# Systèmes Robotiques Hétérogènes et Coopératifs

UPJV, Département EEA

**Fabio MORBIDI**

Laboratoire MIS
Équipe Perception Robotique
E-mail : `fabio.morbidi@u-picardie.fr`

CM: Mercredi et Vendredi 13h30-16h30, salle CURI 8

TD, TP : salle TP203/204

**AU 2025-2026**

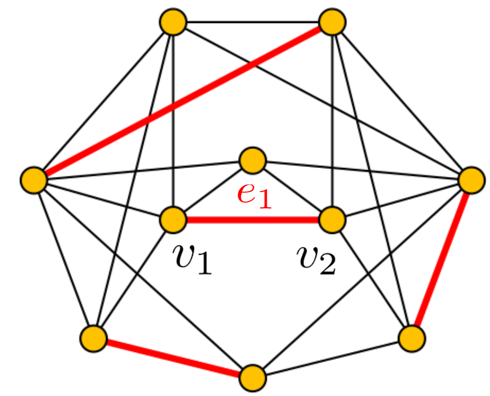
# Plan du cours

1. Introduction aux systèmes multi-agents
2. Théorie des graphes
3. Systèmes dynamiques connectés en réseaux : protocole de consensus
4. Traitement du signal sur graphes

# Théorie des graphes : introduction

- Les graphes fournissent une **abstraction naturelle** pour l‘échange d‘informations entre les agents dans un réseau

- L‘abstraction basée sur la théorie des graphes fournit une description à **haut niveau** de la topologie du réseau par rapport à des objets simples appelés *sommets* et *arêtes*




---

Dans les prochaines slides, nous verrons quelques notions de base de :

- **Théorie algébrique des graphes** (matrice des degrés, matrice d‘adjacence, matrice d‘incidence, matrice laplacienne) [1]

- **Théorie spectrale des graphes** (spectre de la matrice laplacienne) [2]

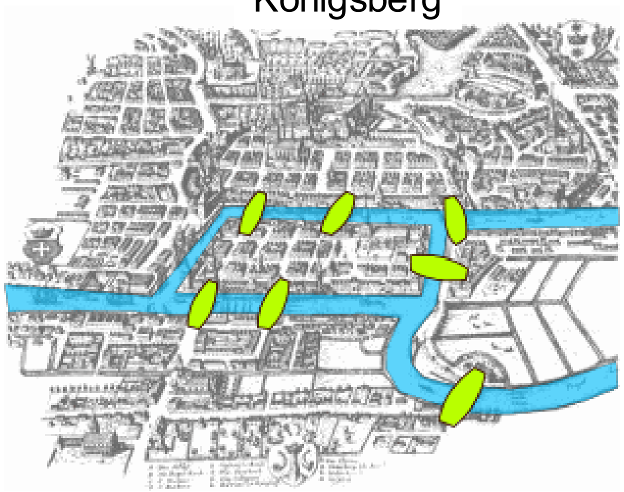

[1] “*Algebraic Graph Theory*“, C. Godsil, G. Royle, Springer, 2001

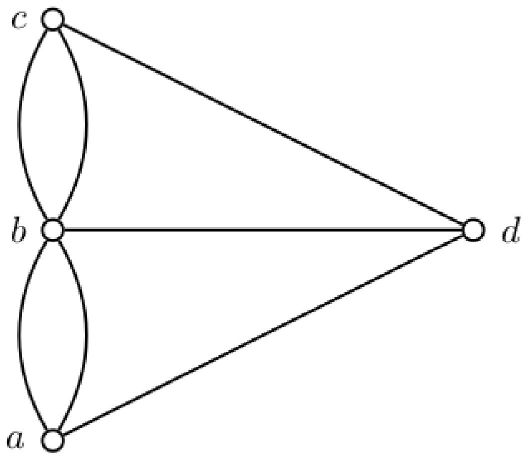
[2] “*Spectral Graph Theory*“, F.R.K. Chung, American Mathematical Society, 1997

# Théorie des graphes : petit historique

- Le problème des ***7 ponts de Königsberg*** est le premier étudié en théorie des graphes
- Trouver un parcours dans la ville qui traverse chaque pont *une fois seulement* (en d'autres terms, trouver un *chemin eulérien* dans le graphe correspondant)
- *Réponse négative* de Leonhard Euler en 1735
- Un chemin eulérien existe si le graphe est connexe et il y a exactement zéro ou deux sommets de *degré impair* (dans le problème de Königsberg, il y a 4 sommets de degré impair)

Königsberg

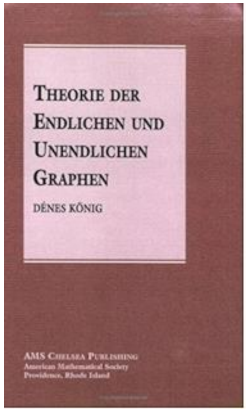




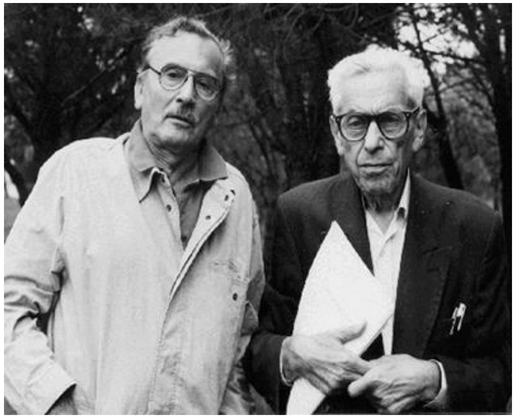
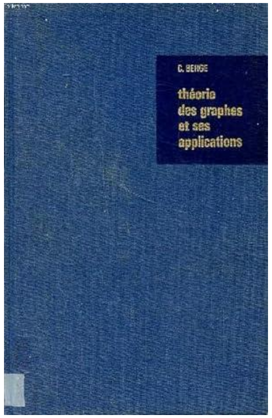

Leonhard Euler
(1707-1783)

# Théorie des graphes : petit historique

- Dénes Kőnig a écrit le **1er livre** sur la théorie des graphes “*Theorie der endlichen und unendlichen Graphen*“ (“*Théorie des graphes finis et infinis*“) en 1936
  - Début de la théorie des graphes comme une *branche indépendante* des mathématiques
- En suivant Kőnig, Claude Berge a écrit le **2e livre** sur la théorie des graphes, “*Théorie des graphes et ses applications*“ en 1958


Dénes Kőnig (1884-1944)


Claude Berge (1926-2002)
avec Paul Erdős (1913-1996)

# Théorie des graphes : notions de base

- Un graphe *fini*, *non orienté*, *simple*, ou **graphe** en bref, est fondé sur l'ensemble fini :

$$V = \{v_1,\, v_2,\, \ldots,\, v_n\}$$ l'ensemble des **sommets** (ou **nœuds**)

- On définit aussi l'ensemble des **arêtes** (ou **arcs**) :

$$E \subseteq V \times V$$ Il consiste en éléments de la forme $\{v_i,\, v_j\}$ ou $v_i v_j$ tels que $i, j \in \{1, 2, \ldots, n\}$, $i \neq j$.

- Le graphe $\mathcal{G}$ est alors défini formellement comme le couple :

$$\mathcal{G} = (V,\, E)$$

# Théorie des graphes : notions de base

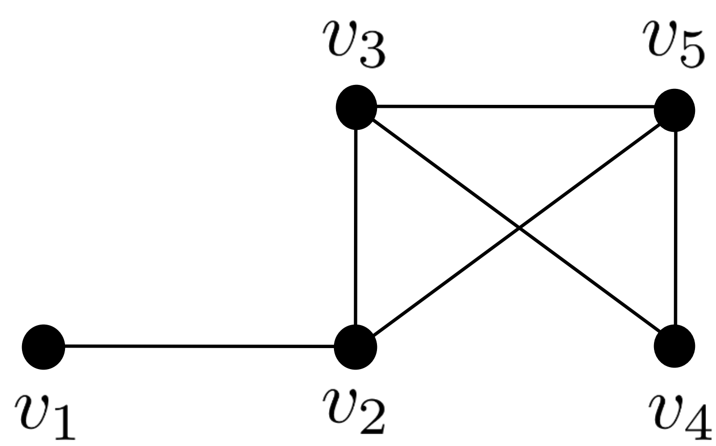
Exemple:

