

CLASE N°20

Los misteriosos flotantes

LOS FLOTANTES

- Revisamos con mayor profundidad los **números de punto (o coma) flotante**
 - Representación de **valores no enteros** en la arquitectura de los computadores
 - Normalmente se usan las definiciones del estándar **IEEE 754**
 - Con **precisión simple** en 32 *bits*
 - Con **precisión doble** en 64 *bits*
 - Valores se representan con un **número fijo, predeterminado de bits**

LOS FLOTANTES



- Esta **limitación** significa:

$$F \neq \mathbb{R}$$

- Hay finitos flotantes, pero infinitos reales
 - Luego, un flotante representa a **más de un real**
 - Un valor real es **aproximado** por un flotante
 - Entre dos números reales hay infinitos números reales; entre dos flotante contiguos no hay **ningún valor**

3

LOS FLOTANTES



- Esta **limitación** significa:

$$F \neq \mathbb{R}$$

- Dos flotantes están separados por un **valor épsilon**
 - ϵ varía para cada flotante: son **más densos** cerca del cero y **más dispersos** para valores altos
 - Pero el **ϵ -relativo** es pequeño
- Pueden producirse **desbordes**
 - Underflow** si es **más pequeño** que el menor número flotante
 - Overflow** si es **más grande** que el mayor número flotante

4

LOS FLOTANTES



- Esta **limitación** significa:

$$F \neq R$$

- Los flotantes incluyen **valores especiales**
 - Inf representa el valor **infinito**
 - NaN representa el valor **inválido o indefinido**
 - NA representa el valor **desconocido**
- Se **pierden algunas propiedades** de los reales
 - La **asociatividad** y la **distributividad** no están garantizadas

5

EL ENTORNO OCTAVE



- Algunos comandos que aprendimos:
 - Podemos cambiar **explícitamente** los **tipos de datos**
 - Tipos enteros con diferente tamaño (en *bits*): `int8()`, `int16()`, `int32()`, etc.
 - Tipos flotantes : `single()`, `double()`
 - Podemos conocer las **limitaciones en la máquina**
 - Épsilon: `eps()`, `eps = eps(1)`
 - Límites: `realmin`, `realmax`
 - Podemos **desplegar mensajes** en pantalla
 - Strings: `'texto'` (otras comillas no compatibles con MatLab®)
 - Función `disp(x)`, siempre agrega fin de línea

6

EL ENTORNO OCTAVE



- Algunos comandos que aprendimos:
 - Usamos raíz cuadrada: `sqrt()`
 - Desplegamos la **representación binaria**: `format bin`
 - Pudimos hacer **bifurcaciones**

```
if(<condición>)      if(<condición>)      if(<condición 1>)
    cuerpo-entonces  cuerpo-entonces  cuerpo-entonces 1
end                  else                  elseif(<condición 2>)
                                cuerpo-sino  cuerpo-entonces 2
                                end          . . .
                                elseif(<condición n>)
                                cuerpo-entonces n
                                else
                                cuerpo-sino
                                end
```

(Otras alternativas sintácticas no son compatibles con MatLab®)

7

EL ENTORNO OCTAVE



- Algunos comandos que aprendimos:
 - Pudimos hacer **iteraciones** y **fijas y condicionadas**

```
for <var> = <expresión>      while(<condición>)
    cuerpo                  cuerpo
end                          end
```

(Otras alternativas sintácticas no son compatibles con MatLab®)

8

PELIGROS



- **Nunca podemos olvidar** las limitaciones de los flotantes:
- Los errores pueden **acumularse y/o propagarse**
- Pueden tener consecuencias nefastas
- La disciplina llamada **análisis numérico** estudia cómo evitar estos problemas de la resolución numérica



9

PRÓXIMA CLASE



- Veremos **más comandos** para el entorno Octave
- Discutiremos cómo usarlos para resolver **problemas de la ingeniería**

10

