.

Una herramienta de Cálculo Computacional para la enseñanza de las Matemáticas

Roberto Rodríguez del Río
Universidad Complutense de Madrid

Cursos de verano

San Lorenzo de El Escorial, julio de 2000

Una herramienta de Cálculo Computacional para la enseñanza de las Matemáticas

Roberto Rodríguez del Río
Departamento de Matemática Aplicada
Facultad de Ciencias Químicas
Universidad Complutense Madrid
rr_delrio@mat.ucm.es

Cursos de Verano
UNIVERSIDAD COMPLUTENSE MADRID
San Lorenzo de El Escorial, julio de 2000

1. Introducción

En la actualidad existen numerosos *asistentes matemáticos* en el mercado, casi todos ellos son aprovechables de una manera o de otra para utilizarlos en la enseñanza de las Matemáticas en alguno de sus niveles. Aquí vamos a presentar uno de estos asistentes: MatLab.

MatLab es un programa ampliamente difundido en la Universidad, no se puede decir lo mismo, sin embargo, de los niveles no universitarios de la enseñanza. Pero, como se decía antes, creemos que hay algunos elementos del programa que se pueden utilizar incluso en niveles tales como el Bachillerato actual, algunas de estas herramientas las analizaremos en estas notas.

La primera versión de MatLab fue escrita a finales de los años setenta, empezándose a utilizar en las universidades de New Mexico y Stanford, tenía como finalidad utilizarse en cursos de teoría de matrices, álgebra lineal y análisis numérico. El nombre MatLab es una abreviatura de las palabras *MATrix LABoratory*. MatLab es un sistema interactivo para cálculos científicos y de ingeniería basado en las matrices. Con él se pueden resolver complejos problemas numéricos sin necesidad de escribir un programa específico para ello, aunque también es posible programar, con un lenguaje de programación bastante sencillo e intuitivo.

Además el programa dispone, dependiendo de la versión, de diferentes módulos *(Toolboxes)* que permiten resolver problemas específicos. En particular, la versión para Estudiante de MatLab[1] dispone, entre otras, de la *(Matlab Symbolic Toolbox)*, que es la herramienta que permite hacer Cálculo Simbólico, está basada en el núcleo del programa Maple y permite disponer, en un sólo programa, de dos asistentes matemáticos muy potentes: MatLab, con sus capacidades de cálculo numérico correspondientes, y Maple, que permite un tratamiento simbólico de los problemas.

En estas notas vamos a ver, en primer lugar, un ejemplo de simulación de un problema de Matemática Aplicada. Después veremos dos de las herramientas interactivas de que dispone la versión Estudiante de MatLab, que pueden utilizarse sin demasiados conocimientos acerca de las generalidades del programa.

2. Un ejemplo: la cadena pesada

Vamos a analizar un ejemplo sobre Teoría de Control[2] desarrollado por N. Petit y P. Rouchon del *Centre Automatique et Systèmes, École des Mines de Paris*, ver [P].

2.1. El modelo

Tenemos una cadena que cuelga de un carrito que podemos mover (ver figura 1). El carrito se desliza, sin rozamiento, sobre un rail de una longitud determinada, vamos a suponer que es $1{,}5L$, donde L es la longitud de la

[1]El contenido de estas notas está basado en la versión 5 de Estudiante, *The Student Edition of MATLAB*, comercializada en España por Prentice Hall.

[2]La teoría matemática de Control, es la parte de las matemáticas en la cual se estudian los procesos de formalización y los métodos de elección del modo (óptimo, en cierto sentido definido de antemano) para ejecutar un proceso dinámico sujeto a control. Por lo general, dicho proceso dinámico puede ser descrito mediante ecuaciones diferenciales, integrales, funcionales y en diferencias finitas que dependen de un sistema de funciones o parámetros, llamados controles y que se han de determinar.

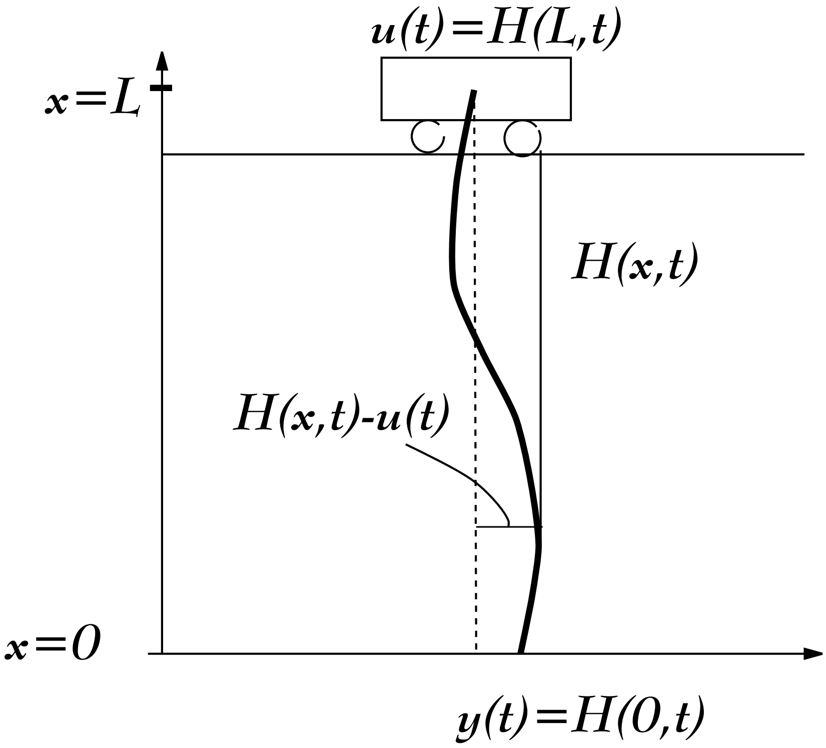

Figura 1: Cadena

cadena. Supondremos también que la masa está uniformemente distribuida a lo largo de la cadena.

Una manera de estudiar el modelo sería suponer que la cadena está formada por una cantidad finita de pequeños péndulos encadenados: sus trayectorias se podrían determinar explícitamente por medio de las trayectorias de sus puntos de unión (de un péndulo con otro). Sin embargo, este procedimiento involucra numerosas derivadas, de manera que si el número de péndulos es muy alto el tratamiento matemático es bastante complicado.

