Cremona
Un programa de dimensionado de cerchas

Mariano Vázquez Espí[*]

Ondara, 14 de abril de 2007.

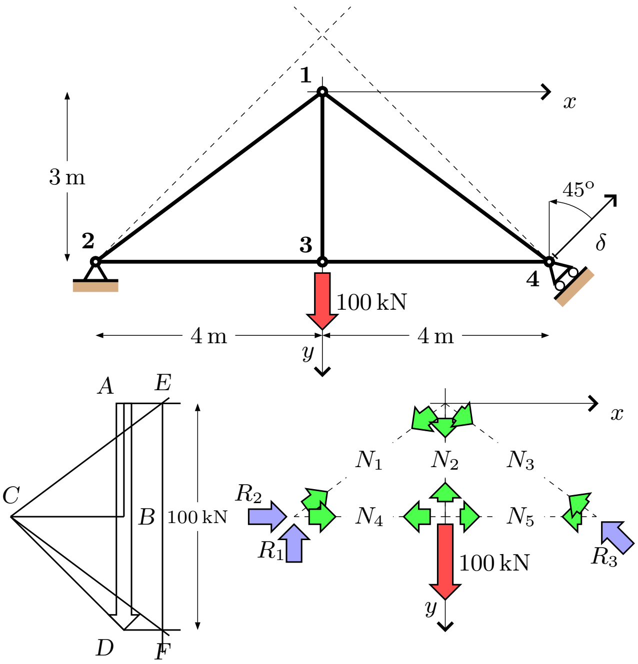

Figura 1.

Para analizar y calcular una cercha como la de la Figura 1, bastan las ecuaciones de equilibrio de la estática, debido a que el número de incógnitas de *fuerza* iguala exactamente el número de ecuaciones disponibles. En efecto, la articulación del apoyo izquierdo cuenta como dos bielas, cada una aportando una fuerza ($\vec{R}_1$, vertical; y $\vec{R}_2$, horizontal); el apoyo sobre el plano inclinado de la derecha (suponiendo que no existe rozamiento) cuenta como otra biela, es decir, otra fuerza de dirección conocida ($\vec{R}_3$, perpendicular al plano de apoyo); por último, la solicitación de cada barra (N_i) es otro par de fuerzas iguales y opuestas desconocidas; en total 8 fuerzas, igual a las ocho ecuaciones que pueden obtenerse considerando las dos ecuaciones cartesianas que pueden escribirse por cada nudo.

En cerchas como ésta —la gran mayoría— las reacciones (R_i) pueden calcularse con ecuaciones de equilibrio global ('tomando momentos' en el punto 2 se obtiene $R_3 = 71$ kN; por lo mismo en el punto 4, $R_1 = 50$ kN; finalmente por equilibrio horizontal, $R_2 = 50$ kN; también podría haberse trazado un funicular con el mismo objetivo).

Una vez determinadas las reacciones, si en cada par de ecuaciones de nudo sólo apareceren dos solicitaciones desconocidas, las ecuaciones pueden resolverse gráficamente mediante poligonos vectoriales de nudo, tal y como se ha hecho en el *diagrama de Cremona* representado en la figura (tales diagramas se deben en realidad a Maxwell[1]). Por ejemplo, el polígono *C*–*B*–*A*–*E*–*C* representa el equilibrio de todas las fuerzas que actuan *sobre* el nudo 2; del mismo modo A–D–F–E–A representa el equilibrio del nudo 3. En diagramas semejantes, todas las fuerzas aparecen representadas a escala y su valor se obtiene *simplemente midiendo*, sin ninguna complicación. En este caso, las solicitaciones resultan ser las indicadas en el Cuadro 1, una distribución simétrica como cabía esperar.

Cuadro 1:

Barra	nudos	N_i kN	$A_{i,\min}$ mm²	A_i	Δ_i mm
1	1–2	-83	463	465	−4,3
2	1–3	100	556	560	2,6
3	1–4	-83	463	465	−4,3
4	2–3	17	93	200	1,6
5	3–4	17	93	200	1,6

N_i : solicitación
$A_{i,\min}$: área estricta, $|N_i|/(0{,}18\,\mathrm{kN/mm^2})$
A_i : área diseñada
Δ_i : alargamiento

Para un acero con límite elastico σ_e de 260 N/mm², tensión admisible de 180 N/mm² (coeficiente de seguridad $\gamma = 1,44$) y módulo de Young de 210 kN/mm², un dimensionado *teórico* podría conducir a las secciones A_i indicadas en el cuadro. Nótese que se trata de *mala* teoría, pues los dos pares comprimidos fracasarían en la vida real debido a la inestabilidad inherente a la compresión (pero ésta es otra historia...); nóte-

[*]Departamento de Estructuras de Edificación de la Universidad Politécnica de Madrid

