

1. Gráficos y Octave.

1.1. ¿Porqué hace falta un programa?

Cuando se usa una computadora, se usan programas como por ejemplo el procesador de texto o el messenger o el navegador. Cada una de estas *cosas* son programas que tienen alguna finalidad, auxiliarnos en la edición de textos, enviar mensajes o acceder a las páginas distribuidas por toda la web.

Para obtener soluciones de problemas (no triviales) de matemática, hacen falta programas que los resuelvan, o que ayuden a resolverlos, así como el editor de texto ayuda a elaborar un informe, pero no lo escribe.

1.2. ¿Porqué hace falta programar?

A medida que se avanza en el aprendizaje, se descubren nuevas cosas, surgen nuevas ideas y aparece la necesidad de realizar algo que el programa no tiene entre sus funciones. Otras veces, se desean realizar tareas fáciles aunque extremadamente rutinarias. En estos casos, enseñar a la computadora, que puede hacer algunas cosas más rápido que las personas, como hacer el proceso o la cuenta que se necesita, puede ser la diferencia entre realizar la tarea o no.

Programar es la forma de enseñarle a una computadora que es lo que se quiere que haga.

1.3. ¿Porqué Octave? ¿Porqué no otro?

Octave es un programa que permite ambas cosas al mismo tiempo. Resuelve problemas matemáticos y permite que se programen nuevas funciones.

Existen muchos programas que tienen un funcionamiento similar. Se eligió *Octave* porque:

- Es de uso libre.
- Se puede usar tanto en linux como en windows.
- Es interactivo.
- Tiene un lenguaje de programación bastante amplio e intuitivo.

La primera parte del curso tiene por objetivo que el alumno se familiarice con *Octave* a partir de situaciones problemáticas planteadas sobre conceptos matemáticos conocidos. Se propondrán soluciones y se estudiarán los comandos necesarios para poner en práctica dichas soluciones.

1.4. Primera situación problemática:

Mostrar cual es el efecto, en el gráfico, de modificar el coeficiente del término de grado uno en una función lineal $f(x) = ax + b$.

Es fácil describir con palabras que el gráfico de la función $f(x)$ es una recta, que a es la pendiente y que si a aumenta el gráfico es una recta más empinada. Pero todos saben que muchos conceptos se asimilan mejor si hay un gráfico que los explique.

Es posible graficar una recta en el pizarrón, en algunos casos hasta se puede usar tizas de colores, pero ciertamente es imposible reflejar la idea de animación usando sólo tiza y pizarrón.

Un abordaje del problema planteado se encuentra en *lineal.m*, no es el único posible, pero lo usaremos como ejemplo para introducir los comandos básicos de *Octave*.

lineal

```
x0=3;
dx=0.1;
b=0;
a0=0;
a1=1;
x=[-x0:dx:x0];
% a0=input("a inicial ==> ");
% a1=input("a final ==> ");
deltaa=(a1-a0)/10;
% axis([-3,3,-3,3]);
figure (1)
hold on
y=a0*x+b;
plot(x,y,'r');
sleep(3);
for a=a0+deltaa:deltaa:a1;
    title(strcat("a_=",num2str(a)))
    plot(x,a*x+b,'b');
    sleep(1)
end;
hold off;
sleep(2);
figure (2)
newplot ()
axis([-3,3,-3,3]);
plot(x,a0*x+b,'r')
hold on
for a=a0+deltaa:deltaa:a1-deltaa;
    plot(x,a*x+b,'.b')
end;
plot(x,a1*x+b,'g')
hold off
```

En el ejemplo se muestran rectas con pendientes entre 0 y 1, se utilizan colores y distintos tipos de líneas para diferenciar las rectas. Primero se grafican rectas a intervalos regulares de tiempo y a continuación se presenta un dibujo que representa el proceso completo.

Para realizar el primer gráfico se necesita saber como graficar una recta y repetir este proceso varias veces.

En las figuras 1 se presentan los gráficos generados por la ejecución de *lineal*.

A continuación se analizará el archivo *lineal.m* línea por línea, ya que esta es la forma que *Octave* lo interpreta, es decir la forma en que ejecuta las instrucciones contenidas en él. La línea:

```
x0=3;
```

representa una **asignación**, el valor que está a la derecha se lo guarda (o asigna) en la **variable** que está a la izquierda. En el lado izquierdo de una asignación siempre hay una variable y en el derecho puede