Otra forma de atacar el problema es considerar el caso continuo. Aparece, en este caso, una ecuación de ondas en la que se considera que interviene, además de las tensiones de la cuerda que se consideran habitualmente en la ecuación de ondas clásica, la gravedad. Esta ecuación, ya estudiada por Daniel Bernoulli en 1738, es la siguiente:

$$\begin{cases} \dfrac{\partial^2 H}{\partial t^2} = \dfrac{\partial}{\partial x}\left(gx\dfrac{\partial H}{\partial x}\right) & ; \text{si } x \in [0, L],\ t > 0 \\ H(L, t) = u(t) \end{cases}$$

donde x es la altura, el eje vertical (ver figura 1); para un t fijo, $H(x,t)$ es la posición del punto de la cadena que se encuentra a la altura x, medida desde el eje X; g es la gravedad; L es la altura a la que se encuentra el carrito, desde el punto más bajo de la cadena (también es la longitud de la cadena, considerando que los desplazamientos horizontales sobre posiciones de equilibrio son pequeños) y $u(t)$ es la posición del carrito con respecto del tiempo, el *control*.

2.2. La solución del problema

Si consideramos las condiciones iniciales $H(x,0) = 0$ y $\dfrac{\partial H}{\partial t}(x,0) = 0$. Se puede demostrar que (ver [P])

$$H(x,t) = \frac{1}{2\pi}\int_0^\pi \left[y(t + 2\sqrt{\frac{x}{g}}\operatorname{sen}\theta) + y(t - 2\sqrt{\frac{x}{g}}\operatorname{sen}\theta) \right] d\theta$$

donde $y(t) = H(0,t)$, es la posición del extremo libre de la cadena. Esta relación nos muestra que hay una correspondencia uno a uno entre soluciones (diferenciables) de la ecuación y funciones (derivables) $y(t)$. Por otra parte, para cada función $y(t)$ (dos veces derivable, al menos), podemos calcular el *control* de la siguiente forma:

$$u(t) = H(L,t) = \frac{1}{2\pi}\int_0^\pi \left[y(t + 2\sqrt{\frac{L}{g}}\operatorname{sen}\theta) + y(t - 2\sqrt{\frac{L}{g}}\operatorname{sen}\theta) \right] d\theta$$

Así, por ejemplo, si queremos saber cómo hay que mover el carrito, es decir, calcular $u(t)$, para llevar el sistema desde una posición de equilibrio $H \equiv 0$ hasta otra $H \equiv D$. Construimos una función $y(t)$ que valga 0 si $t \leq 0$ y D para t suficientemente grande (al menos $t > 4\sqrt{\frac{L}{g}}$) y calculamos $u(t)$ con la fórmula anterior.

2.3. La implementación en MatLab

Vamos a hacer los cálculos, con MatLab, de un caso particular. Construimos una función $y(t)$ con las condiciones descritas anteriormente

$$y(t) = \begin{cases} 0 & ; \text{si } t < 0 \\ 1{,}5L(\frac{t}{T})^2(3 - 2(\frac{t}{T})) & ; \text{si } 0 \leq t \leq T \\ 1{,}5L & ; \text{si } t > T \end{cases}$$

con $T = 4\sqrt{L/g}$.

Para trabajar más cómodamente con la función, la guardamos en un fichero `.m`, que son los ficheros en los que se guardan los programas de MatLab. Para ello, habría que escribir con algún editor de texto el siguiente programa,

```
function y=extremo(t)

% funcion y(t). Extremo de la cadena

L=1.5; % longitud de la cadena en m
g=9.81; % gravedad en m/s^2
T=4*sqrt(L/g); % tiempo de transferencia

y=0.*(t<0)+(3*L/2)*((t/T).^2).*(3-2*(t/T)).*((0<=t)&(t<=T))+...
   (3*L/2).*(t>T);
```

(Hemos elegido $L = 1{,}5$ metros y $T = 4\sqrt{L/g}$. Todo lo que se escribe después de % son comentarios que el programa no interpreta cuando se ejecuta. Sobre la sintaxis, se puede consultar [R] o [V].)

Para dibujar la gráfica de $y(t)$

```
>>t=-1:0.001:2.5; % generamos una tabla de valores para t
>>y=extremo(t); % sustituimos en y(t)
>>plot(t,y) % dibujamos (t,y)
```

y obtenemos la gráfica de la figura 2.

La construcción de la función $u(t)$, el control del sistema, es algo más complicada, por estar definida mediante una integral. Una forma de hacerlo sería utilizar el siguiente programa (guardándolo con el nombre `control.m`):

