MECAPAC3D
INTRODUCCIÓN

Mecánica
ETSICCP Madrid

24 de febrero de 2003

Índice

Índice de figuras

1. Introducción

El objetivo de mecapac es definir una serie de funciones en Maple que hagan sencillo el estudio de sistemas mecánicos, permitiendo obtener las ecuaciones que rigen su movimiento, realizar la integración (numérica) de las mismas y representar gráficamente en tres dimensiones el movimiento de los mismos.

La formulación de las ecuaciones utilizada es la formulación lagrangiana, dado que es la que mejor se adapta a un tratamiento sistemático.

En lo que sigue se hace una introducción a mecapac desde el punto de vista de usuario.

2. Parametrización del movimiento

Dado que las magnitudes fundamentales en la formulación lagrangiana son la energía cinética y la potencial, es necesario definir las variables que para cada elemento de un sistema mecánico permitan obtener las mismas.

En principio bastaría con conocer de cada sólido la posición de su centro de masa y su velocidad angular.

Si se quiere sistematizar la obtención de la velocidad angular suele ser mejor obtenerla a partir de la rotación que nos da la orientación del sólido. Por otro lado si queremos hacer representaciones gráficas de los sólidos es necesario conocer la orientación del mismo, siendo entonces la velocidad angular una consecuencia del conocimiento de la matriz de rotación que relaciona los ejes fijos y los ejes del sólido.

Para la obtención de las matrices de rotación en los casos más usuales se ha definidio una funciones auxiliar que permite la obtención de las mismas.

Esta función, denominada rota, devuelve la matriz de rotación que corresponde a la rotación respecto de un eje coordenado de valor arbitrario.

Un ejemplo sería:

```
r1 := rota(s,1) ;
```

que se corresponde con la rotación respecto del eje x de valor s.

Si la orientación del sólido no se puede obtener como rotación de un eje coordenado solamente, se puede utilizar la función anterior combinando distintas rotaciones respecto de ejes coordenados.

Teniendo en cuenta que la forma de obtener una rotación compuesta es mediante el producto de las matrices de rotación correspondientes, se puede proceder definiendo las diversas matrices de giros elementales combinándolas posteriormente.

En el caso de que los los giros sean relativos, los productos se hacen por la derecha, mientras que si son giros absolutos se multiplican por la izquierda (ver problema 62 de prácticas 2001–2002) .

Una vez conocida la posición y orientación del sólido se calculan (internamente dentro de las funciones de mecapac) la velocidad de su centro de masa y la velocidad angular, y a partir de ahí se obtienen las energías.

3. Ecuaciones del movimiento

Utilizando las ecuaciones de Lagrange, las ecuaciones se obtienen a partir de la lagrangiana del sistema, que como se sabe es la diferencia entre las energías cinética y potencial.

La energía cinética de un sistema se obtiene simplemente sumando las energías cinéticas de cada uno de los elementos que la forman. De forma similar se procede con la energía potencial.

La forma de proceder con mecapac es la siguiente:

1. Definir las coordenadas generalizadas del sistema en una lista que se denominará cg.
2. Definir utilizando variables, las magnitudes auxiliares necesarias, como pueden ser matrices de rotación, coordenadas de puntos, constantes, vectores, etc...
3. Definición mediante variables de los elementos que forman el sistema mecánico.
4. Definición de la variable sistema que contiene una lista con los elementos anteriormente definidos.
5. Obtención de la energía cinética del sistema mediante fT asignándola a una variable que se denominará T.
6. Obtención de la energía potencial del sistema mediante fV asignándola a una variable que se denominará V.
7. Obtención de la lagrangiana que se debe denominar L.
8. Obtención de las ecuaciones de Lagrange (utilizando ec_l o ec_lag)
9. Asignación de valores numéricos a los parámetros que queden sin asignar.
10. Integración numérica del problema mediante la función fint asignando el resultado a la variable res.
11. Representación de resultados mediante odeplot.
12. Animación del resultado con la función dibu3 (si la lagrangiana dependiera del tiempo de forma explícita mediante dibu2).

3.1. Ligaduras anholónomas

En el caso de existir ligaduras anholónomas caracterizadas como expresiones lineales en las velocidades generalizadas (catastásicas), también es posible obtener las ecuaciones y su resolución.

En este caso el procedimiento de resolución difiere del anterior en los siguientes aspectos:

1. Deben definirse las ligaduras que tiene el sistema. Estas ligaduras se definen en una lista que se denominará *Phi*. Al escribir tanto sean coordenadas como velocidades, no se incluirá de forma explícita la dependencia del tiempo, escribiendo las velocidadas generalizadas como coordenadas con un 1 yuxtapuesto, (ej.: x y $x1$).

2. Una vez definidas las ligaduras, se obtiene la expresión de las reacciones asociadas a esas ligaduras; tantas como coordenadas generalizadas tenga el sistema. Esas reacciones son las que se incluirán en las ecuaciones del movimiento. La función para obtener las reacciones es Fc(). (este paso no es indispensable).

3. La obtención de las ecuaciones del movimiento se realiza mediante la función ec_lagr.

4. La integración de las ecuaciones se obtiene mediante la función fintr.

4. Componentes

Los componentes se definen creando una variable por cada uno de ellos, la cual contiene una lista, cuyo primer elemento es el nombre del tipo de componente y el resto de elementos, son los parámetros necesarios para poder definir su situación, sus caracteristicas dinámicas y su representación gráfica.

Entre los componentes que se pueden usar para definir un sistema mecánico, podemos distinguir dos tipos:

- Componentes mecánicos propiamente dichos. En este grupo se encuadran aquellos elementos que influyen en el comportamiento dinámico del correspondiente sistema, es decir, aquellos que tienen o energía potencial, o energía cinética o ambas.

- Componentes gráficos. En este grupo se incluyen aquellos elementos que sólo tienen representación visual. El objetivo del uso de los mismos es facilitar la representación gráfica, pudiendo dibujar ejes, ángulos, vectores, etc...

Los componentes que se encuentran disponibles en este momento son los siguientes, empezando por los componentes mecánicos:

4.1. Punto

Permite definir masas puntuales sometidas al campo gravitatorio dando sus tres coordenadas cartesianas y su masa.

```
p1 := [punto,X,Y,Z,m] ;
```

Una representación esquemática puede verse en la figura 1

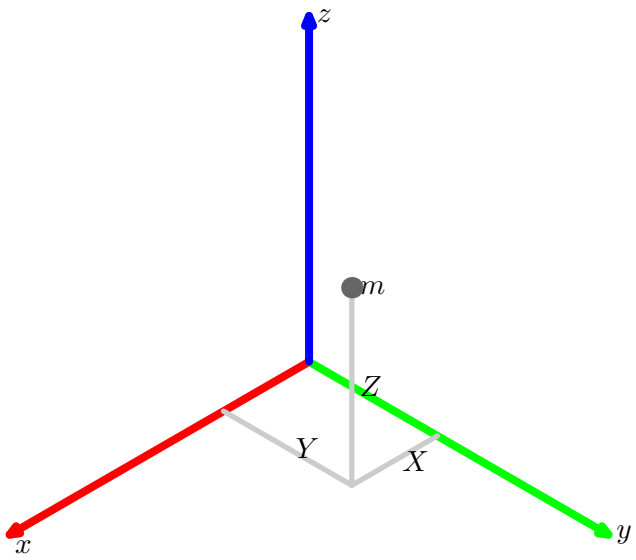


Figura 1: Punto

4.2. Varilla

Sirve para definir varillas a partir de:

- las coordenadas de su centro de masa (tipo de variable lista)
- la matriz de rotación entre unos ejes fijos y unos ejes ligados a la varilla situados en su centro de masa
- la masa de la misma
- la longitud de la varilla

Un ejemplo de definición de varilla sería:

```
v1 := [varilla,xgvar,rotp,m,h] ;
```

Los ejes locales que describen la posición de la varilla son tales que el eje z está situado según el eje de la misma. (Ver figura 2)

4.3. Aro

La definición de un aro se hace de forma prácticamente idéntica a la de una varilla (como en general para los sólidos en tres dimensiones).

Hace falta conocer lo siguiente:

- las coordenadas de su centro de masa (tipo de variable lista)

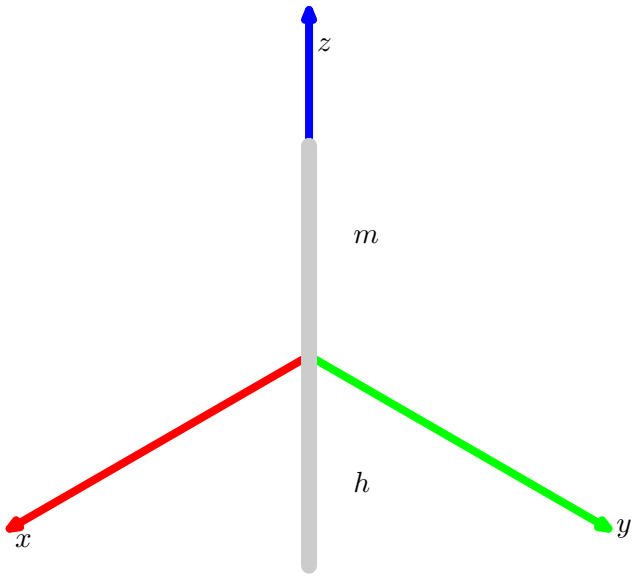


Figura 2: Varilla

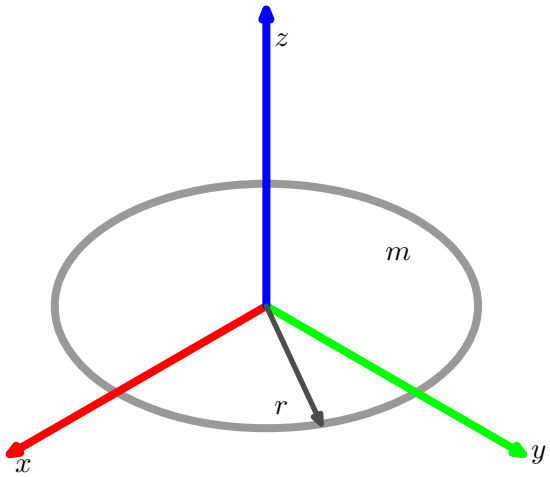


Figura 3: Aro

- la matriz de rotación entre unos ejes fijos y unos ejes ligados a la varilla situados en su centro de masa
- la masa de la misma
- el radio del aro

Un ejemplo de definición de aro sería:

```
a1 := [aro,xgvar,rotp,m,r] ;
```

En este caso el eje z del aro es perpendicular al mismo (ver figura 3).