[1]Los procedimientos gráficos para el análisis de cerchas constituían la única herramienta práctica disponible hasta la aparición de máquinas de cálculo. Las *figuras recíprocas*, introducidas por Maxwell en 1864, constituyen el núcleo del análisis gráfico de solicitaciones. Sin embargo, su exposición resultó tan abstracta que su enorme utilidad pasó desapercibida. Un lustro despues, Jenkin (1869), mostró numerosos ejemplos prácticos de la aplicación de las *figuras recíprocas* de Maxwell, si bien se guió para su trazado de las reglas inventadas por W. P. Taylor, un artesano de la época que trabajaba para un constructor. Paralelamente, en Suiza, la obra *Die graphische Statik* de K. Culmann (1866 y 1875) contiene una exposición del método 'de las secciones' y del de 'los nudos', aunque sin apoyarse en la formulación de Maxwell. La estática gráfica llegó al Mediterráneo de la mano del italiano Cremona —*Le figure reciproche nella statica grafica*, 1872—, razón por la cual el *diagrama de Maxwell* fue conocido como *diagrama de Cremona* en la Península Ibérica.

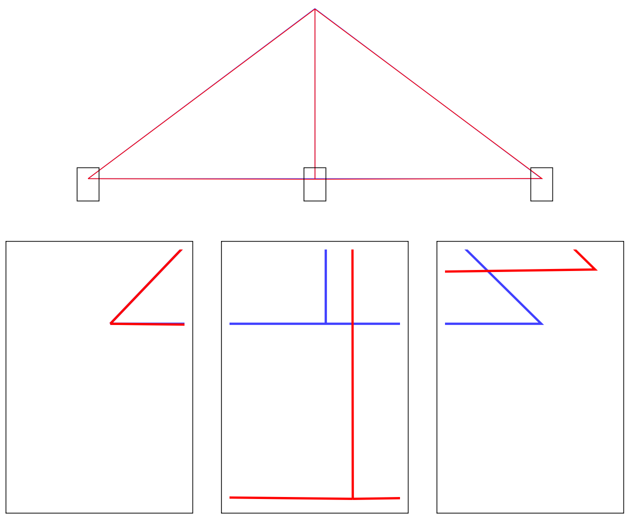

FIGURA 2.
En azul la geometría inicial; en rojo, la posición de equilibrio, con la estructura deformada.

se también que las barras 2–3 y 3–4 se dimensionan con una barra cilíndrica de 16 mm de diámetro, aunque podría ponerse una menos gruesa. Sea como fuere, una vez dimensionadas las barras pueden calcularse sus alargamientos Δ_i, tal y como se muestra en el cuadro.

Para averiguar la deformada un método divertido e instructivo consiste en dibujarla mediante un programa (como AUTOCAD), cuya precisión en el trazo permita distinguir tan pequeños alargamientos (desde luego mediante funciones ZOOM y cosas semejantes, de otro modo no se aprecia diferencia entre la geometría inicial y la deformada, véase la FIGURA 2, arriba). Una vez trazada la deformada, pueden medirse los desplazamientos de los nudos: por ejemplo, el apoyo derecho sube por el plano inclinado 4,5 mm, lo que determina el valor del desplazamiento δ de la FIGURA 1. Se puede comprobar el requisito de rigidez y, en su caso, dar el diseño global por válido y pasar al diseño de detalle… *Este es el método más exacto posible.*[2]

Cálculo analítico

El cálculo analítico permite escribir las dos ecuaciones correspondientes a cada nudo (componentes horizontales y verticales) y calcular las reacciones a la vez que las solicitaciones. Como siempre, hay que operar bajo un convenio de signos; también hay que convenir si se manejan fuerzas *sobre* el nudo o las fuerzas que el nudo *ejerce* sobre el resto. Aquí opto por la primera posibilidad. Con estas consideraciones, la ecuación 'horizontal' tendrá la forma:

$$X_l(\vec{Q}_l) + \sum_i X_l(\vec{N}_i) + \sum_j X_l(\vec{R}_j) = 0$$

en la que el operador $X_l(\cdot)$ representa la componente horizontal de la fuerza implicada en el nudo l: ya sea la carga $\vec{Q}_l$, las solicitaciones $\vec{N}_i$ de las barras que acometen al nudo, o las reacciones $\vec{R}_j$ de cada biela que el nudo tenga. Nótese que las bielas pueden estar orientadas según los ejes cartesianos (como en el apoyo izquierdo del ejemplo), pero no siempre pasa eso (tal ocurre en el apoyo derecho). La otra ecuación de cada nudo es formalmente igual, si se sustituye $X_l(\cdot)$ por $Y_l(\cdot)$, la componente vertical.

