

CLASE N°19

Introducción a la Programación Científica

EL ENTORNO OCTAVE

- Octave
 - **Entorno integrado** orientado a la **Programación Científica**
 - **Interpretado**, permite **trabajo interactivo** con el intérprete
 - **Orientado** a la **resolución numérica** y **visualización de datos**
 - Utiliza un lenguaje de **alto nivel** con muchas **funcionalidades nativas**
 - Cuenta con muchas **bibliotecas** que extienden sus herramientas
 - Con **buena compatibilidad** con el entorno MatLab®
 - Licencia **GNU/GPL**
 - Viene con una **extensa documentación** que debemos aprender a usar

EL ENTORNO OCTAVE



- Algunos comandos que aprendimos:
 - Creación de **variables** es por medio de la **asignación**
 - Escalares, vectores o matrices**

```
> escalar = 5
escalar = 5
> vector1 = -10:3.5:10, vector2 = [2 5 7 pi e]
vector1 =
-10.000    -6.500    -3.000     0.500     4.000     7.500
vector2 =
 2.0000    5.0000    7.0000    3.1416    2.7183
> matriz = [1 2 3; 4 5 6]
matriz =
 1  2  3
 4  5  6
```
 - El **operador** : funciona como la función `range()` de Python, pero puede generar valores no enteros
 - Conoce **varias constantes** nativamente (π , número de Euler)

3

EL ENTORNO OCTAVE



- Algunos comandos que aprendimos:
 - Muchas **funciones nativas**
 - Trigonómicas, como `sin()`, `cos()`
 - Comunes, como `abs()`, `exp()`
 - Para crear vectores y matrices, como `zeros()`, `ones()`, `eye()`, `polyval()`
 - Generadores de números aleatorios, como `rand()`
 - Para procesar matrices, como `det()`, `inv()`
 - La **función `plot()`** permite graficar datos en 2D
 - `plot(x, y)` asume puntos (x_i, y_i)
 - Calcula ejes (casi siempre) adecuados
 - Podemos **personalizar** el gráfico **cambiando sus propiedades**

4

EL ENTORNO OCTAVE



- Algunas propiedades de los gráficos:
 - Hay funciones que cambian las propiedades del **gráfico activo**
 - Podemos darle título con `title()`
 - Podemos cambiar el nombre de los ejes con `xlabel()` e `ylabel()`
 - Podemos “administradores” (*handlers*) de **objetos gráficos**
 - De la figura con `gcf()`
 - De los ejes con `gca()`
 - Usamos los administradores para cambiar propiedades
 - Usando `set(<handler>, <propiedad>, <valor>)`
 - Límites de los ejes: `xlim, ylim`
 - Marcas en el eje de las abscisas: `xtick, xticklabel`

5

EL ENTORNO OCTAVE



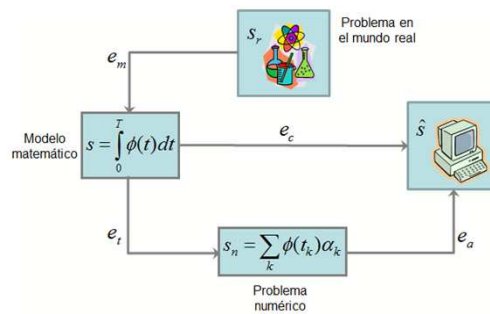
- Siempre hay un gráfico activo
 - Que es **reiniciado** cada vez que se “plotea”
 - Para plotear dos funciones en un mismo gráfico, debemos cambiar el estado de “sostenido” con `hold on`
 - Usamos `hold off` para liberarlo
- Podemos escribir **programas** en cualquier editor
 - Colección de comandos para el intérprete
 - Se le entrega con la función `source()`
 - Normalmente con la extensión `.m`
 - Podemos comentarlos con `#`, `#{`, `#}`
- Podemos cambiar como se muestran los números
 - Cambiando el estado `format` a `long`, `short`, u otras posibilidades

6

LIMITACIONES



- La **resolución numérica** introduce **errores**
 - Son **inevitables** y debemos **tener siempre presente** su existencia
 - Tienen **distintas fuentes**

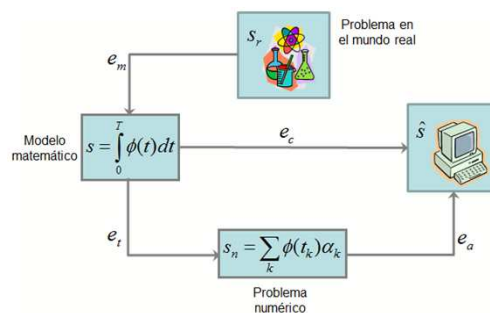


7

LIMITACIONES



- La **resolución numérica** introduce **errores**
 - Por ahora, nos interesa el **error computacional**
 - Error de truncamiento + error de aproximación
 - Puede medirse como error **absoluto** o **relativo**



8

PRÓXIMA CLASE



- Veremos **más comandos** para el entorno Octave
- Discutiremos con más detalle el **error de aproximación** y de dónde proviene
- Sabremos de las consecuencias que puede tener que un ingeniero **“olvide”** que estos errores existen

9

¿CONSULTAS?



10