`diag`: diagramas de solicitaciones

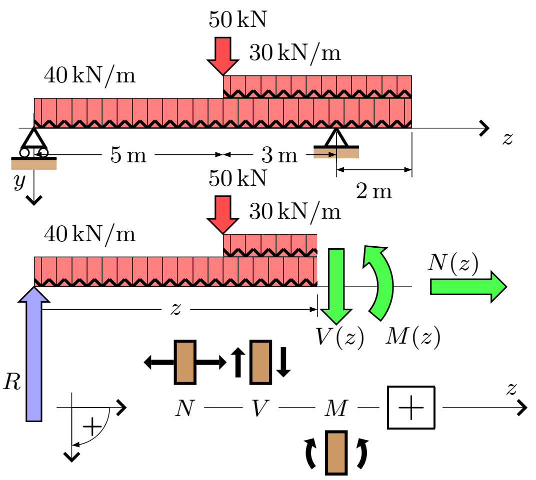

Figura 1: Conjunto de fuerzas

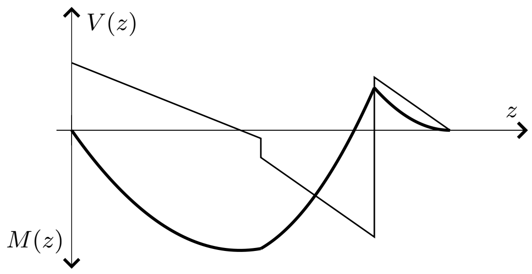

Figura 2: Diagramas de $M(z)$ y $V(z)$

Un conjunto de fuerzas como el de la figura 1 *puede estar* en equilibrio si se verifican que la fuerza y el momento resultantes respecto a un punto arbitrario son nulos. Para el equilibrio real *tiene* que existir una estructura mecánica que permita la conexión física de las acciones y reacciones que aquellas fuerzas abstractas representan. Bajo ellas, la estructura habrá de resistir solicitaciones en cada uno de sus puntos. Estás últimas pueden ponerse al descubierto considerando *sólo* una parte del conjunto de fuerzas mediante un corte imaginario, que divida a la estructura en dos. Para que uno cualquiera de los dos subconjuntos de fuerzas esté a su vez en equilibrio, deben aparecer *al otro lado* del corte *tensiones* cuyas resultantes de fuerza y momento equilibren las resultantes del subconjunto. Estas nuevas fuerzas que *sólo* aparecen en cortes imaginarios son las solicitaciones que soporta la estructura. *Como cada corte tiene siempre dos caras, las solicitaciones siempre aparecen a pares.*

Una representación estándar consiste en referir las fuerzas exteriores a una linea recta entre dos puntos significativos del conjunto o de la estructura, denominada *directriz*. Los cortes se dan perpendicularmente a ella y las solicitaciones en ellos se describen mediante una solicitación normal (fuerza paralela a la directriz y normal al corte), una solicitación transversal (perpendicular a la directriz y tangente al corte) y un momento referido a la intersección entre el corte y la directriz.

Cálculo analítico. Para el cálculo analítico es necesario un convenio de signos muy riguroso. Suelo emplear los de la figura. **¡Mucho ojo!** Tanto el de fuerzas abstractas como el de solicitaciones **¡son distintos!** Por ejemplo, una fuerza horizontal positiva (que apunta hacia la derecha) *es* una solicitación normal positiva en un corte vertical (una *tracción*) **sólo si está a la derecha del corte**. Si estuviera al otro lado seguiría siendo una fuerza positiva, pero respecto al corte representaría una *compresión*, una solicitación normal negativa. En consecuencia, las solicitaciones pueden calcularse desde la derecha o la izquierda. Mi costumbre es calcularlas como resultantes de las fuerzas exteriores que quedan a la *izquierda* de un corte vertical (suponiendo que la directriz sea horizontal).

Fijando el origen de coordenadas en el extremo izquierdo de la directriz, con el eje de las abscisas sobre ella, y denominando a cada una de las fuerzas exteriores como Z_i e Y_i según sean horizontales o verticales, aplicadas en puntos de la directriz a una distancia z_i del origen, las resultantes normal, transversal y de momento a una distancia z, tienen las siguientes expresiones:

$$\sum_{z_i \le z} Z_i \qquad \sum_{z_i \le z} Y_i \qquad \sum_{z_i \le z} Y_i(z - z_i)$$

Para que representen las solicitaciones a la *izquierda* del corte es necesario todavía un cambio de signo según los convenios anteriores; por tanto, las solicitaciones son:

$$N(z) = -\sum_{z_i \le z} Z_i \quad V(z) = -\sum_{z_i \le z} Y_i \quad M(z) = -\sum_{z_i \le z} Y_i(z - z_i)$$

Como funciones matemáticas, pueden representarse en un par de ejes cartesianos, como se muestra en la figura 2. Nótese que para $M(z)$ se emplea el eje de ordenadas hacia abajo, en vez de hacia arriba. Hay buenas razones para ello: de este modo el diagrama de momentos presenta el mismo aspecto que un hilo sometido a las mismas fuerzas (y la construcción del hilo es un estupendo modo de trazarlo).

