Revisión de Conceptos Básicos

- **Normas:** Normas vectoriales y matriciales. Productos interiores.
- **Errores:** Errores computacionales. Propagación de errores.

Normas vectoriales

- La manera usual de expresar el *tamaño* de un vector es mediante una norma.

- **Definición.** Sea V un espacio vectorial. Se llama **norma** sobre V a cualquier función

$$\begin{aligned} \|\cdot\| : V &\longrightarrow \mathbb{R} \\ \boldsymbol{v} &\longmapsto \|\boldsymbol{v}\| \end{aligned}$$

  que satisfaga:

  1. $\|\boldsymbol{v}\| \geq 0 \quad \forall \boldsymbol{v} \in V$ (positividad),
  2. $\|\boldsymbol{v}\| = 0 \iff \boldsymbol{v} = \boldsymbol{0}$ (no degeneración),
  3. $\|k\boldsymbol{v}\| = |k| \, \|\boldsymbol{v}\| \quad \forall \boldsymbol{v} \in V, \quad \forall k \in \mathbb{R}$ (homogeneidad),
  4. $\|\boldsymbol{v} + \boldsymbol{w}\| \leq \|\boldsymbol{v}\| + \|\boldsymbol{w}\| \quad \forall \boldsymbol{v}, \boldsymbol{w} \in V$ (desigualdad triangular).

- A un espacio vectorial V provisto de una norma $\|\cdot\|$ se le llama **espacio vectorial normado** y se le denota $(V, \|\cdot\|)$.

Normas vectoriales (cont.)

- **Ejemplos.** Dados $V = \mathbb{R}^n$ (espacio de vectores *columna* de n componentes reales) y $\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$, se definen las siguientes normas:
  - **Norma euclideana:** $\|\boldsymbol{x}\|_2 := \left(\sum_{i=1}^{n} |x_i|^2\right)^{\frac{1}{2}}$.
  - **Norma infinito**: $\|\boldsymbol{x}\|_\infty := \max_{1 \le i \le n} |x_i|$.
  - **Norma uno:** $\|\boldsymbol{x}\|_1 := \sum_{i=1}^{n} |x_i|$.
- Por comodidad de notación, muchas veces escribiremos un vector columna como $\boldsymbol{x} = (x_1 \cdots x_n)^{\mathrm{t}} \in \mathbb{R}^n$.

Distancia entre vectores. Equivalencia de normas

- Toda norma sobre un espacio vectorial V induce una **distancia**:

$$\operatorname{dist}(\boldsymbol{v}, \boldsymbol{w}) := \|\boldsymbol{v} - \boldsymbol{w}\|, \qquad \boldsymbol{v}, \boldsymbol{w} \in V.$$

- Una sucesión $\{\boldsymbol{v}_n\}_{n\in\mathbb{N}} \subset V$ **converge** a $\boldsymbol{v} \in V$ si $\operatorname{dist}(\boldsymbol{v}_n, \boldsymbol{v}) \to 0$:

$$\boldsymbol{v}_n \to \boldsymbol{v} \quad \Longleftrightarrow \quad \|\boldsymbol{v}_n - \boldsymbol{v}\| \to 0.$$

- Dos normas $\|\cdot\|_*$ y $\|\cdot\|_\bullet$ sobre un espacio vectorial V son **equivalentes** si existen constantes C_1 y C_2 tales que

$$C_1\|\boldsymbol{v}\|_* \leq \|\boldsymbol{v}\|_\bullet \leq C_2\|\boldsymbol{v}\|_* \qquad \forall \boldsymbol{v} \in V.$$

- **Teorema.** Todas las normas sobre un espacio de dimensión finita son equivalentes.

- **Corolario.** Si $\|\cdot\|_*$ y $\|\cdot\|_\bullet$ son dos normas cualesquiera en un espacio de dimensión finita, entonces

$$\|\boldsymbol{v}_n - \boldsymbol{v}\|_* \to 0 \quad \Longleftrightarrow \quad \|\boldsymbol{v}_n - \boldsymbol{v}\|_\bullet \to 0.$$

Normas matriciales inducidas.

