Solución a la segunda práctica voluntaria — Técnicas Numéricas
Francisco R. Villatoro y Carmen M. García — 12 de Noviembre de 2001

SOLUCIÓN DE LA SEGUNDA PRACTICA (PUNTUACIÓN: 0.2)

La respuesta a la pregunta que se realiza en esta práctica, ¿qué pasará cuando deje de sujetar la estructura?

(a) El lado izquierdo caerá hacia abajo,

(b) El lado derecho caerá hacia abajo,

(c) No pasará nada y la estructura se mantendrá horizontal,

tiene como respuesta correcta la (c), es decir, la estructura se mantendrá estable, es decir, siempre que los pesos que se coloquen en ambos brazos laterales de la estructura sean iguales, independientemente de su posición en dichos brazos, la estructura se mantendrá estable.

Las fotos de la figura muestran gráficamente este hecho.

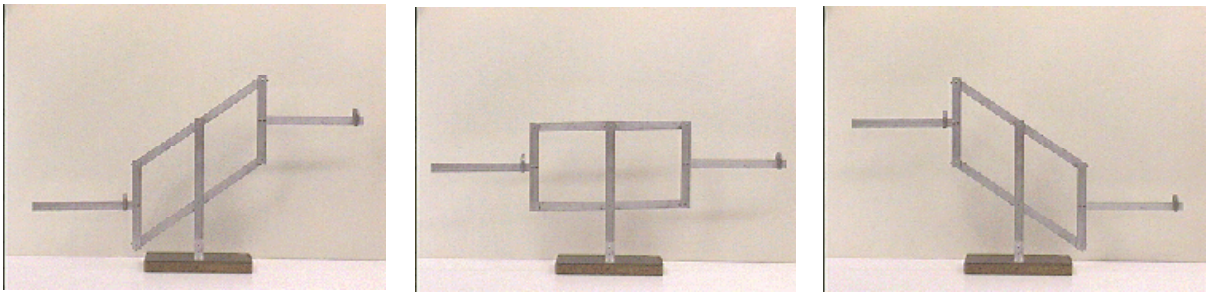


Figure 1. Tres posiciones de equilibrio de una estructura estable incluyendo dos pesos iguales en sus barras extremas. © "The Physics Question of the Week," of the University of Maryland Department of Physics.

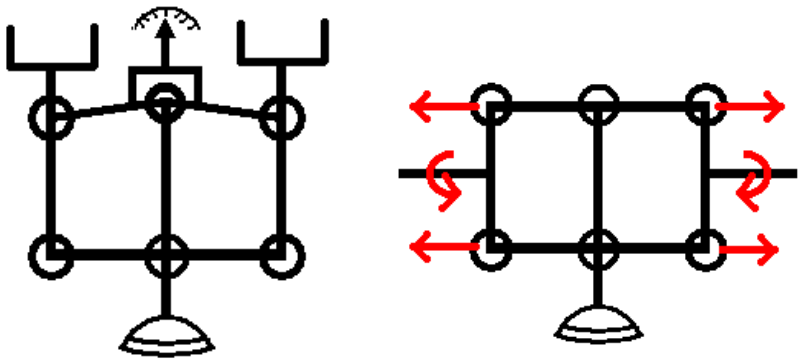
La razón por la cual esta estructura permanece en equilibrio para cualquier posición de sus brazos es que la posición vertical del centro de masas de la figura no cambia independientemente de la posición de los brazos y de las masas que se coloquen en éstos, si se colocan igual cantidad a ambos lados.

Los que han estudiado una asignatura de Mecánica y Mecanismos reconoceréis esta estructura como una balanza de Roberval (ver la parte izquierda de la figura). Las balanzas se construyen para que la posición del peso en la bandeja de la balanza no afecte al equilibrio de la misma.

En cuanto a un modelo matemático de este problema, que es una estructura hiperestática, he de indicar que al aplicar directamente las ecuaciones para las fuerzas y los momentos en los nodos de la estructura se obtiene siempre la misma ecuación. Esto se conoce como paradoja de Roberval. Por ello se requiere un análisis físico más cuidadoso.