4.4. Disco

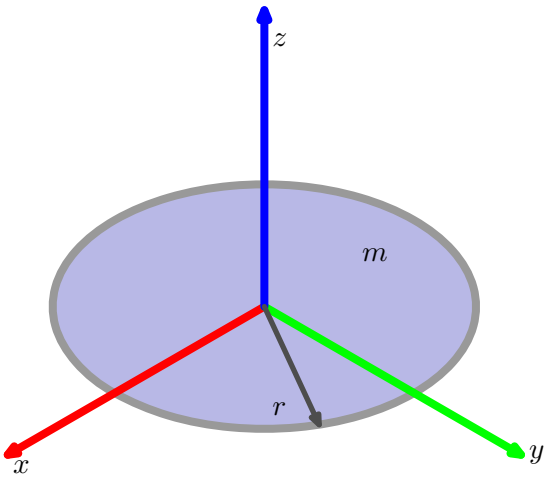


Figura 4: Disco

Es idéntica al aro cambiando el texto aro por disco.
Un ejemplo de disco sería:

```
d2 := [disco,xgvar,rotp,m,r] ;
```

Puede verse un esquema en la figura 4

4.5. Semidisco

Es similar al aro cambiando el texto aro por semidisco pero teniendo en cuenta que la vriable xgvar representa la posición del centro de masa del semidisco.

Un ejemplo de semidisco sería:

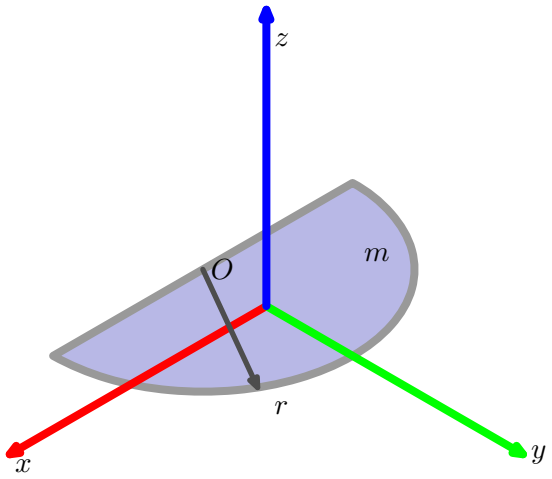


Figura 5: Semidisco

```
d2 := [semidisco,xgvar,rotp,m,r] ;
```

Puede verse un esquema en la figura 5

4.6. Semiaro

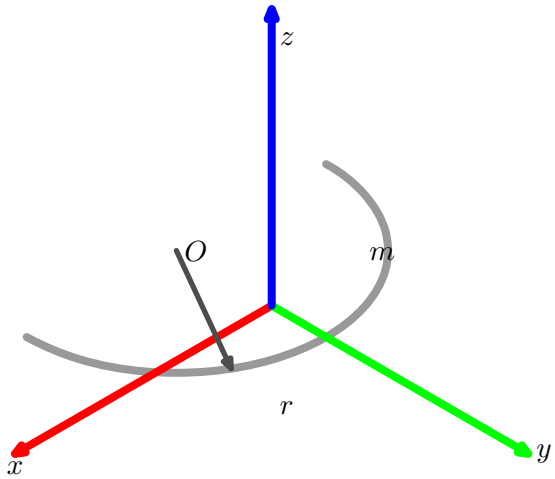


Figura 6: Semiaro

Es simlar al semidsco.
Un ejemplo de semiaro sería:

```
d2 := [semiaro,xgvar,rotp,m,r] ;
```

Igual que en el semidisco hay que situar el centro de masa del semiaro. Puede verse un esquema en la figura 6

4.7. Esfera

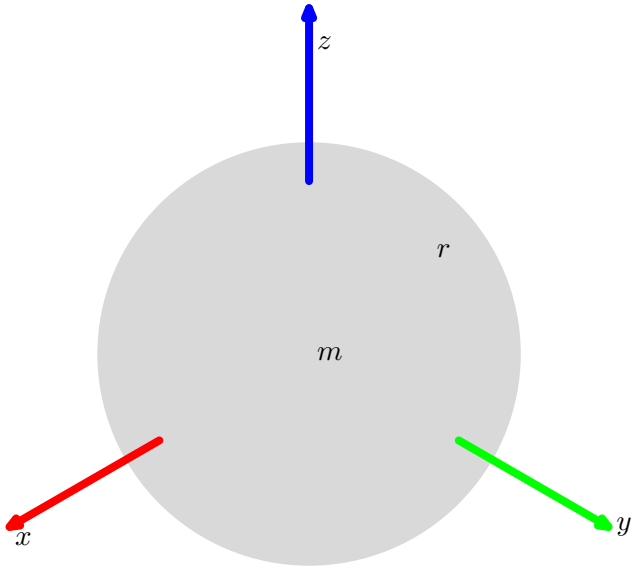


Figura 7: Esfera

Una esfera se define dando los siguientes elementos:

- Coordenadas de su centro de masa (tipo de variable lista)
- la matriz de rotación entre unos ejes fijos y unos ejes ligados a la esfera situados en su centro de masa
- la masa de la misma
- longitud del radio

Un ejemplo de esfera sería:

```
c1 := [esfera,xgvar,rotp,m,r] ;
```

(Ver figura 7)

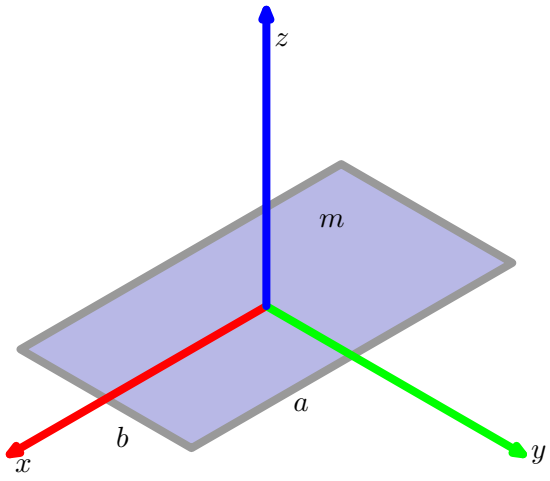


Figura 8: Rectángulo

4.8. Rectángulo

Un rectángulo se define dando los siguientes elementos:

- Coordenadas de su centro de masa (tipo de variable lista)
- la matriz de rotación entre unos ejes fijos y unos ejes ligados al rectángulo situados en su centro de masa
- la masa del mismo
- longitud de la arista según el eje x
- longitud de la arista según el eje y

Un ejemplo de rectángulo sería:

```
c1 := [rectangulo,xgvar,rotp,m,a,b] ;
```

Los ejes del rectángulo son tales que los ejes x e y son paralelos a sus aristas y situados en el plano del rectángulo, mientras que el eje z es perpendicular por su centro (haciendo que el triedro sea directo) (Ver figura 8)

4.9. Hexaedro

Un hexaedro se define dando los siguientes elementos:

- Coordenadas de su centro de masa (tipo de variable lista)
- la matriz de rotación entre unos ejes fijos y unos ejes ligados al hexaedro situados en su centro de masa

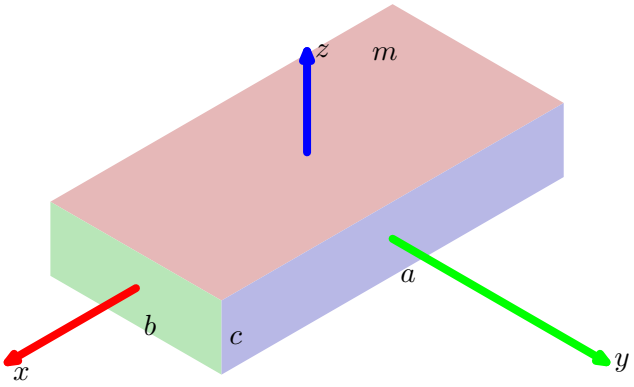


Figura 9: Hexaedro

- la masa del mismo
- longitud de la arista según el eje x (ligado al hexaedro)
- longitud de la arista según el eje y (ligado al hexaedro)
- longitud de la arista según el eje z (ligado al hexaedro)

Un ejemplo de hexaedro sería:

```
c1 := [hexaedro,xgvar,rotp,m,a,b,c] ;
```

Los ejes del hexaedro son paralelos a sus aristas (Ver figura 9)

4.10. Cilindro

El cilindro es similar al hexaedro excepto que en vez de dar como datos geométricos las longitudes de los tres lados, se dan el radio y la altura.

Un ejemplo de cilindro sería:

```
a1 := [cilindro,xgvar,rotp,m,r,h] ;
```

Los ejes del cilindro son tales que el eje z va según el eje del cilindro (Véase figura 10).

4.11. Cono

Se define del mismo modo que el cilindro. Recuérdese que las coordenadas cartesianas que se dan son las del centro de masa del cono.

Un ejemplo:

```
c1 := [cono,xgvar,rotp,m,r,h] ;
```

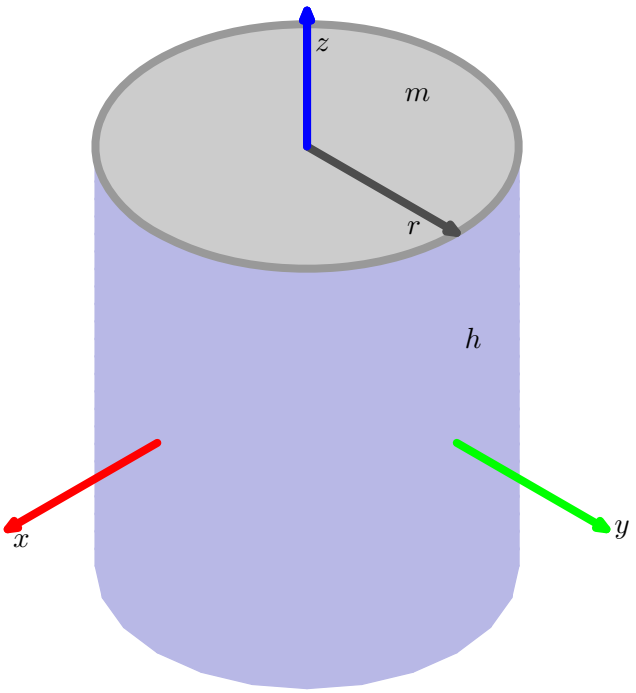


Figura 10: Cilindro

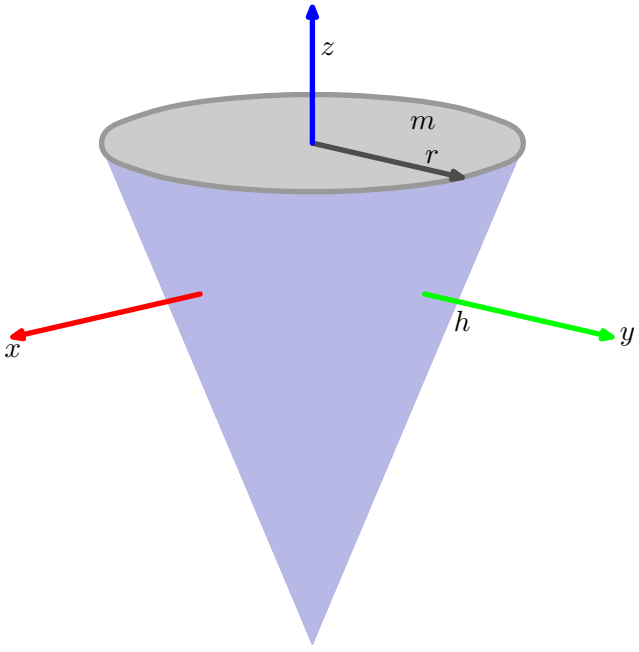


Figura 11: Cono

En este caso el eje z del cono va según su eje y con z creciente según aumenta la sección del cono (Figura 11).