$$\mathcal{G} = (V,\, E)$$

avec

$$V = \{v_1,\, v_2, \ldots,\, v_5\}$$

$$E = \{v_1v_2, v_2v_3, v_3v_4, v_3v_5, v_2v_5, v_4v_5\}$$




- S'il y a une arête entre deux sommets $v_i$ et $v_j$, ils sont appelés **adjacents** qui s'écrit $v_i \sim v_j$. Dans ce cas, $v_iv_j$ est appelée **incidente** avec sommets $v_i, v_j$
- Le **voisinage** $\mathcal{N}(i) \subseteq V$ du sommet $v_i$ est l'ensemble $\{v_j \in V \mid v_iv_j \in E\}$ c'est-à-dire, l'ensemble des sommets adjacents à $v_i$. Dans l'exemple ci-dessus :

$$\mathcal{N}(1) = \{v_2\}, \quad \mathcal{N}(5) = \{v_2,\, v_3,\, v_4\}$$

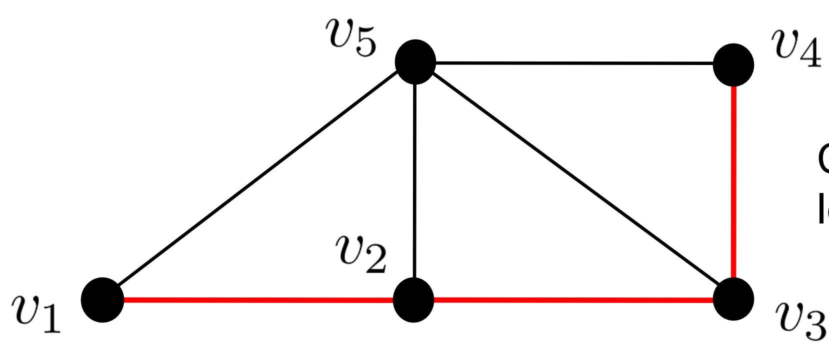
# Théorie des graphes : notions de base

- Un **chemin de longueur** $m$ dans $\mathcal{G}$ est donné par la séquence de sommets distincts

$$v_{i_0},\ v_{i_1},\ \ldots,\ v_{i_m}$$

tels que pour $k \in \{0,\ 1,\ \ldots, m-1\}$, les sommets $v_{i_k}$ et $v_{i_{k+1}}$ sont adjacents

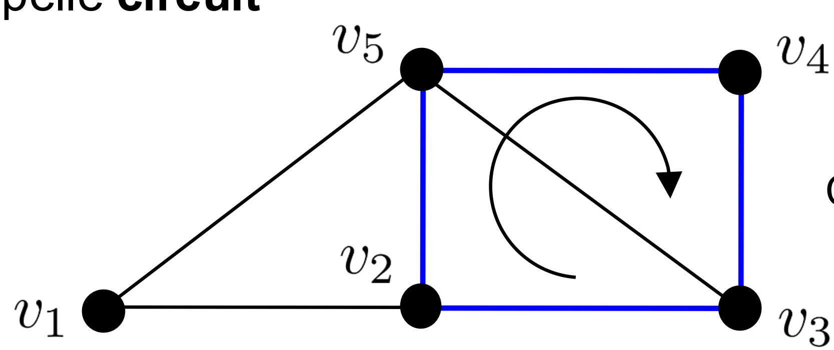
Exemple:




Chemin de longueur 3 entre les sommets 1 et 4 (rouge)

- Si tous les sommets du chemin sont distincts sauf le sommet initial et final, le chemin s'appelle **circuit**

Exemple:

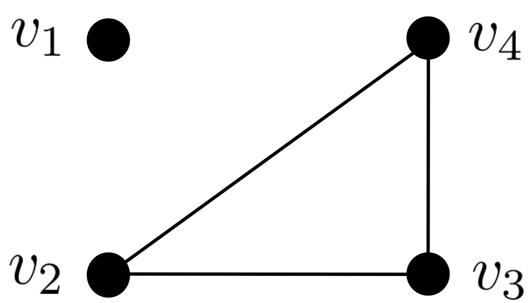



Circuit de longueur 4 (bleu)

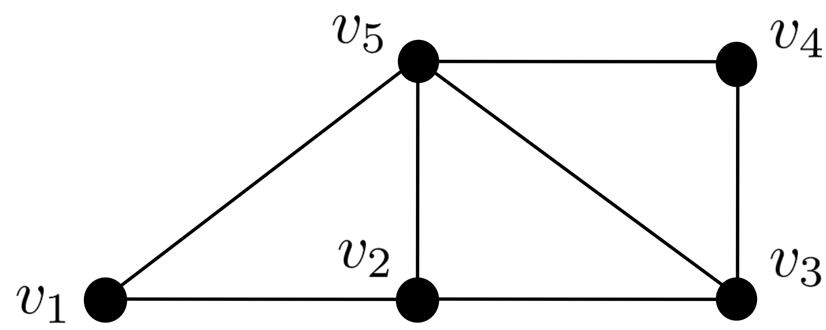
# Théorie des graphes : notions de base

- Un graphe $\mathcal{G}$ est **connexe** s'il existe un chemin entre tout couple de sommets dans $V(\mathcal{G})$ (c'est-à-dire, s'il est d'un seul tenant). Sinon, on dit que le graphe est **non connexe**

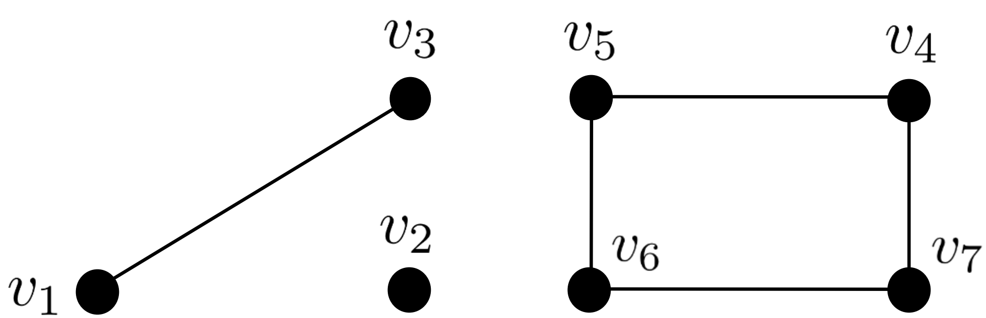
Exemples:




Non connexe (2 composantes connexes)




Connexe

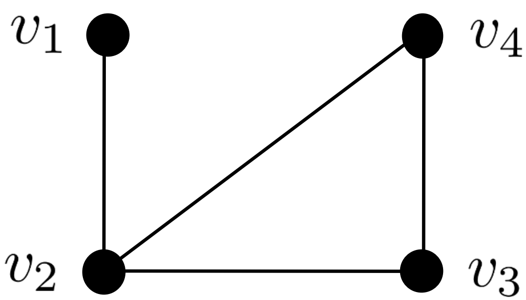



Non connexe (3 composantes connexes)

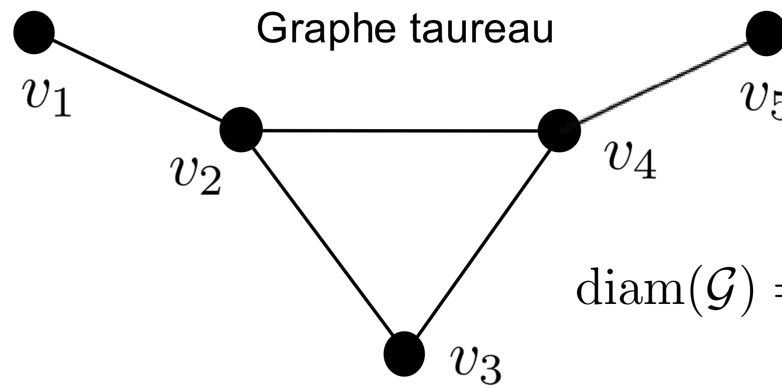
# Théorie des graphes : notions de base

- Le **diamètre** d'un graphe $\mathcal{G}$, noté $\mathrm{diam}(\mathcal{G})$, est la *plus grande distance possible* qui puisse exister entre deux de ses sommets (la distance entre deux sommets est définie par la longueur d'un plus court chemin entre ces deux sommets)

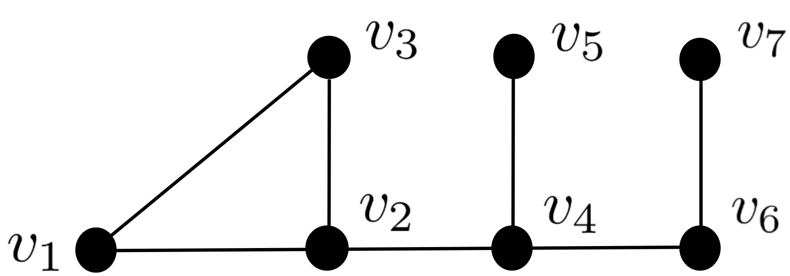
<u>Exemples</u>:




$\mathrm{diam}(\mathcal{G}) = 2$




$\mathrm{diam}(\mathcal{G}) = 3$



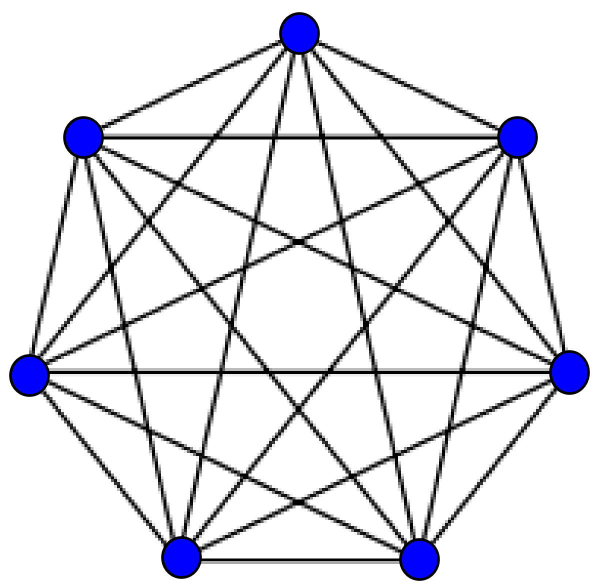

$\mathrm{diam}(\mathcal{G}) = 4$

# Familles de graphes

- $K_n$ : ***Graphe complet*** (ou ***complètement connexe***)

  Dans un graphe complet chaque sommet est relié à tous les autres (tous les sommets sont adjacents deux à deux)

Exemple: $K_7$ (sept sommets)

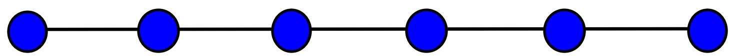


# Familles de graphes

- $P_n$ : ***Graphe chaîne***

$P_n = (\{v_1, \ldots, v_n\},\ E_P)$, où $v_i v_j \in E_P$ si et seulement si $j = i + 1$,
$i \in \{1, \ldots, n-1\}$

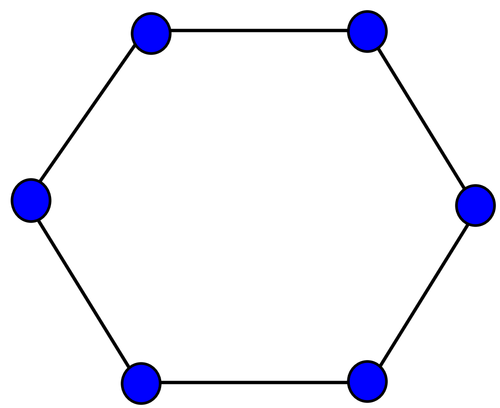
<u>Exemple</u>: $P_6$



- $C_n$ : ***Graphe cycle***

$C_n = (\{v_1, \ldots, v_n\},\ E_C)$, où $v_i v_j \in E_C$ si et seulement si
$i - j \equiv \pm 1 \mod n$

<u>Exemple</u>: $C_6$



<u>Remarque</u> :

$a$ et $b$ sont congruents modulo $n$ :

$$a \equiv b \mod n$$

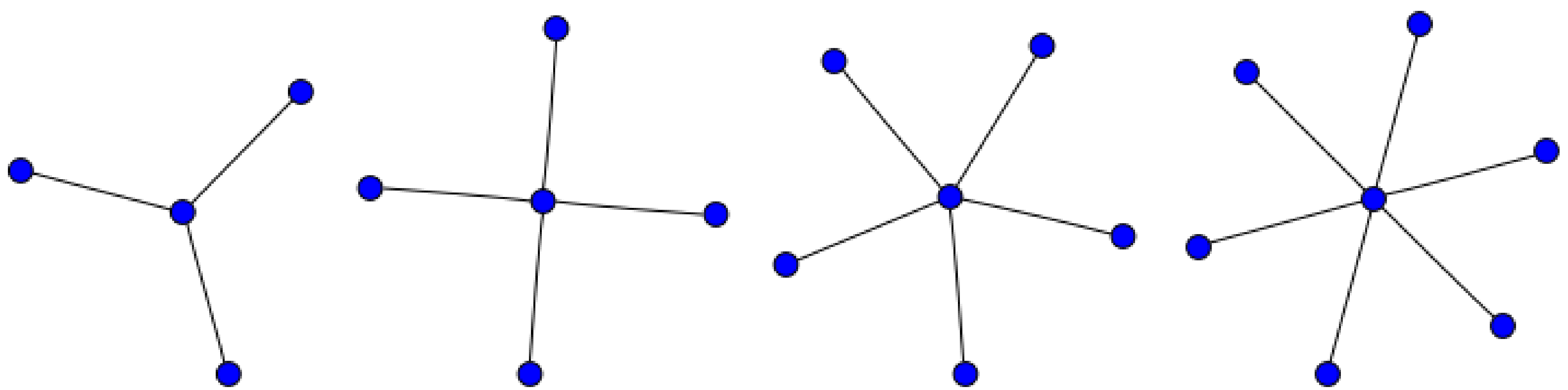
si $a - b$ est divisible par $n$, c'est-à-dire si $a = b + kn$ avec $k$ entier

# Familles de graphes

- $S_n$ : ***Graphe étoile***

  $S_n = (\{v_1, \ldots, v_n\},\, E_S)$ où $v_i v_j \in E_S$ si et seulement si $i = 1$ ou $j = 1$

<u>Exemples</u>:



$S_4$ $S_5$ $S_6$ $S_7$

**Remarque** : par convention, le sommet 1 est le sommet au centre de l'étoile

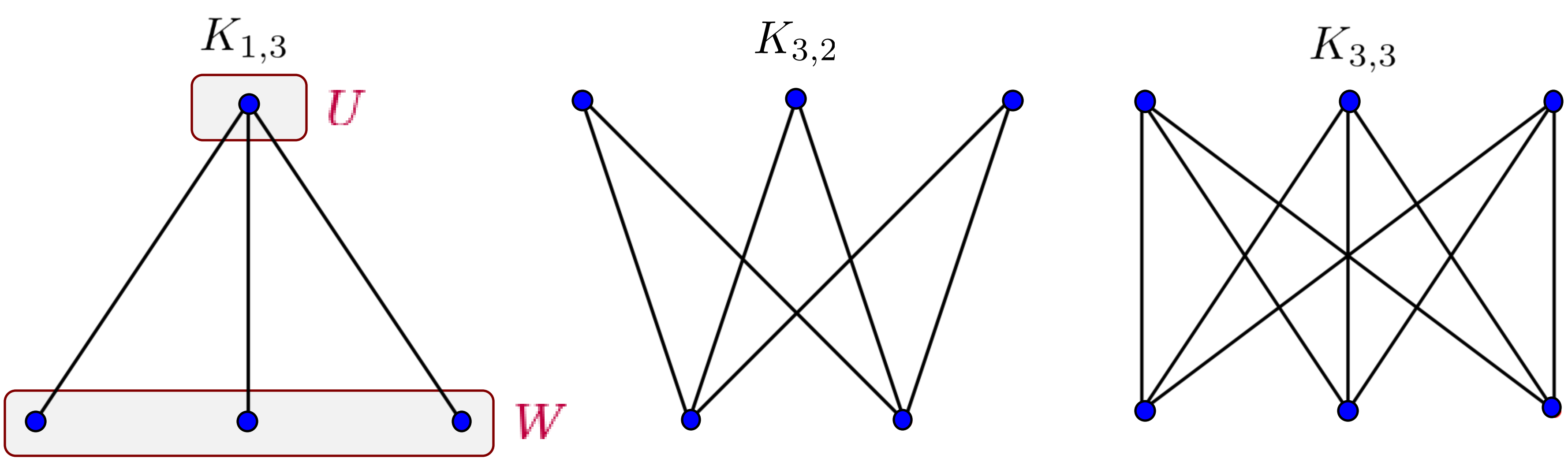
# Familles de graphes

- $K_{m,n}$ : ***Graphe biparti complet (ou biclique)***

• Dans un graphe biparti complet, il existe une partition de l'ensemble de sommets $V$ en deux sous-ensembles $U$ et $W$ ($V = U \cup W$) telle que *chaque sommet* de $U$ est relié à *chaque sommet* de $W$. En d'autres termes : $\mathcal{G} = (U \cup W, U \times W)$

