 0.$

\* Observación: *r* es la distancia entre dos puntos. El uso más común de **LAWR** es definir una ley fuerza-desplazamiento en muelles no lineales, de forma que *r* es la distancia entre los puntos entre los que se sitúa el muelle. Consultar la instrucción **SPRING** para más detalles.

**LINE2** *i j*

Restricción que expresa que los puntos *i, j* se mantienen alineados con el origen.

*i*: Primer punto (entero)

*j*: Segundo punto (entero)

LINE3  $i j k$

Restricción que expresa que los puntos  $i, j, k$  se mantienen alineados.

$i$ : Primer punto (entero)

$j$ : Segundo punto (entero)

$k$ : Tercer punto (entero)

MASSP  $i m$

Define una partícula con masa.

$i$ : Punto en el que está situada la partícula (entero).

$m$ : Masa de la partícula (real).

MAT  $i E rho sy Hi Hk$

Define un tipo de material.

$i$ : Número identificador del material (entero).

$E$ : Módulo elástico (real)

$rho$ : Densidad (real).

$sy$ : Tensión de fluencia (real).

$Hi$ : Módulo de endurecimiento isótropo (real).

$Hk$ : Módulo de endurecimiento cinemático (real).

MIDPOINT2

Integrador temporal. Regla del punto medio implícita.

MIDPOINT3

Integrador temporal. Algoritmo energía-momento.

MOV  $i A1 B1 C1 law1 A2 B2 C2 law2 A3 B3 C3 law3$

Restricción que expresa que la posición de un cierto punto  $i$  viene dada por una cierta ley analítica.

Esta restricción se formula como la intersección de tres planos móviles (suponiendo que su intersección es en todo momento un punto):

$$A1x + B1y + C1z = law1(t)$$

$$A2x + B2y + C2z = law2(t)$$

$$A3x + B3y + C3z = law3(t)$$

donde  $A1, B1, C1, A2, B2, C2, A3, B3, C3$  (reales) son los coeficientes descritos anteriormente.

$law1, law2, law3$  (enteros) son los números que identifican las tres leyes analíticas

\* Observación: Una aplicación típica en la que los tres planos se intersectan en un punto es especificar planos Y-Z, X-Z y X-Y que tengan movimientos de traslación pura; por ejemplo:

$$x = law1(t)$$

$$y = law2(t)$$

$$z = law3(t)$$

por lo que  $A1 = B2 = C3 = 1$  y  $B1 = C1 = A2 = C2 = A3 = B3 = 0$ .

\* Observación:  $law1$ ,  $law2$ ,  $law3$  deben ser definidos previamente mediante la instrucción **LAW**

**MOVHIS** *i coord his*

Restricción que expresa que la coordenada *coord* del punto *i* debe seguir la historia temporal definida en *his*.

*i*: Punto afectado por la restricción (entero)

*coord*: Coordenada absoluta (carácter). Posibles valores son X,Y o Z.

*his*: Historia temporal (entero)

\* Observación: La historia *his* debe definirse previamente con su correspondiente instrucción **HISTORY**.

**NEWMARK** *gamma beta*

Método de integración temporal  $\beta$ -Newmark. *gamma*, *beta*: Parámetros  $(\gamma, \beta)$  del integrador (reales)

La regla trapezoidal (método de aceleración constante) se obtiene para  $\gamma = 1/2$  y  $\beta = 1/4$ ; el método de aceleración lineal para  $\gamma = 1/2$ ,  $\beta = 1/6$ .

**NNODE** *N*

*N*: Número total de puntos (entero)

**NODE** *i x y z vx vy vz*

*i*: Número identificador del punto (entero)

*x*: Coordenada X inicial (real).

*y*: Coordenada Y inicial (real).

*z*: Coordenada Z inicial (real).

*vx*: Velocidad inicial según X (real).

*vy*: Velocidad inicial según Y (real).

*vz*: Velocidad inicial según Z (real).

PENALTY *alpha*

*alpha*: Parámetro de penalización (real)

PLANE *i wall*

Restricción que expresa que el punto *i* se encuentra en todo momento contenido en el plano rígido *wall*.