4.12. Muelle

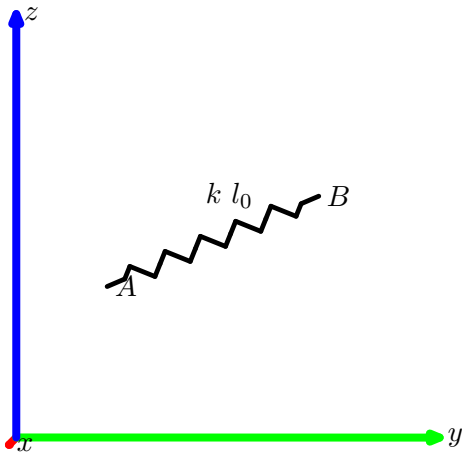


Figura 12: Muelle

El muelle se define dando los siguientes valores :

- Coordenadas de un extremo (lista)
- Coordenadas del otro extremo (lista)
- Constante del resorte
- longitud natural

Un esquema del mismo puede verse en la figura 12.
Ejemplo:

```
c1 := [muelle,x1,x2,k,l0] ;
```

4.13. Subsistemas

Como uno de los componentes disponibles existe uno especial que permite definir sistemas de referencia auxiliares sobre los que definir nuevos componentes.

De este modo se puede construir una jerarquía de objetos utilizando los sistemas auxiliares que se deseen.

El subsistema se define dando los siguientes valores :

- Las coordenadas del nuevo origen (tipo de variable lista)

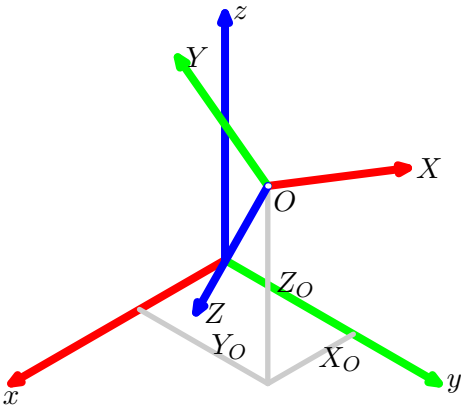


Figura 13: Subsistema2

- La matriz de rotación que permite pasar de los ejes originales a los nuevos ejes.
- Una lista de objetos que van referidos al nuevo sistema

Puede verse un sistema auxiliar en la figura 13 y un ejemplo de como se definiría a continuación:
Ejemplo:

```
s1 := [subsistema2,x0,rot,[obj1,obj2]] ;
```

Obsérvese que la la palabra clave usada es subsistema2.
Entre los componentes auxiliares están los siguientes:

4.14. Ángulo

Permite representar ángulos menores de 180°. Los valores necesarios para definirlo son tres puntos, definidos mediante sus coordenadas, y un radio para trazar el arco.

- Coordenadas de un extremo del ángulo (tipo de variable lista)
- Coordenadas del vértice del ángulo (tipo de variable lista)
- Coordenadas del otro extremo del ángulo (tipo de variable lista)
- Radio del arco dibujado.

Ejemplo:

```
a3 := [angulo,x1,x2,r] ;
```

Ver ilustración en figura 14

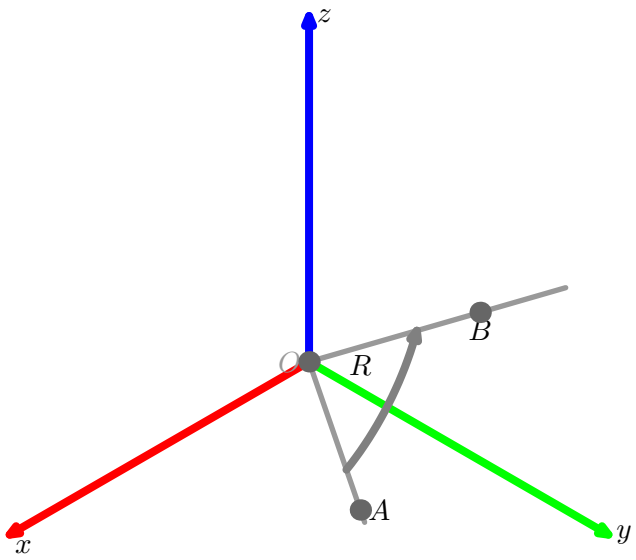


Figura 14: Ángulo

4.15. Segmento

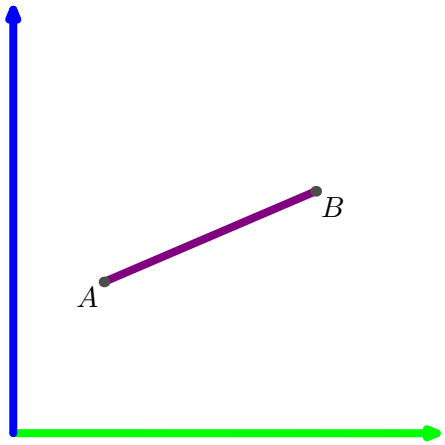


Figura 15: Segmento

Permite representar segmentos de longitud cualquiera. Puede utilizarse para representar ejes, o cualquier recta auxiliar. Los datos necesarios son

- Coordenadas de un extremo del segmento (lista)
- Coordenadas del otro extremo del segmento (lista).
- Color con el que se dibuja el segmento.

Ejemplo:

```
a3 := [angulo,[0,0,0],[3,0,0],green] ;
```

Ver figura 15

4.16. Vector

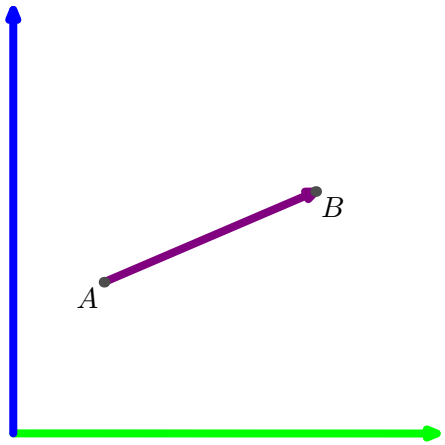


Figura 16: Vector

Permite representar vectores.
Los datos que hacen falta son:

- Punto de inicio (lista)
  - Punto del extremo (lista) o
  - Coordenadas del vector (vector)
- Color de representación del vector

Ejemplo:

```
v3 := [vector,[0,sin(alpha),-cos(alpha)],
       vector([0.,0.4*sin(alpha),0.4*cos(alpha)]),green] ;
```

Ver figura 16

4.17. Gráfico

Para los casos en que los elementos gráficos predefinidos no sean suficientes, es posible incorporar en el dibujo curvas definidas directamente mediante maple.

Estas curvas deben generarse mediante la función de maple `spacecurve`, así es posible incorporar curvas situadas donde se desee y arbitrariamente orientadas.

El componente que permite esta posibilidad se denomina `grafico` y requiere los siguientes valores:

- Posición del origen del sistema en el que está definida la curva (tipo lista)
- Rotación de los ejes de la curva
- Curva generada mediante `spacecurve`

Si quisiéramos incluir una hélice como representación gráfica en el sistema mecánico se procedería del siguiente modo:

```
gr1 := spacecurve([2*cos(t),2*sin(t),t],t=0..8) :
elgraf := [grafico,[0,0,0],rota(0,1),gr1]:
```

En la figura 17 se puede ver el resultado de el uso de este elemento.

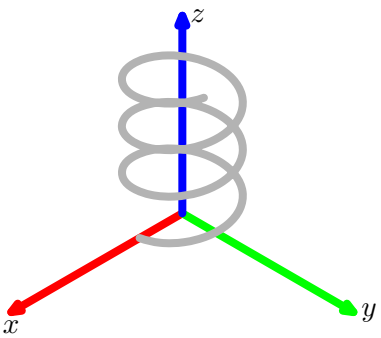


Figura 17: Gráfico

4.18. Texto

Este componente permite situar textos en posiciones arbitrarias en el espacio.

Los datos necesarios son:

- Punto de colocación (lista)
- Textos (entre dobles comillas)

t3 := [texto,[X,Y,Z],"Texto"] :
Ver figura 18

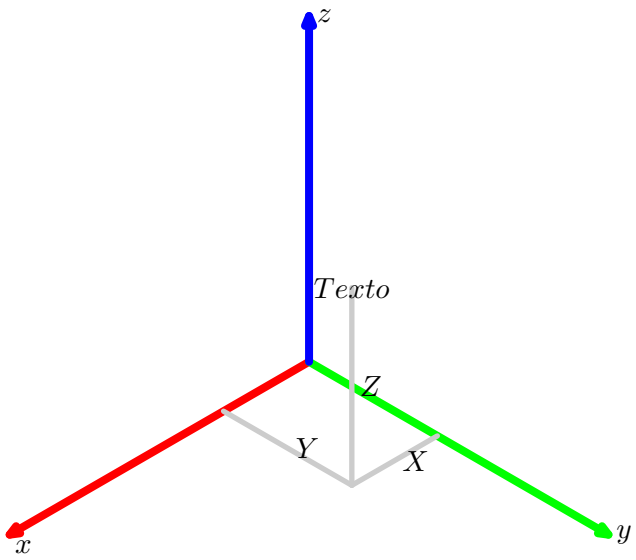


Figura 18: Texto

5. Funciones disponibles

5.1. Introducción

Previamente al uso de cualquier función de las descritas a continuación, es necesario en primer lugar cargar la libreria mecapac3d y las librerías opcionales que se desee (habitualmente `linalg`, `plots`, y `plottools`)

```
restart;
with(linalg):with(plots):with(plottools):
libname := "/opt/meca3d",libname ;
with(mecapac3d):
```

También debe definirse una variable denominada cg que contiene las coordenadas generalizadas del sistema:

```
cg := [alpha,beta] ;
```

A continuación definir los diversos componentes mecánicos y gráficos que definen el sistema y agruparlos en una variable denominada sistema de la forma siguiente:

```
sistema := [v1,a2,c3] ;
```

Las funciones disponibles tanto como ayuda para definir objetos, como para obtener resultados son las siguientes:

5.2. rota

Esta función permite obtener la matriz de rotación entre dos sistemas ortonormales al pasar de uno a otro girando respecto uno de los ejes coordenados.