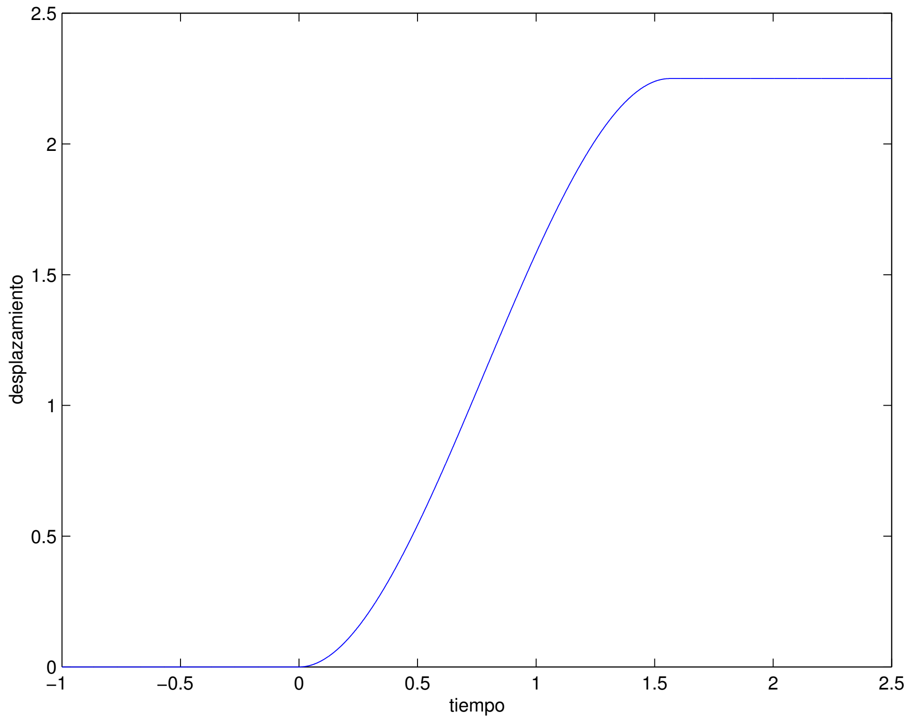

Figura 2: Gráfica de $y(t)$

```
function u=control(t)

% funcion u(t). Control: movimiento del carrito.
L=1.5; % longitud de la cadena en m.
g=9.81; % gravedad en m/s^2
k=length(t); % n'umero de puntos de t
theta=linspace(0,pi,100); % discretizaci'on de theta
% bucle para calcular todos los valores de u
% usando la regla del trapecio para aproximar la integral
% mediante el comando trapz
for i=1:k
   f=extremo(t(i)+2*sqrt(L/g)*sin(theta))+...
      extremo(t(i)-2*sqrt(L/g)*sin(theta));
   u(i)=(1/(2*pi))*trapz(theta,f);
end
```

Ahora para dibujar las gráficas de las dos funciones, $y(t)$ y $u(t)$, en la misma figura. Primero calculamos los valores de $u(t)$ para el mismo rango de t

```
>>t=-1:0.001:2.5;
>>u=control(t);
```

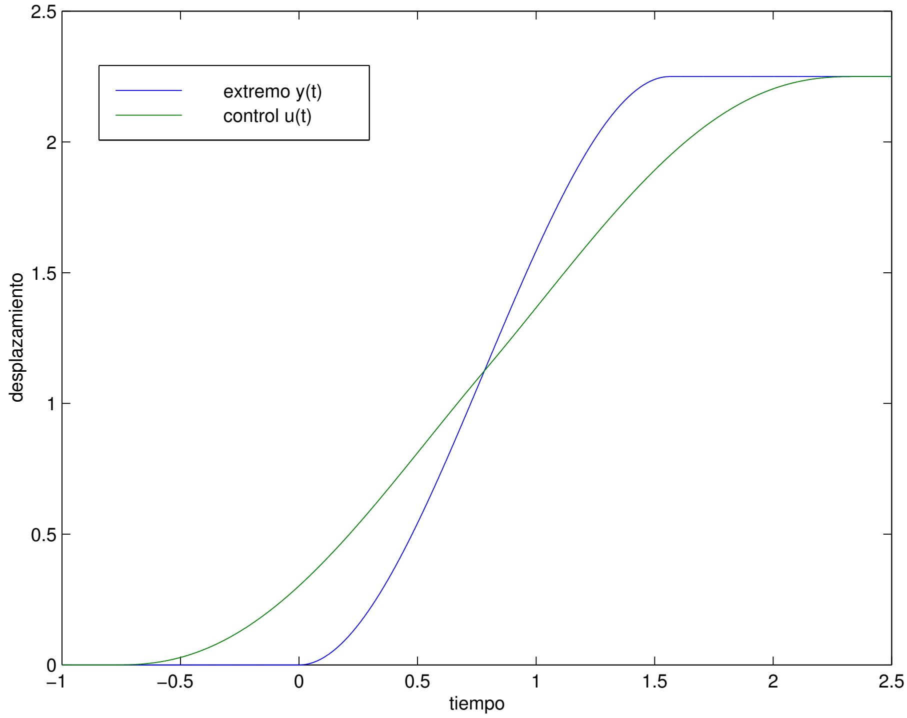

Figura 3: Gráficas de $y(t)$ y $u(t)$

Y, ahora, dibujamos las dos gráficas juntas

```
>>plot(t,y,t,u) % gr'aficas
>>xlabel('tiempo'),ylabel('desplazamiento') % etiquetas ejes
>>legend('extremo y(t)','control u(t)') % leyenda
```

Y obtenemos las gráficas de la figura 3.

Por último, el programa que han desarrollado N. Petit y P. Rouchon, cuyo código completo se incluye en el apéndice que hay al final de estas notas, permite ver de una manera dinámica el movimiento de la cadena y del carrito que lo controla. Para ejecutarlo hay que escribir `>>cadena` en la línea de comandos. Aparecerá una ventana gráfica interactiva (ver figura 4). Para que el carrito se mueva, disponemos de una *referencia* en forma de triángulo, en la parte de abajo de la ventana, para que funcione, hay que desplazar esta referencia al lugar deseado con el ratón, el carrito se moverá de tal forma que la base de la cadena acabe justo en el sitio en el que hemos puesto la referencia. Además de esto, se puede modificar la velocidad del movimiento, usando el deslizador que hay en el centro arriba. Hay otros dos botones, que no aparecen en nuestro dibujo: el botón $\boxed{\text{GRAFICAS}}$ produce las gráficas de las funciones $y(t)$ y $u(t)$, extremo libre de la cadena y carrito respectivamente, durante los últimos instantes de tiempo del proceso, como en la figura 3; y el botón $\boxed{\text{SALIR}}$, que sirve para salir del programa.

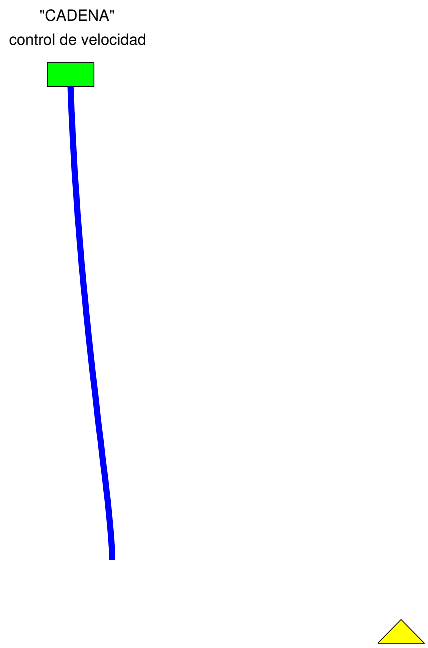

Figura 4: Ventana del programa `cadena.m`

3. Herramientas interactivas

El programa MatLab dispone de algunas herramientas interactivas elementales que se pueden utilizar sin saber prácticamente nada acerca del programa. Esto hace que puedan resultar interesantes para utilizarlas en algunos temas de Bachillerato. Vamos a ver dos de ellas: una que sirve para calcular sumas de Riemann y, por tanto, para aproximar numéricamente el valor de una integral de una función de una variable; y otra que es una especie de calculadora gráfica de funciones.