*i*: Número identificador del punto (entero)

*wall*: Número identificador del plano (entero)

\* Observación: El plano *wall* debe definirse previamente con su correspondiente instrucción WALL.

PLANE4 *i j k l*

Restricción que expresa que los puntos *i*, *j*, *k* y *l* deben ser coplanarios.

*i*: Primer punto (entero)

*j*: Segundo punto (entero)

*k*: Tercer punto (entero)

*l*: Cuarto punto (entero)

PRINTACEL *n N1 N2 N3 ... Nn*

Guarda el vector aceleración absoluta ( $a_x, a_y, a_z$ ) de *n* puntos, siendo éstos los identificados como *N1*, *N2*, ... *Nn*, en un archivo llamado **acel.mbs**

La estructura del archivo **acel.mbs** es:

Primera columna: Tiempo (*t*)

Segunda columna:  $a_x$  del punto *N1*

Tercera columna:  $a_y$  del punto *N1*

Cuarta columna:  $a_z$  del punto *N1*

Quinta columna:  $a_x$  del punto *N2*

Sexta columna:  $a_y$  del punto *N2*

.....

PRINTANGLE *i j k*

Guarda el ángulo (en grados) que forman las direcciones definidas por las direcciones *ij* y *ik*.

*i*: Primer punto (entero)

*j*: Segundo punto (entero)

*k*: Tercer punto (entero)

PRINTEL

Guarda variables relacionadas con los elementos elastoplásticos unidimensionales (“truss”) en el archivo **elem.mbs**

La estructura del archivo `elem.mbs` es:

Primera columna: Tiempo

Segunda columna: Deformación del PRIMER elemento

Tercera columna: Tensión del PRIMER elemento

Cuarta columna: Deformación plástica del PRIMER elemento.

Quinta columna: Parámetro de endurecimiento isótropo del PRIMER elemento.

Sexta columna: Longitud total del PRIMER elemento.

Séptima columna: Deformación of the SEGUNDO elemento.

.....

\* Observación: En esta versión no es posible guardar magnitudes de elementos individuales. PRINTEL guarda los datos correspondientes a todos los elementos elastoplásticos unidimensionales que se han definido.

#### PRINTENER

Guarda la energía vs. tiempo en un fichero llamado `ener.mbs`

La estructura del fichero `ener.mbs` es:

Primera columna: Tiempo ( $t$ )

Segunda columna: Energía cinética

Tercera columna: Energía potencial

Cuarta columna: Energía de restricción

Quinta columna: Energía total (cinética + potencial)

Sexta columna: Energía total (cinética + potencial + e. de restricción)

#### PRINTFINAL

Guarda el número de puntos, con sus posiciones y velocidades, en el instante final de la integración, en el archivo llamado `final.mbs`

La estructura del archivo `final.mbs` imita la definición de los puntos propia del archivo de entrada: `NNODE n`

NODE 1  $x_1$   $y_1$   $z_1$   $vx_1$   $vy_1$   $vz_1$

NODE 2  $x_2$   $y_2$   $z_2$   $vx_2$   $vy_2$   $vz_2$

.....

NODE  $n$   $x_n$   $y_n$   $z_n$   $vx_n$   $vy_n$   $vz_n$

#### PRINTITER

Guarda el número de iteraciones en cada paso de integración vs. tiempo en un fichero llamado `iter.mbs`

La estructura del fichero `iter.mbs` es:

Primera columna: Tiempo ( $t$ )  
Segunda columna: Número de iteraciones.

**PRINTNODEST**  $n$   $N1$   $N2$   $N3$  ...  $Nn$

Guarda las tres componentes absolutas de la tensión de Cauchy ( $\sigma_x, \sigma_y, \sigma_z$ ) de  $n$  puntos, siendo éstos los identificados como  $N1, N2, \dots Nn$ , en el archivo llamado **nodest.mbs**

La estructura del archivo **nodest.mbs** es:

Primera columna: Tiempo ( $t$ )

Segunda columna: Tensión  $\sigma_x$  del punto  $N1$

Tercera columna: Tensión  $\sigma_y$  del punto  $N1$

Cuarta columna: Tensión  $\sigma_z$  del punto  $N1$

Quinta columna: Tensión  $\sigma_x$  del punto  $N2$

Sexta columna: Tensión  $\sigma_y$  del punto  $N2$

.....