Si trasladamos las componentes conocidas a la derecha, el aspecto de las ecuaciones cambia a:

$$\sum_i n_{ki} N_i + \sum_j r_{kj} R_j = B_k$$

en la que k es el número de ecuación considerada. Tanto n_{ki} como r_{kj} representan la contribución al equilibrio de las fuerzas en barras y bielas; si la barra o la biela no acomete al nudo que se considera en la ecuación k, su aportación será nula (n_{ki} ó r_{kj} nulo). B_k representa la componente de fuerza en la ecuación k cambiada de signo, es decir, $-X_l(\vec{Q}_l)$ ó $-Y_l(\vec{Q}_l)$. Las incognitas son los módulos, N_i y R_j, de las fuerzas, es decir, $\{x_1 \dots x_{e+v}\} = \{N_1, \dots N_e, R_1, \dots R_v\}$, siendo e el número de barras (o elementos), y v el de bielas o condiciones de sustentación (vínculos); en consecuencia podemos escribir el sistema de todas las ecuaciones como

$$\sum_{i=1}^{e+v} A_{ki} x_i = B_k \qquad \text{para } k = 1 \dots 2n$$

siendo n el número de nudos. O en forma matricial, mucho más compacta:

$$[A_{ki}]\,\{x_i\} = \{B_k\} \qquad \text{o bien} \qquad \mathbf{Ax} = \mathbf{B}$$

Resuelto el sistema, se obtienen tanto el valor de las solicitaciones como el de las reacciones. Nótese que en condiciones normales $2n = e+v$ y si la estructura es viable el sistema tiene una única solución. Si por el contrario $2n > e+v$ el sistema no tiene, *en general*, solución: la estructura es una mecanismo capaz de moverse dinámicamente bajo la acción de cargas *genéricas* —pero véanse los ejemplos más adelante. Finalmente, si $2n < e+v$ y la estructura es viable, existirán infinitas soluciones posibles. En los dos últimos casos, y con técnicas algebraicas bien conocidas —como el método de los mínimos cuadrados— puede extraerse alguna solución (si existe alguna) matemáticamente significativa (y puede que estructuralmente también), pero su *análisis* sólo puede hacerse considerando la rotura o la deformación elástica, y en general es necesario *fijar* el dimensionado de las barras *previamente*: el *método de análisis* ha de ser necesariamente hiperestático, al contrario que en el primer caso donde las ecuaciones de la estática bastan (método isostático).

Cálculo automático

¿Matrices? ¿Sumatorios? ¡Calculadora! Puede… Sigo calculando cerchas con lápiz y papel (incluso, en ocasiones, sin regla, a mano alzada), pero es tal la *mayoritaria fascinación* que ejercen hoy las máquinas de

[2] Aunque la aparición de los primeros ordenadores propició la paulatina desaparición de los métodos gráficos, la muy posterior aparición y divulgación de los programas de dibujo hacen posible volver a ellos, pues la que siempre fue su relativa desventaja, la imprecisión del trazo, ha desaparecido completamente: entre estudiantes de arquitectura e ingeniería, supuestamente proclives al dibujo, cabría esperar su 'renacimiento'.

cálculo que quizá esté justificado ofrecer al público un programa automático: aquí va el que hice para una de las mías, la HP48GX.

Los nudos son simplemente puntos, de manera que pueden representarse mediante una lista de números complejos. Cada número $(x;y)$ contiene las coordenadas en los ejes elegidos. La secuencia de puntos en la lista los numera implícitamente. Para la estructura de la FIGURA 1, los puntos se representan por la lista:

```
{ (0;0) (-4;3) (0;3) (4;3) }
```

objeto que puede almacenarse bajo un nombre, por ejemplo, `P`.

Las barras conectan nudos a pares, de manera que dando el nudo inicial y final quedan determinadas: un formato conveniente es una lista de vectores de la forma `[` n_i n_f `]`, siendo n_i y n_f la posición de los nudos de la barra en la lista `P`. Las barras de la figura se representan como:

```
{ [1 2][1 3][1 4][2 3][3 4] }
```

y quedan numeradas implícitamente por su posición en la secuencia. Lo más seguro es almacenarla en una variable, por ejemplo en `B`.

Por razones precisas, las bielas están emparentadas tanto con barras como con cargas. Cada biela representa una fuerza desconocida a medias: desconocemos su magnitud pero no su dirección. Por ejemplo, en la figura, cualquiera que sea su valor, $\vec{R_3}$ aportará al equilibrio del nudo componentes horizontal y vertical de valor $-0,71R_3$. De hecho, el valor $-0,71$ es el coeficiente r_{k8} ($k = 7$ ó $k = 8$) de las ecuaciones analíticas del nudo 4. Estos dos coeficientes pueden agruparse en un número complejo y antecederles con el número del nudo en el que se sitúa la biela. Como la fuerza es desconocida, sólo interesa la proporción entre sus componentes, de manera que el apoyo sobre el plano inclinado de la figura puede representarse con una lista `{ 4 (-1;-1) }` (aunque `{ 4 (-,71;-,71) }` valdría igual de bien, pero sería más largo de escribir), y el conjunto de las bielas por la lista de todas ellas:

```
{ { 2 (0;-1) } { 2 (1;0) } { 4 (-1;-
1) } }
```

Nótese que cada reacción tiene su propio convenio de signos, el dibujado. Sin embargo, en la lista de bielas el signo de sus componentes se rige por el convenio general de los ejes xy. Si la reacción resulta negativa, actúa en sentido contrario al dibujado. La lista de vínculos se almacena en lugar seguro, por ejemplo en `V`.