3.1. Sumas de Riemann

Esta herramienta se activa con el comando `rsums`. La sintaxis del comando es `rsums(f)`, donde `f` es una función simbólica (para introducir una función simbólica sólo hay que escribirla entre apóstrofes, de esta manera se distingue de una variable con contenido numérico.) Aparece una ventana gráfica (ver figura 5). Debajo de la gráfica hay un deslizador (que no se aprecia en nuestra figura) que permite cambiar con el ratón el número de rectángulos usados para aproximar el área de la curva, desde 2 hasta 256 rectángulos como máximo. En la parte superior aparece la función junto con el resultado numérico de la suma.

Veamos un ejemplo

```
>>f='10*x*exp(-5*x^2)' % creamos una funci'on
f =
    10*x*exp(-5*x^2)
```

Para comparar los resultados que obtengamos con las aproximaciones por sumas de Riemann con el valor exacto, vamos a calcular éste, por ejemplo, usando el comando

```
>>vpa(int(f,0,1),6) % evaluamos la integral de 0 a 1
ans =
    .993262
```

(El número 6 escrito como segundo argumento del comando `vpa` sirve para especificar la precisión del cálculo.)

```
>>rsums(f)  % aproximaci'on de Riemann desde 0 hasta 1
```

Y aparece la figura interactiva mencionada antes, en principio aparecen 10 rectángulos, aunque este número se puede cambiar como se ha indicado antes. Esta herramienta tiene una limitación, sea cual sea la función sólo la representa en el intervalo $[0, 1]$, pero esto se puede arreglar como se muestra en el ejercicio siguiente.

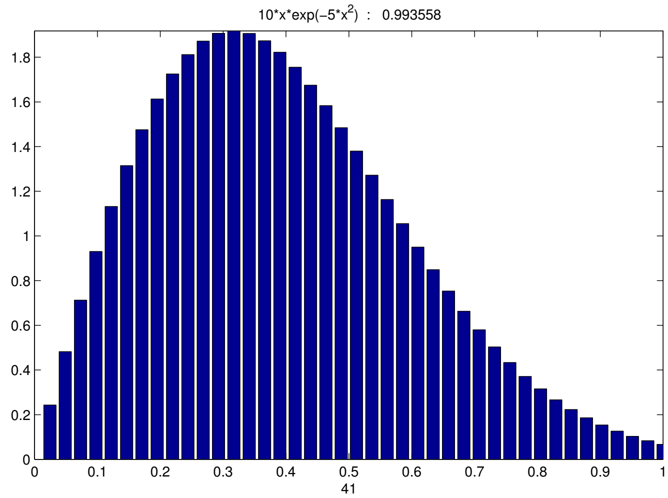

Figura 5: Sumas de Riemann para la función $f(x) = 10xe^{-5x^2}$

Ejercicio 1 *A pesar de que el comando* `rsums` *sólo aproxima la integral en el intervalo* $[0, 1]$*, también se puede utilizar para aproximar en un intervalo* $[a, b]$ *cualquiera, basta con encontrar el cambio de variable adecuado:*

a) Encontrar el cambio de variable que convierte la integral $\int_a^b f(x)dx$ *en la integral* $\int_0^1 g(t)dt$*.*

b) Utilizar el cambio anterior para aproximar la siguiente integral, usando `rsums`*,*

$$\int_1^2 \log(x)dx$$

Utilizar el comando `vpa(int(funcion,a,b),6)` *para calcular la integral con 6 decimales exactos y comparar el resultado con el obtenido mediante las sumas de Riemann.*

3.2. Calculadora de Funciones

Otra herramienta interesante de la que dispone MatLab, es la Calculadora de Funciones (`funtool`). Se activa escribiendo el comando `>>funtool`. Aparecen tres nuevas ventanas, dos de ellas conteniendo sendas gráficas de funciones y, la tercera, es la calculadora (figura 6).

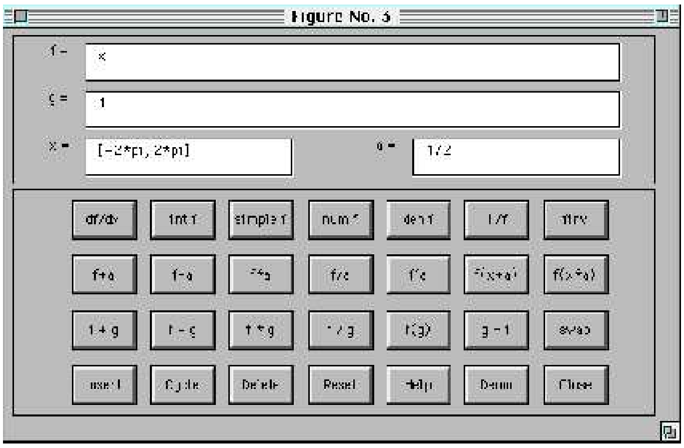

Figura 6: Calculadora de Funciones

FUNTOOL es una calculadora gráfica interactiva, que manipula funciones de una variable. En todo momento hay dos funciones que aparecen en las dos gráficas, $f(x)$ y $g(x)$. El resultado de la mayoría de las operaciones afecta a $f(x)$.

En las ventanitas etiquetadas con `"f ="` y `"g ="` se puede escribir y cambiar las funciones que aparecen por defecto, para introducir en su lugar nuevas funciones. Lo mismo que la etiquetada con `"x ="`, que puede ser cambiada para introducir un nuevo dominio. Y la `"a ="`, en la que se puede introducir un nuevo parámetro.

La fila superior de teclas de la calculadora son operaciones que sólo afectan a la función $f(x)$. Estas operaciones son:

`df/dx` - Derivada de $f(x)$.
`int f` - Integral de $f(x)$.
`simple f` - Simplifica la expresión, si es posible.
`num f` - Extrae el numerador de una expresión racional.
`den f` - Lo mismo, pero ahora el denominador.
`1/f` - Reemplaza $f(x)$ por $\frac{1}{f(x)}$.

`finv` - Calcula la inversa de $f(x)$.
Evidentemente las operaciones `int(f)` y `finv` no siempre funcionan.