Las expresiones anteriores indican que, sobre una directriz recta, las fuerzas Z_i solo producen solicitación normal. Por el contrario, las fuerzas Y_i producen tanto solicitación transversal como momento. De hecho, es fácil ver que se cumple $M(z) = \int_0^z V(z)\,\mathrm{d}z$, salvo el valor de la constante de integración ($\sum Y_i z_i$, nula si hay equilibrio): M es la integral de V o bien, V es la derivada de M.

Cálculo automático. Cuando en una expresión matemática aparece el símbolo $\sum$ lo que cabe esperar es una secuencia repetitiva y tediosa de operaciones. Puede ser útil una máquina para realizarlas. Lo que sigue son programas para la 'mía', una HP48GX (en otras *hp* puede que los programas también funcionen, pero no lo he comprobado —*hp* nunca me ha dado un duro: ¡podía regalarme al menos una HP de las nuevas!).

En la mayoría de los casos prácticos, sólo interesan tres tipos de carga transversal a la directriz (las cargas paralelas se tratan de similar manera pero no las considero aquí): la uniformemente repartida a lo largo de toda la directriz, $\mathbf{p}$; la puntual, $\mathbf{Q}$, en z_Q; y un trozo de carga uniforme $\mathbf{p}$ entre z_i y z_f ($z_i < z_f$). Un conjunto de varias de tales cargas puede representarse con comodidad mediante una lista de reales, complejos y listas de tres números: $\mathbf{p}$, $(\mathbf{Q};x_Q)$, y {$\mathbf{p}$ z_i z_f}. De este modo las cargas de la figura 2, se codifican como `{ 40 (50;5) { 30 5 10 } }`, que puede guardarse en una variable, por ejemplo `Q`. La contribución a $V(z)$ de cada una de estas cargas tiene por expresión:

$$\Delta_{\mathbf{p}} V(z) = -pz \qquad \Delta_{\mathbf{Q},z_Q} V(z) = \begin{cases} -\mathbf{Q} & \text{si } z_Q \leq z \\ 0 & \text{en otro caso} \end{cases}$$

$$\Delta_{\mathbf{p},z_i,z_f} V(z) = \begin{cases} -\mathbf{p}(z - z_i) & \text{si } z_i \leq z \text{ y } z < z_f \\ -\mathbf{p}(z_f - z_i) & \text{si } z_f \leq z \\ 0 & \text{en otro caso} \end{cases}$$

Las contribuciones a $M(z)$ son similares (pero más complicadas): considere su deducción como un útil ejercicio. Para calcular M y V sólo necesitamos recorrer la lista `Q` secuencialmente e ir sumando las contribuciones de cada miembro de la lista. Los programas para ello los distribuyo mediante un *autoinstalable*, `DIAG`, (las instrucciones figuran al final).

Una vez instalado todo en algún directorio, pulsando `CST` aparecerán `V`, `M`, `Q`, `R>C`, `OVER`, y `ROT`. Los dos primeros son los fundamentales: ambos funcionan cogiendo de la pila la lista de cargas (en el nivel 2) y la abscisa z (en el nivel 1), y devolviendo el valor de $M(z)$ ó $V(z)$. Se trata, por tanto, de dos funciones, `M(X,Y)` y `T(X,Y)`, que pueden manejarse como cualquier otra función: puede emplear sobre ellas las diversas funciones de `SOLVE` o `SIMBOLIC` (pero no son *analíticas* en la acepción ***hp***). `Q` permite recuperar o almacenar las cargas. (`R>C` convierte dos valores de la pila en un número complejo; `OVER` duplica el objeto en el nivel 2; y `ROT` permuta circularmente los tres primeros pisos de la pila, dejándo en el 1 lo que hubiera en el 3: son instrucciones estándar, consulte el manual.)