Por último, las cargas pueden expresarse con un convenio similar: cada carga es una lista formada por el nudo en que se aplica y por el número complejo formado por sus componentes; todas ellas se agrupan en una lista. En el caso de la figura, con una única carga:

```
{ { 3 (0;100) } }
```

que se almacena en `Q`. Una misma cercha puede estar sometida a distintas cargas independientes, almacenadas en variables distintas: `Q1`, `Q2`, ...

Los tres primeros objetos, `P`, `B` y `V`, definen una cercha particular. Junto con las cargas de un caso contienen toda la información relevante para el análisis de ese caso. Nótese lo apropiado que resulta diseñar programas que *leen* información, en vez de hacer estúpidas preguntas sobre el número de nudos de la cercha y otras zarandajas: la misma información puede usarse siempre que sea necesario (y con distintos propósitos) sin necesidad de teclearla otra vez. Además si se cometieron errores al escribirla por vez primera, basta corregirlos para intentarlo una segunda.

Ahora lo único que resta es escribir el programa que forme las matrices **A** y **B** y resuelva el sistema para obtener **x**; los programas para ello (y otros útiles para la comprobación de la estructura) se muestran en el anejo final, indicando además las instrucciones necesarias para almacenarlos en la memoria de la calculadora.

Uso de los programas

Una vez escritos y almacenados todos los programas y la variable `CST` en un 'directorio'[3], y con la 'ruta actual' apuntando a él, pulsando [CST] aparecerán [CREM], [LONGI], [MULV], [DIVVE], [DOT] y [→OBJ]. Los últimos son útiles aunque no esenciales y se explican más adelante; es `Cremona` el que se encarga de escribir y resolver el sistema de ecuaciones. Con [NXT], aparecerán el resto de definiciones que se hayan almacenado en `CST`.

En cualquier 'subdirectorio' desde allí, con [VAR] se tendrá acceso a las variables donde se haya ido almacenando la información de una cercha particular (en el ejemplo: [P], [B], etc.).[4]

`Cremona.` Recoge de la pila una lista de puntos, una de barras, una de vínculos y una de cargas, y deja en la pila un vector con las solicitaciones de las barras y otro con la magnitud de las reacciones. Para el ejemplo de la figura, la secuencia sería:

[VAR] [P] [B] [V] [Q] [CST] [CREM]

que deja en la pila el vector de solicitaciones (en la posición 2) y el de reacciones (en la 1), en todo concordantes con las calculadas previamente. Como las solicitaciones son de gran utilidad, lo suyo es guardarlas, por ejemplo en `N`.[5]

Dimensionado. Para el acero considerado en el ejemplo, pueden obtenerse las áreas estrictas con la secuencia [N] `0,18` [÷], que arroja en la pila un vector de áreas en mm^2 (con los mismos valores que los indicados en el CUADRO 1, en la columna $A_{i,\min}$). Este vector se puede editar para hacer retoques razonables[6], y de paso suprimir el irrazonable signo negativo, heredado de las solicitaciones. Como de costumbre, lo mejor es almacenarlo, por ejemplo en `A`.

[3] Probablemente el mejor sitio es `HOME` si se va a usar mucho. Un lugar razonable es un directorio específico, como `CERCHAS`, del que 'cuelguen' los directorios que contengan las cerchas objeto de cálculo.

[4] Con [VAR] en el directorio de programas se tendrá acceso directo al resto de los programas que figuran en el Anejo: [STOP], [SOLE] y [OPVEC].

[5] En lo que sigue omitiré las pulsaciones de [CST] o [VAR], dando por supuesto que se pulsará lo que convenga para acceder a la variable que se menciona.

[6] Para ajustar a catálogos comerciales, por ejemplo.

Con las áreas decididas y las longitudes de las barras se puede calcular el volumen de la cercha. `Longitudes` suministra el vector de longitudes de barras tras coger de la pila la lista de puntos y la de barras. El producto escalar de esos dos vectores da el volumen de la cercha, pero en $\mathrm{mm}^2 \cdot \mathrm{m}$, dado que longitudes y áreas están en distintas unidades (ciertamente puede usarse la misma unidad en ambos casos). En definitiva, la secuencia:

`P` `B` `LONGI` `A` `DOT` 1E6 `÷` 7850 `×`

arroja en la pila el peso de la cercha, en kilogramos en este caso.