La segunda fila de teclas trasladan y reescalan la función $f(x)$ según el valor del parámetro `a`. Las operaciones son:
`f + a` - Reemplaza $f(x)$ por $f(x) + a$.
`f - a` - Reemplaza $f(x)$ por $f(x) - a$.
`f * a` - Reemplaza $f(x)$ por $a.f(x)$.
`f / a` - Reemplaza $f(x)$ por $\frac{f(x)}{a}$.
`f ^ a` - Reemplaza $f(x)$ por $f(x)^a$.
`f(x+a)` - Reemplaza $f(x)$ por $f(x + a)$.
`f(x*a)` - Reemplaza $f(x)$ por $f(ax)$.

La tercera fila de teclas son operaciones en las que intervienen las dos funciones $f(x)$ y $g(x)$. Las operaciones son:
`f + g` - Reemplaza $f(x)$ por $f(x) + g(x)$.
`f - g` - Reemplaza $f(x)$ por $f(x) - g(x)$.
`f * g` - Reemplaza $f(x)$ por el producto $f(x).g(x)$.
`f / g` - Reemplaza $f(x)$ por $\frac{f(x)}{g(x)}$.
`f(g)` - Reemplaza $f(x)$ por la composición $f(g(x))$.
`g = f` - Reemplaza $g(x)$ por $f(x)$.
`swap` - Intercambia $f(x)$ y $g(x)$.

Las tres primeras teclas de la cuarta fila producen una lista de funciones. La tecla `Insert` añade la función actual a la lista. La tecla `Cycle` hace aparecer todas las funciones de la lista. Y la tecla `Delete`, borra la función actual de la lista. La lista de funciones está en un fichero interno que se llama `fxlist` que lleva, por defecto, una serie de funciones interesantes.

La tecla `Reset` devuelve los valores de f, g, x, a y `fxlist` a sus valores iniciales por defecto. La tecla `Help` imprime en pantalla ayuda (lo mismo que en estas instrucciones pero en inglés).

La tecla `Demo` propone un curioso problema: ¿es posible generar la función $\text{sen}(x)$ sin tocar el teclado, utilizando sólo el ratón y pulsando las teclas adecuadas de la calculadora? La `Demo` lo hace con un `Reset` y nueve "clics"del ratón y se propone intentar hacerlo con menos "clics". Si alguien lo consigue ruegan ponerse en contacto con la siguiente dirección electrónica:

moler@mathworks.com. ¿Habrá premio? Por último, con la tecla `Close` se acaba el juego.

Ejercicio 2 *¿Cuáles son los pasos que se siguen en la* `Demo` *para obtener la función* $\mathrm{sen}(x)$*?*

Intentar obtener la función $\mathrm{sen}(x)$ *con menos clics que en la* `Demo`*. (Es posible hacerlo, no es una broma).*

A continuación se ha incluido un ejemplo de una práctica para realizar utilizando la calculadora de funciones.

4. Una práctica con la calculadora de funciones

En esta práctica vamos a ver cómo es posible construir las gráficas algunas funciones a partir del conocimiento de las gráficas de otras. Para ello vamos a utilizar una herramienta interactiva del programa MatLab, la *Calculadora de Funciones*. Para activarla, hay que escribir

```
>>funtool
```

Aparecerán tres ventanas, dos de ellas son ventanas gráficas, en las que aparecen dos funciones, y la tercera (ver figura 6) es la ventana en la que introduciremos las funciones y ejecutaremos los comandos, pulsando los botones de la calculadora con la ayuda del ratón.

4.1. Gráficas de funciones opuestas

Las funciones $f(x) = x^3$ y $g(x) = -x^3$ se dice que son dos funciones opuestas. Para dibujarlas, las introducimos en la calculadora de funciones, una en la ventana `f=x^3` y la otra en `g=-x^3`, pulsando `Intro` cada vez que cambiemos cada formula, para que se dibuje. Obtendremos las dos gráficas de la figura 7.

Como se puede apreciar en las gráficas, las dos son simétricas con respecto del eje X.

Ejercicio 1 *Dibujar las gráficas de las siguientes parejas de funciones opuestas, como se ha descrito anteriormente:*

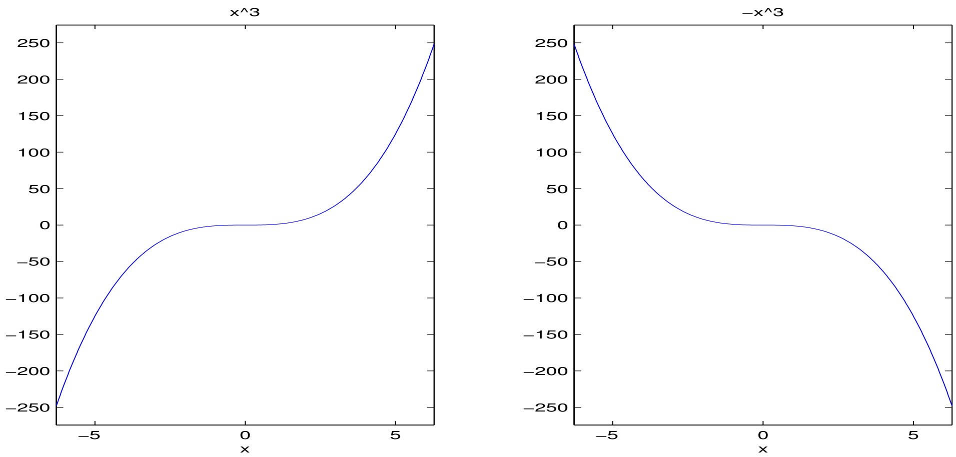

Figura 7: Gráficas de dos funciones opuestas.

a) $f(x) = x$ *y* $g(x) = -x$*.*

b) $f(x) = x^2 - 5x + 2$ *y* $g(x) = -x^2 + 5x - 2$*.*

c) $f(x) = e^x$ *y* $-e^x$*. (Para escribir* e^x *en MatLab, se usa la función exponencial* `exp(x)`*.)*

d) $f(x) = \frac{1}{x}$ *y* $g(x) = -\frac{1}{x}$*.*

4.2. Gráficas de funciones pares entre sí

Decimos que dos funciones son pares entre sí, cuando se cumple $f(x) = g(-x)$ para todo x. Por ejemplo, $f(x) = e^x$ y $g(x) = e^{-x}$.