Ejemplo:

```
r1 := rota(Pi/2,2) ;
```

Así se obtiene la matriz de rotación respecto al eje y al girar $\pi/2$.

5.3. fT

Esta función permite obtener la energía cinética del sistema y se usa de la siguiente forma:

```
T := fT(sistema) ;
```

En este caso la energía ha sido almacenada en la variable T.

Debe obtenerse una función de las coordenadas generalizadas y sus derivadas. Éstas van representadas con el nombre de la coordenada concatenada con un 1 que indica la primera derivada (velocidad generalizada).

5.4. fV

Esta función nos da la energía potencial del sistema y se usa de forma similar a la energía cinética:

```
V := fV(sistema) ;
```

En este caso la energía potencial ha sido almacenada en la variable V. Actualmente las energías potenciales que se tienen en cuenta son las de origen gravitatorio (en función de la variable g) y asociada a la variable z y las de los muelles.

5.5. ec_lag

Esta función nos devuelve las ecuaciones de Lagrange del sistema. Es necesario que la Lagrangiana del sistema esté en una variable denominada L y que las coordenadas generalizadas estén en cg.

Ejemplo:

```
ecua := ec_lag()  ;
```

Las ecuaciones quedan almacenadas en la variable ecua.

5.6. ec_lagr

Esta función nos devuelve las ecuaciones de Lagrange del sistema en el caso de tratarse de un sistema con ligaduras anholónomas. Previamente deben haberse calculado las reacciones asociadas a las ligaduras mediante la función Fc().

Ejemplo:

```
ecua := ec_lagr()  ;
```

Las ecuaciones quedan almacenadas en la variable ecua.

5.7. ec_l

Esta función nos da la ecuación de Lagrange correspondiente a una coordenada generalizada. Si queremos obtener todas las ecuaciones, un modo alternativo al uso de la función `ec_lag` es utilizando la instrucción de Maple map.

```
ecua  := map(ec_l,cg)  ;
```

Las ecuaciones quedan almacenadas en la variable ecua.

5.8. intprim

Esta función nos da las integrales primeras asociadas a las posibles coordenadas cíclicas.

Quedan expresadas con el convenio anteriormente indicado sobre las derivadas de las coordenadas generalizadas.

```
integprim   := intprim()  ;
```

5.9. intjacobi

Para los casos en que exista la integral de jacobi, esta función nos devuelve su expresión. En caso de que no exista no devuelve nada.

La expresión que devuelve está formulada sin representar la dependencia del tiempo ni en las coordenadas generalizadas ni en las velocidades generalizadas.

Estas últimas se representan añadiendo un 1 a las coordenadas generalizadas.

Se calcula partiendo de la expresión de la lagrangiana y

Ejemplo:

```
intjac   := intjacobi()  ;
```

5.10. Fc

Esta función nos da las reacciones asociadas a las ligaduras anholónmas proyectadas según las coordenadas generalizadas. Se obtiene una lista con tantos elementos como coordenadas. En el caso de existir diversas ligaduras, según cada coordenada se obtiene la suma de las reacciones asociadas a cada ligadura según esa coordenada. Anteriormente deben haberse definido las ligaduras mediante la variable *Phi*.

Quedan expresadas con el convenio anteriormente indicado sobre las derivadas de las coordenadas generalizadas.

```
reacc   := Fc()  ;
```

5.11. intjacobi

5.12. fint

Esta función integra de forma numérica las ecuaciones del movimiento. Es necesario, por tanto, que no existan parámetros, sin asignar valores numéricos en las mismas. El cálculo de las ecuaciones del movimiento, las integrales primeras, etc..., puede realizarse con parámetros, pero la integración numérica, obviamente, no.

El argumento de la función es una lista con los valores de las coordenadas y velocidades inciales, ordenadas según las coordenadas generlizadas.

Ejemplo:

```
res   := fint([0.1,0.2,0.3,0.4]) ;
```

En este ejemplo se trata de integrar un sistema con dos grados de libertad y valores iniciales 0,1 para la primera coordenada, 0,2 para la velocidad generalizada asociada a la primera coordenada, 0,3 valor de la segunda coordenada y 0,4 velocidad inicial de la segunda coordenada.

5.13. fintr

Esta función es similar a fint. La diferencia es que fintr es la función que se debe usar para sistemas con ligaduras anholónomas.

El argumento de la función es una lista con los valores de las coordenadas y velocidades inciales, ordenadas según las coordenadas generlizadas.

Ejemplo:

```
res   := fintr([0.1,0.2,0.3,0.4]) ;
```

En este ejemplo se trata de integrar un sistema con dos grados de libertad y valores iniciales 0,1 para la primera coordenada, 0,2 para la velocidad generalizada asociada a la primera coordenada, 0,3 valor de la segunda coordenada y 0,4 velocidad inicial de la segunda coordenada.

5.14. f_lin

Esta función permite linealizar las ecuaciones del movimiento. Generalmente las ecuaciones diferenciales obtenidas no son ecuaciones lineales. En muchos casos para estudiar propiedades del movimiento, o para simplificar la solución del problema se recurre a la linealización.

Un ejemplo clásico es el del estudio de las oscilaciones respecto a una posición de equilibrio.

La función se aplica a una ecuación y en función de las coordenadas generalizadas conocidas (cg) se linealiza la ecuación. Las ecuaciones a linealizar suelen ser las obtenidas a través de la función `ec_l` por lo que la dependencia del tiempo debe estar expresada de forma explícita. En el resultado se mantiene esta forma.

Si se desea linealizar un conjunto de ecuaciones se debe utilizar la función auxiliar de maple `map`.

Ejemplo:

```
ecua := ec_lag() :
ecl1 := f_lin(ecua[1]) :
ecl := map(f_lin,ecua) :
```

5.15. fG

Esta función permite representar posiciones de un sistema en función de las coordenadas generalizadas. Estas se dan como argumento en una lista y evaluadas en coma flotante.

Ejemplo:

```
fG([evalf(Pi/3),0.6]) ;
```

En este ejemplo se obtiene la representación gráfica del sistema para los valores $\pi/3$ y 0,6 de las coordenadas generalizadas.

Al ser esta función exclusivamente de representación, no depende de las magnitudes dinámicas del sistema, es decir, no es necesario haber calculado las ecuaciones, ni la Lagrangiana, ni las energías.

Es una función util para comprobar que la definición cinemática del sistema en función de las coordenadas generalizadas es la correcta.

Si en el sistema existen componentes cuya representación dependa del tiempo o de las derivadas, esta función no se puede aplicar.

5.16. dibu3

Esta función genera animaciones del movimiento del sistema en función de la resolución numérica del mismo (que debe estar almacenada en la variable res).

Utiliza dos argumentos, el tiempo a integrar (representar) y el número de cuadros a utilizar.

Ejemplo:

```
dibu3(2,30) ;
```

En este caso se obtiene una animación que representa 2 segundos y utiliza 30 imágenes para obtener la misma.

5.17. dibu2

Es una función similar a dibu3 aunque bastante más lenta. Su ventaja radica en que podemos representar elementos cuyas posiciones dependan de las derivadas de las coordenadas o del tiempo explicitamente.

Es el caso de sistemas en los que se utilicen coordenadas móviles, como por ejemplo, en el caso en el que haya algún tipo de movimiento impuesto.

6. Ejemplos

6.1. Tiro parabólico

Este ejemplo consiste en el tiro parabólico de una partícula de masa m según el plano oyz.

A continuación figura el código de `maple` con las órdenes usadas y los resultados obtenidos.

Tiro Parab´olico

```
> restart:
> with(linalg):with(plots):with(plottools):

Warning, the protected names norm and trace have been redefined and
unprotected

Warning, the name changecoords has been redefined

Warning, the name arrow has been redefined

> libname := "/meca3d",libname :
> with(mecapac3d):
```

Definicion de coordenadas generalizadas

```
> cg := [y,z] :
```

Definicion de objetos (una particula y dos segmentos)

```
> m1 := [punto,0,y,z,m]:
> s1 := [segmento,[0,0,0],[0,6,0],green]:
> s2 := [segmento,[0,0,0],[0,0,2],blue]:

> sistema := [m1,s1,s2] :
```

```
> T := fT(sistema);
```

$$T := \frac{1}{2}\, m\, (y1^2 + z1^2)$$

```
> V := fV(sistema);
```

$$V := m\, g\, z$$

```
> L := T - V ;
```

$$L := \frac{1}{2}\, m\, (y1^2 + z1^2) - m\, g\, z$$

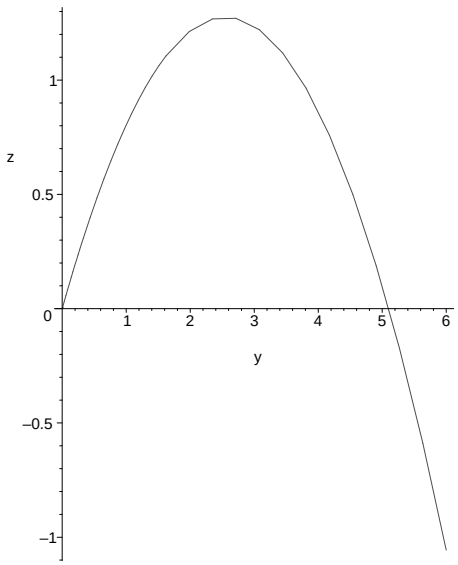
```
> ecua := ec_lag() ;
```

$$ecua := [m\, (\tfrac{\partial^2}{\partial t^2}\, \mathrm{y}(t)),\ m\, (\tfrac{\partial^2}{\partial t^2}\, \mathrm{z}(t)) + m\, g]$$

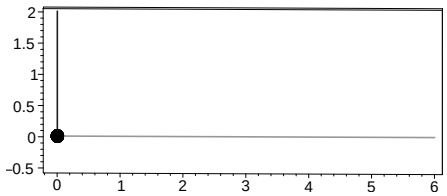
```
> m := 1 : g := 9.8 :
> res := fint([0,5,0,5]):
> Lmax := 2*10*5/9.8;
```

$$Lmax := 10{,}20408163$$

```
> odeplot(res,[y(t),z(t)],0..1.2);
```




```
> dibu3(1.1,10);
```



6.2. Péndulo simple

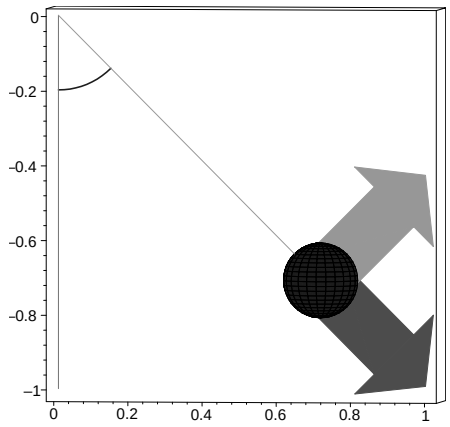
Este es el péndulo simple con una masa puntual en el plano *oyz*.

