

Notas para los alumnos de Matemáticas II del grado en Químicas  
 Universidad de Murcia. Curso 2011-2012  
 G. Garrigós y A.Pallares

## El paquete plotdf

El paquete adicional “plotdf” incorpora el comando del mismo nombre para el análisis de una ecuación diferencial ordinaria (EDO) de primer orden

$$y'(x) = G(x, y)$$

o para un sistema de dos ecuaciones diferenciales autónomas de primer orden

$$\begin{cases} x'(t) = F(x(t), y(t)) \\ y'(t) = G(x(t), y(t)) \end{cases}$$

(obsérvese que el caso de una ecuación diferencial ordinaria se puede estudiar como un sistema de dos ecuaciones añadiendo la ecuación  $x'(t) = 1$ , o lo que es lo mismo, poniendo  $x = t$ . Esto es lo que hará el comando **plotdf** cuando sólo le indiquemos una ecuación diferencial)

El resultado que produce el comando plotdf es una ventana gráfica donde aparece una región del plano  $x$ ,  $y$  y dentro de esta

1. el campo de vectores  $(F(x,y), G(x,y))$  que representan la direcciones de las tangentes a las soluciones que pasan en cada punto,
2. la trayectoria de la solución  $(x(t), y(t))$  que pasa por un punto (que podemos elegir cuando lanzamos el comando),
3. cada vez que marcamos un punto de la región haciendo “click” en el ratón, se traza la trayectoria de la solución que pasa por ese punto.

Para poder usar el comando lo primero que debemos hacer es cargarlo con la orden

```
load("plotdf");
```

### Un primer ejemplo con una EDO

(recordad el ejercicio 8 de la hoja de problemas)

Se trata de analizar la ecuación  $x'(t) = 0.2x(5-x)$  con la condición inicial  $x(0)=1$

1ª opción. Sin declarar variables: hay que usar ecuaciones  $y'(x)=f(x,y)$  así que tendremos que asignar las letras  $x$  para el tiempo  $t$  e  $y$  para la variable  $x$ .

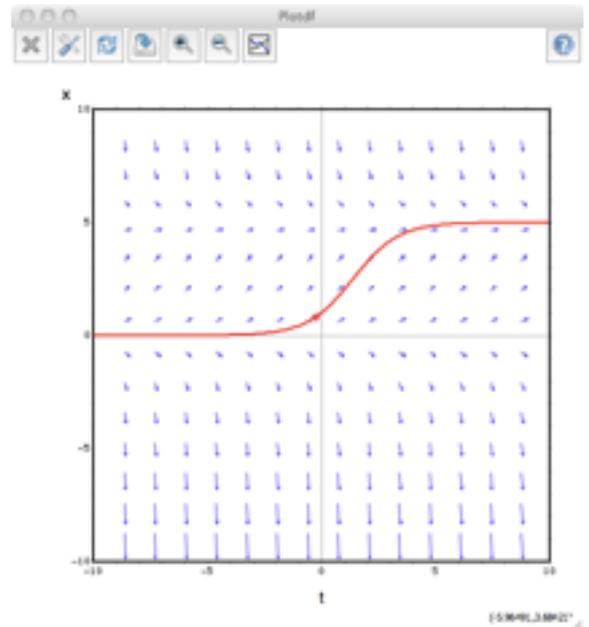
```
plotdf(0.2*y*(5-y), [trajectory_at, 0, 1]);
```

2ª opción. Declarando las variables: después del valor de  $x'(t)$  se señala que las variables son  $t$  y  $x$ .