Para hacer su gráfica con la Calculadora, primero introducimos en `f=exp(x)` y en `a=-1`. Si pulsamos ahora, `g=f`, tendremos las dos iguales. Pulsamos `f(x*a)` y `swap`, para que las intercambie y obtendremos las gráficas de 8.

En estas gráficas podemos apreciar que, ahora, las dos son simétricas con respecto del eje Y.

Ejercicio 2 *Dadas las siguientes funciones* $f(x)$*, dibujarlas y dibujar también* $g(x) = f(-x)$*, utilizando el procedimiento descrito antes:*

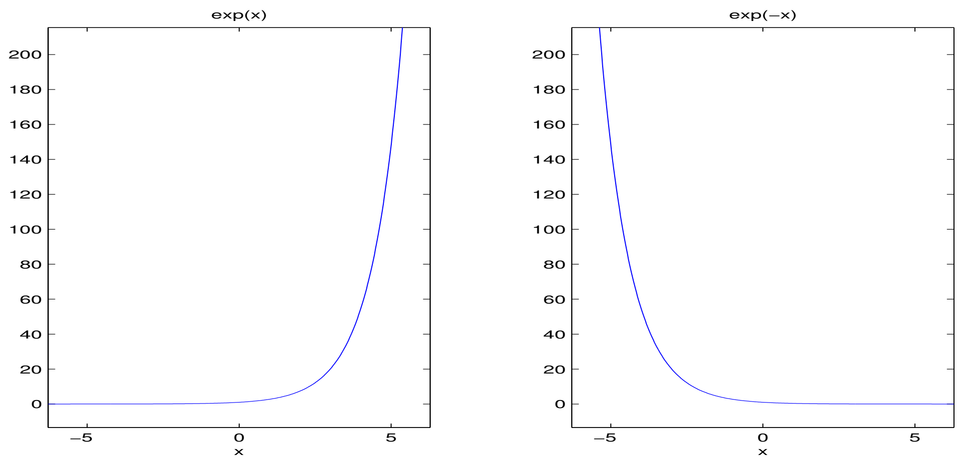

Figura 8: Gráficas de dos funciones pares entre sí.

a) $f(x) = x$.

b) $f(x) = 3^x$.

c) $f(x) = x^4 - 2x^2$.

d) $f(x) = \mathrm{sen}(x)$. *(Escribir* `sin(x)`*.)*
¿Qué ocurre en el apartado c)? ¿Por qué?

4.3. Composición de funciones. Función inversa

Sean las funciones $f(x) = x^3$ y $g(x) = x - 3$. Se llama **composición de funciones** a la operación consistente en aplicar primero una de las funciones y, a su imagen, aplicarle la otra función. Así,

$$f\left(g(x)\right) = f\left(x - 3\right) = \left(x - 3\right)^3$$

Para hacerlo con MatLab, escribimos `f=x^3`, `g=x-3` y pulsamos $\boxed{\texttt{f(g)}}$ (de paso, saldrá la gráfica de la composición en la ventana de la izquierda.)

También se podría componer al revés

$$g\left(f(x)\right) = g\left(x^3\right) = x^3 - 3$$

Ejercicio 3 *Dadas las siguientes parejas de funciones, $f(x)$ y $g(x)$. Calcular las composiciones $f(g(x))$ y $g(f(x))$:*

a) $f(x) = x^2 + x + 1$ *y* $g(x) = \dfrac{1}{x^2}$.

b) $f(x) = e^{x-1}$ *y* $g(x) = \dfrac{x}{x-1}$.

Ejercicio 4 *Sea la función $f(x) = x^3$. Dibujar la gráfica de la función $|f(x)|$ y de la función $f(|x|)$. Para calcular las dos funciones, considerar la función $g(x) = |x|$ (valor absoluto,* `abs(x)`*) y calcular las dos composiciones $f(g(x))$ y $g(f(x))$.*

Dada una función $f(x)$, se dice que la función $g(x)$ es la **inversa** de $f(x)$, si

$$f(g(x)) = x \quad \text{y} \quad g(f(x)) = x \quad \text{para todo } x$$

A la función inversa se la denota $f^{-1}(x)$, y puede no existir.

Por ejemplo, si $f(x) = e^x$, escribimos `f=exp(x)`, pulsamos `Intro` y ahora, para calcular la inversa, `finv`. Que nos da como resultado la función $g(x) = \log(x)$ (logaritmo neperiano). Si dibujamos las gráficas de las dos funciones, introduciéndolas en `f=` y `g=`, respectivamente, obtendremos las gráficas de la figura 9, en la que se puede apreciar que las dos gráficas son simétricas con respecto de la recta $y = x$.

Ejercicio 5 *Calcular las inversas de las siguientes funciones, si existen, y dibujar las gráficas de la función y su inversa:*

a) $f(x) = \dfrac{1}{x}$. *¿Qué ocurre en este caso? ¿Por qué?*

b) $f(x) = 2x - 3$.

c) $f(x) = x^3$.

d) $f(x) = \tan(x)$ *(función tangente, en este caso, cambiar el intervalo en* `x=[-pi/2,pi/2]`*.)*

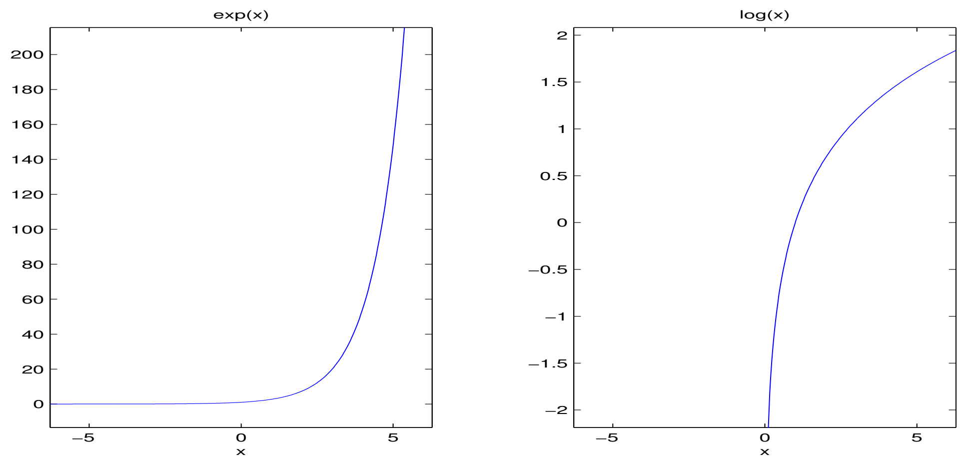

Figura 9: Gráficas de dos funciones inversas.