Requisito de rigidez

Para comprobar si además de ligera es rígida puede emplearse el principio de los trabajos virtuales, que reza así:

$$\sum \vec{F}_i \cdot \vec{\delta}_i^\star = \sum N_j \cdot \Delta_j^\star$$

en la que $\vec{F}$ y N son un conjunto de fuerzas exteriores en los nudos y de solicitaciones en las barras que *están en equilibrio*; por su parte, $\vec{\delta}^\star$ y $\Delta^\star$ son un conjunto de desplazamientos de los nudos y los alargamientos que producen en las barras (*alargamientos compatibles*). El superíndice $\star$ simboliza que uno y otro conjunto *no necesitan guardar más relación entre sí* que referirse a la misma cercha. (De uno de los conjuntos se suele decir que es *virtual* o *imaginario* respecto al otro.)

Para calcular el desplazamiento δ de la figura empleamos la misma cercha, pero cargada con una fuerza unidad en el mismo punto, sentido y dirección de δ. Esto determina las fuerzas exteriores y las solicitaciones a emplear en la ecuación anterior. Como la deformación es arbitraria respecto a las fuerzas, podemos escoger la deformación real de la estructura inicial, de la que conocemos o podemos calcular los alargamientos de las barras y sólo desconocemos los desplazamientos. En la ecuación anterior el primer término quedará como $\mathbf{1}\,\mathrm{kN}{\cdot}\delta$ puesto que, a parte de la fuerza unidad, sólo existen las reacciones, aplicadas en puntos y direcciones *sin* desplazamiento, y con contribución nula al trabajo virtual exterior. El segundo término quedará como $\sum N_j^\star \Delta_j$, donde $N_j^\star$ representa las solicitaciones en las barras bajo la carga unitaria en la dirección de δ, y Δ_j los alargamientos reales bajo la carga de 100kN, virtuales e imaginarios *para la carga unidad* —pero bien reales para los 100kN. En consecuencia,

$$\delta = \frac{1}{\mathbf{1}\,\mathrm{kN}} \sum N_j^\star \Delta_j$$

es decir, el *producto escalar* del *vector* 'solicitaciones' producido por la fuerza unidad y el *vector* 'alargamientos' bajo 100 kN (salvo el factor $\mathbf{1}\,\mathrm{kN}^{-1}$ necesario para la coherencia de las unidades).

Con la máquina la secuencia de cálculos es como sigue. La carga unidad se representa como `{ { 4 (0,71;-0,71) } }`, como siempre almacenada en una variable, por ejemplo `QV`. La secuencia:

`P` `B` `V` `QV` `CREM` `DROP`

deja en la pila el vector $\{N_j^\star\}$. Ahora hace falta formar el vector de los alargamientos y realizar el producto escalar (como se hizo para calcular el peso). Pero también se pueden calcular automáticamente. La fórmula para una barra es:

$$\Delta = \frac{N}{EA} L$$

El problema es que tanto las solicitaciones como las áreas y las longitudes de las barras están almacenadas en forma de vectores y necesitamos un programa que calcule un vector a partir de otros dos, de tal manera que el nuevo tenga por componentes la división o el producto de las correspondientes componentes de los dos primeros. Esto es lo que hacen `DivVec` y `MulVec`: toman dos vectores de la pila y arrojan en ella el resultado de dividir o multiplicar *componente a componente*.[7] Ambos usan `OpVec`, que es la versión general: coge dos vectores de la pila y un *programa*, y arroja en ella el resultado de evaluar el programa sobre las componentes correspondientes de ambos vectores, sucesivamente.

La secuencia `N` `A` `DIVVE` arroja en la pila el vector de tensiones; 210 `÷` calcula el vector de deformaciones; por último, `P` `B` `LONGI` `MULV` multiplica cada deformación por la longitud y deja en la pila los alargamientos.

Ahora `DOT` ejecuta el producto escalar entre las solicitaciones de la carga unidad (que estaban en la pila hace rato) y los alargamientos recien calculados. El resultado es cuanto sube el extremo derecho por el plano inclinado (o cuanto baja, si el valor es negativo) ¿En que unidades? $\mathrm{kN/mm^2/(kN/mm^2)} \times \mathrm{m}$, es decir, en metros.

Resolución de ecuaciones

Hasta ahora nada he dicho de `Soleq`. Se trata de un escueto programa (¡con una sola instrucción!), que es ejecutado por `Cremona` casi al final, cuando las matrices $\mathbf{A}$ y $\mathbf{B}$ han sido ya formadas. En la HP48GX dividir matrices *tiene* sentido. Si el sistema de ecuaciones se expresa como $\mathbf{Ax} = \mathbf{B}$, formalmente $\mathbf{x} = \mathbf{B}/\mathbf{A}$ es la solución buscada; la división aquí simboliza la *inversión de una matriz*, lo que sólo tiene sentido, en principio, para matrices cuadradas. Por ello, si `Soleq` contuviera `« / »`, `Cremona` resolvería sólo aquellas cerchas con igual número de incógnitas (barras más bielas) que ecuaciones, y solamente éstas. Pero si en `Soleq` aparece `LSQ` en vez de `/` (como es el caso), `Cremona` sugerirá una solución si es que existe alguna, cualquiera que sea la relación entre el número de barras y bielas, de una parte, y el de nudos por la otra. Lo que cambia aquí es simplemente el método de resolución de ecuaciones. `LSQ` es el acrónimo del inglés ***l**east **sq**uares*, es decir, 'mínimos cuadrados'; y con este método es posible obtener una solución cuando $2n \neq e+v$.