**PRINTPOS**  $n$   $N1$   $N2$   $N3$  ...  $Nn$

Guarda las coordenadas ( $x, y, z$ ) de  $n$  puntos, siendo éstos los identificados como  $N1, N2, \dots Nn$ , en el archivo llamado **pos.mbs**

La estructura del archivo **pos.mbs** es:

Primera columna: Tiempo ( $t$ )

Segunda columna: Coordenada  $x$  del punto  $N1$

Tercera columna: Coordenada  $y$  del punto  $N1$

Cuarta columna: Coordenada  $z$  del punto  $N1$

Quinta columna: Coordenada  $x$  del punto  $N2$

Sexta columna: Coordenada  $y$  del punto  $N2$

.....

**PRINTREAC**  $n$   $N1$   $N2$   $N3$  ...  $Nn$

Guarda las tres componentes absolutas de la fuerza de reacción ( $R_x, R_y, R_z$ ) de  $n$  puntos, siendo éstos los identificados como  $N1, N2, \dots Nn$ , en el archivo llamado **reac.mbs**

La estructura del archivo **reac.mbs** es:

Primera columna: Tiempo ( $t$ )

Segunda columna: Reacción  $R_x$  del punto  $N1$

Tercera columna: Reacción  $R_y$  del punto  $N1$

Cuarta columna: Reacción  $R_z$  del punto  $N1$

Quinta columna: Reacción  $R_x$  del punto  $N2$

Sexta columna: Reacción  $R_y$  del punto  $N2$

.....

**PRINTSTEP**



Guarda los pasos de tiempo adoptados en un archivo llamado `step.mbs`

`PRINTVEL`  $n$   $N1$   $N2$   $N3$  ...  $Nn$

Guarda las velocidades ( $v_x, v_y, v_z$ ) de  $n$  puntos, siendo éstos los identificados como  $N1, N2, \dots, Nn$ , en un archivo llamado `vel.mbs`

La estructura del fichero `vel.mbs` es:

Primera columna: Tiempo ( $t$ )

Segunda columna:  $v_x$  del punto  $N1$

Tercera columna:  $v_y$  del punto  $N1$

Cuarta columna:  $v_z$  del punto  $N1$

Quinta columna:  $v_x$  del punto  $N2$

Sexta columna:  $v_y$  del punto  $N2$

.....

`QUASISTATIC`  $h$

Permite ejecutar un cálculo dinámico con una alta disipación numérica (empleando el método de Euler implícito, BDF1) que progresa hasta que el sistema queda en reposo.

$h$ : Paso de tiempo empleado por el integrador temporal.

`RIGIDBAR`  $i$   $j$   $mass$

Define una barra rígida de masa  $mass$  entre los puntos  $i$  a  $j$ .

\* Observación: Puesto que se trata de una barra unidimensional, no se define su sección.

$i$ : Primer punto (entero)

$j$ : Segundo punto (entero)

$mass$ : Masa (real)

`RIGIDBAR1`  $i$   $x$   $y$   $z$   $mass$

Define una barra rígida de masa  $mass$  entre el punto  $i$  y el punto fijo definido por las coordenadas  $(x, y, z)$ .

\* Observación: Puesto que se trata de una barra unidimensional, no se define su sección.

$i$ : Primer punto (entero)

$x, y, z$ : Coordenadas del segundo punto (reales)

$mass$ : Masa (real)

`RIGIDBODY`  $i$   $j$   $k$   $l$   $mass$   $x_{ij}$   $y_{ij}$   $z_{ij}$   $x_{ik}$   $y_{ik}$   $z_{ik}$   $x_{il}$   $y_{il}$   $z_{il}$   $A$   $B$   $C$   $fname$   $l1$   $l2$   
 $l3$

Define un sólido rígido tridimensional de masa  $mass$  mediante cuatro puntos no coplanarios ni alineados en grupos de tres, siendo uno de ellos (el  $i$ ) el centro de masas. Es necesario definir a priori un sistema de ejes cuerpo que sea principal de inercia y que tenga origen en el centro de masas  $G \equiv i$ .