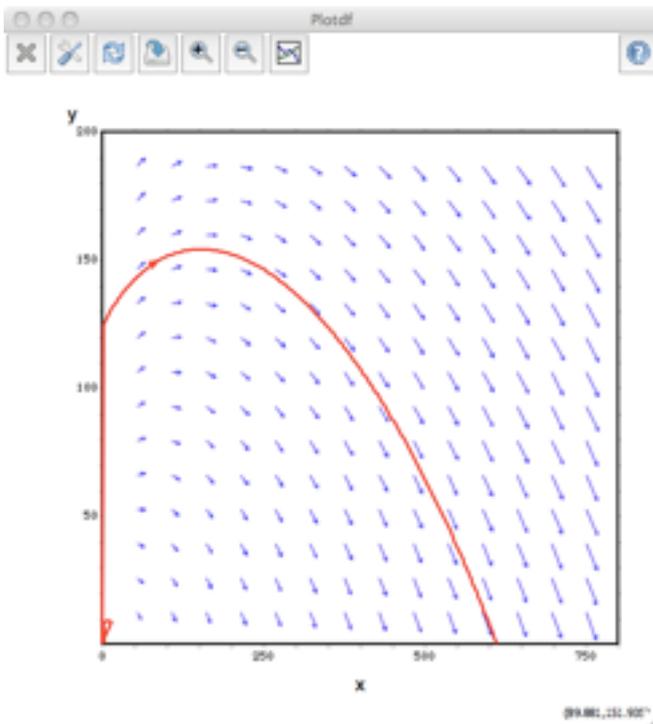
```
plotdf(0.2*x*(5-x), [t, x], [trajectory_at, 0, 1]);
```

En los dos casos se obtiene casi el mismo resultado, la única diferencia va a estar en los nombres de los ejes.

Si vais marcando con el ratón en distintos puntos, obtendréis más trayectorias (soluciones)



### Un ejemplo con un sistema de dos ecuaciones EDO



(recordad el ejercicio 22 de la hoja de problemas)

Se trata de analizar las soluciones del sistema de ecuaciones de un modelo competitivo

$$\begin{cases} x' = 3x + 4y \\ y' = -2x + 2y \end{cases}$$

con la condición inicial  $x(0) = 90$  e

$y(0) = 150$ .

En este caso como los valores iniciales de  $x$  e  $y$  son grandes, vamos a definir la región donde ver el campo de vectores y las trayectorias con las opciones  $[x,0,800]$  ( $x$  varía entre 0 y 800) e  $[y,0,200]$  ( $y$  varía entre 0 y 200):

```
plotdf([3*x+4*y,-2*x+2*y],[x,0,800],[y,0,200],[trajectory_at,90,150]);
```

## Dos cuestiones:

P: ¿Cómo se trazan las trayectorias (soluciones)?

R: Se usan métodos de aproximación numérica para ir construyendo una lista de puntos en virtud de las diferencias

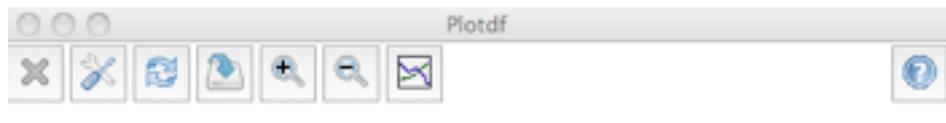
$$\begin{cases} x(t + \Delta t) - x(t) = x'(\xi_t)\Delta t = F(x(\xi_t), y(\xi_t))\Delta t \\ y(t + \Delta t) - y(t) = y'(\xi_t)\Delta t = G(x(\xi_t), y(\xi_t))\Delta t \end{cases}$$

y de hacer aproximaciones de las mismas. El programa usa el método de Adams-Moulton o un Runge-Kutta 4 (opcional).

P: ¿Para que sirven los botones que hay en la ventana?

R: La ventana tiene vida propia (es un programa independiente) y se pueden cambiar los parámetros de la representación gráfica sin tener que volver a wxMaxima y ejecutar el comando.

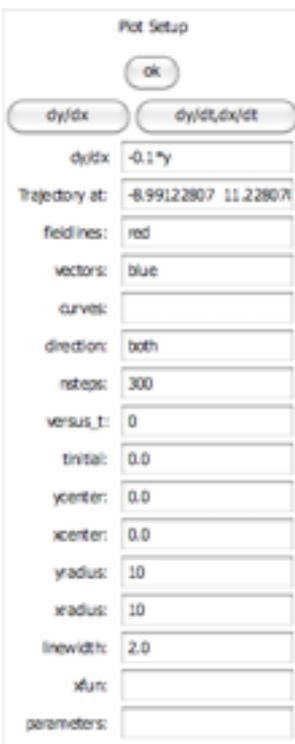
## Opciones del comando y del programa `plotpdf`:



**Cerrar** el programa `plotpdf`. Importante: ES NECESARIO CERRARLO PARA PODER SEGUIR EJECUTANDO ORDENES EN wxMAXIMA.



**Herramientas** del programa `plotpdf` :



- **dy/dx** : permite cambiar la ecuación diferencial ordinaria (en el ejemplo `dy/dx := -0.1*y`)
- **dy/dt, dx/dt** : permite cambiar las ecuaciones del sistema autónomo de ecuaciones diferenciales.
- **trajectory\_at** : permite fijar las condiciones iniciales de una trayectoria (solución) particular. Esta opción puede establecerse cuando se ejecuta el comando:
 

```
plotdf(-0.1*y, [trajectory_at, -8.9912, 11.228]);
```
- **fieldlines** : es el color de las trayectorias, por defecto es rojo (red).
- **vectors**: es el color de los vectores del campo ( $F(x,y), G(x,y)$ ), por defecto es azul (blue).
- **curves**: es el color de las trayectorias del campo ortogonal ( $-G(x,y), F(x,y)$ ), por defecto no se calculan ni dibujan porque el campo está vacío. Si se le asigna un color (por ejemplo: yellow) dibuja las trayectorias ortogonales en color amarillo.
- **direction**: Es la dirección de la trayectoria (solución) aproximada desde el punto de coordenadas iniciales. Las opciones son



★**forward** (hacia delante) : el método numérico elabora una aproximación de la solución avanzando en el tiempo desde **tinitial** (por defecto es 0.0) a pasos **tstep** (por defecto es 0.1) **nstep** (número de pasos que por defecto es 100). Con los valores por defecto t varía entre 0 y 10.

★**backward** (hacia atrás) : el método numérico elabora una aproximación de la solución retrocediendo en el tiempo desde **tinitial** a pasos **tstep**, **nstep** veces. Con los valores por defecto t varía entre 0 y -10.

★**both** (en los dos sentidos) : el método numérico elabora la aproximación avanzando y retrocediendo en el tiempo desde **tinitial** a pasos **tstep**, **nstep** veces en los dos sentidos. Con los valores por defecto t varía entre -10 y 10.

Esta opción sirve para observar el sentido en el que se recorren las soluciones cuando avanza o retrocede el tiempo t. También sirve, reduciendo el tamaño del paso y aumentando el número de pasos, para tener mejores aproximaciones de las trayectorias (soluciones). Por defecto la opción es both.

Las opciones pueden incluirse cuando se ejecuta el comando poniéndolas entre corchetes. En particular la opción tstep no aparece en el menú de configuración y sólo puede cambiarse al ejecutar el comando en wxMaxima.

- **nstep** : ya se ha explicado con la dirección. Puede ser útil para trazar trayectorias más o menos largas (en el tiempo).
- **versus\_t** : Se usa para crear una segunda ventana gráfica con el gráfico de las componentes x(t) e y(t) de la solución del sistema correspondiente a la condición inicial seleccionada. Importante: cada vez que se marca en la ventana de plotdf y se genera una trayectoria, se modifica la condición inicial en plotpdf. El valor por defecto es 0 y no crea esta segunda ventana, cualquier otro valor hace que se cree la ventana. La opción puede incluirse entre corchetes al ejecutar el comando en wxMaxima. OTRA POSIBILIDAD ES USAR EL BOTÓN  DEL PROGRAMA

- **tinitial** : ya se ha explicado con la dirección.
- **ycenter ; xcenter**; Son las coordenadas del punto que corresponde al centro de la ventana. Por defecto son las del origen 0.0 0.0;
- **yradius ; xradius**; Son las longitudes de los semiejes (radios), e.d. la mitad de la longitud de cada eje. Por defecto el valor de los dos radios es 10. IMPORTANTE: UNA ALTERNATIVA A ESTAS OPCIONES ES EJECUTAR EL COMANDO plotdf CON LAS OPCIONES [x, xinferior, xsuperior] e [y, yinf, ysup] como en el ejemplo anterior.