P´endulo simple

```
> restart:
> with(linalg):with(plots):with(plottools):
```

```
Warning, the protected names norm and trace have been redefined and
unprotected

Warning, the name changecoords has been redefined

Warning, the name arrow has been redefined
```

```
> libname := "/meca3d",libname :
> with(mecapac3d):
> cg := [alpha] :
> m1 := [punto,0,sin(alpha),-cos(alpha),m]:
> a2 := [angulo,[0,0,-1],[0,0,0],[0,sin(alpha),-cos(alpha)],0.2]:
> v2 :=
> [vector,[0,sin(alpha),-cos(alpha)],vector([0.,0.4*sin(alpha),-0.4*cos(
> alpha)]),red]:
> v1 :=
> [vector,[0,sin(alpha),-cos(alpha)],vector([0.,0.4*cos(alpha),0.4*sin(a
> lpha)]),green]:

> sistema := [m1,a2,v1,v2] :

> fG([evalf(Pi/4)]);
```




```
> T := fT(sistema);
```

$$T := \frac{1}{2}\, m \left(\cos(\alpha)^2\, \alpha 1^2 + \sin(\alpha)^2\, \alpha 1^2\right)$$

```
> V := fV(sistema);
```

$$V := -m\, g \cos(\alpha)$$

```
> L := simplify(T - V) ;
```

$$L := \frac{1}{2}\, m \left(\alpha 1^2 + 2\, g \cos(\alpha)\right)$$

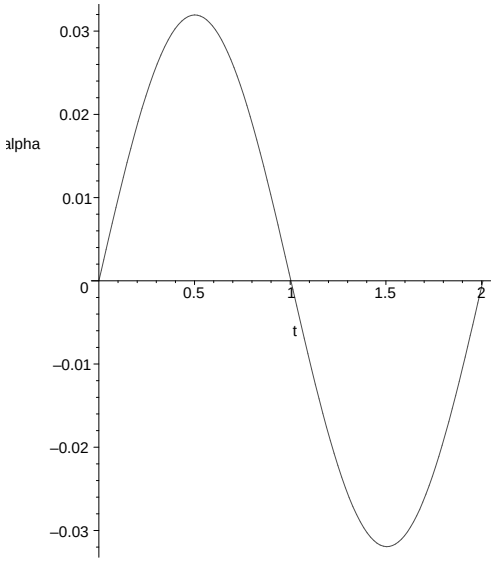
```
> ecua := map(ec_l,cg) ;
```

$$ecua := [m\,(\frac{\partial^2}{\partial t^2}\,\alpha(t)) + m\,g\sin(\alpha(t))]$$

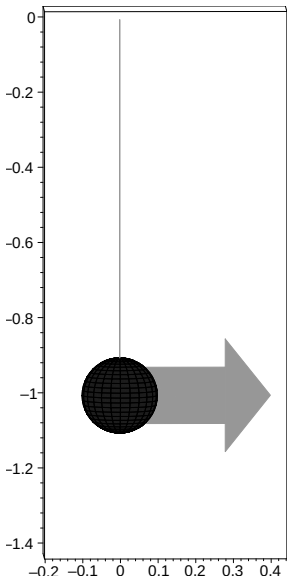
```
> m := 1 : g := 9.8 :
> res := fint([0,0.1]):
> periodo := evalf(2*Pi*sqrt(1/9.8));
```

$$periodo := 2{,}007089923$$

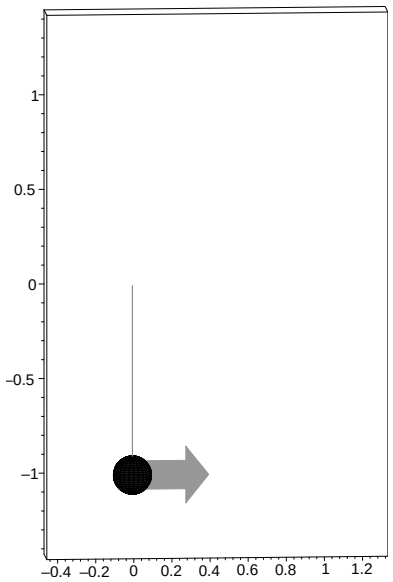
```
> odeplot(res,[t,alpha(t)],0..2.007);
```




```
> dibu3(2.1,10);
```




```
> res := fint([0,sqrt(4*9.8)]):
> dibu3(4.1,20);
```



6.3. Péndulo con varilla

Este péndulo es similar al anterior, pero se introduce una varilla (sin masa) con el fin de observar la forma como se situa la varilla. (centro de masa y rotación).

Péndulo con una varilla de masa m y longitud 1

```
> restart:
> with(linalg):with(plots):with(plottools):
```

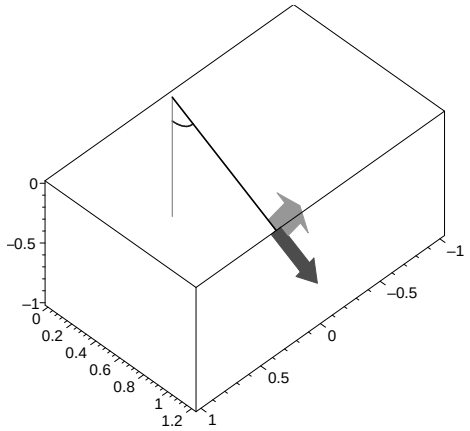
```
Warning, the protected names norm and trace have been redefined and
unprotected

Warning, the name changecoords has been redefined

Warning, the name arrow has been redefined
```

```
> libname := "/meca3d",libname :
> with(mecapac3d):
> cg := [alpha] :
> xg := [0,sin(alpha)/2,-cos(alpha)/2]:
> r1 := rota(alpha,1);
```

$$r1 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

```
> va := [varilla,xg,r1,m,1]:
> a2 := [angulo,[0,0,-1],[0,0,0],[0,sin(alpha),-cos(alpha)],0.2]:
> v2 :=
> [vector,[0,sin(alpha),-cos(alpha)],vector([0.,0.4*sin(alpha),-0.4*cos(
> alpha)]),red]:
```

```
> v1 :=
> [vector,[0,sin(alpha),-cos(alpha)],vector([0.,0.4*cos(alpha),0.4*sin(a
> lpha)]),green]:

> sistema := [va,a2,v1,v2] :

> fG([evalf(Pi/3)]);
```




```
> T := fT(sistema);
```

$$T := \frac{1}{2}\, m\, (\frac{1}{4}\cos(\alpha)^2\, \alpha 1^2 + \frac{1}{4}\sin(\alpha)^2\, \alpha 1^2) + \frac{1}{24}\, \alpha 1^2\, m$$

```
> V := fV(sistema);
```

$$V := -\frac{1}{2}\, m\, g\cos(\alpha)$$

```
> L := simplify(T - V );
```

$$L := \frac{1}{6}\, m\, (\alpha 1^2 + 3\, g\cos(\alpha))$$

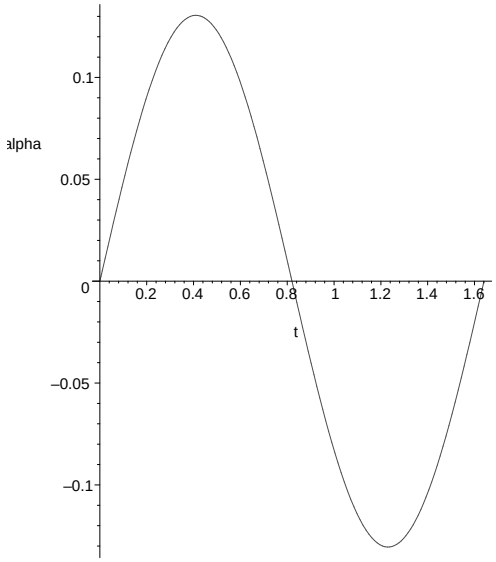
```
> ecua := map(ec_l,cg) ;
```

$$ecua := [\frac{1}{3}\, (\frac{\partial^2}{\partial t^2}\, \alpha(t))\, m + \frac{1}{2}\, m\, g\sin(\alpha(t))]$$

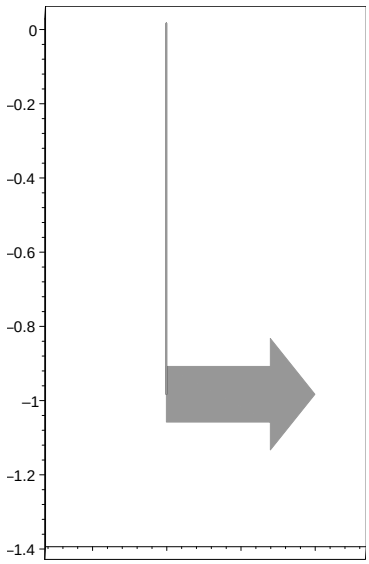
```
> m:=10 : g := 9.8:
> res := fint([0,0.5]):
> periodo := evalf(2*Pi*sqrt(0.33333333/4.9));
```

$$periodo := 1{,}638782052$$

```
> odeplot(res,[t,alpha(t)],0..1.64);
```




```
> dibu3(1.641,20);
```



6.4. Disco rodando sobre plano inclinado

Se trata de un disco que rueda sobre un plano inclinado. Se toma como parámetro para definir el sistema la distancia desde el punto de contacto entre disco y plano hasta el comienzo del plano inclinado.