$i$ : Primer punto, que debe coincidir con el centro de masas  $G$  (entero)

$j$ : Segundo punto (entero)

$k$ : Tercer punto (entero)

$l$ : Cuarto punto (entero)

$mass$ : Masa (real)

$x_{ij}, y_{ij}, z_{ij}$ : Coordenadas  $x, y, z$  respectivamente, relativas a un sistema de ejes cuerpo, del vector posición del punto  $j$  respecto del punto  $i$  (reales)

$x_{ik}, y_{ik}, z_{ik}$ : Coordenadas  $x, y, z$  respectivamente, relativas a un sistema de ejes cuerpo, del vector posición del punto  $k$  respecto del punto  $i$  (reales)

$x_{il}, y_{il}, z_{il}$ : Coordenadas  $x, y, z$  respectivamente, relativad a un sistema de ejes cuerpo, del vector posición del punto  $l$  respecto del punto  $i$  (reales)

$A$ : Momento principal de inercia, según el eje del cuerpo  $x$  (real).

$B$ : Momento principal de inercia, según el eje del cuerpo  $y$  (real).

$C$ : Momento principal de inercia, según el eje del cuerpo  $z$  (real).

$fname$ : Nombre del archivo con el fichero gráfico (cadena de caracteres). Aunque  $fname$  puede ser cualquier objeto gráfico reconocible por GEOMVIEW que se encuentre en el directorio de trabajo, existen varios objetos genéricos, que son:

- Elipsoide de inercia:  $fname=inerelip$

- Prisma de inercia:  $fname=inerpris$ . Prisma cuyas dimensiones principales son los inversos de las raíces cuadradas de los correspondientes momentos principales de inercia. Se puede interpretar como la versión prismática del elipsoide de inercia.

- Elipsoide genérico:  $fname=elip$ , orientado según los ejes cuerpo (direcciones principales).

- Prisma genérico:  $fname=pris$ , orientado según los ejes cuerpo (direcciones principales).

$l1$   $l2$   $l3$ : Dimensiones del objeto gráfico según los ejes cuerpo ( $x, y, z$  respectivamente) (reales). Tiene efecto sobre los objetos definidos con *elip* y *pris*; para el resto de objetos, estos valores pueden ser arbitrarios, y no producen ningún efecto.

**RIGIDBODY5**  $i$   $j$   $k$   $l$   $m$   $mass$   $x_{ij}$   $y_{ij}$   $z_{ij}$   $x_{ik}$   $y_{ik}$   $z_{ik}$   $x_{il}$   $y_{il}$   $z_{il}$   $A$   $B$   $C$   $fname$   $l1$   
 $l2$   $l3$

Define un sólido rígido tridimensional de masa  $mass$  mediante cinco puntos. Los cuatro primeros no deben ser coplanarios ni alineados en grupos de tres, y el primero de ellos (el  $i$ ) debe coincidir con el centro de masas. Es necesario definir a priori un sistema de ejes cuerpo que sea principal de inercia y que tenga origen en el centro de masas  $G \equiv i$ .

\* Observación: En la práctica, el programa asigna una masa muy pequeña al quinto punto e introduce tres restricciones de distancia constante a los puntos  $i$   $j$   $k$ , por lo que es importante asegurar que estos tres últimos puntos no estén alineados.

Los argumentos de esta instrucción son los mismos que los correspondientes a RIGIDBODY, salvo que se añade el nuevo punto  $l$ .

**RIGIDBODY2D**  $i$   $j$   $k$   $mass$   $x_{ij}$   $y_{ij}$   $x_{ik}$   $y_{ik}$   $A$   $B$   $fname$   $l1$   $l2$

Define un sólido rígido bidimensional de masa  $mass$  mediante tres puntos no coplanarios ni alineados, siendo uno de ellos ( $i$ ) el centro de masas. Es necesario definir a priori un sistema de ejes cuerpo que sea principal de inercia y que tenga origen en el centro de masas  $G \equiv i$ . El plano  $xy$  del sistema de ejes cuerpo debe coincidir con el propio plano del cuerpo.