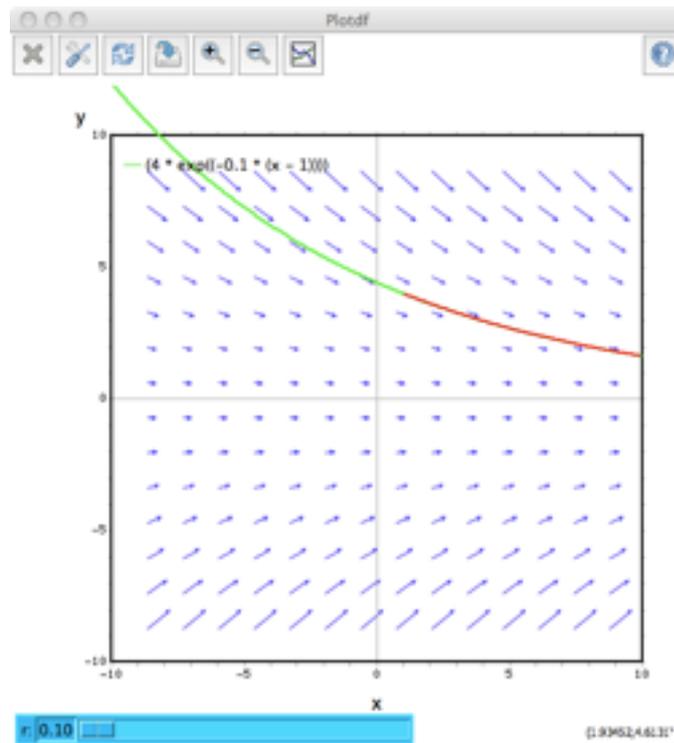
```
plotdf([3*x+4*y, -2*x+2*y], [x, 0, 800], [y, 0, 200], [trajectory_at, 90, 150]);
```

- **linewidth** : es el grosor de las trayectorias, por defecto es 2.0 puntos de pantalla.
- **xfun** : es una función f(x), cuya gráfica aparece representada en la ventana. Es interesante cuando conocemos soluciones estacionarias del sistema o la solución exacta de la EDO correspondiente a una condición inicial y queremos representarlas para compararlas con la soluciones (aproximadas) que produce plotdf.
- **parameters** : son los parámetros a utilizar en cualquiera de las opciones anteriores o en la definición de las ecuaciones. Su utilización debe simultanearse con la opción **sliders** (deslizadores) que permite cambiar los valores de los parámetros en la misma ventana de plotdf moviendo los deslizadores con el ratón.

Ejemplo de uso de **direction**, **xfun**, **parameters** y **sliders**:

```
plotdf(-r*y,[xfun,"4*exp(-r*(x-1))"],[direction, forward],
[parameters,"r=0.1"],[sliders,"r=0.1:2"]);
```

representa la solución aproximada y la solución exacta de  $y'(x) = -ry(x)$ , con condición inicial  $y(1) = 4$  donde podemos ir variando la velocidad de decrecimiento dada por el parámetro  $r$  que variará entre 0.1 y 2 en la barra de deslizamiento.



La solución real está en verde y la aproximada en rojo.



**IMPORTANTE:** para borrar trayectorias y actualizar las gráficas basta con marcar el botón de **actualizar/restaurar**

- **width** : Es el ancho de la ventana del programa plotdf medido en puntos (píxeles) de la pantalla. Por defecto el valor es 500.
- **height** : Es el alto de la ventana del programa plotdf medido en puntos (píxeles) de la pantalla. Por defecto el valor es 500.