4.4. Traslaciones y dilataciones

Ejercicio 6 *Sea la función* $f(x) = x^2 - 3x + 1$, *introducirla en* `f=`. *Queremos descubrir para qué sirven las teclas* $\boxed{\texttt{f+a}}$, $\boxed{\texttt{f-a}}$ *y* $\boxed{\texttt{f(x+a)}}$. *Para ello, probar con diferentes valores del parámetro* `a=` *(positivos y negativos.)*

Ejercicio 7 *Sea la función* $f(x) = \cos(x)$, *introducirla en* `f=`. *Queremos descubrir ahora para qué sirven las teclas* $\boxed{\texttt{f*a}}$, $\boxed{\texttt{f/a}}$ *y* $\boxed{\texttt{f(x*a)}}$. *Para ello, probar con diferentes valores del parámetro* `a=` *(positivos, negativos, mayores que 1 y menores que 1.)*

5. Recursos en Internet

En Internet se pueden encontrar diversas direcciones que ofrecen información, ejemplos de programas, instrucciones, etc., acerca del programa MatLab. La lista que se ofrece a continuación no es desde luego exhaustiva, lo cual sería simplemente imposible, dado el elevado número de direcciones en las que aparecen referencias a este programa. Pero sí puede dar algunas ideas de por dónde empezar.

- `http://www.mathworks.com`: La página web de la compañía que ha creado MatLab. Se puede encontrar información sobre actualizaciones, cuestiones técnicas referenctes a la instalación del programa y muchos enlaces relacionados con el programa.

- `http://www.matcam.upm.es/cmoreno/metodos/matlab/icematr1/index.htm` Página del Prof. Carlos Moreno del Departamento de Matemática e Informática Aplicadas a la Ingeniería Civil. Esc. Técnica Superior de Ingenieros de Caminos, Canales y Puertos. UPM.

- `http://www.math.duke.edu/modules/materials` En esta página se pueden encontrar ejemplos de muchas prácticas de diferentes niveles de dificultad y diversos temas de Cálculo fundamentalmente, en muchos casos aparecen preparadas para realizarlas con varios programas: Maple, Mathematica y Matlab, principalmente.

- `http://texas.math.ttu.edu/~gilliam/m5399/symbolic.html` Una descripción con muchos ejemplos de la *MatLab Symbolic Toolbox*.

- `http://www.ius.cs.cmu.edu/help/Math/vasc_help_matlab.html` y la dirección `http://www.unm.edu/cirt/info/software/apps/matlab.html` contienen guías muy detalladas sobre MatLab.

- `http://www.mat.ucm.es/deptos/ma/ich/manual-matlab.html` Una introducción muy esquemática pero muy completa a MatLab, en español, realizada por el Prof. Uwe Brauer del Departamento de Matemática Aplicada de la Universidad Complutense de Madrid.

- `http://www.matematicas.net` En esta dirección, en la que se pueden encontrar diversos recursos de Matemáticas (casi todos en español), se pueden encontrar apuntes y cursos sobre MatLab. La mantiene un grupo de estudiantes y profesores, vinculados a la UNED.

- `http://web.mit.edu/18.06/www` y la dirección `http://www.math.columbia.edu/∼psorin/linear_ algebra/Matlab/index.html` Contienen una *toolbox* con comandos de Álgebra Lineal que se pueden utilizar junto con el libro de G. Strang, *Introduction to Linear Algebra*, Segunda Edición, Wellesley-Cambridge Press, Wellesley MA, 1998.

También tenemos la posibilidad de buscar nosotros mismos más direcciones de interés. Utilizando algún buscador como Altavista, por ejemplo, si queremos buscar algo relacionado con Series de Fourier con MatLab, podemos poner en la ventana del buscador: `Fourier & Matlab`, y nos aparecerá una extensa lista conteniendo estas dos palabras. Lamentablemente, ocurre con los buscadores de Internet que, al ser tan grande la cantidad de información que circula actualmente, mucha de ella es poco útil. Pero con un poco de paciencia se pueden encontrar cosas interesantes.