$i$ : Primer punto, que debe coincidir con el centro de masas  $G$  (entero)  
 $j$ : Segundo punto (entero)  
 $k$ : Tercer punto (entero)  
 $mass$ : Masa (real)

$x_{ij}, y_{ij}$ : Coordenadas  $x, y$  respectivamente, relativas a un sistema de ejes cuerpo, del vector posición del punto  $j$  respecto del punto  $i$  (real)  
 $x_{ik}, y_{ik}$ : Coordenadas  $x, y$  respectivamente, relativas a un sistema de

ejes cuerpo, del vector posición del punto  $k$  respecto del punto  $i$  (real)

$A$ : Momento principal de inercia, según el eje del cuerpo  $x$  (real).

$B$ : Momento principal de inercia, según el eje del cuerpo  $y$  (real).

$fname$ : Nombre del archivo con el fichero gráfico (cadena de caracteres). Aunque  $fname$  puede ser cualquier objeto gráfico reconocible por GEOMVIEW que se encuentre en el directorio de trabajo, existen varios objetos genéricos, que son:

- Elipsoide de inercia:  $fname=inerelip$

- Prisma de inercia:  $fname=inerpris$ . Prisma cuyas dimensiones principales son los inversos de las raíces cuadradas de los correspondientes momentos principales de inercia. Se puede interpretar como la versión prismática del elipsoide de inercia.

- Elipsoide genérico:  $fname=elip$ , orientado según los ejes cuerpo (direcciones principales).

- Prisma genérico:  $fname=pris$ , orientado según los ejes cuerpo (direcciones principales).

$l1$   $l2$  : Dimensiones del objeto gráfico según los ejes cuerpo ( $x, y, z$  respectivamente) (reales). Tiene efecto sobre los objetos definidos con  $elip$  y  $pris$ ; para el resto de objetos, estos valores pueden ser arbitrarios, y no producen ningún efecto.

#### RK4

Integrador temporal Runge-Kutta explícito de orden 4.

#### SPRING $i$ $j$ $lawr$

Define un muelle no lineal entre los puntos  $i$  y  $j$

$i$ : Primer punto (entero)

$j$ : Segundo punto (entero)

$lawr$ : Número identificador de la ley analítica que define la relación fuerza-posición relativa entre los dos puntos (entero).

\* Observación: La historia  $lawr$  debe definirse previamente con su correspondiente instrucción **LAWR**.

#### SPRING1 $i$ $x$ $y$ $z$ $lawr$

Define un muelle no lineal entre el punto  $i$  y el punto definido por las coordenadas  $x$   $y$   $z$

$i$ : Primer punto (entero)

$x y z$ : Coordenadas del segundo punto (reales)  
 $lawr$ : Número identificador de la ley analítica que define la relación fuerza-posición relativa entre los dos puntos (entero).

\* Observación: La historia  $lawr$  debe definirse previamente con su correspondiente instrucción **LAWR**.

**SURFCON**  $i$

Restricción que expresa que el punto  $i$  se mantenga sobre una determinada superficie fija. En esta versión del programa, la superficie debe ser definida en el propio código fuente.

**TIE**  $i j$

Iguala los tres grados de libertad  $(x, y, z)$  de los puntos  $i, j$ .

$i$ : Primer punto (entero)

$j$ : Segundo punto (entero)

**TIE1**  $i x y z$

Iguala los tres grados de libertad del punto  $i$  y del punto fijo definido por las coordenadas  $x, y, z$

$i$ : Primer punto (entero)

$x, y, z$ : Coordenadas del segundo punto (reales)

**TIME**  $t0 tf ts h sc hmin hmax$

$t0$ : Origen de tiempos (real).

$tf$ : Tiempo final (real).

$ts$ : Paso de tiempo de muestreo (real). Sin uso en esta versión.

$h$ : Paso de tiempo inicial (real).

$sc$ : Variable de control de paso de tiempo (entero):

$sc = 0$  (Paso de tiempo constante)

$sc = 1$  (Paso de tiempo variable)

$hmin$ : Paso de tiempo mínimo (real) Cuando hay contactos, es el paso de tiempo adoptado en la fase de contacto.

$hmax$ : Paso de tiempo máximo (real)

**TITLE** *Character strings*

Título o comentarios. Es válido cualquier número de cadena de caracteres, aunque debe evitarse el uso de palabras que coincidan literalmente con instrucciones (por ejemplo, **SPRING**, **TIME**, etc.)