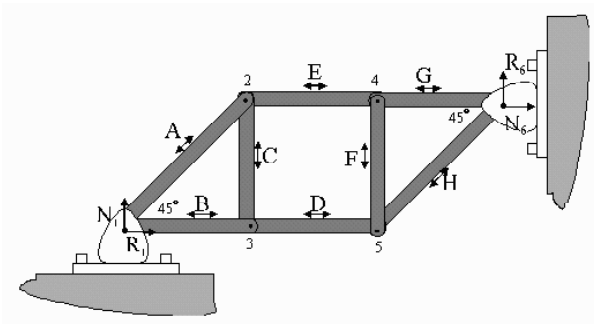
Como se indica en la parte derecha de la figura los momentos en los brazos generan fuerzas horizontales en cada uno de los nodos externos que se balancean entre sí, no afectando al centro de masas.

En cuanto a la estructura de 8 enlaces del enunciado, mostrada en la siguiente figura



(a) (b)

Figure 2. Dibujo esquemático de una balanza de Roberval (a) y fuerzas y momentos en la misma indicadas por flechas (b).



aplicando el equilibrio estático de fuerzas (sin tener en cuenta los momentos) en todos los nodos se obtiene el sistema de ecuaciones

$$1)\quad \rightarrow\ R_1 - \frac{1}{\sqrt{2}}A - B = F_{1x} \qquad \uparrow\ N_1 - \frac{1}{\sqrt{2}}A = F_{1y}$$

$$2)\quad \rightarrow\ \frac{1}{\sqrt{2}}A - E = F_{2x} \qquad \uparrow\ \frac{1}{\sqrt{2}}A + C = F_{2y}$$

$$3)\quad \rightarrow\ B - D = F_{3x} \qquad \uparrow\ -C = F_{3y}$$

$$4)\quad \rightarrow\ E - G = F_{4x} \qquad \uparrow\ F = F_{4y}$$

$$5)\quad \rightarrow\ D - \frac{1}{\sqrt{2}}H = F_{5x} \qquad \uparrow\ -F - \frac{1}{\sqrt{2}}H = F_{5y}$$

$$6)\quad \rightarrow\ N_6 + G + \frac{1}{\sqrt{2}}H = F_{6x} \qquad \uparrow\ R_6 + \frac{1}{\sqrt{2}}H = F_{6y}$$

que conducen al sistema lineal

$$\begin{bmatrix} 1 & 0 & -1/\sqrt{2} & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1/\sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1/\sqrt{2} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/\sqrt{2} & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R_1 \\ N_1 \\ A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \\ N_6 \\ R_6 \end{bmatrix} = \begin{bmatrix} F_{1x} \\ F_{1y} \\ F_{2x} \\ F_{2y} \\ F_{3x} \\ F_{3y} \\ F_{4x} \\ F_{4y} \\ F_{5x} \\ F_{5y} \\ F_{6x} \\ F_{6y} \end{bmatrix}$$

cuya matriz de coeficientes tiene inversa

$$A^{-1} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & -1 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{2} & 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & -1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

por lo que la estructura es estable (como se indicó en el enunciado).

El resto de la práctica es sencillo, salvo la implementación del algoritmo de factorización LU con pivotaje parcial.

```
function [L,U,P] = lufact(A),
%
% function [L,U,P] = lufact(A),
%   FACTORIZACION DOOLITTLE CON PIVOTAJE PARCIAL
%
% Datos: A- matriz orden NxN
% Resultado: Matices L, U, P tales que
%    L.U = P.A
% donde L es triangular inferior unitaria
%       U es triangular superior
%       P es matriz de permutaciones de filas
```

```
%
% Se utiliza pivotaje parcial sobre A
%  en R se guardan los intercambios de filas
%  C es un vector temporal para los intercambios

N=size(A);
if (N(1)~=N(2)), error('A debe ser cuadrada.'); end
N=length(A);

L=zeros(N,N);  % reserva espacio para matriz triangular inferior
U=zeros(N,N);  % reserva espacio para matriz triangular superior
R=1:N;         % guarda intercambios de filas

TOL = N^2*eps; % Tolerancia para saber qué es cero.

% Para cada uno de los pasos
for q=1:N,
  % se determina la fila pivote para la columna q-esima
  [maximo,j] = max(abs(A(q:N,q)));
  % se intercambian las filas q-esima y (j+q-1)-esima
  C=A(q,:); A(q,:)=A(j+q-1,:); A(j+q-1,:)=C;
  % MUY IMPORTANTE: tambien hay que intercambiar L
  C=L(q,:); L(q,:)=L(j+q-1,:); L(j+q-1,:)=C;
  r=R(q); R(q)=R(j+q-1); R(j+q-1)=r;

  % Se rellena la q-esima fila de U
  U(q,q:N) = A(q,q:N) - L(q,1:q-1)*U(1:q-1,q:N);
  %for k=q:N,
    % la q-esima fila de L por la k-esima columna de U es igual a A(q,k)
    %U(q,k) = A(q,k) - L(q,1:q-1)*U(1:q-1,k);
  %end

  % >A es singular?
  if abs(U(q,q))<TOL, 'A es (próxima a) singular.', clear L U P; break; end

  % Se rellena la q-esima columna de L
  L(q,q)=1;
  L(q+1:N,q) = ( A(q+1:N,q) - L(q+1:N,1:q-1)*U(1:q-1,q) )/U(q,q);
  %for k=q+1:N,
    % la k-esima fila de L por la q-esima columna de U es igual a A(k,q)
    %L(k,q) = ( A(k,q) - L(k,1:q-1)*U(1:q-1,q) )/U(q,q);
  %end
end

P = sparse ( (1:N)', R', ones(N,1), N, N);
```