- Toda norma vectorial $\|\cdot\|$ sobre $\mathbb{R}^n$ induce una **norma matricial** sobre $\mathbb{R}^{n\times n}$ (espacio de matrices cuadradas $n \times n$):

$$\|\boldsymbol{A}\| := \max_{\boldsymbol{x}\in\mathbb{R}^n:\ \boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|}, \qquad \boldsymbol{A} \in \mathbb{R}^{n\times n}.$$

  Es fácil verificar que esto define efectivamente una norma sobre $\mathbb{R}^{n\times n}$.

  Esta norma se dice que es una norma matricial **inducida** por la norma vectorial y se denota con el mismo símbolo que la norma vectorial.

- **Proposición.** Toda norma matricial inducida por una norma vectorial satisface las siguientes propiedades:

$$\|\boldsymbol{A}\boldsymbol{x}\| \leq \|\boldsymbol{A}\|\,\|\boldsymbol{x}\| \qquad \forall \boldsymbol{A} \in \mathbb{R}^{n\times n}, \quad \forall \boldsymbol{x} \in \mathbb{R}^n \qquad \text{(compatibilidad)},$$

$$\|\boldsymbol{A}\boldsymbol{B}\| \leq \|\boldsymbol{A}\|\|\boldsymbol{B}\| \qquad \forall \boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{n\times n} \qquad \text{(submultiplicatividad)}.$$

Normas matriciales (cont.)

- Hay formas más sencillas de calcular algunas de las normas matriciales más usuales.

Proposición. Sea $\boldsymbol{A} = (a_{ij}) \in \mathbb{R}^{n\times n}$. Entonces:

- $\|\boldsymbol{A}\|_\infty = \max\limits_{1\leq i\leq n} \left(\sum\limits_{1\leq j\leq n} |a_{ij}| \right)$ (máxima suma de filas).
- $\|\boldsymbol{A}\|_1 = \max\limits_{1\leq j\leq n} \left(\sum\limits_{1\leq i\leq n} |a_{ij}| \right)$ (máxima suma de columnas).
- $\|\boldsymbol{A}\|_2 = \sqrt{\rho\left(\boldsymbol{A}^{\mathrm{t}}\boldsymbol{A}\right)}$ (norma espectral).

Valores y vectores propios

- **Definición.** Dada una matriz cuadrada $\boldsymbol{A}$, diremos que un número $\lambda \in \mathbb{C}$ es un **valor propio** (o **autovalor**) de $\boldsymbol{A}$ si existe un vector no nulo, $\boldsymbol{v} \neq 0$, tal que

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}.$$

  A $\boldsymbol{v}$ se le llama **vector propio** (o **autovector**) asociado al valor propio λ.

  λ es un valor propio si y sólo si es raíz de la **ecuación característica**

$$\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = 0.$$

- **Definición.** El **espectro** de $\boldsymbol{A}$ es el conjunto de todos sus valores propios:

$$\sigma(\boldsymbol{A}) := \left\{ \lambda \in \mathbb{C} : \lambda \text{ es un valor propio de } \boldsymbol{A} \right\}.$$

- **Definición.** El **radio espectral** de $\boldsymbol{A}$ es el máximo módulo de sus autovalores:

$$\rho(\boldsymbol{A}) := \max_{\lambda \in \sigma(\boldsymbol{A})} |\lambda|.$$

- **Teorema.** Si $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ es simétrica ($\boldsymbol{A}^{\mathrm{t}} = \boldsymbol{A}$), entonces $\|\boldsymbol{A}\|_2 = \rho(\boldsymbol{A})$.

Producto interior

- Sea V un espacio vectorial real. Se llama **producto interior** sobre V a cualquier función

$$\begin{aligned} \langle \cdot, \cdot \rangle : \; V \times V \quad &\longrightarrow \quad \mathbb{R} \\ (\boldsymbol{v}, \boldsymbol{w}) \quad &\longmapsto \quad \langle \boldsymbol{v}, \boldsymbol{w} \rangle \end{aligned}$$

  que satisfaga:

  1. $\langle \boldsymbol{v}, \boldsymbol{v} \rangle \geq 0 \qquad \forall \boldsymbol{v} \in V$ (positividad),
  2. $\langle \boldsymbol{v}, \boldsymbol{v} \rangle = 0 \quad \Longleftrightarrow \quad \boldsymbol{v} = \boldsymbol{0}$ (no degeneración),
  3. $\langle \boldsymbol{v}, \boldsymbol{w} \rangle = \langle \boldsymbol{w}, \boldsymbol{v} \rangle \qquad \forall \boldsymbol{v}, \boldsymbol{w} \in V$ (simetría),
  4. $\langle \boldsymbol{v}, k_1 \boldsymbol{w}_1 + k_2 \boldsymbol{w}_2 \rangle = k_1 \langle \boldsymbol{v}, \boldsymbol{w}_1 \rangle + k_2 \langle \boldsymbol{v}, \boldsymbol{w}_2 \rangle$
     $\forall \boldsymbol{v}, \boldsymbol{w}_1, \boldsymbol{w}_2 \in V, \quad \forall k_1, k_2 \in \mathbb{R}$ (linealidad).