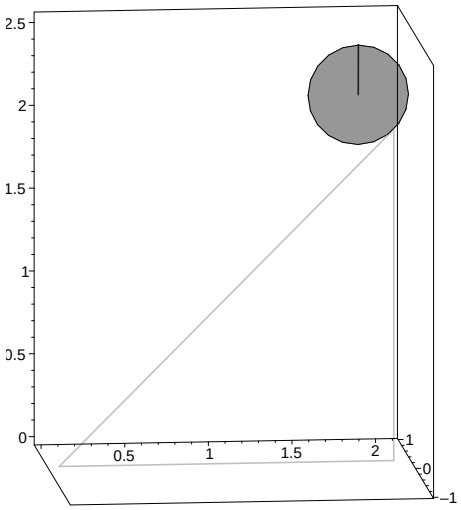
Disco rodando sobre plano inclinado

```
> restart:
> with(linalg):with(plots):with(plottools):

Warning, the protected names norm and trace have been redefined and
unprotected
```

```
Warning, the name changecoords has been redefined

Warning, the name arrow has been redefined
> libname := "/meca3d",libname :
> with(mecapac3d):
```

Disco rodando sobre plano inclinado

```
> cg := [s] :
> h := 2 : alpha := evalf(Pi/4) :r := 0.3:
> r1 := [segmento,[0,0,0],[0,h/tan(alpha),0],grey] :
> r2 := [segmento,[0,h/tan(alpha),0],[0,h/tan(alpha),h],grey]
:
> r3 := [segmento,[0,0,0],[0,h/tan(alpha),h],grey] :
> xg := [0,s*cos(alpha)-r*sin(alpha),s*sin(alpha)+r*cos(alpha)]:
> rot1 := rota(Pi/2,2):
> rot2 := rota(-s/r,1):
> rottot := evalm(rot2 &* rot1):
> a1 := [disco,xg,rottot,m,r]:
> sistema := [a1,r1,r2,r3] :
> fG([h/sin(alpha)]);
```




```
> T := fT(sistema);
```

$$T := ,7500000000\, s1^2\, m$$

```
> V := fV(sistema);
```

$$V := m\, g\, (,7071067813\, s + ,2121320343)$$

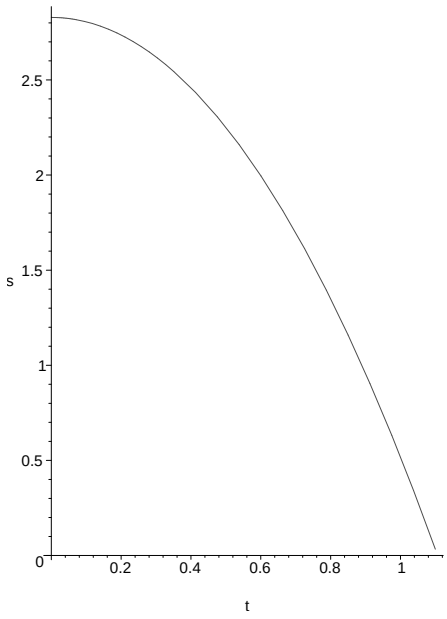
```
> L := simplify(T - V );
```

$$L := ,7500000000\, s1^2\, m - ,7071067813\, m\, g\, s - ,2121320343\, m\, g$$

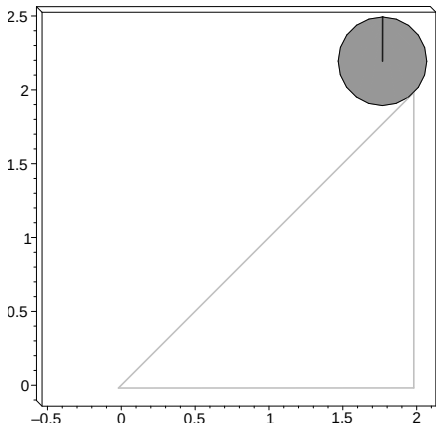
```
> ecua := ec_lag() ;
```

$$ecua := [1{,}500000000\,(\frac{\partial^2}{\partial t^2}\,\mathrm{s}(t))\,m + {,}7071067813\,m\,g]$$

```
> m:=10 :g:=9.8:
> res := fint([2/sin(alpha),0.]):
> odeplot(res,[t,s(t)],0..1.10);
```




```
> dibu3(1.10,20);
```



6.5. Péndulo esférico

Es un péndulo formado por una masa puntual y una varilla sin masa. Se integran dos movimientos, correspondientes al movimiento del extremo en un plano vertical o en un plano horizontal.

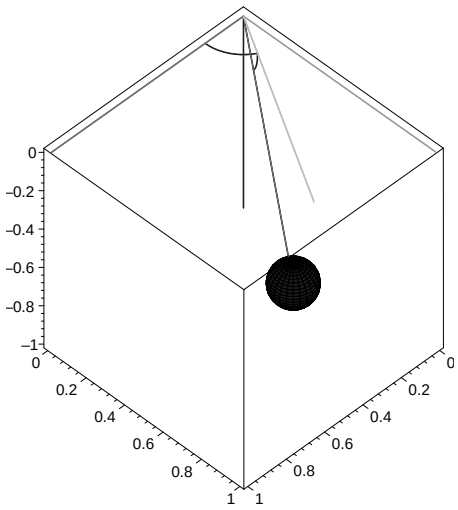
P´endulo esf´erico

```
> restart:
```

```
> with(linalg):with(plots):with(plottools):

Warning, the protected names norm and trace have been redefined and
unprotected

Warning, the name changecoords has been redefined

Warning, the name arrow has been redefined

> libname := "/meca3d",libname :

> with(mecapac3d):
```

Péndulo con una varilla sin masa y longitud l en 3 dimensiones

```
> cg := [theta,phi] :

> xg := [cos(phi)*cos(theta)/2,cos(phi)*sin(theta)/2,-sin(phi)/2]:

> r1 := rota(-Pi/2-phi,1):r2 := rota(-Pi/2+theta,3):

> rtot := evalm(r2 &*r1):

> va := [varilla,xg,rtot,0,1]:

> m1 := [punto,cos(phi)*cos(theta),cos(phi)*sin(theta),-sin(phi),1]:

> s1 := [segmento,[0,0,0],[1,0,0],red] :

> s2 := [segmento,[0,0,0],[0,1,0],green] :

> s3 := [segmento,[0,0,0],[0,0,-1],blue] :

> s4 := [segmento,[0,0,0],[cos(theta),sin(theta),0],grey] :

> a1 := [angulo,[1,0,0],[0,0,0],[cos(theta),sin(theta),0],0.2]:
> a2 :=
> [angulo,[cos(phi)*cos(theta),cos(phi)*sin(theta),-sin(phi)],[0,0,0],[c
> os(theta),sin(theta),0],0.2]:

> sistema := [va,m1,s1,s2,s3,s4,a1,a2] :


> fG([evalf(Pi/3),evalf(Pi/4)]);
```



```
> T := fT(sistema);
```

$$T := \frac{1}{2}\left(-\sin(\phi)\,\phi 1\cos(\theta) - \cos(\phi)\sin(\theta)\,\theta 1\right)^2 + \frac{1}{2}\left(-\sin(\phi)\,\phi 1\sin(\theta) + \cos(\phi)\cos(\theta)\,\theta 1\right)^2 + \frac{1}{2}\cos(\phi)^2\,\phi 1^2$$

```
> V := fV(sistema);
```

$$V := -g\sin(\phi)$$

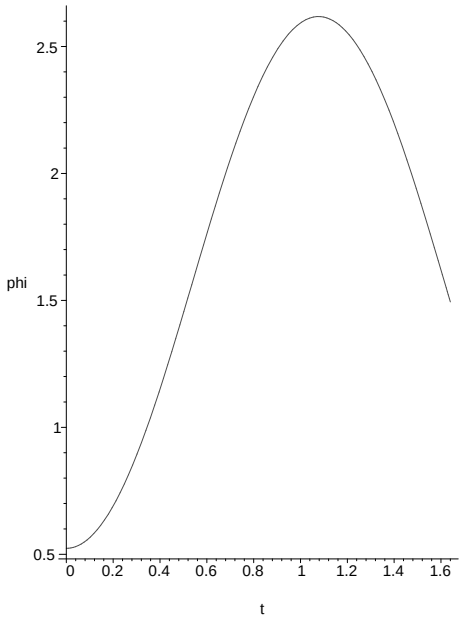
```
> L := simplify(T - V );
```

$$L := \frac{1}{2}\cos(\phi)^2\,\theta 1^2 + \frac{1}{2}\,\phi 1^2 + g\sin(\phi)$$

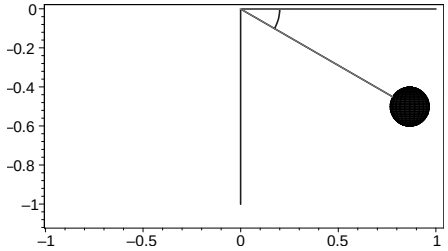
```
> ecua := map(ec_l,cg) ;
```

$$ecua := [-2\cos(\phi(t))\left(\tfrac{\partial}{\partial t}\,\theta(t)\right)\sin(\phi(t))\left(\tfrac{\partial}{\partial t}\,\phi(t)\right) + \cos(\phi(t))^2\left(\tfrac{\partial^2}{\partial t^2}\,\theta(t)\right),$$
$$\left(\tfrac{\partial^2}{\partial t^2}\,\phi(t)\right) + \cos(\phi(t))\left(\tfrac{\partial}{\partial t}\,\theta(t)\right)^2\sin(\phi(t)) - g\cos(\phi(t))]$$

```
> g:=9.8:
> res := fint([0.,0.,Pi/6,0.]):
> odeplot(res,[t,phi(t)],0..1.64);
```

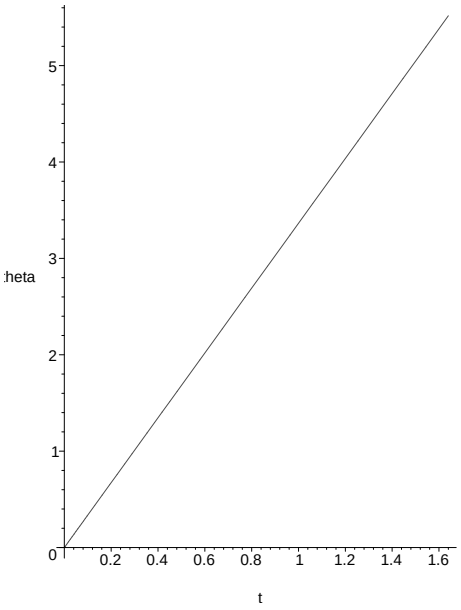



```
> dibu3(1.8,40);
```

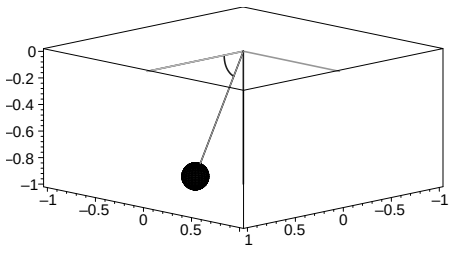


Movimiento circunferencial

```
> alpha0 := Pi/3 : theta0 := sqrt(g/sin(alpha0)):
> res := fint([0.,theta0,alpha0,0.]):
> odeplot(res,[t,theta(t)],0..1.64);
```




```
> dibu3(1.8,40);
```



6.6. Disco rodando sobre una catenaria

Disco rodando sobre una catenaria con uso de elementos gráficos arbitrarios y representación de la velocidad del centro del disco.