**TRIANG\_NEO** *i j k mat mass*

Define un elemento triangular de deformación constante (CST) delimitado por los puntos  $i, j, k$ , de masa  $mass$  y con un material  $mat$  representado por un modelo neo-Hookeano. El espesor inicial del elemento se calcula a partir de su densidad, masa y área. Admite grandes desplazamientos y deformaciones. No tiene disponible la formulación conservativa.

*i*: Primer punto (entero)

*j*: Segundo punto (entero)

*k*: Tercer punto (entero)

*mat*: Número identificador del material (entero)

*mass*: Masa (real)

**TRIANG\_SV** *i j k mat mass*

Define un elemento triangular de deformación constante (CST) delimitado por los puntos  $i, j, k$ , de masa  $mass$  y con un material  $mat$  representado por un modelo de Saint Venant-Kirchhoff. El espesor inicial del elemento se calcula a partir de su densidad, masa y área. Admite grandes desplazamientos y deformaciones. No tiene disponible la formulación conservativa.

*i*: Primer punto (entero)

*j*: Segundo punto (entero)

*k*: Tercer punto (entero)

*mat*: Número identificador del material (entero)

*mass*: Masa (real)

**WALL** *i A B C D*

Define una pared rígida, fija, lisa y plana dada por

$$Ax + By + Cz + D = 0$$

*i*: Número identificador de la pared (entero).

*A, B, C, D* (reales). Coeficientes de la ecuación que define el plano.

## 5. Ejemplo de aplicación. Problema de los tres cuerpos.

El programa es capaz de analizar la dinámica de sistemas “clásicos” en el sentido ingenieril, como máquinas (con componentes rígidos o deformables), estructuras sometidas a cargas transitorias, etc. Pero también es capaz de

analizar sistemas dinámicos más académicos, como movimientos de partículas sometidas a fuerzas generales.

Para ilustrar esta capacidad se considera un problema de tres cuerpos (idealizados mediante partículas) entre los que actúan fuerzas de tipo gravitatorio. Considerando dos partículas cuya posición está definida por los vectores  $(\mathbf{x}_i, \mathbf{x}_j)$ , la expresión general de la fuerza que actúa sobre la partícula  $i$  debida a la acción de la partícula  $j$  es:

$$\mathbf{f}_{i,j} = G \frac{m_i m_j}{r_{i,j}^3} \mathbf{r}_{i,j} = -\mathbf{f}_{j,i} \quad ; \quad \mathbf{r}_{i,j} = \mathbf{x}_j - \mathbf{x}_i$$

donde  $G$  es la constante gravitatoria,  $m_i, m_j$  son las masas de las dos partículas y  $r_{i,j} = \|\mathbf{r}_{i,j}\|$ .

Los datos numéricos considerados en la prueba han sido  $G = 5 \text{ m}^3/\text{kg s}^2$ ,  $m_1 = 2 \text{ kg}$ ,  $m_2 = m_3 = 1 \text{ kg}$ . En el instante inicial, las posiciones y velocidades de las tres partículas son:

$$\begin{aligned} \mathbf{x}_1 &= (0, 0, 0) & \mathbf{v}_1 &= (0, -1, 0) \\ \mathbf{x}_2 &= (1, 0, 0) & \mathbf{v}_2 &= (0, 1, 0) \\ \mathbf{x}_3 &= (-4, 0, 0) & \mathbf{v}_3 &= (0, 1, 0,5) \end{aligned}$$

Con estas condiciones iniciales y dado que el sistema es aislado, la velocidad del centro de masas es constante y dada por  $\mathbf{v}_G = (0, 0, 0,5)$ .

La trayectoria de las tres partículas calculada con el algoritmo energía-momento con un paso de tiempo  $h = 0,008 \text{ s}$  es la que se muestra en la Figura 1. La Figura 2 muestra la evolución calculada de la energía cinética y de la total; ésta última se conserva de forma exacta.