Producto interior (cont.)

- **Ejemplo.** El **producto escalar** en $\mathbb{R}^n$:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^{\mathrm{t}} \boldsymbol{y} = \sum_{i=1}^{n} x_i y_i, \qquad \boldsymbol{x} = (x_1 \cdots x_n)^{\mathrm{t}},\ \boldsymbol{y} = (y_1 \cdots y_n)^{\mathrm{t}} \in \mathbb{R}^n.$$

- Todo producto interior sobre un espacio vectorial V induce una norma:

$$\|\boldsymbol{v}\| := \sqrt{\langle \boldsymbol{v}, \boldsymbol{v} \rangle}, \qquad \boldsymbol{v} \in V.$$

  En particular, el producto escalar en $\mathbb{R}^n$ induce la norma euclideana:

$$\sqrt{\boldsymbol{x}^{\mathrm{t}} \boldsymbol{x}} = \|\boldsymbol{x}\|_2, \qquad \boldsymbol{x} \in \mathbb{R}^n.$$

- **Teorema. Desigualdad de Schwarz:**

$$|\langle \boldsymbol{v}, \boldsymbol{w} \rangle| \leq \|\boldsymbol{v}\|\, \|\boldsymbol{w}\| \qquad \forall \boldsymbol{v}, \boldsymbol{w} \in V.$$

- **Definición.** Dos vectores $\boldsymbol{v}$ y $\boldsymbol{w}$ son **ortogonales** cuando $\langle \boldsymbol{v}, \boldsymbol{w} \rangle = 0$:

$$\boldsymbol{v} \perp \boldsymbol{w} \qquad \Longleftrightarrow \qquad \langle \boldsymbol{v}, \boldsymbol{w} \rangle = 0.$$

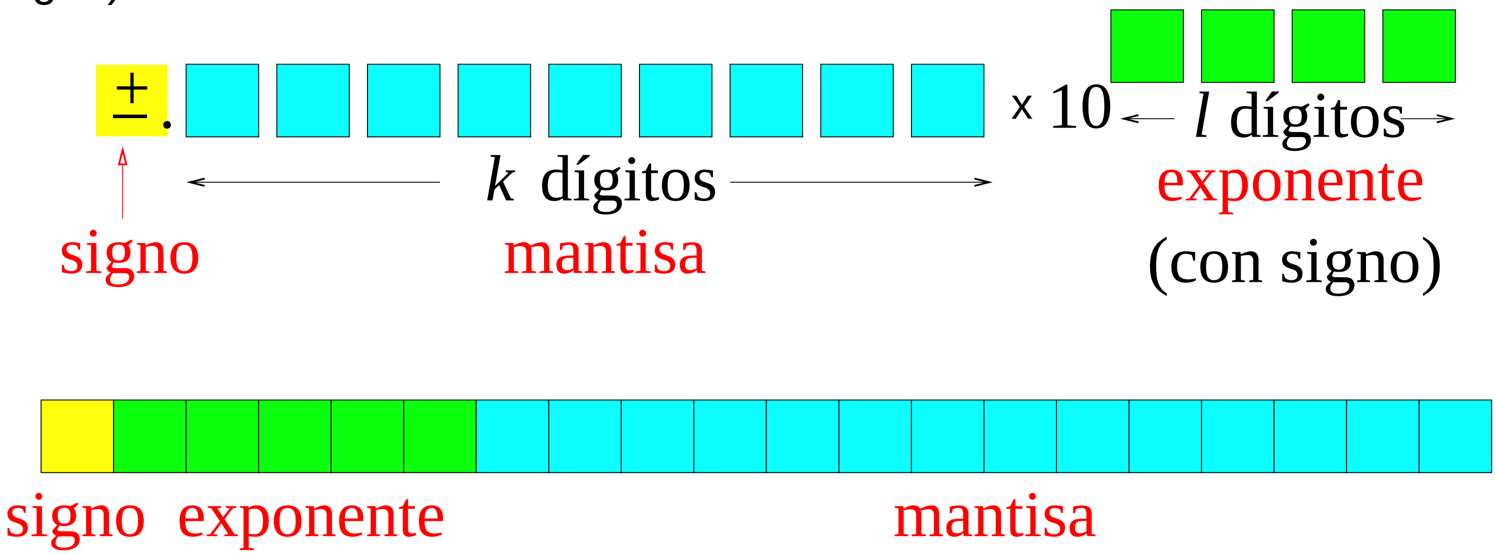
Representación de números reales en un sistema de punto flotante