De momento, con la secuencia `Q` `10` `V` obtendrá en la pila `-600` que es la solicitación que *existiría* en $z = 10\,\text{m}$: el sistema no está en equilibrio: ¡falta calcular e incluir en `Q` las reacciones! El resultado debe interpretarse en el sentido de que las dos reacciones deberían *equilibrar* 600 kN. Con la secuencia `Q` `10` `M` se obtiene un momento de $-2{,}625\,\text{mkN}$, momento que deben aportar las reacciones. Con `OVER` `÷` averiguamos que la resultante de las cargas está a unos 4,38 m del extremo derecho, es decir, `2` `−`, a 2,38 m del apoyo derecho. `OVER` `×` `8` `÷` nos da el valor de la reacción en el apoyo izquierdo; `ENTER` `+/-` `ROT` `+`, el valor de la reacción derecha. Ahora basta con: `8` `R>C` `SWAP` `0` `R>C` `Q` `+` `+` `↰` `Q` para incluir ambas: el conjunto de fuerzas exteriores está completo. (Al principio todo esto parece raro... pero te acostumbras.) Ahora, la secuencia `Q` `10` `M` arroja `0` indicando que hay equilibrio de momentos; igual ocurre con `Q` `10` `V`.

Con secuencias como `Q` z `M` se obtiene el valor $M(z)$: para 5 m se tienen 390,63 mkN. Almacenando en la variable `EQ` la expresión `'T(Q,X)'`, con `SOLVR` puede encontrar las abscisas z para las cuales $V(z) = 0$ y, por tanto, $M(z)$ máximo o mínimo. Aunque, mejor aún, con `PLOT` puede dibujar completamente los diagramas... (Pero tenga en cuenta que debe dibujar `'-M(Q;X)'` para ajustar los ejes usados aquí para $M(z)$ con los de la HP48GX. `PLOT` puede ser muy lento: configure `STEP` en el menú `PLOT OPTIONS`.)

Para facilitarme la vida, con `CST` `NXT` se dispone de `PV` y `PM`. `PM` contiene `'-M(Q;X)'` y `PV`, `'T(Q,X)'`. Aunque normalmente manejo varios conjuntos de fuerzas (denominados como `Q1`, `Q2`,...), convengo comigo mismo en almacenar en `Q` la carga 'en uso'; de este modo, secuencias como `PM` `'EQ'` `STO` `PLOT` me permiten producir gráficas con gran ahorro de pulsaciones.

No hay previsión para momentos externos ni otros tipos de carga: serán bienvenidas sugerencias al respecto.

Para observar *qué* hacen los programas, y por tanto entenderlo, puede usar `PRG` `NEXT` `RUN` `DBUG`. *Conviene que lo haga: toda la responsabilidad al usar un programa es suya...*

Anejo: `DIAG`

```
«
  « → ←lq ←x
    «
      « ←x *
      »
      « OBJ→ ←x
        IF >
        THEN DROP 0
        END
      »
      « → v
        «
          IF 'v(2)'
EVAL ←x ≤
          THEN
            IF 'v(3)'
EVAL ←x <
            THEN
'v(1)*(v(3)-v(2))' EVAL
            ELSE
'v(1)*(←x-v(2))' EVAL
            END
          ELSE 0
          END
        »
      » Driver
    »
  » 'V' STO
  « → ←lq ←x
    «
      « ←x SQ * 2 /
      »
      « OBJ→ ←x DUP2
        IF >
        THEN 3 DROPN 0
        ELSE - NEG *
        END
      »
      « → v
        «
          IF 'v(2)'
EVAL ←x ≤
          THEN
            IF 'v(3)'
 EVAL ←x <  THEN 'v(1)*
(v(3)-v(2))*(←x
-(v(3)+v(2))/2)' EVAL
            ELSE
'v(1)*SQ(←x-v(2))/2'
EVAL
            END
          ELSE 0
          END
        »
      » Driver
    »
  »'M' STO
  « → cp cq cpp
    « ←lq OBJ→ 0
      WHILE OVER
      REPEAT ROT → q
        « q q TYPE → qt
          «
            CASE qt
0 ==
              THEN cp
              END
qt 1 ==
              THEN cq
              END
qt 5 ==
              THEN
IF q SIZE 3 ==
THEN cpp
ELSE "BAD LIST SIZE"
DOERR
END
              END
"BAD FORMAT"DOERR
            END
          »
        » EVAL +
SWAP 1 - SWAP
      END SWAP DROP
NEG
    »
  » 'Driver' STO 'V
'(Q;X)' 'PV' STO '-M(
Q;X)' 'PM' STO { V
M Q R→C OVER ROT
PV PM } 'CST' STO
»
```

`Install`: Sitúe lo anterior en la pila (si lo tiene almacenado en `DIAG`, ejecute `DIAG` `RCL`), vaya al directorio dónde desee instalar los programas y pulse `EVAL`.