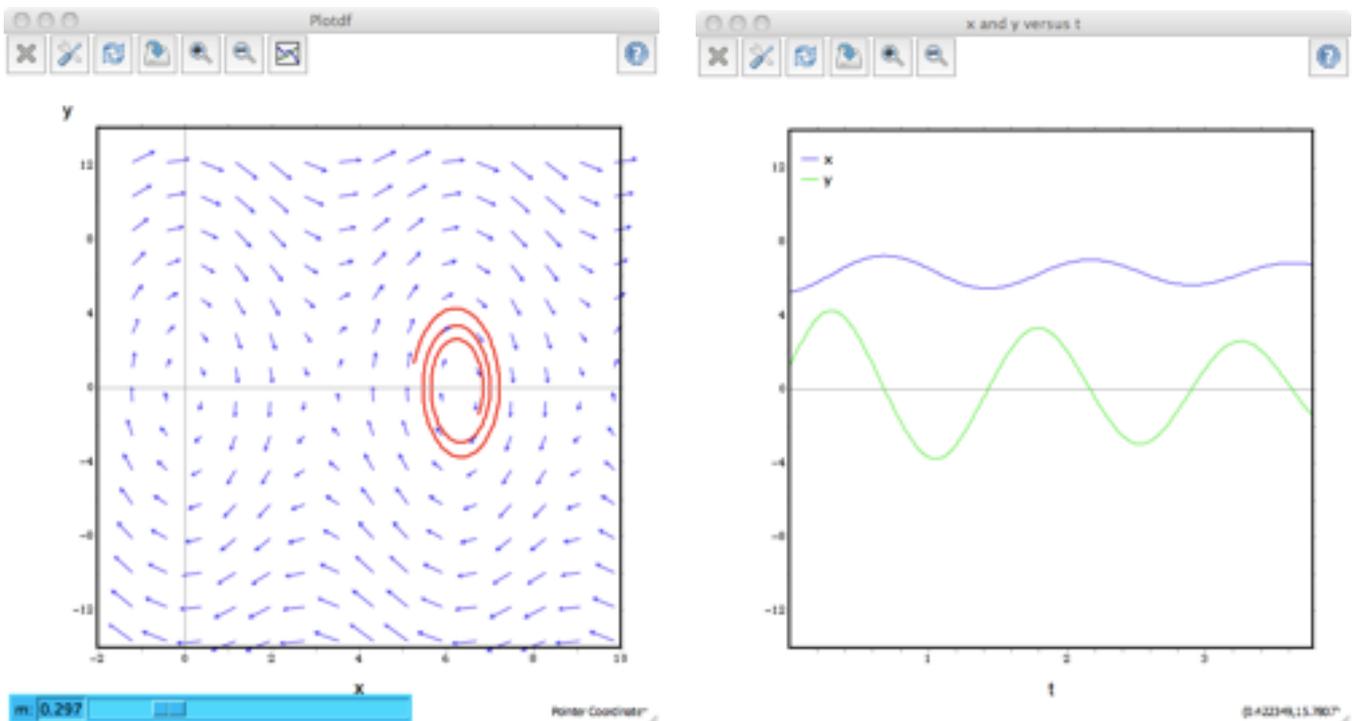


• **ZOOM** (AUMENTAR Y DISMINUIR LAS ESCALAS DE LAS GRÁFICAS. (SEGUN EL SISTEMA OPERATIVO NO VA DEMASIADO BIEN))

Ejemplo de uso de las opciones **tstep**, **nsteps** y **versus\_t**:

El campo de direcciones de un péndulo amortiguado, incluyendo la trayectoria (solución) para unas condiciones iniciales dadas, con una barra de deslizamiento que se puede utilizar para cambiar el valor de la masa,  $m$ , y con el gráfico de las dos variables de estado como funciones del tiempo:

```
plotdf([y,-g*sin(x)/l - b*y/m/l], [x,2,10],[y,-14,14],
[parameters,"g=9.8,l=0.5,m=0.3,b=0.05"],
[trajectory_at,5.26,1.27],[tstep,0.01],[direction,forward],
[nsteps,300],[sliders,"m=0.1:1"], [versus_t,1]);
```



## Otros paquetes comandos : dynamics

En los manuales/ficheros de ayuda de máxima podéis encontrar documentación sobre el paquete **dynamics** que incluye el comando **rk** correspondiente al método de Runge-Kutta de cuatro puntos para la aproximación de soluciones de sistemas autónomos de EDO de cualquier tamaño. Produce una lista/tabla de puntos/vectores de la trayectoria (solución) que se ha aproximado con las coordenadas de la variable independiente (tiempo  $t$ ) y las variables dependientes ( $x, y, z, \dots$ ) que aparecen en el sistema.