- Lo describiremos, por sencillez, en el sistema decimal, aunque los computadores representan los números en sistema binario:

$$
\begin{array}{rcl}
783.375 & \longrightarrow & +.783375 \times 10^{3} \\
0.00843215 & \longrightarrow & +.843215 \times 10^{-2} \\
-1.23456 & \longrightarrow & -.123456 \times 10^{1} \\
-0.999999 & \longrightarrow & -.999999 \times 10^{0}
\end{array}
$$

Almacenamiento de números reales en computador

- En el computador se almacenan el signo, la mantisa y el exponente (con su signo) del número real.



- La cantidad de dígitos k que se utilizan para almacenar la mantisa es fija. Por lo tanto sólo pueden almacenarse a lo sumo k dígitos de mantisa de un número real.
- La cantidad de dígitos l que se utilizan para almacenar el exponente también es fija. Por lo tanto hay un exponente positivo máximo y otro exponente negativo mínimo que pueden llegar a representarse.

Overflow

- Si se intenta almacenar un número cuyo exponente positivo es mayor que el máximo representable en el sistema, se produce un error fatal: **overflow**. Por lo tanto existe un máximo número positivo que puede representarse.

- En MATLAB el máximo número representable es aproximadamente 10^{308}.
  Si se intenta representar un número mayor, el computador devuelve `Inf` (infinito) como una indicación de que se produjo un overflow:

```
>> a=2^1023
a =
    8.98846567431158e+307

>> b=2^1024
b =
    Inf
```

Underflow

- Si se intenta almacenar un número cuyo exponente negativo es menor que el mínimo representable se produce un error leve: **underflow**. Por lo tanto existe un mínimo número positivo que puede representarse. Si se representa un número positivo menor que ese mínimo, el computador almacena `0`:

- En MATLAB el mínimo número positivo representable es aproximadamente 10^{-323}:

```
>> c=2^-1074
c =
    4.940656458412465e-324

>> d=2^-1075
d =
    0
```

Error de redondeo

- Si se intenta almacenar un número con más dígitos de mantisa que la cantidad máxima k que permite el sistema, sólo se almacenan los primeros k dígitos.

  Por ello, al representar un número real en un sistema de punto flotante, en general se comete un pequeño **error de redondeo**.

- En MATLAB se almacenan entre 15 y 16 dígitos decimales de mantisa:

```
>> e=123456789.0123456789
e =
    1.234567890123457e+08

>> f=pi

f =
    3.14159265358979
```

Operaciones en punto flotante

- El computador calcula cada una de las cuatro operaciones matemáticas fundamentales, suma ($+$), resta ($-$), producto ($*$) y cociente ($/$), de manera tal que el resultado que se obtiene tiene correctos el máximo número k de dígitos representables.

- Por ello, en cada operación matemática también se comete un pequeño error de redondeo:

```
>> g=1.0000000001*1.0000000001
g =
    1.00000000020000

>> h=1+10^-17
h =
    1
```

Constante de precisión

- Como se ve en el ejemplo anterior, si se calcula en punto flotante $1+\varepsilon$, con ε suficientemente pequeño, debido al error de redondeo el resultado que se obtiene es 1.
- La **constante de precisión** (o **unidad de redondeo**) del computador se define como el mínimo número $\varepsilon > 0$ tal que, en punto flotante,
$$1+\varepsilon \neq 1.$$
- Se demuestra que el **error relativo** que comete el computador al representar un número real o al hacer una operación elemental en punto flotante es menor o igual que la contante de precisión.

Constante de precisión (cont.)