Por ejemplo, ¿qué ocurre si el apoyo derecho se sustituye por una articulación como en la Figura 3? Los vínculos cambian, tenemos ahora cuatro bielas, es decir, la lista:

[7] Justamente lo contrario de lo que se entiende por multiplicar vectores (producto vectorial): en nuestro contexto de cálculo, objetos como las 'longitudes' unas veces viene bien que sean vectores, pero en otras, es más adecuado considerarlos como listas de números.

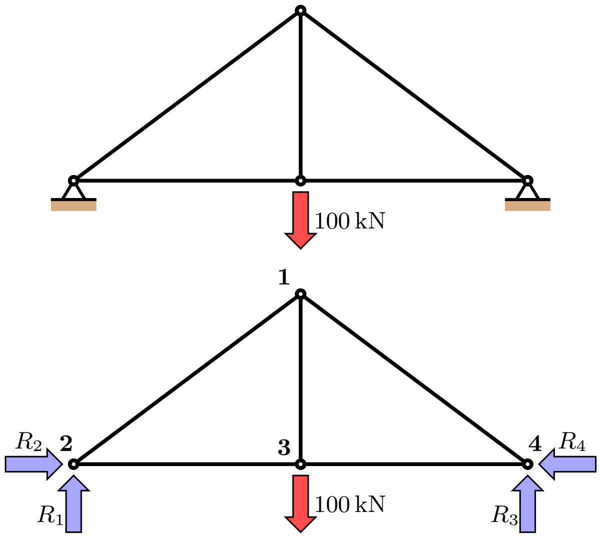

FIGURA 3.

```
{ { 2 (0;-1) }{ 2 (1;0) }{ 4 (0;-1)
}{ 4 (-1;0) } }
```

almacenada, por ejemplo, en `V1`. Ahora, la secuencia `P` `B` `V1` `Q` `CREM` arroja en la pila nuevas reacciones y solicitaciones. Las reacciones son $R_1 = R_3 = 50\,\mathrm{kN}$ y $R_2 = R_4 = 33\,\mathrm{kN}$. Las solicitaciones calculadas son `[-83 100 -83 33 33]`. Puede comprobarse que globalmente —o nudo a nudo— hay equilibrio entre todas las fuerzas implicadas, pero ¿tiene sentido *físico*? Un teórico purista contestará ‘‘no’’ según el siguiente argumento:

> Puesto que en los nudos 2 y 4 hay articulaciones *fijas*, la distancia entre estos nudos es fija igualmente y, en consecuencia, las barras 2–3 y 3–4 no pueden experimentar alargamiento, ni deformación ni tensión ni, por tanto, solicitación: las solicitaciones facilitadas por `Cremona` son incorrectas, pues en esas barras es fácil ver que la solicitación es nula.

Sin embargo, un@ diseñador@ más interesad@ en el proyecto de estructuras y en predecir como se comportarán, podría optar por un argumento diferente:

> Supongamos que el coeficiente de seguridad en el diseño de cada pieza sea $\gamma = 2$. Si diseño *las estructuras que soportan la cercha*, a través de las articulaciones 2 y 4, para resistir una fuerza horizontal de $33\gamma = 66\,\mathrm{kN}$ y con suficiente ductilidad, cuando la carga en la cercha sea $100\gamma = 200\,\mathrm{kN}$, estas estructuras comenzarán a *ceder* (si no lo han hecho antes), aumentando la distancia entre 2 y 4. En consecuencia, las barras 2–3 y 3–4 comenzarán a alargarse y entrarán en tracción *a partir de ese momento*. `Cremona` me informa que la máxima tracción en esas barras será de $33\gamma = 66\,\mathrm{kN}$. Y de hecho una vez que diseñe las barras puedo incluso calcular el alargamiento experimentado entre 2 y 4 hasta que la cedencia de las articulaciones se detiene, gracias a la ‘entrada’ en tensión de los tirantes.

¿Existen las articulaciones que *nunca* se mueven? Si su respuesta es ‘‘sí’’ es usted un purista y `Cremona`, con la actual versión de `Soleq`, no le será de utilidad: sustituya `LSQ` por `/`. Cuando intente resolver esta última cercha recibira el mensaje:[8]

```
/ Error:
INVALID DIMENSION
```

Si su respuesta es ‘‘no’’, ¡bienvenid@ al grupo! Este programa, en su actual versión, podría darle en ocasiones pistas interesantes.