• Si $U$ contient $m$ sommets et $W$ contient $n$ sommets, le graphe est noté $K_{m,n}$

Exemples:




**Remarque**: $K_{1,n} = S_{n+1}$ (graphe étoile)

# Graphes réguliers

- Un **graphe régulier** est un graphe où tous les sommets ont le *même nombre de voisins*, c'est-à-dire le même degré

- Un graphe régulier dont les sommets sont de degré $k$ est appelé un graphe $k$-**régulier** (ou graphe régulier de degré $k$)

<u>Exemples</u>:

$C_n$ : 2-régulier

$K_n$ : $(n-1)$-régulier

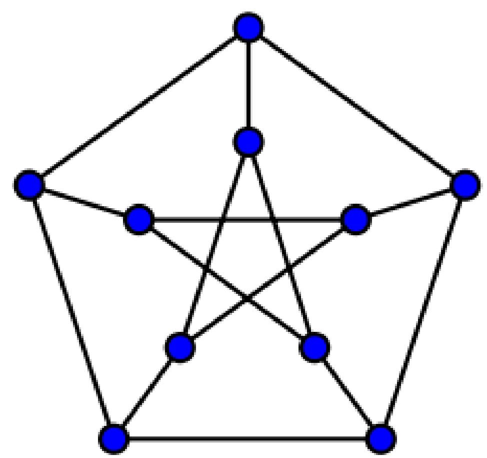
$K_{n,n}$ : $n$-régulier

Graphe de Petersen : 3-régulier

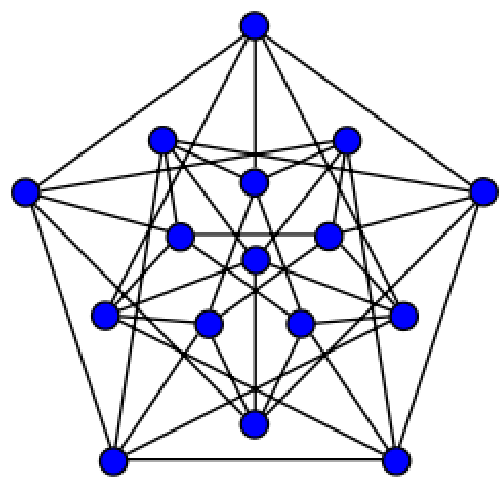
Graphe de Dürer : 3-régulier

Graphe de Clebsch : 5-régulier

Graphe de Schläfli : 16-régulier



Graphe de Petersen



Graphe de Clebsch

# Généralisation de la notion de graphe

**Graphes orientés (digraphes)**

- Si on associe des *directions* aux arêtes d'un graphe, l'interconnexion résultante n'est plus considérée comme un graphe non orienté
- Un ***graphe orienté*** (ou ***digraphe***), noté $\mathcal{D} = (V,\, E)$, peut être obtenu de deux manières différentes :

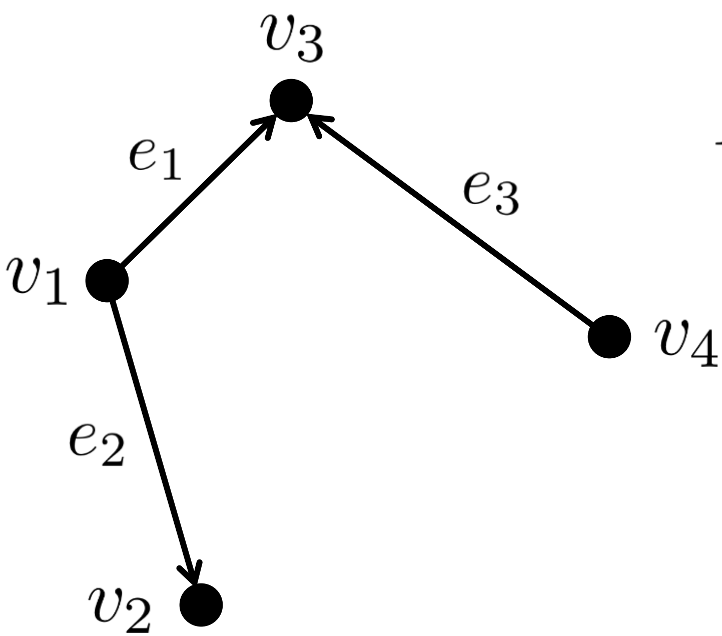
1. On supprime la condition que l'ensemble des arêtes $E$ contient des *couples non ordonnées* de sommets :

   Si le couple ordonné $(v_i,\, v_j) \in E$, alors on dit que $v_i$ est la **queue** (point de départ de la flèche) de l'arête, tandis que $v_j$ est la **tête** (point terminal de la flèche)

2. On associe une orientation $o : E \rightarrow \{-1,\, 1\}$ à l'ensemble non ordonné des arêtes

# Graphes orientés

<u>Exemple</u>:




$$V = \{v_1,\ v_2,\ v_3,\ v_4\}$$

$$E = \{(v_1,\ v_3),\ (v_1, v_2),\ (v_4,\ v_3)\}$$

ou plus simplement

$$E = \{e_1,\ e_2,\ e_3\}$$

# Graphes orientés

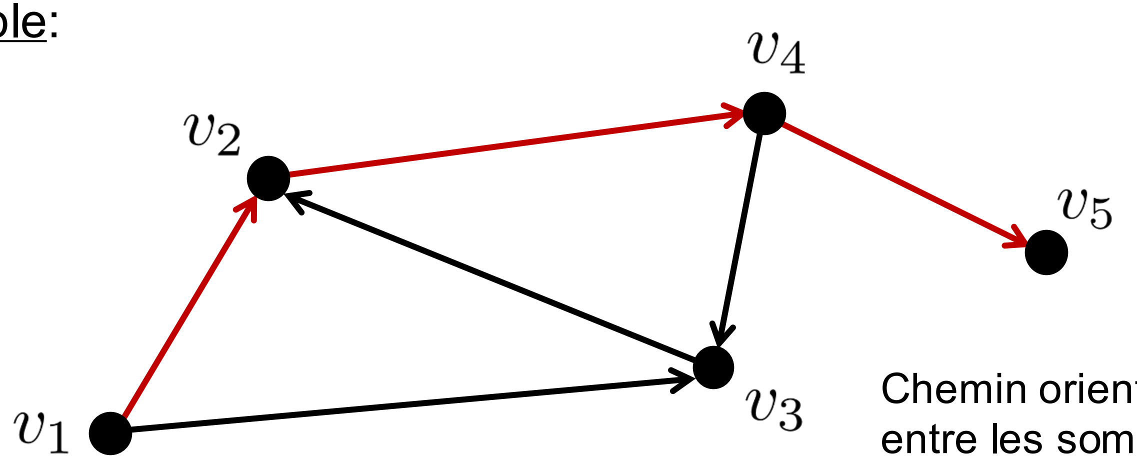
- Les notions d'**adjacence**, de **voisinage,** de **chemin** et de **connexité** vues précédemment peuvent être aisément étendues aux digraphes

  Par exemple, un **chemin orienté** de longueur $m$ dans le digraphe $\mathcal{D}$ est donné par la sequence de sommets distincts :

$$v_{i_0},\ v_{i_1},\ \ldots,\ v_{i_m}$$

  telle que, pour $k \in \{0,\, 1,\, \ldots, m-1\}$, nous avons $(v_{i_k}, v_{i_{k+1}}) \in E(\mathcal{D})$

Exemple:




Chemin orienté (rouge) de longueur 3 entre les sommets 1 et 5

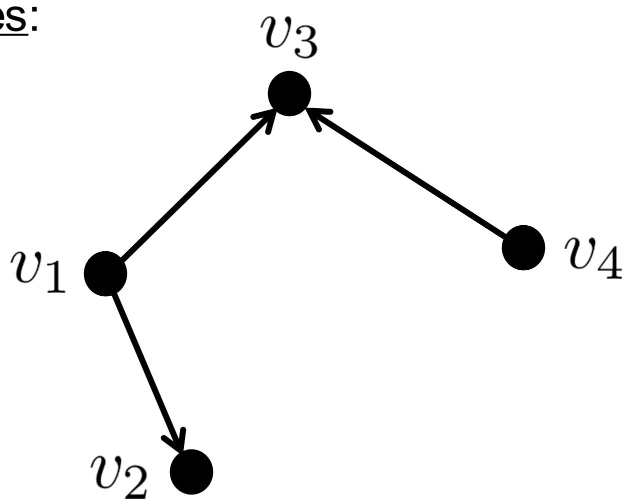
# Graphes orientés

- On dit que un digraphe est :