- En MATLAB la constante de precisión se llama `eps` y es aproximadamente 10^{-16}:

```
>> eps
ans =
    2.220446049250313e-16

>> p=(1+eps)-1
p =
    2.220446049250313e-16

>> q=(1+eps/2)-1
q =
    0
```

Cancelación excesiva

- Las operaciones de producto y cociente en punto flotante dan siempre resultados confiables, en el sentido que todos los dígitos del resultado que se obtienen son correctos (siempre que no se produzca overflow o underflow).
- Las operaciones de suma y resta usualmente dan resultados con todos sus dígitos correctos. Sin embargo, esto no es así cuando se restan números muy parecidos.

  Este fenómeno se denomina **cancelación excesiva**.

Cancelación excesiva (cont.)

```
>> r=.123456789012345;
>> s=.12345678901234;
>> t=r-s  % El valor exacto es t=5.e-15
t =
     4.996003610813204e-15

>> u=12345678901234567;
>> v=12345678901234566;
>> w=u-v  % El valor exacto es w=1
w =
     2
```

Propagación de errores

- Las integrales

$$I_n := \int_0^1 x^n e^{x-1}\, dx, \qquad n = 1, 2, \ldots$$

se relacionan mediante la expresión

$$I_n = 1 - nI_{n-1}, \qquad n = 2, 3, \ldots$$

que se obtiene fácilmente por integración por partes.

- Esta expresión permite calcular esas integrales recursivamente a partir del valor de I_1, que se calcula exactamente también por partes:

$$I_1 = \int_0^1 xe^{x-1}\, dx = xe^{x-1}\Big|_0^1 - \int_0^1 e^{x-1}\, dx = e^{-1} = 0.36787944...$$

- Si se parte del valor aproximado $\hat{I}_1 \approx 0.367879$ (correctamente redondeado a 6 dígitos) se obtiene:

$$\hat{I}_2 = 0.264242, \qquad \cdots \qquad \hat{I}_9 = -0.06848 < 0 \quad \textbf{!!!}$$

Propagación de errores (cont.)

- Dado que la expresión $I_n = 1 - nI_{n-1}$ es exacta para números reales, el error en $\hat{I}_9$ se debe necesariamente a la propagación del error en $\hat{I}_1$.
- Para estudiar esta propagación sea:

$$\epsilon = I_1 - \hat{I}_1 = e^{-1} - 0.367879 \approx 0.44 \times 10^{-6}.$$

Entonces:

$$\begin{aligned}
\hat{I}_1 &= I_1 - \epsilon, \\
\hat{I}_2 &= 1 - 2\hat{I}_1 = 1 - 2(I_1 - \epsilon) = I_2 + (-2)(-\epsilon), \\
\hat{I}_3 &= 1 - 3\hat{I}_2 = 1 - 3[I_2 + (-2)(-\epsilon)] = I_3 + (-3)(-2)(-\epsilon), \\
&\vdots \\
\hat{I}_9 &= 1 - 9\hat{I}_8 = I_9 + (-9)\cdots(-3)(-2)(-\epsilon) = I_9 - 9!\epsilon.
\end{aligned}$$

Por lo tanto,

$$I_9 - \hat{I}_9 = 9!\epsilon \approx 362\,880 \times 0.44 \times 10^{-6} \approx 0.16 \qquad \text{e} \qquad \hat{I}_9 = -0.06848 \quad \textbf{!}$$

Propagación de errores (cont.)

- Un algoritmo como el anterior, donde el error crece en cada paso, se dice que es **inestable**.

  Este tipo de algoritmos debe evitarse pues propaga de manera catastrófica los errores en los datos, como se ve en el ejemplo.

- La relación anterior puede reformularse para obtener el siguiente algoritmo:

$$\left| \begin{array}{l} I_N \ : \ \text{dato}, \\ I_{n-1} := \dfrac{1 - I_n}{n}, \qquad n = N, N-1, \ldots, 3, 2. \end{array} \right.$$

- Un análisis semejante al anterior permite demostrar que, en el paso n-ésimo de este algoritmo, el error del paso anterior se divide por n. En consecuencia este algoritmo no sólo no amplifica los errores, sino que los reduce.

- Un algoritmo como éste, que no amplifica los errores, se dice que es **estable**.

- Por ejemplo, si se toma $N = 20$ e $\hat{I}_{20} = 0$, verifique que mediante este algoritmo se obtiene $\hat{I}_9 = 0.091\,612\,292\,990$, con error menor que 10^{-12}.