El fichero de entrada de datos para este problema es el que se incluye a continuación:

```
TITLE Problema de los tres cuerpos
TIME 0. 10. 1. 0.008 01. 1.
MIDPOINT3
PRINTPOS 1 3
PRINTENER
LAWR 1 10. 0. 0. 0. 0. 0. 1. 0.
LAWR 2 5. 0. 0. 0. 0. 0. 1. 0.
NNODE 3
NODE 1 0. 0. 0. 0. -1. 0.
NODE 2 1. 0. 0. 0. 1. 0.
NODE 3 -4. 0. 0. 0. 1. 0.5
MASSP 1 2.
```

MASSP 2 1.  
MASSP 3 1.  
SPRING 1 2 1  
SPRING 1 3 1  
SPRING 2 3 2

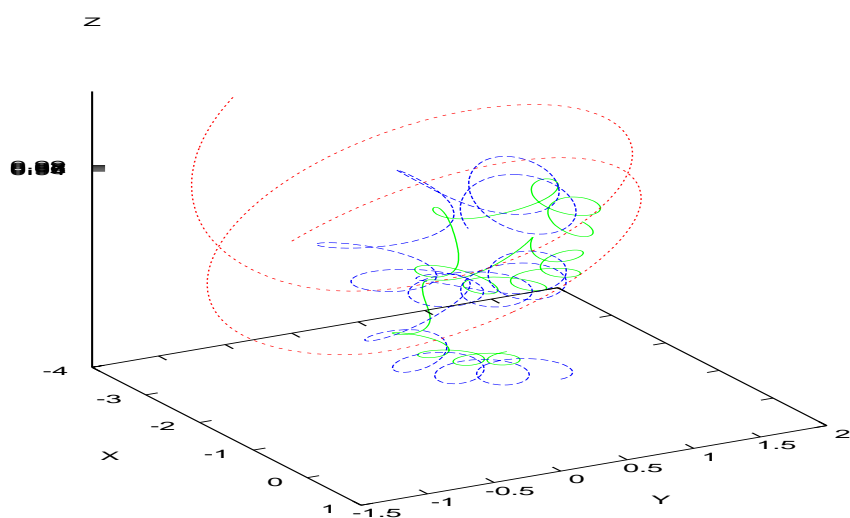


Figura 1: *Problema de los tres cuerpos. Trayectorias.*



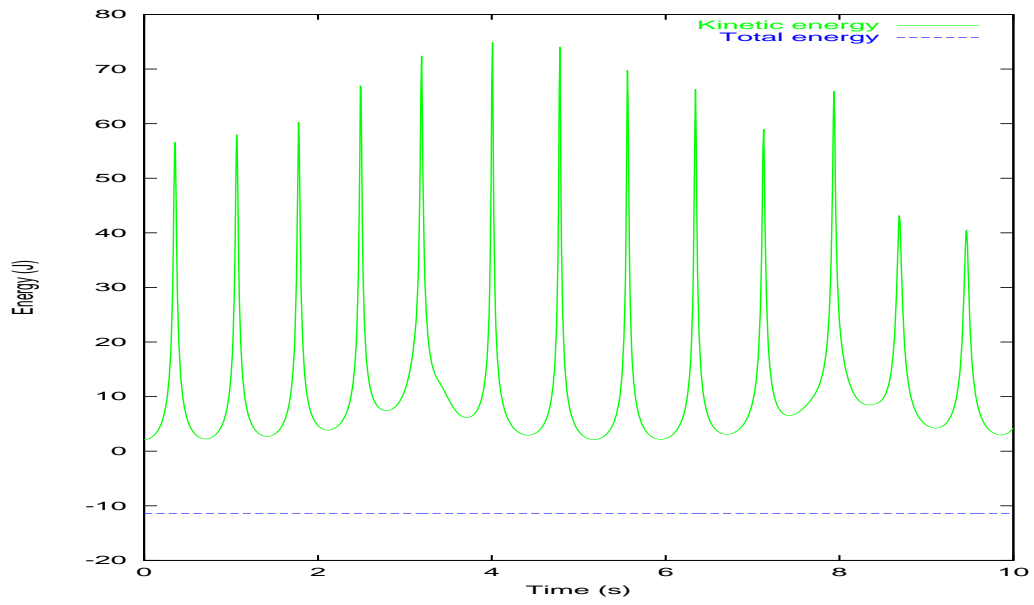


Figura 2: Problema de los tres cuerpos. Evolución de la energía.