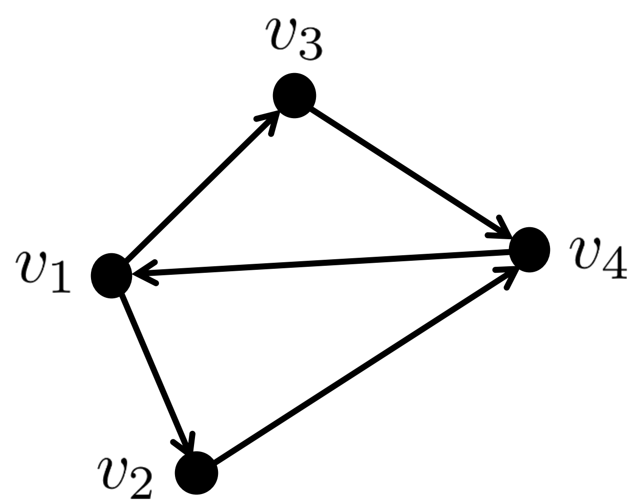
  **a)** **Fortement connexe** : si entre tout couple de sommets, il y a un chemin orienté

  **b)** **Faiblement connexe** : s'il est connexe lorsqué il est consideré comme un graphe, c'est-à-dire comme un graphe *sans orientation*

Exemples:




Digraphe *faiblement connexe* mais **pas** fortement connexe



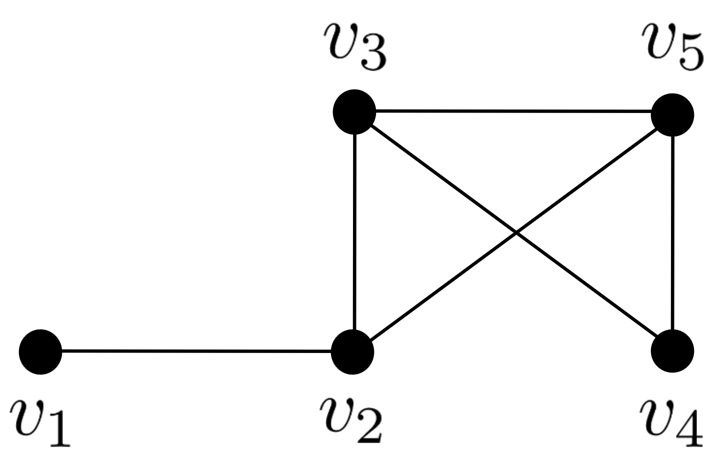

Digraphe *fortement connexe* (et, par consequent, aussi faiblement connexe)

# Graphes et matrices

- Outre une représentation graphique en termes de *sommets* et d'*arêtes*, chaque graphe admet une représentation sous forme ***matricielle***

- Pour un graphe non orienté $\mathcal{G}$, le **degré** d'un sommet $v_i$, noté $d(v_i)$, est la cardinalité du voisinage $\mathcal{N}(i)$, c'est-à-dire le nombre de sommets qui sont adjacents au sommet $v_i$ dans $\mathcal{G}$

<u>Exemple</u>:




$$d(v_1) = 1$$
$$d(v_2) = 3$$
$$d(v_3) = 3$$
$$d(v_4) = 2$$
$$d(v_5) = 3$$

# Matrice des degrés et matrice d'adjacence

- La **matrice des degrés** d'un graphe $\mathcal{G}$ est la matrice diagonale $n \times n$ qui contient le degré de chaque sommet de $\mathcal{G}$ sur la diagonale :

$$\boldsymbol{\Delta}(\mathcal{G}) = \mathrm{diag}(d(v_1),\, d(v_2), \ldots, d(v_n)).$$

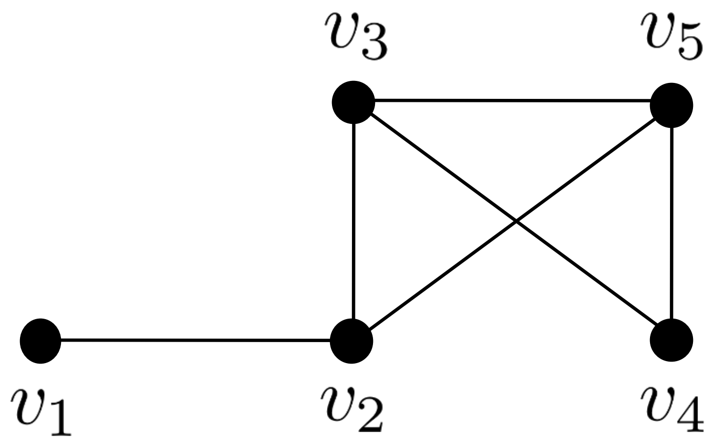
- La **matrice d'adjacence** $\mathbf{A}(\mathcal{G})$ est la matrice symétrique $n \times n$ qui codifie les relations d'adjacence dans le graphe $\mathcal{G}$ :

$$[\mathbf{A}(\mathcal{G})]_{ij} = \begin{cases} 1 & \text{si } \; v_i v_j \in E, \\ 0 & \text{sinon.} \end{cases}$$

Élément $(i,\, j)$ de la matrice $\mathbf{A}(\mathcal{G})$

# Matrice des degrés et matrice d'adjacence

Exemple:




Matrice symétrique

$$\boldsymbol{\Delta}(\mathcal{G}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}, \qquad \mathbf{A}(\mathcal{G}) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

# Matrice d'incidence

- Soit $\mathcal{G}^o$ le digraphe obtenu en associant une orientation *arbitraire* aux arêtes du graphe $\mathcal{G}$. La **matrice d'incidence** $\mathbf{D}(\mathcal{G}^o)$ est une matrice $n \times m$ définie comme suit ($n =$ nombre de sommets; $m =$ nombre d'arêtes) :

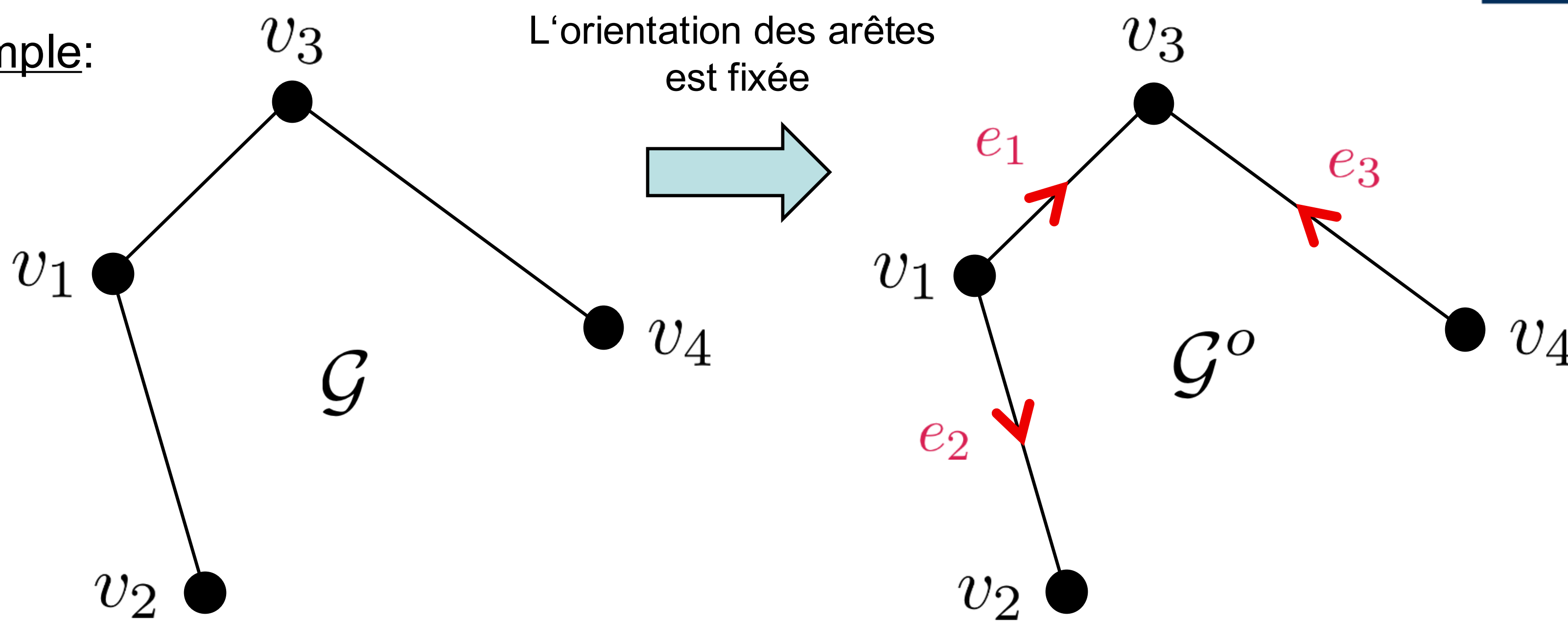
$$[\mathbf{D}(\mathcal{G}^o)]_{ij} = \begin{cases} -1 & \text{si } v_i \text{ est la queue de } e_j, \\ 1 & \text{si } v_i \text{ est la tête de } e_j, \\ 0 & \text{sinon.} \end{cases}$$