Apéndice

Código del programa de simulación de la cadena pesante `cadena.m`

```
%  Programa de simulaci'on de movimiento mediante rat'on de un sistema
%  de cadena pesante.
%  Desarrollado por
%  Ecole des Mines de Paris,
%  Centre Automatique et Systemes
%  60 Bd St Michel
%  75272 Paris
%    Nicolas Petit: petit@cas.ensmp.fr
%    Pierre Rouchon: rouchon@cas.ensmp.fr
%
% Aprox. lineal (Modelo estudiado por Daniel Bernoulli 1738 )
%     D_tt H  =  D_x (  g x D_x H ) = 0
%     H(x=L,t)= u(t) (control, posici'on horizontal del carrito)
%  y(t)=H(0,t) posici'on del extremo de la cadena
%
%  F'ormula del movimiento:
%  H(x,t)=1(2 pi)int_0^pi[y(t-2sqrt(x/g)sin s)+y(t+2sqrt(x/g)sin s)]ds
%
%
clear all;
L=1.;  % longitud de la cadena en  m
A_max= 1.5*L ; % amplitud horizontal del movimiento en  m
g=9.81; % gravedad en  m/s^2
T_max=2*sqrt(A_max/g);  % intervalo de movimiento en s.
DT=2*sqrt(L/g); scale=min(11., 11/max(A_max*14/19,L));

% discretizaci'on del espacio
dx=L/40.; x=[0:dx:L]; nx=length(x); H=zeros(1,nx);
% paso temporal
dt0=DT/50; dt=dt0;
dtmin=DT/1000; %paso m'inimo
dtmax=DT/3.; %paso m'aximo
```

```
YY=0*[-DT:dt0:DT];
mt=length(YY);
m0=(mt+1)/2.;
mm=[1:mt-1];

Y=0*[-5*DT:dt:0];
D=Y;t=Y;
nt=length(Y);
nn=[1:nt-1];

figure(1),
clf;
go=1;
mouse=0;
set(gca,'position',[0 0 1
1],'visible','off','Xlim',[0 19],'Ylim',[0 14]....
   ,'nextplot','add')
set(gcf,'CloseRequestFcn','go=0; closereq;',...
    'WindowButtonUpFcn',['mouse=0;set(ref,''facecolor'',''y'') ;'...
        'set(gcf,''WindowButtonMotionFcn'','''');']);

%  carrito
Xt=1+[0   1  1  0  ]+scale*H(nx);
Yt=12+[0   0  1/2  1/2  ];
trol=fill(Xt,Yt,'g','EraseMode','background');

% cadena
Xc=1.5 + scale*H;
Yc=12+scale*(x-1);
chain=line(Xc,Yc,'EraseMode','background','LineWidth', 4, 'Color', [0 0 1 ]);

%  referencia
Xr=1+[0   1  0.5  0  ] + scale*H(nx);
Yr=0.3+[0   0  1/2  0  ];
ref=fill(Xr,Yr,'y', 'EraseMode','xor',...
        'ButtonDownFcn',['mouse=1; set(ref,''facecolor'',''r'');' ...
           'set(gcf,''WindowButtonMotionFcn'',''1;'')']);

Hplay=uicontrol('style','push',...
                'units','normalized','position',[15/19 12.6/14 2/19 .5/14], ....
                'string','GRAFICAS','callback', ...
                ['figure(2);clf;' ...
                 'plot(t,Y,t,D);' ...
                 'xlabel(''tiempo'');' ...
```

```
                 'ylabel(''desplazamiento'');' ...
                 'legend(''EXTREMO y(t)'',''CARRITO u(t)'');'...
                 'pause;figure(1);'...
                ]);

set(Hplay,'visible','off');

Hrun=uicontrol('style','push',...
                'units','normalized','position',[1/19 12.7/14 2/19 .5/14], ...
                'string','stop','callback', ...
                ' ; ' ,'visible','off');

% t'itulos
text(9.5,13.5,'"CADENA"','color',[0 0 0],...
   'EraseMode','background','HorizontalAlignment','center');
text(9.5,13,'control de velocidad','color',[0 0 0],'EraseMode','background',...
   'HorizontalAlignment','center');
p1=[7.5/19 12.6/14 4/19 0.2/14];
Hslider=uicontrol(figure(1),'units','normalized','style','slider','position',...
            p1,'min',log(dtmin),'max',log(dtmax),'value',log(dt0),...
         'callback',['buf=get(Hslider,''value'');dt=exp(buf);']);

figure(1);
pause(1)
set(Hrun,'string','SALIR','callback','go=0;');
set(Hplay,'visible','on')
set(Hrun,'visible','on')

t1=T_max/5;
t2=t1;
seuil=A_max/5;
y1=0;
y2=0;

dth=pi/100;
th=[0:dth:pi-dth];
Sth=sin(th);

tt=0.;
y1_old=y1;
buf(1,1)=1.5;
while (go >0)
    if mouse > 0
       buf=get(gca,'CurrentPoint');
    end
```

```
    A=(buf(1,1)-1.5)/scale;
    A=max(0,A);
    A=min(A_max,A);

   y1=(y1+dt*y2/t1)/(1+dt/t1);
   t2b=t2*max(1,abs(A-y2)/seuil);
   y2=(y2+dt*A/t2b)/(1+dt/t2b);

   tt=tt+dt;
   if (tt >= dt0);
      ii=floor(tt/dt0);
      dy1=(y1-y1_old)/tt;
      for iii=1:ii;
         YY(mm)=YY(mm+1);YY(mt)=y1_old+iii*dt0*dy1;
      end;
      y1_old=y1-dy1*(tt-ii*dt0);
      tt=0.;
   end;

    for ix=1:nx;
       tau=2*sqrt(x(ix)/g);
       s=tau*Sth;
       ns=floor((mt-m0)*s/DT);
       H(ix)=(dth/2/pi)*sum(YY(m0+ns)+YY(m0-ns));
    end

    Y(nn)=Y(nn+1); Y(nt)=H(1);
    D(nn)=D(nn+1); D(nt)=H(nx);
    t(nn)=t(nn+1); t(nt)=t(nt)+dt;

    Xt=1+[0   1  1  0  ] + scale*H(nx);
    Xr=1+[0   1  0.5  0  ] + scale*A;
    Xc=1.5 + scale*H;
    set(trol, 'XData',Xt);
    set(chain,'XData',Xc);
    set(ref,'XData',Xr);
    drawnow;

end;

close(figure(1));
close(figure(2));

return
```

Referencias

[Ch] Chen, K.; Giblin, P. e Irving, A. *Mathematical Explorations with MATLAB*. Cambridge University Press, Cambridge, 1999.

[G] García de Jalón, J. et al. *Aprenda MatLab 5.2 como si estuviera en primero.* Esc. Sup. Ing. Industriales, Universidad de Navarra, 1998. (Estos apuntes se pueden conseguir, en formato `.pdf`, en la dirección `http://193.146.240.173/cursos/matlab/matlab52.pdf`)

[H] Harman, Thomas L.;Dabney, J. y Richert, N. *Advanced Engineering Mathematics using MATLAB V.4*. PWS, Boston, 1997.

[P] Petit, N. y Rouchon, P. *Motion Planing for Heavy Chain Systems*, en vías de publicación.

[R] Rodríguez del Río, R. *Matemáticas en el Aula de Informática.* En el curso *Temas relevantes en la Matemática actual: el reto de la Enseñanza Secundaria.* Dirigido por el Prof. E. Zuazua. Servicio de Publicaciones del MEC-UIMP, en vías de publicación.

[S] Sigmon, K. *MATLAB Primer.* 1989. (Apuntes clásicos sobre Matlab, algo anticuados, se pueden conseguir en la dirección `http://www.mathworks.com` también se pueden conseguir mediante anonymous ftp de `ftp://math.ufl.edu`, directorio `pub/matlab`, fichero `primer35.tex`)

[V] Varios Autores. *MATLAB Edición de estudiante. Versión 4. Guía de usuario.* The Mathworks Inc.-Prentice Hall, Madrid, 1998. (Esta es, de momento, la última edición en español de esta guía).