Disco que rueda sobre una catenaria

```
> restart ;
> with(linalg):with(plots):with(plottools):

Warning, the protected names norm and trace have been redefined and
unprotected

Warning, the name changecoords has been redefined

Warning, the name arrow has been redefined
```

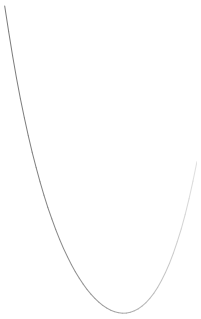
```
> libname := "/meca3d",libname :
> with(mecapac3d):

> cg := [x] ;
```

$$cg := [x]$$

```
> a:=2;
```

$$a := 2$$

```
> gr1 := spacecurve([0,t,a*cosh(t/a)],t=-4..4):
> display(gr1);
```



```
> acota := proc(u,v,w,ut,vt,wt)
> _v1 := [segmento,[u,v,0],[0,v,0],grey,"_v1"] :
> _v2 := [segmento,[u,v,0],[u,0,0],grey,"_v1"] :
> _v3 := [segmento,[u,v,0],[u,v,w],grey,"_v1"] :
> _tx := [texto,[u/2,v,0],ut] :
> _ty := [texto,[u,v/2,0],vt] :
> _tz := [texto,[u,v,w/2],wt] :
> _p1 := [punto,u,v,w,0]:
> _p1,_v1,_v2,_v3,_tx,_ty,_tz:
> end:

Warning, `_v1` is implicitly declared local to procedure `acota`

Warning, `_v2` is implicitly declared local to procedure `acota`

Warning, `_v3` is implicitly declared local to procedure `acota`

Warning, `_tx` is implicitly declared local to procedure `acota`

Warning, `_ty` is implicitly declared local to procedure `acota`

Warning, `_tz` is implicitly declared local to procedure `acota`

Warning, `_p1` is implicitly declared local to procedure `acota`

> elgraf := [grafico,[0,0,0],rota(0,1),gr1]:
```

```
> triedro := proc(long)
> _v1 := [vector,[0,0,0],[long,0,0],red,"_v1"] :
> _v2 := [vector,[0,0,0],[0,long,0],green,"_v1"] :
> _v3 := [vector,[0,0,0],[0,0,long],blue,"_v1"] :
> _tx := [texto,[long,0,0],"x"] :
> _ty := [texto,[0,long,0],"y"] :
> _tz := [texto,[0,0,long],"z"] :
> _v1,_v2,_v3,_tx,_ty,_tz:
> end :
```

```
Warning, `_v1` is implicitly declared local to procedure `triedro`

Warning, `_v2` is implicitly declared local to procedure `triedro`

Warning, `_v3` is implicitly declared local to procedure `triedro`

Warning, `_tx` is implicitly declared local to procedure `triedro`

Warning, `_ty` is implicitly declared local to procedure `triedro`

Warning, `_tz` is implicitly declared local to procedure `triedro`
```

Inclusi´on del disco de radio r que rueda sobre la catenaria

```
> d1 :=
> [disco,[0,x-r*sin(arctan(sinh(x/a))),a*cosh(x/a)+r*cos(arctan(sinh(x/a
> )))],evalm(rota(-a*sinh(x/a),1)&*rota(Pi/2,2)),m,r,"d1"]:
```

Velocidad del centro del disco

```
> vel :=
> vector(3,map(u->diff(u,t),[0,x(t)-r*sin(arctan(sinh(x(t)/a))),a*cosh(x
> (t)/a)+r*cos(arctan(sinh(x(t)/a)))]));
```

$$vel := \left[0,\ (\tfrac{\partial}{\partial t}\,\mathrm{x}(t)) - \frac{1}{2}\,\frac{r\cosh(\frac{1}{2}\,\mathrm{x}(t))\,(\frac{\partial}{\partial t}\,\mathrm{x}(t))}{\sqrt{1+\sinh(\frac{1}{2}\,\mathrm{x}(t))^2}} + \frac{\frac{1}{2}\,r\sinh(\frac{1}{2}\,\mathrm{x}(t))^2\cosh(\frac{1}{2}\,\mathrm{x}(t))\,(\frac{\partial}{\partial t}\,\mathrm{x}(t))}{(1+\sinh(\frac{1}{2}\,\mathrm{x}(t))^2)^{(3/2)}},\right.$$

$$\left.\sinh(\frac{1}{2}\,\mathrm{x}(t))\,(\frac{\partial}{\partial t}\,\mathrm{x}(t)) - \frac{1}{2}\,\frac{r\sinh(\frac{1}{2}\,\mathrm{x}(t))\cosh(\frac{1}{2}\,\mathrm{x}(t))\,(\frac{\partial}{\partial t}\,\mathrm{x}(t))}{(1+\sinh(\frac{1}{2}\,\mathrm{x}(t))^2)^{(3/2)}}\right]$$

```
> s1 :=
> map(fi_t,[vector,[0,x-r*sin(arctan(sinh(x/a))),a*cosh(x/a)+r*cos(arcta
> n(sinh(x/a)))],eval(vel),red]);
```

$$s1 := \left[vector,\ \left[0,\ x - \frac{r \sinh(\frac{1}{2}x)}{\sqrt{\%1}},\ 2\cosh(\frac{1}{2}x) + \frac{r}{\sqrt{\%1}} \right],\ \left[0,\ x1 - \frac{1}{2}\frac{r\cosh(\frac{1}{2}x)\,x1}{\sqrt{\%1}} + \frac{\frac{1}{2}r\sinh(\frac{1}{2}x)^2\cosh(\frac{1}{2}x)\,x1}{\%1^{(3/2)}},\ \sinh(\frac{1}{2}x)\,x1 - \frac{1}{2}\frac{r\sinh(\frac{1}{2}x)\cosh(\frac{1}{2}x)\,x1}{\%1^{(3/2)}} \right],\ red \right]$$

$$\%1 := 1 + \sinh(\frac{1}{2}x)^2$$

```
> m := 10 : r := 1 :
> sistema := [elgraf,triedro(5),d1,acota(0,x,a*cosh(x/a),"
> ","X","Y"),s1] :
> acota(0,x,a*cosh(x/a),"","X","Y");
```

$[punto,\ 0,\ x,\ 2\cosh(\frac{1}{2}x),\ 0],\ [segmento,\ [0,\ x,\ 0],\ [0,\ x,\ 0],\ grey,\ \text{“\_v1”}],$

$[segmento,\ [0,\ x,\ 0],\ [0,\ 0,\ 0],\ grey,\ \text{“\_v1”}],$

$[segmento,\ [0,\ x,\ 0],\ [0,\ x,\ 2\cosh(\frac{1}{2}x)],\ grey,\ \text{“\_v1”}],\ [texto,\ [0,\ x,\ 0],\ \text{“”}],$

$[texto,\ [0,\ \frac{1}{2}x,\ 0],\ \text{“X”}],\ [texto,\ [0,\ x,\ \cosh(\frac{1}{2}x)],\ \text{“Y”}]$

```
> fG([3.]);
```

```
Plotting error, non-numeric vertex definition
```

```
> T := simplify(fT(sistema));
```

$$T := \frac{5}{4}\frac{x1^2\,(-4\,\mathrm{csgn}(\cosh(\frac{1}{2}x))\cosh(\frac{1}{2}x)^2 + 1 + 6\cosh(\frac{1}{2}x)^4)}{\cosh(\frac{1}{2}x)^2}$$

```
> V := simplify(fV(sistema));
```

$$V := 10\,\frac{g\,(2\,\mathrm{csgn}(\cosh(\frac{1}{2}x))\cosh(\frac{1}{2}x)^2 + 1)\,\mathrm{csgn}(\cosh(\frac{1}{2}x))}{\cosh(\frac{1}{2}x)}$$

```
> L := T - V ;
```

$$L := \frac{5}{4}\frac{x1^2\,(-4\,\mathrm{csgn}(\cosh(\frac{1}{2}x))\cosh(\frac{1}{2}x)^2 + 1 + 6\cosh(\frac{1}{2}x)^4)}{\cosh(\frac{1}{2}x)^2} - \frac{10\,g\,(2\,\mathrm{csgn}(\cosh(\frac{1}{2}x))\cosh(\frac{1}{2}x)^2 + 1)\,\mathrm{csgn}(\cosh(\frac{1}{2}x))}{\cosh(\frac{1}{2}x)}$$

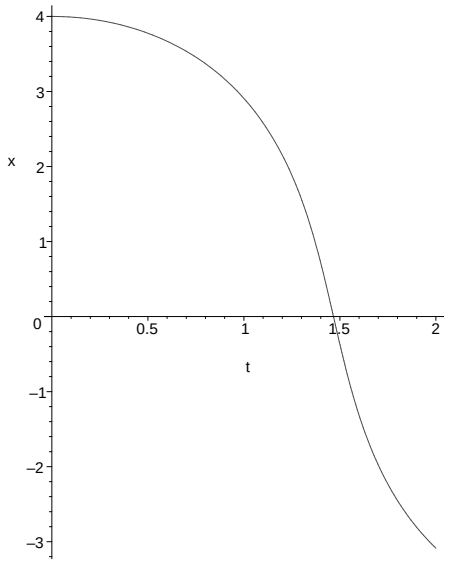
```
> ecua := simplify(ec_lag()) ;
```

$$ecua := [\frac{5}{4}(-8\,(\frac{\partial^2}{\partial t^2}\,\mathrm{x}(t))\ \%2\,\mathrm{csgn}(\cosh(\frac{1}{2}\,\mathrm{x}(t))) + 2\,(\frac{\partial^2}{\partial t^2}\,\mathrm{x}(t))\cosh(\frac{1}{2}\,\mathrm{x}(t))$$
$$+ 12\,(\frac{\partial^2}{\partial t^2}\,\mathrm{x}(t))\cosh(\frac{1}{2}\,\mathrm{x}(t))^5 - (\frac{\partial}{\partial t}\,\mathrm{x}(t))^2\sinh(\frac{1}{2}\,\mathrm{x}(t))$$
$$+ 6\,(\frac{\partial}{\partial t}\,\mathrm{x}(t))^2\sinh(\frac{1}{2}\,\mathrm{x}(t))\cosh(\frac{1}{2}\,\mathrm{x}(t))^4 - 8\,(\frac{\partial}{\partial t}\,\mathrm{x}(t))\ \%2\ \%1 + 4\,(\frac{\partial}{\partial t}\,\mathrm{x}(t))^2\ \%2\ \%1$$
$$+ 32\,g\cosh(\frac{1}{2}\,\mathrm{x}(t))^4\,\mathrm{csgn}(\cosh(\frac{1}{2}\,\mathrm{x}(t)))\ \%1 + 8\,g\ \%2\sinh(\frac{1}{2}\,\mathrm{x}(t))$$
$$+ 8\,g\ \%1\cosh(\frac{1}{2}\,\mathrm{x}(t))^2 - 4\,g\,\mathrm{csgn}(\cosh(\frac{1}{2}\,\mathrm{x}(t)))\sinh(\frac{1}{2}\,\mathrm{x}(t))\cosh(\frac{1}{2}\,\mathrm{x}(t)))\Big/$$
$$\cosh(\frac{1}{2}\,\mathrm{x}(t))^3]$$