**Remarque:**

$\mathbf{D}(\mathcal{G}^o)$ capture non seulement la relation d'adjacence, mais aussi l'information sur l'***orientation*** que le graphe possède désormais

# Matrice d'incidence

Exemple:




$$\mathbf{D}(\mathcal{G}^o) = \begin{bmatrix} -1 & -1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix}$$

Sommets $v$ $\downarrow$

Arêtes $e$ $\rightarrow$

# Matrice d‘incidence

$$\mathbf{D}(\mathcal{G}^o) = \begin{bmatrix} \boxed{\begin{matrix} -1 \\ 0 \\ 1 \\ 0 \end{matrix}} & \begin{matrix} -1 \\ 1 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ 1 \\ -1 \end{matrix} \end{bmatrix}$$

**Remarques :**

- La somme des éléments de chaque colonne de $\mathbf{D}(\mathcal{G}^o)$ est zéro
- La matrice d‘incidence d‘un **digraphe** $\mathcal{D}$ peut être définie de manière similaire, en omettant l‘étape de pre-orientation des arêtes nécessaire pour les graphes non orientés
  - Dans ce cas, la matrice d‘incidence est notée $\mathbf{D}(\mathcal{D})$

# Matrice laplacienne

La **matrice laplacienne** associée au graphe non orienté $\mathcal{G}$ est définie comme suit :

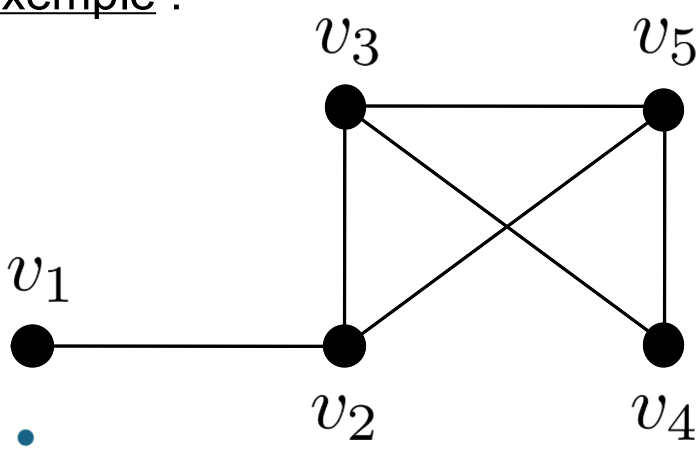
$$\mathbf{L}(\mathcal{G}) = \mathbf{\Delta}(\mathcal{G}) - \mathbf{A}(\mathcal{G}).$$

Il découle de cette definition, que pour tout graphe la somme des éléments dans *chaque ligne* de la matrice laplacienne est **zero** :

$$\mathbf{L}(\mathcal{G})\mathbf{1} \; = \; \mathbf{0} \quad \text{avec} \quad \mathbf{1} = [1,\, 1,\, \ldots,\, 1]^T.$$

Pierre-S. Laplace (1749-1827)

<u>Exemple</u> :




$$\mathbf{L}(\mathcal{G}) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{bmatrix}.$$

# Matrice laplacienne : définition alternative

Étant donnée une *orientation arbitraire* de l'ensemble des arêtes $E(\mathcal{G})$, la **matrice laplacienne** du graphe $\mathcal{G}$ peut être définie de façon alternative à partir de la matrice d'incidence, comme suit :

$$\mathbf{L}(\mathcal{G}) = \mathbf{D}(\mathcal{G}^o)\,\mathbf{D}^T(\mathcal{G}^o)$$

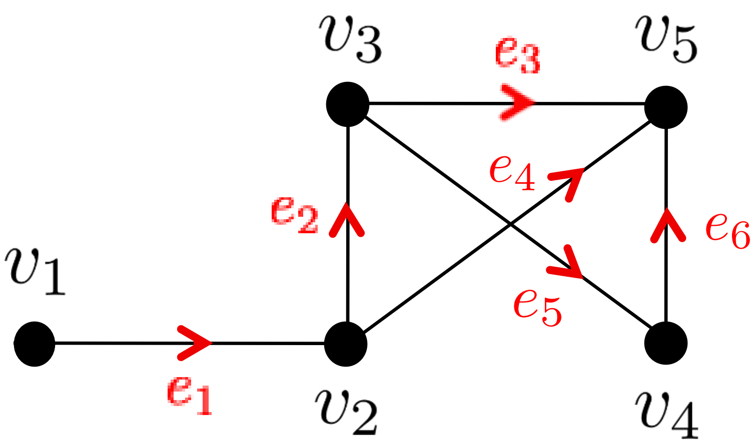
Cette définition montre que $\mathbf{L}(\mathcal{G})$ est une matrice :

- Symétrique (c'est-à-dire, $\mathbf{L}(\mathcal{G}) = \mathbf{L}^T(\mathcal{G})$)
- Semi-définie positive (c'est-à-dire, $\mathbf{x}^T\,\mathbf{L}(\mathcal{G})\,\mathbf{x} \geq 0,\ \forall\,\mathbf{x} \in \mathbb{R}^n$)

**Remarque :**
Les deux définitions que nous avons données sont *équivalentes* et du moment que la notion d'orientation n'est pas nécessaire dans la première, on conclut que la matrice laplacienne est **indépendante de l'orientation**

# Matrice laplacienne : définition alternative

Exemple:




$$\mathbf{D}(\mathcal{G}^o) = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$$\mathbf{L}(\mathcal{G}) = \mathbf{D}(\mathcal{G}^o)\,\mathbf{D}^T(\mathcal{G}^o) = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & -1 & 3 \end{bmatrix}$$

# Matrice laplacienne : d'autres propriétés

- Nous avons que

$$\mathrm{trace}(\mathbf{L}) = \sum_{i=1}^{n} [\mathbf{L}]_{ii} = 2m$$

avec $m$ le nombre d'arêtes de $\mathcal{G}$

- La matrice laplacienne d'un graphe $\mathcal{G}$ peut être écrite comme la *somme de produits extérieurs* de vecteurs :

$$\mathbf{L} = \sum_{(i,j)\in E} (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T$$

où $\mathbf{e}_i \in \mathbb{R}^n$ est le vecteur unitaire élementaire dans la direction $i$ (par exemple $\mathbf{e}_1 = [1,\, 0,\, \ldots, 0]^T$ et $\mathbf{e}_2 = [0,\, 1,\, 0, \ldots, 0]^T$)

# Matrice laplacienne pondérée

- Étant donné un *graphe pondéré* $\mathcal{G} = (V,\, E,\, w)$ la **matrice laplacienne pondérée** est définie comme suit :

$$\mathbf{L}_w(\mathcal{G}) \;=\; \mathbf{D}(\mathcal{G}^o)\,\mathbf{W}\,\mathbf{D}^T(\mathcal{G}^o)$$

où

$$\mathbf{W} = \mathrm{diag}(w(e_1),\, w(e_2), \ldots, w(e_m)) \in \mathbb{R}^{m\times m}\,,$$

avec poids :

