

Resolución del problema puntuable de 22/04/97 (oscilaciones lineales con varios g.d.l.)

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

Warning, new definition for norm

Warning, new definition for trace

Debido a que en las ecuaciones de Lagrange intervienen derivadas respecto a las coordenadas generalizadas, las velocidades generalizadas y respecto al tiempo, es necesario derivar de distintas maneras según el caso.

Esto se lleva a cabo estableciendo el siguiente convenio:

Dada una coordenada generalizada, la correspondiente velocidad es la misma coordenada añadiendo ".t" a su denominación. Cuando queramos derivar respecto del tiempo, cambiamos las coordenadas (y las velocidades) haciendo explícita su dependencia del tiempo (x por x(t) y xt por diff(x(t),t)).

Para poder realizar estas manipulaciones nos apoyamos en las siguientes funciones.

```
> f_e := x->x=x(t);
> fi_e := x->x(t)=x;
> f_d := x->x.t=diff(x(t),t);
> fi_d := x->diff(x(t),t)=x.t;
> f_dd := x->x.t.t=diff(x(t),t,t);
> fi_dd := x->diff(x(t),t,t)=x.t.t;
```

$$f_e := x \rightarrow x = x(t)$$

$$fi_e := x \rightarrow x(t) = x$$

$$f_d := x \rightarrow x.t = \frac{\partial}{\partial t} x(t)$$

$$fi_d := x \rightarrow \frac{\partial}{\partial t} x(t) = x.t$$

$$f_dd := x \rightarrow x.t.t = \frac{\partial^2}{\partial t^2} x(t)$$

$$fi_dd := x \rightarrow \frac{\partial^2}{\partial t^2} x(t) = x.t.t$$

Con las funciones anteriores y supuesta conocida la Lagrangiana (L) podemos definir una aplicación "ec_l" que operando sobre una coordenada generalizada (x , del conjunto de coordenadas cg) nos devuelve la ecuación de Lagrange correspondiente.

```
> ec_l := x -> diff(subs(map(f_e,cg), map(f_d,cg), diff(L,x.t)),t)
-subst(map(f_e,cg), map(f_d,cg), diff(L,x));
```

$$ec_l := x \rightarrow \left(\frac{\partial}{\partial t} \text{subs}(\text{map}(f_e, cg), \text{map}(f_d, cg), \text{diff}(L, x.t)) \right) - \text{subs}(\text{map}(f_e, cg), \text{map}(f_d, cg), \text{diff}(L, x))$$

Establecimiento de las coordenadas generalizadas a usar.

```
> cg := [x.1,x.2];
```

$$cg := [x_1, x_2]$$

```
> T := (1/2)*m.1*x.1.t^2+(1/2)*m.2*x.2.t^2 +(1/2)* m *((x.2.t - e*omega*sin(omega*t))^2 + (e*omega*cos(omega*t))^2);
```

$$T := \frac{1}{2} m_1 x_1 t^2 + \frac{1}{2} m_2 x_2 t^2 + \frac{1}{2} m ((x_2 t - e \omega \sin(\omega t))^2 + e^2 \omega^2 \cos(\omega t)^2)$$

Definimos la expresión de la energía potencial (en función de las coordenadas generalizadas sin la dependencia de las coordenadas del tiempo)

```
> V := m*g*e*sin(omega*t) +(1/2)*k*x.1^2 +(1/2)*k*(x.2-x.1)^2;
```

$$V := m g e \sin(\omega t) + \frac{1}{2} k x_1^2 + \frac{1}{2} k (x_2 - x_1)^2$$

Cálculo de la lagrangiana (es importante que se llame L para poder utilizar a continuación la función "ec_l")

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

Cálculo de las ecuaciones de Lagrange (todas de una vez aplicando "ec_l" sobre las coordenadas generalizadas).

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

```
> ecua := map(simplify,ecua);
```

$$ecua := [m_1 \left(\frac{\partial^2}{\partial t^2} x_1(t) \right) + 2 k x_1(t) - k x_2(t), \\ m_2 \left(\frac{\partial^2}{\partial t^2} x_2(t) \right) + m \left(\frac{\partial^2}{\partial t^2} x_2(t) \right) - m e \omega^2 \cos(\omega t) + k x_2(t) - k x_1(t)]$$

Obtenemos ahora las matrices K y M y el vector de fuerzas F , identificando los coeficientes de las ecuaciones:

```
> dim:=nops(cg):
```

```
> K:=matrix(dim,dim):
```

```
> for i from 1 to dim do for j from 1 to dim do K[i,j] := coeff(
subs(map(fi_dd,cg), map(fi_e,cg), ecua[i]),cg[j]): od; od;
```

```
> M:=matrix(dim,dim):
```

```

> for i from 1 to dim do for j from 1 to dim do M[i,j] := coeff(
subs(map(fi_dd,cg), map(fi_e,cg), ecua[i]),cat(cg[j],t.t)): od;
od;
> cgg:=[op(cg),op(map(x->cat(x,t.t),cg))];
      cgg := [x1, x2, x1tt, x2tt]

> F:=vector(dim):
> for i from 1 to dim do ecuc:=subs(map(fi_dd,cg),map(fi_e,cg),ecua[i]):
F[i]:=-subs(map(x->x=0,cgg),ecuc): od:
> print(M,K,F);
      
$$\begin{bmatrix} m1 & 0 \\ 0 & m2 + m \end{bmatrix}, \begin{bmatrix} 2k & -k \\ -k & k \end{bmatrix}, [0, m e \omega^2 \cos(\omega t)]$$


```

datos numéricos:

```

> datos:={e=.1, m=40, m.1=8000., m.2=5000., k=8000., omega=(900/60)*2*Pi};
datos := {e = .1, m = 40, k = 8000., ω = 30 π, m1 = 8000., m2 = 5000.}