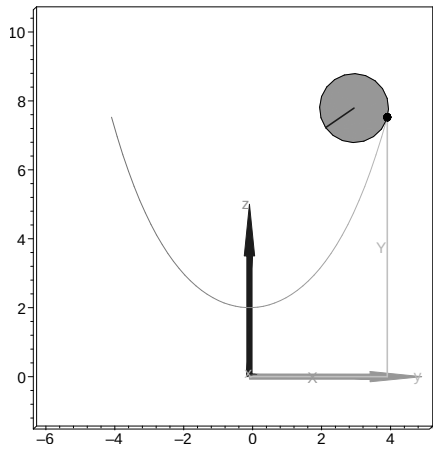
$$\%1 := \mathrm{csgn}(1,\ \cosh(\frac{1}{2}\,\mathrm{x}(t)))$$

$$\%2 := \cosh(\frac{1}{2}\,\mathrm{x}(t))^3$$

```
> g:=9.8 :
> res := fint([4,0]):
> odeplot(res,[t,x(t)],0..2);
```




```
> dibu2(4,100);
```



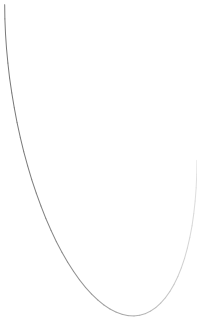
6.7. Disco sobre una circunferencia

Disco que rueda sobre una circunferencia (resuelto mediante ligaduras)

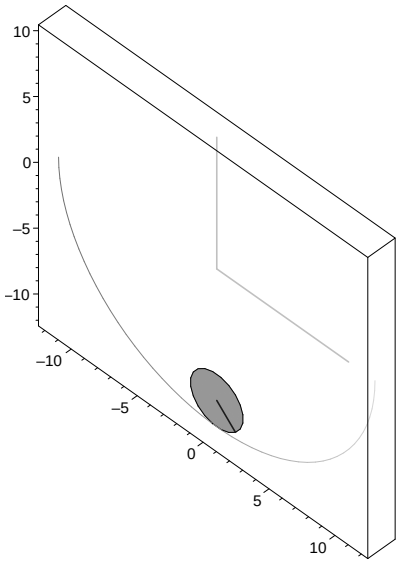
```
> restart:
> with(linalg):with(plots):with(plottools):
```

```
Warning, the protected names norm and trace have been redefined and
unprotected

Warning, the name changecoords has been redefined

Warning, the name arrow has been redefined
```

```
> libname := "meca3d",libname :
> with(mecapac3d):
> cg := [r,theta,beta] :
> s1 := [segmento,[0,0,0],[0,10,0],grey] :
> s2 := [segmento,[0,0,0],[0,0,10],grey] :
> xg := [0,r*sin(theta),-r*cos(theta)]:
> rot1 := rota(Pi/2,2):
> rot2 := rota(beta,1):
> rottot := evalm(rot2 &* rot1):
> a1 := [disco,xg,rottot,R,R]:
> R:=2 :rf:=10:
> gr1 :=
> spacecurve([0,(R+rf)*sin(t),-(R+rf)*cos(t)],t=-Pi/2..Pi/2,thickness=2)
> :
> elgraf := [grafico,[0,0,0],rota(0,1),gr1]:
> display(gr1);
```



```
> sistema := [a1,s1,s2,elgraf] :
> fG([10.,0.,evalf(Pi/4)]);
```




```
> T := fT(sistema);
```

$$T := (r1\,\sin(\theta) + r\,\cos(\theta)\,\theta1)^2 + (-r1\,\cos(\theta) + r\,\sin(\theta)\,\theta1)^2 + 2\,\beta1^2$$

```
> V := fV(sistema);
```

$$V := -2\,g\,r\,\cos(\theta)$$

```
> L := simplify(T - V );
```

$$L := r1^2 + r^2\,\theta1^2 + 2\,\beta1^2 + 2\,g\,r\,\cos(\theta)$$

```
> Phi := [r1,(r+R)*theta1+beta1*R] ;
```

$$\Phi := [r1,\ (r+2)\,\theta1 + 2\,\beta1]$$

```
> reacc :=Fc();
```

$$reacc := \left[-2\,\mathrm{r}(t)\,(\tfrac{\partial}{\partial t}\,\theta(t))^2 - 2\,g\cos(\theta(t)),\ 2\,\frac{(\mathrm{r}(t)+2)\,\mathrm{r}(t)^2\left(-(\tfrac{\partial}{\partial t}\,\mathrm{r}(t))\,(\tfrac{\partial}{\partial t}\,\theta(t)) - \frac{1}{2}\,\frac{(\mathrm{r}(t)+2)\,(-4\,\mathrm{r}(t)\,(\tfrac{\partial}{\partial t}\,\theta(t))\,(\tfrac{\partial}{\partial t}\,\mathrm{r}(t)) - 2\,g\,\mathrm{r}(t)\sin(\theta(t)))}{\mathrm{r}(t)^2}\right)}{3\,\mathrm{r}(t)^2+4\,\mathrm{r}(t)+4} \right]$$

```
> reacc[1];
```

$$-2\,\mathrm{r}(t)\,(\tfrac{\partial}{\partial t}\,\theta(t))^2 - 2\,g\cos(\theta(t))$$

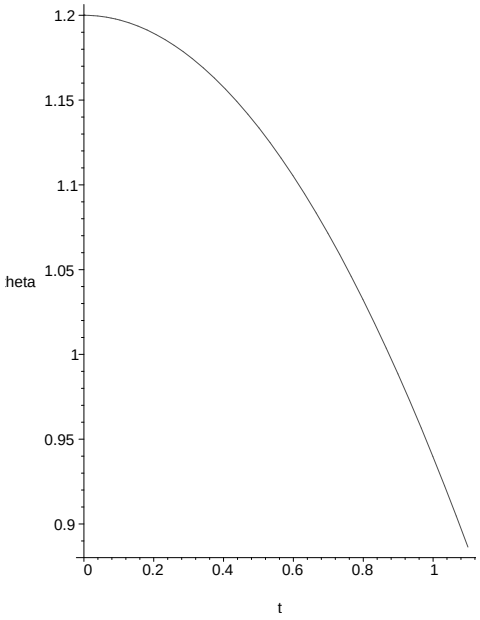
```
> simplify(ec_lagr());
```

$$[2\,(\tfrac{\partial^2}{\partial t^2}\,\mathrm{r}(t)), 2\mathrm{r}(t)^2(5\,\mathrm{r}(t)\,(\tfrac{\partial}{\partial t}\,\theta(t))\,(\tfrac{\partial}{\partial t}\,\mathrm{r}(t)) + 2\,(\tfrac{\partial}{\partial t}\,\mathrm{r}(t))\,(\tfrac{\partial}{\partial t}\,\theta(t)) + 3\,\mathrm{r}(t)^2\,(\tfrac{\partial^2}{\partial t^2}\,\theta(t))$$
$$+ 4\,\mathrm{r}(t)\,(\tfrac{\partial^2}{\partial t^2}\,\theta(t)) + 4\,(\tfrac{\partial^2}{\partial t^2}\,\theta(t)) + 2\,g\,\mathrm{r}(t)\sin(\theta(t)))\,/(3\,\mathrm{r}(t)^2+4\,\mathrm{r}(t)+4), -4($$
$$-3\,(\tfrac{\partial^2}{\partial t^2}\,\beta(t))\,\mathrm{r}(t)^2 - 4\,(\tfrac{\partial^2}{\partial t^2}\,\beta(t))\,\mathrm{r}(t) - 4\,(\tfrac{\partial^2}{\partial t^2}\,\beta(t)) + (\tfrac{\partial}{\partial t}\,\mathrm{r}(t))\,(\tfrac{\partial}{\partial t}\,\theta(t))\,\mathrm{r}(t)^2$$
$$+ g\,\mathrm{r}(t)^2\sin(\theta(t)) + 4\,\mathrm{r}(t)\,(\tfrac{\partial}{\partial t}\,\theta(t))\,(\tfrac{\partial}{\partial t}\,\mathrm{r}(t)) + 2\,g\,\mathrm{r}(t)\sin(\theta(t)))\,/($$
$$3\,\mathrm{r}(t)^2+4\,\mathrm{r}(t)+4)]$$

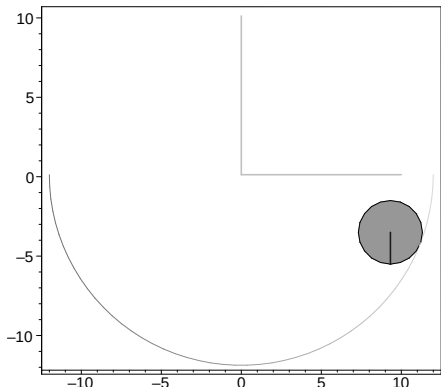
```
> m:=10 :g:=9.8:
> res := fintr([10,0,1.2,0.,0.,0.]):
> res(0.1);
```

$$[t = ,1,\ \beta(t) = ,0159286208899611752,\ \tfrac{\partial}{\partial t}\,\beta(t) = ,318517462777871907,\ \mathrm{r}(t) = 10.,$$
$$\tfrac{\partial}{\partial t}\,\mathrm{r}(t) = 0.,\ \theta(t) = 1,19734522985167312,\ \tfrac{\partial}{\partial t}\,\theta(t) = -,0530862437963119799]$$

```
> odeplot(res,[t,theta(t)],0..1.10);
```




```
> dibu3(6,20);
```



Índice alfabético