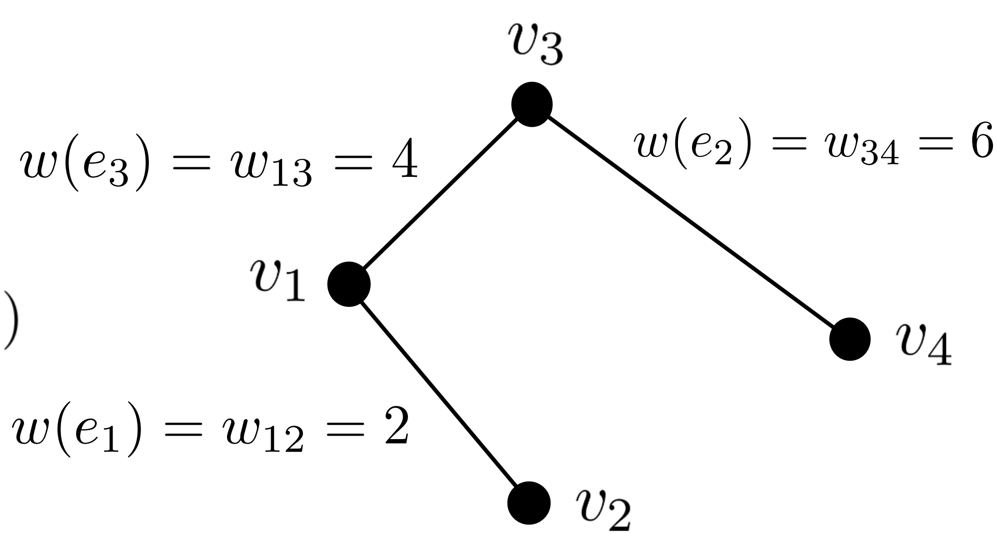
$$w(e_j) > 0,\;\; j \in \{1, 2, \ldots, m\}.$$

Exemple:

$$V = \{v_1,\, v_2,\, v_3,\, v_4\}$$

$$E = \{e_1,\, e_2,\, e_3\}$$

$$\mathbf{W} \;=\; \mathrm{diag}(w(e_1),\, w(e_2),\, w(e_3))$$



# Matrice laplacienne pondérée (digraphes)

Soit $\mathcal{D} = (V,\, E,\, w)$ un graphe orienté pondéré.

- Pour la **matrice d'adjacence**, on peut choisir :

$$[\mathbf{A}(\mathcal{D})]_{ij} = \begin{cases} w_{ij} & \text{si } (v_j,\, v_i) \in E(\mathcal{D}), \\ 0 & \text{sinon}, \end{cases}$$

- Pour la **matrice des degrés** $\mathbf{\Delta}(\mathcal{D})$, on peut faire le choix suivant :

$$\mathbf{\Delta}(\mathcal{D}) = \operatorname{diag}(d_{\text{in}}(v_1),\, d_{\text{in}}(v_2), \ldots, d_{\text{in}}(v_n)),$$

où

$$d_{\text{in}}(v_i) = \sum_{\{j \,|\, (v_j,\, v_i) \in E(\mathcal{D})\}} w_{ij},$$

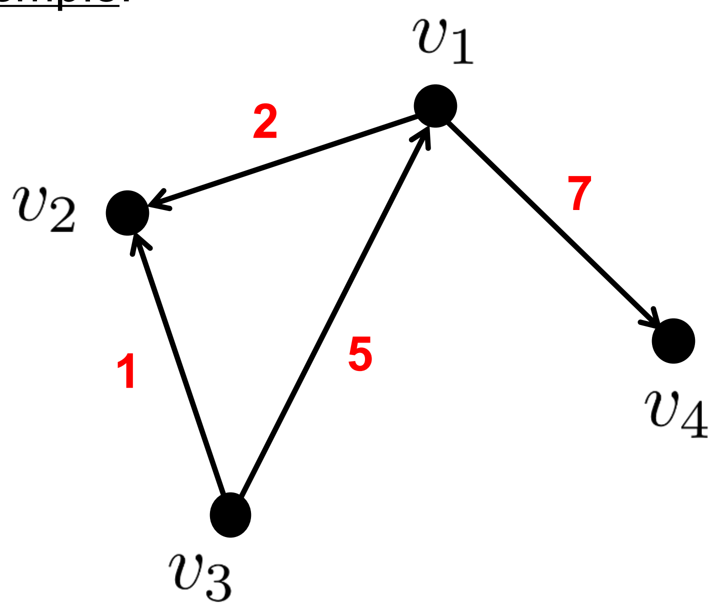
est le **degré pondéré entrant** du sommet $v_i$

- La **matrice laplacienne pondérée** (degré entrant) est alors définie comme :

$$\mathbf{L}(\mathcal{D}) = \mathbf{\Delta}(\mathcal{D}) - \mathbf{A}(\mathcal{D})$$

# Matrice laplacienne pondérée (digraphes)

Exemple:




$$\mathbf{\Delta}(\mathcal{D}) = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 \end{bmatrix}$$

$$\mathbf{A}(\mathcal{D}) = \begin{bmatrix} 0 & 0 & 5 & 0 \\ 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{L}(\mathcal{D}) = \begin{bmatrix} 5 & 0 & -5 & 0 \\ -2 & 3 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -7 & 0 & 0 & 7 \end{bmatrix}$$

Non symétrique !!

# Matrice laplacienne pondérée (digraphes)

- On peut toujours partitionner la matrice d'incidence d'un digraphe $\mathcal{D}$ comme suit :

$$\mathbf{D} = \mathbf{D}_{\odot} + \mathbf{D}_{\otimes}$$

où $\mathbf{D}_{\odot}(\mathcal{D}) \in \mathbb{R}^{n \times m}$ est la *matrice d'incidence entrante*, définie par

$$[\mathbf{D}_{\odot}]_{ij} = \begin{cases} -1 & \text{si } v_i \text{ est la queue de } e_j, \\ 0 & \text{sinon,} \end{cases}$$

et $\mathbf{D}_{\otimes}(\mathcal{D}) \in \mathbb{R}^{n \times m}$ est à la *matrice d'incidence sortante*, définie par

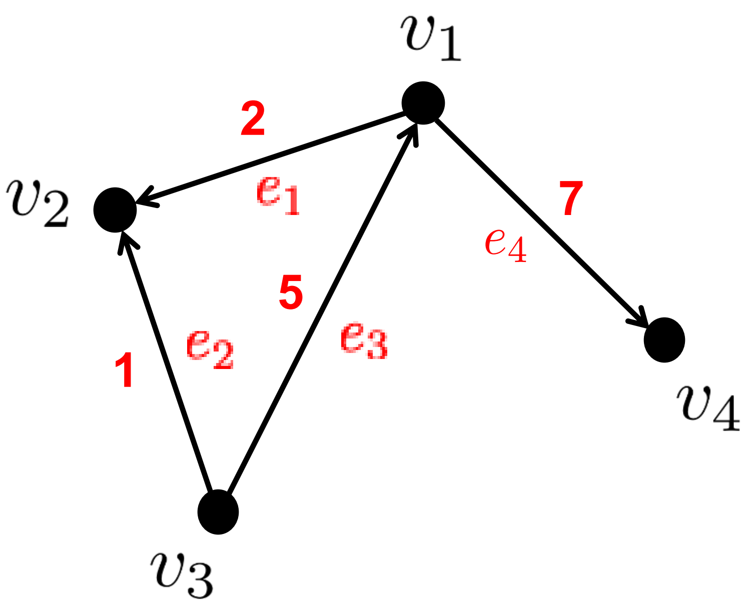
$$[\mathbf{D}_{\otimes}]_{ij} = \begin{cases} 1 & \text{si } v_i \text{ est la tête de } e_j, \\ 0 & \text{sinon.} \end{cases}$$

- On peut ainsi définir la **matrice laplacienne pondérée** (degré entrant) en fonction de la *matrice d'incidence* et de la *matrice d'incidence sortante*, comme :

$$\mathbf{L}(\mathcal{D}) = \mathbf{D}_{\otimes}(\mathcal{D})\mathbf{W}\,\mathbf{D}^{T}(\mathcal{D})$$

# Matrice laplacienne pondérée (digraphes)

Exemple (revisité) :




$$\mathbf{D}(\mathcal{D}) = \begin{bmatrix} -1 & 0 & 1 & -1 \\ 1 & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{D}_{\odot}(\mathcal{D}) = \begin{bmatrix} -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{D}_{\otimes}(\mathcal{D}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{W} = \mathrm{diag}(2,\, 1,\, 5,\, 7)$$

$$\begin{aligned} \mathbf{L}(\mathcal{D}) &= \mathbf{D}_{\otimes}(\mathcal{D})\,\mathbf{W}\,\mathbf{D}^T(\mathcal{D}) \\ &= \begin{bmatrix} 5 & 0 & -5 & 0 \\ -2 & 3 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -7 & 0 & 0 & 7 \end{bmatrix} \end{aligned}$$

# Théorie algébrique et spectrale des graphes

- La **théorie algébrique des graphes** associe des *objets algébriques* (par ex. la matrice des degrés, d'adjacence, laplacienne) à un graphe
- La **théorie spectrale des graphes** étudie les *valeurs et les vecteurs propres* associés à la matrice d'adjacence et à la matrice laplacienne d'un graphe

---

Petit rappel ….