```

Valores numéricos de las matrices:

```

> map2(subs,datos,M);map2(subs,datos,K);map2(subs,datos,F);
      
$$\begin{bmatrix} 8000. & 0 \\ 0 & 5040. \end{bmatrix}$$

      
$$\begin{bmatrix} 16000. & -8000. \\ -8000. & 8000. \end{bmatrix}$$

      
$$[0, 3600.0 \pi^2 \cos(30 \pi t)]$$


```

Ecuación característica para el cálculo de los autovalores

```

> ec.car:=collect(det(K-lambda*M),lambda);
      eccar := (m1 m2 + m1 m) λ2 + (-2 k m2 - 2 k m - m1 k) λ + k2
> ec.car:=subs(datos,ec.car);
      eccar := .40320000 108 λ2 - .144640000 109 λ + .64000000 108
> lambda:=[solve(ec.car,lambda)];
      λ := [.5169826370, 3.070318950]

```

Las frecuencias propias son las raíces cuadradas de los autovalores:

```

> Omega:=map(sqrt,lambda);
      Ω := [.7190150464, 1.752232562]

```

Obtenemos ahora los vectores propios v_i , a partir de las matrices características para cada autovalor λ_i , y formamos la matriz modal A (vectores propios por filas):

```
> for i from 1 to dim do mcar.i:=map2(subs,datos,evalm(K-lambda[i]*M)):
v.i:=vector(dim,[1,-mcar.i[1,1]/mcar.i[1,2]]): if (i=1) then A:=v.i
else A:=stack(A,v.i) fi; od;
```

$$mcar1 := \begin{bmatrix} 11864.13890 & -8000. \\ -8000. & 5394.407510 \end{bmatrix}$$

$$v1 := [1, 1.483017363]$$

$$mcar2 := \begin{bmatrix} -8562.55160 & -8000. \\ -8000. & -7474.407508 \end{bmatrix}$$

$$v2 := [1, -1.070318950]$$

```
> evalm(A);
```

$$\begin{bmatrix} 1 & 1.483017363 \\ 1 & -1.070318950 \end{bmatrix}$$

Expresión del cambio de coordenadas geométricas x_i a coordenadas normales u_i

```
> u:=vector(dim):u=inverse(transpose(A))*cg;
```

```
> evalm(");
```

$$u = \begin{bmatrix} .4191844782 & .3916444516 \\ .5808155218 & -.3916444516 \end{bmatrix} \&* [x1, x2]$$

$$[u_1, u_2] = [.4191844782 x1 + .3916444516 x2, .5808155218 x1 - .3916444516 x2]$$

Régimen permanente, mediante la expresión $(K - M \omega^2)^{-1} F$:

```
> x.p:=evalf( map2(subs,datos,evalm(inverse(K-M*omega^2)*F) ) );
```

$xp :=$

$$[.8938458308 \cdot 10^{-7} \cos(94.24777962 t), -.0007937926585 \cos(94.24777962 t)]$$

y en coordenadas normales

```
> u.p = evalm(transpose(A) &* x.p);
```

$up =$

$$[-.0007937032739 \cos(94.24777962 t), .0008497438837 \cos(94.24777962 t)]$$

PROCEDIMIENTO ALTERNATIVO

Asignamos los valores numéricos y evaluamos los autovalores y vectores propios directamente mediante la función "Eigenvals":

```
> assign(datos);
> lambda:=evalf(Eigenvals(K,M,vects));
      λ := [3.070318951, .5169826369]
```

Las frecuencias propias son las raíces cuadradas de los autovalores:

```
> Omega:=map(sqrt,lambda);
```

(nótese que aquí se obtienen ordenados de mayor a menor, al contrario de la ordenación habitual para los modos normales de vibración)

```
Ω := [1.752232562, .7190150464]
```

La matriz "vects" contiene los vectores propios (modos normales de vibración) en columnas:

```
> evalm(vects);
      [ .9343009385  -0.6743009390 ]
      [ -1.000000000  -1.000000000 ]
```

Construimos la matriz modal A (vectores propios en filas) trasponiendo "vects" y normalizando, de forma que la primera componente de cada modo sea la unidad:

```
> A:=transpose(vects);
> for i from 1 to dim do fac:=A[i,1]: for j from 1 to dim do A[i,j]
:= A[i,j]/fac: od; od;
> evalm(A);
      [ 1.000000000  -1.070318951 ]
      [ 1.000000000  1.483017362 ]
```

Comprobamos que efectivamente que v_i son vectores propios con autovalores λ_i

```
> for i from 1 to dim do v.i := row(A,i): evalm(K &* v.i)=scalarmul(M
&* v.i,lambda[i]): od;
      v1 := [1.000000000, -1.070318951]
      [24562.55161, -16562.55161] = [24562.55161, -16562.55162]
      v2 := [1.000000000, 1.483017362]
      [4135.86110, 3864.138900] = [4135.861095, 3864.138901]
```

Diagonalización de las matrices M , K , y C y comprobación de los resultados

```
> MD:=evalm(A &* M &* transpose(A));
      MD := [ 13773.73659  .1 10-5 ]
            [ .1 10-5  19084.67610 ]
```

```
> KD:=evalm(A &* K &* transpose(A));  
       $KD := \begin{bmatrix} 42289.76448 & .00001 \\ .6 \cdot 10^{-5} & 9866.446178 \end{bmatrix}$   
  
> evalm(KD &* inverse(MD));  
       $\begin{bmatrix} 3.070318951 & .3631018421 \cdot 10^{-9} \\ .3980776986 \cdot 10^{-9} & .5169826370 \end{bmatrix}$ 
```