Algunos alumnos han realizado el problema aplicando directamente el método de triangulación de Gauss, cogiendo como matriz L la matriz de los multiplicadores y como matriz U la matriz A resultante.

En dicho caso el código sería como el siguiente. Los alumnos pueden comprobar que los resultados son también válidos, aunque cuando P no coincide entre ambos algoritmos el resultado para L y U puede ser diferente.

```
function [L,U,P] = lufactgauss(A),
%
% function [L,U,P] = lufact(A),
%   FACTORIZACION DOOLITTLE CON PIVOTAJE PARCIAL
%
% Datos: A- matriz orden NxN
% Resultado: Matices L, U, P tales que
%    L.U = P.A
% donde L es triangular inferior unitaria
%       U es triangular superior
%       P es matriz de permutaciones de filas
%
% Se utiliza pivotaje parcial sobre A
%  en R se guardan los intercambios de filas
%

N=size(A);
if (N(1)~=N(2)), error('A debe ser cuadrada.'); end
N=length(A);

% Guardaremos L y U en la parte sub- y supra-diagonal de A

R=1:N;          % guarda intercambios de filas

TOL = N^2*eps; % Tolerancia para saber qué es cero.

% Para cada uno de los pasos
for q=1:N,
  % se determina la fila pivote para la columna q-esima
  [maximo,j] = max(abs(A(q:N,q)));
  % se intercambian las filas q-esima y (j+q-1)-esima
  C=A(q,:); A(q,:)=A(j+q-1,:); A(j+q-1,:)=C;
  r=R(q); R(q)=R(j+q-1); R(j+q-1)=r;

  % >A es singular?
  if abs(A(q,q))<TOL, error('A es (próxima a) singular.'); end
```

```
  % Calculo del multiplicador (L en subdiagonal de A)
  %for k=q+1:N,
  %  A(k,q)=A(k,q)/A(q,q);              % multiplicador
  %  A(k,q+1:N) = A(k,q+1:N)-A(k,q)*A(q,q+1:N);
  %end
  A(q+1:N,q)=A(q+1:N,q)./A(q,q);
  A(q+1:N,q+1:N) = A(q+1:N,q+1:N)-A(q+1:N,q)*A(q,q+1:N);
end

L= tril(A,-1) + eye(N,N);
U= triu(A,0);
P = sparse ( (1:N)', R', ones(N,1), N, N);
```

Los alumnos interesados en aprender a programar BIEN en Matlab deberían estudiar con cuidado estos códigos, en los que se ha hecho uso de toda la potencia que tiene Matlab para describir operaciones matriciales. Para los productos fila-columna se ha presentado un código sin bucles `for`, aunque se ha incluido el código con un sólo bucle `for`, aunque sólo como comentario para facilitar su comprensión. Los alumnos que han realizado esta práctica normalmente han utilizado dos bucles `for` anidados. Para matrices grandes, en el intérprete de Matlab o cuando se utiliza el compilador de Matlab, el código presentado es más eficiente que el que utiliza estos bucles `for`.