**Définition :**
Un vecteur non nul $\mathbf{u} \in \mathbb{C}^n$ est un *vecteur propre* de la matrice $\mathbf{A} \in \mathbb{R}^{n \times n}$ si et seulement si il existe un scalaire $\lambda \in \mathbb{C}$ tel que :

$$\mathbf{A}\,\mathbf{u} = \lambda\,\mathbf{u}$$

où $\lambda$ s'appelle *valeur propre* associée à $\mathbf{u}$.

- On trouve les $n$ valeurs propres et vecteur propres de $\mathbf{A}$ en résolvant le système d'équations :

$$(\mathbf{A} - \lambda\,\mathbf{I}_n)\mathbf{u} = \mathbf{0}$$

# Spectre de la matrice laplacienne

Soit $\mathbf{L}(\mathcal{G})$ la matrice laplacienne du graphe non orienté $\mathcal{G}$. Cette matrice est *symétrique* et *semi-définie positive.* Par consequent, ses $n$ *valeurs propres réelles* peuvent être ordonnées de la manière suivante :

$$0 = \lambda_1(\mathbf{L}(\mathcal{G})) \leq \lambda_2(\mathbf{L}(\mathcal{G})) \leq \ldots \leq \lambda_n(\mathbf{L}(\mathcal{G})).$$

**Théorème:**

Le graphe $\mathcal{G}$ est **connexe** si et seulement si :

$$\lambda_2(\mathbf{L}(\mathcal{G})) > 0$$

En outre, le nombre de *composantes connexes* d'un graphe non-orienté $\mathcal{G}$ est égal à la *multiplicité algébrique* de la valeur propre 0 de la matrice laplacienne $\mathbf{L}(\mathcal{G})$

# Spectre de la matrice laplacienne

- **Il n'est pas aisé** de trouver le spectre, c'est-à-dire l'ensemble des valuers propres $\{\lambda_1(\mathbf{L}),\, \lambda_2(\mathbf{L}), \ldots, \lambda_n(\mathbf{L})\}$, de la matrice laplacienne d'un *graphe* $\mathcal{G}$ *quelconque*

- Toutefois, les valeurs propres (et les vecteurs propres) de la matrice laplacienne de certaines familles de graphes admettent une expression de forme fermée

---

**Graph complet**

Puisque $\mathbf{L}(K_n) = -\mathbf{1}\mathbf{1}^T + n\mathbf{I}_n$, le spectre de $\mathbf{L}(K_n)$ est celui de $-\mathbf{1}\mathbf{1}^T$ décalé de $n$. Le spectre de la matrice $\mathbf{1}\mathbf{1}^T$ est $\{0,\, \ldots,\, 0,\, n\}$, donc le spectre de la matrice laplacienne de $K_n$ est :

$$\{0,\, n,\, \ldots,\, n\}$$

**Graphe chaîne**

Les valeurs propres de la matrice laplacienne de $P_n$ pour $n \geq 3$, sont :

$$2 - 2\cos\left(\frac{\pi k}{n}\right),\;\; k \in \{0,\, 1, \ldots,\, n-1\}$$

# Spectre de la matrice laplacienne

**Graphe cycle**

Les valeurs propres de la matrice laplacienne de $C_n$ pour $n \geq 3$, sont :

$$2 - 2\cos\left(\frac{2\pi k}{n}\right), \; k \in \{0, 1, \ldots, n-1\}$$

Exemple :

Le spectre de $\mathbf{L}(C_3)$ est $\left\{2 - 2\cos(0),\, 2 - 2\cos\left(\frac{2\pi}{3}\right),\, 2 - 2\cos\left(\frac{4\pi}{3}\right)\right\} = \{0, 3, 3\}$

**Graphe étoile**

Le spectre de la matrice laplacienne de $S_n$ est :

$$\{0, 1, \ldots, 1, n\}$$

où la valeur propre 1 a multiplicité algébrique $n-2$ (c'est-à-dire, la valeur propre 1 est répétée $n-2$ fois)

# Spectre de la matrice laplacienne

## Graphe biparti complet

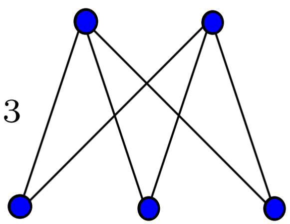
Le spectre de la matrice laplacienne du graphe $K_{m,n}$ est :

$$\{0,\, n, \ldots, n,\; m, \ldots, m,\, n+m\}$$

avec une multiplicité algébrique $1,\, m-1,\, n-1,\, 1,$ respectivement

Exemple :

$K_{2,3}$



Le spectre de $\mathbf{L}(K_{2,3})$ est $\{0,\, 3,\, 2,\, 2,\, 5\}$

## Graphe de Petersen

Le spectre de la matrice laplacienne du graphe de Petersen est $(n=10)$:

$$\{0,\, 2,\, 2,\, 2,\, 2,\, 2,\, 5,\, 5,\, 5,\, 5\}$$

## Graphe de Clebsch

Le spectre de la matrice laplacienne du graphe de Clebsch est $(n=16)$:

$$\{0,\, 4, \ldots, 4,\, 8, \ldots, 8\}$$

avec une multiplicité algébrique 1, 10, 5, respectivement

# Spectre de la matrice laplacienne

$\lambda_2(\mathbf{L}(\mathcal{G}))$ est la deuxième valeur propre la plus faible de la matrice laplacienne ($\lambda_1(\mathbf{L}(\mathcal{G}))$ est toujours zéro)

$\lambda_2(\mathbf{L}(\mathcal{G}))$ est appelée valeur de Fiedler (ou connexité algébrique) et le vecteur propre associé $\mathbf{u}_2$, vecteur de Fiedler du graph $\mathcal{G}$

Miroslav Fiedler
(1926 – 2015)

**Remarque :**

La valeur de Fiedler est importante non seulement comme **mesure de robustesse** ou **du niveau de connexité** d'un graphe, mais aussi pour les **propriétés de convergence** d'un *algorithme distribué* remarquable :

$\boxed{\textbf{Protocole de consensus}}$

# Toolbox de Matlab pour la théorie des graphes

**À partir de Matlab R2015b**

# Quelques commandes de la toolbox

## Classe `graph`

G = graph(A) uses the square symmetric matrix A as an adjacency matrix and constructs a weighted graph with edges corresponding to the nonzero entries of A.

G = graph(S,T) constructs a graph with edges specified by the node pairs (S,T).
S and T must have the same number of elements or be scalars.

G = graph(S,T,WEIGHTS) also specifies edge weights with the numeric array WEIGHTS.
WEIGHTS must have the same number of elements as S and T, or can be a scalar.

G = graph(S,T,...,'OmitSelfLoops') does not add self-loops to the graph

**graph properties** :

Edges - Table containing edge information
Nodes - Table containing node information

# Quelques commandes de la toolbox



## graph methods:

`numnodes` - Number of nodes in a graph
`numedges` - Number of edges in a graph

`addnode` - Add nodes to a graph
`rmnode` - Remove nodes from a graph
`addedge` - Add edges to a graph
`rmedge` - Remove edges from a graph

`degree` - Degree of nodes in a graph
`neighbors` - Neighbors of a node in a graph
`subgraph` - Extract an induced subgraph

`adjacency` - Adjacency matrix of a graph
`incidence` - Incidence matrix of a graph
`laplacian` - Graph Laplacian

# Quelques commandes de la toolbox

**graph methods:**

| | |
|---|---|
| `bfsearch` | - Breadth-first search (BFS) |
| `dfsearch` | - Depth-first search (DFS) |
| `shortestpath` | - Compute shortest path between two nodes |
| `shortestpathtree` | - Compute single source shortest paths |
| `distances` | - Compute all pairs distances |
| `nearest` | - Compute nearest neighbors (at a distance D) of a node |
| `conncomp` | - Compute connected components of a graph |
| `minspantree` | - Compute minimum spanning tree of a graph |
| `isisomorphic` | - Determine whether two graphs are isomorphic |
| `plot` | - Plot an undirected graph |

- De la même façon pour les **graphes orientés**, il existe la classe **`digraph`**