La cercha de la FIGURA 3 será calificada por muchos como *hiperestática* indicando con ello que sólo la consideración de las deformaciones de las barras con secciones previamente definidas, permite calcular con ‘rigor y precisisón’ las solicitaciones. Esto es cierto bajo la *carga de servicio* de 100 kN, pues el comportamiento debe ser elástico. Pero ya hemos visto que con sólo las ecuaciones de la estática, es posible obtener solicitaciones en equilibrio que serán proporcionales a las de la rotura de la cercha si y sólo si ajustamos el diseño a esas solicitaciones. A fin de cuentas, `Cremona` permite obtener un diseño seguro (pero no sabemos si rígido y habría que comprobar que dúctil) incluso en esta situación *hiperestática*. Todo depende de si nos permitimos *pensar isostáticamente*.

Cerchas funiculares

En esencia, la suma $R_2 + N_4$ sobre el nudo 2 debe equilibrar la componente horizontal de N_1, que es (bajo 100kN) de 66 kN. `Cremona` sugiere dividir el trabajo entre la articulación y el tirante a partes iguales. Hay infinitas alternativas a ese reparto, pero ¿hay alguna alternativa mejor a la de `Cremona`? Depende de cuanto *cuesten* el tirante y la articulación; no estoy pensando en costes monetarios —muy poco interesantes. Por ejemplo, si la estructura que ha de soportar la cercha es el suelo y tiene suficiente resistencia, podemos encomendarle todo el trabajo: $R_2 = 66$ y $N_4 = 0$ sería una solución posiblemente mejor. Este diseño consiste, de hecho, en suprimir las dos barras del tirante, véase la FIGURA 4. Ahora en la lista de barras sólo quedan tres: `{ [1 2][1 3][1 4] }`. Con esta lista almacenada en `BF`, la secuencia `P` `BF` `V1` `Q` `CREM` arroja en la pila nuevas reacciones y solicitaciones:

```
[-83 100 -83]
[50 67 50 67]
```

Es fácil en este caso comprobar a mano que estos son los resultados correctos para esta peculiar estructura. Se trata de hecho, de una estructura *funicular*: si además de 100kN verticales hubiera algo de carga horizontal en el nudo 3, el pendolón 1–3 adaptaría su posición al igual que lo hace un hilo. De hecho, un purista diría que esta cercha ni siquiera es *estructura*, pero entonces ¿qué son los cables o barras de los que cuelga una lámpara?

[8] Es lo habitual en los programas que se intercambian por dinero: pruebe a que alguno de ellos calcule (sin recurrir a sortilegios) un cable del que cuelga una lámpara...

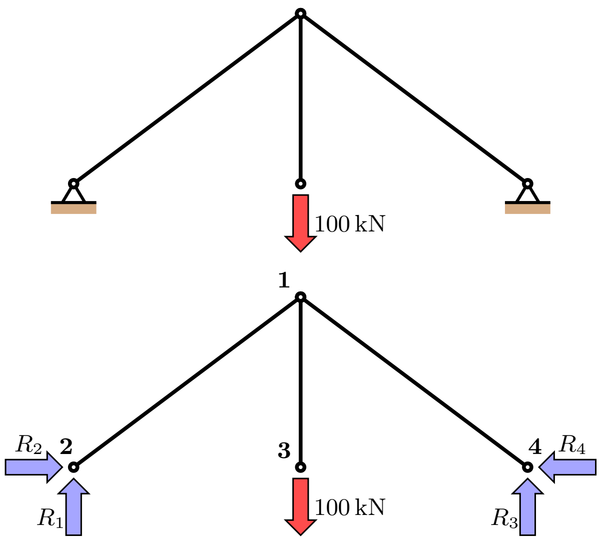

FIGURA 4.

Diseño de cerchas

En resumen, `Cremona` ofrece para *cualquier estructura de barras* (de codales y tirantes), un conjunto de solicitaciones y reacciones en equilibrio estático con las acciones, si es que existe algún conjunto semejante. La solución puede ser estable o inestable y averiguarlo es tarea de quien usa el programa (piense en la cercha simétrica respecto al eje horizontal de la última cercha considerada, pero con los 100kN dirigidos todavía hacia abajo). Además, con áreas adecuadas (iguales o mayores que las que resultan de $\gamma N_i/\sigma_\mathrm{e}$), ese conjunto será proporcional a las solicitaciones en la rotura de la cercha; la razón de la proporción entre éstas y aquéllas será justamente el coeficiente de seguridad γ. Faltará comprobar la ductilidad y la rigidez de la estructura. Puesto que dado un esquema `Cremona` permite obtener secciones bastante razonables para las barras, `Cremona` es en realidad un programa de *dimensionado* de cerchas.