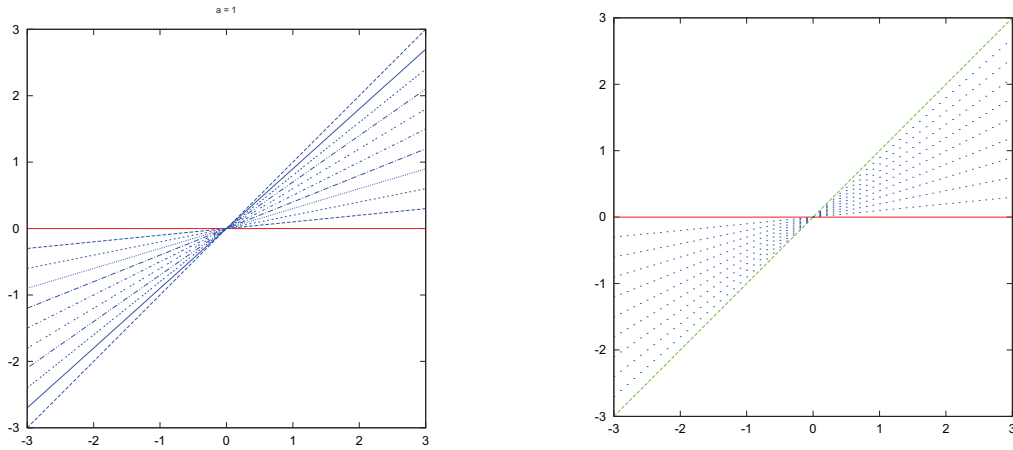


Figura 1: Salidas de lineal.m

escribirse una **constante**, una variable o una operación que puede involucrar constantes, funciones o variables.

Las constantes pueden ser números, vectores, matrices, etc, Las variables pueden tener nombres de longitud variable.

El signo "=" significa que se debe evaluar la expresión de la derecha y se **guarda** en la variable. De ninguna manera es una comparación entre las expresiones a ambos lados del signo "=". Más adelante se estudiarán los **operadores** de comparación.

La línea

```
x=[ -x0 : dx : x0 ];
```

es una asignación que en el lado derecho tiene un **vector**, definido por un **rango** (o lista de elementos) cuyo primer elemento es $-x_0$, el último es x_0 y todos están separados por dx . El rango puede ser creciente, como en el ejemplo, o decreciente, en este último caso el incremento debe ser negativo. Si se omite el **incremento** (dx), *Octave* asume que es 1.

```
y=a0*x+b ;
```

Es otra asignación, pero en este caso el lado derecho hay una multiplicación de un número a_0 por un vector x , cuyo resultado es un vector que se obtiene multiplicando cada elemento del vector x por a_0 . La suma entre un vector $a * x$ y un escalar b se obtiene sumando a cada componente del vector $a_0 * x$, el escalar b .

Para graficar se usa el comando:

```
plot(x , y , ' r ' );
```

donde x e y son vectores, el tercer argumento indica características especiales (**opciones**) que tendrá el dibujo. En este caso particular, los pares (x_i, y_i) son puntos de una recta y el gráfico se hará con color rojo.

Observación 1 *Octave no grafica funciones, grafica pares de puntos y los une con segmentos de recta.*

Al ejecutar *lineal* se observa una pequeña pausa entre el primer gráfico y los siguientes. Esto se obtiene usando la instrucción:

```
sleep (3);
```

El comando:

```
for a=a0+deltaa : deltaa : a1;
```

se utiliza para realizar algo un conjunto de acciones varias veces, en cada una de ellas a tomará los valores del rango **a0+deltaa:deltaa:a1**.

La sintaxis del comando **for** es:

```
for variable=rango
    acciones
end
```

Observando detenidamente los gráficos generados por la función *lineal* se ve que la escala en el eje de las ordenadas cambia a medida que se dibujan nuevas rectas. Esto se debe a que *Octave* elige los intervalos de graficación en forma automática. Para fijar el área del gráfico se debe usar la función **axis**.

En *lineal.m* se tiene:

```
% axis ([ -3, 3, -3, 3]);
```

el **%** le dice a *Octave* que la línea no debe ser considerada. Las líneas que comienzan con **%** son llamadas comentarios. La sintaxis del comando **axis** es:

```
axis ([ xmin , xmax , ymin , ymax , zmin , zmax ] );
```

Cada vez que se ejecuta *lineal*, se obtiene el mismo resultado. Para cambiar el resultado hay que modificar el archivo antes de ejecutarlo nuevamente. Para facilitar la interacción entre el usuario y el **script** *Octave* permite modificar algún parámetro mientras el script se está ejecutando. La línea:

```
% a0=input("a inicial ==> ");
```

muestra el mensaje *a inicial ==>* en la pantalla y queda a la espera de una respuesta del usuario, el valor ingresado es almacenado en la variable *a0* para ser utilizada cuando se necesite.

Ejercicio 1 *Usando como ejemplo lineal.m explicar el efecto que produce variar b en la ecuación de la recta.*

Ejercicio 2 *Proponer otra forma de abordar el problema planteado.*