Desde un punto de vista más abstracto, `Cremona` resuelve con su mejor criterio *cualquier* sistema de ecuaciones con fuerzas que esté asociado a una cercha. Técnicamente, *su mejor criterio* consiste en obtener el vector de incógnitas con menor norma euleriana y que satisface el sistema, si es que existe alguno… (¿Qué significa 'norma euleriana'? En esencia el módulo de un vector, pero consulte un buen libro de matemáticas…) Desafortunadamente, la mínima norma euleriana del vector de solicitaciones y reacciones no puede relacionarse positivamente con ninguna propiedad estructural que sea deseable minimar. Por tanto, los diseños propuestos por el programa no serán en general óptimos.

Finalmente…

`STOP` es el último programa del que tengo que decir algo. Se trata de una versión generalizada de las rutinas de 'aritmética en memoria' (inspeccione ↰ `VAR` `ARITH`): toma de la pila un *valor*, una *expresión algebraica* y un *programa* y almacena en *expresión algebraica* el resultado de evaluar el *programa* sobre *expresión algebraica* y *valor*. Si piensa hacer uso de `STOP` en sus propios programas, note que la *expresión algebraica* debe corresponder a una ubicación precisa de la memoria (es decir, ese misma *expresión algebraica* debe ser admitida por `STO` como segundo argumento sin producir error), y que *programa* debe tomar dos argumentos de la pila y devolver un valor (mire los ejemplos de uso en `Cremona`).

`STOP`, `OpVec`, `MulVec`, `DivVec` y `Soleq` son programas de utilidad general y su lugar es `HOME` o algún directorio específico desde donde 'cuelguen' los programas que operan con vectores y matrices (y desde dónde podrán estos últimos invocar a aquellos).

Para observar *qué* hace cualquiera de los programas, y por tanto entenderlos, puede usar `PRG` `NXT` `RUN` `DBUG`. *Conviene que lo haga: toda la responsabilidad al usar un programa es suya…*

Anejo

Código fuente de los programas

Sitúe la ruta actual en el directorio elegido para los programas y ejecute las siguientes instrucciones.

```
« 1 4
   START 4 ROLL DUP
      IF TYPE 6 ==
      THEN EVAL
      END
   NEXT → p b v c
   « p SIZE b SIZE v SIZE c SIZE → nn nb
nv nc
      « nn 2 * DUP nb nv + 2 →LIST 0 CON
SWAP  + 0 CON → a q
         « 1 nb
            FOR i 'b(i)' EVAL → b
               « 'RE(p(b(2)))-RE(p(b(1)))'
EVAL 'IM(p(b(2)))-IM(p(b(1)))' EVAL → lx ly
                  « 'ABS(p(b(2))-p(b(1)))'
EVAL INV DUP 'lx' STO* 'ly' STO* lx
'a(2*b(1)-1;i)' STO lx NEG 'a(2*b(2)-1;i)'
STO ly 'a(2*b(1);i)' STO ly NEG
'a(2*b(2);i)' STO
                  »
               »
            NEXT 1 nv
            FOR i 'v(i)' EVAL → v
               « 'RE(v(2))/ABS(v(2))' EVAL
'a(2*v(1)-1;i+nb)' STO 'IM(v(2))/ABS(v(2))'
EVAL 'a(2*v(1);i+nb)' STO
               »
            NEXT 1 nc
            FOR i 'c(i)' EVAL → v
               « 'RE(v(2))' EVAL
'q(2*v(1)-1)'
                  « -
                  » STOP 'IM(v(2))' EVAL
'q(2*v(1))'
                  « -
                  » STOP
               »
            NEXT q a Soleq OBJ→ DROP nv
 + →ARRY → r
            « nb  + →ARRY r
            »
         »
      »
   »
»
'Cremona'
[ENTER]
[STO]
« → o
   «DUP EVAL ROT o EVAL SWAP STO
   »
»
'STOP'
[ENTER]
[STO]
« LSQ
»
'Soleq'
[ENTER]
[STO]
« 1 2 START SWAP DUP
      IF TYPE 6 ==
      THEN EVAL
      END
   NEXT → p b
   « 1 b SIZE
      FOR i 'b(i)' EVAL → b
         « 'ABS(p(b(2))-p(b(1)))' EVAL
         »
      NEXT b SIZE  + →ARRY
   »
»
'Longitudes'
[ENTER]
[STO]
« → o
   « OBJ→ EVAL →LIST SWAP OBJ→ EVAL
→LIST SWAP 2 o DOLIST OBJ→  + →ARRY
   »
»
'OpVec'
[ENTER]
[STO]
«
   « *
   » OpVec
»
'MulVec'
[ENTER]
[STO]

«
   « /
   » OpVec
»
'DivVec'
[ENTER]
[STO]
{ Cremona
Longitudes MulVec
DivVec { "DOT"
« DOT
» } { "OBJ→"
« "OBJ→" DROP
» }
añada más información a su gusto
}
'CST'
[ENTER]
[STO]
```