



## Licence « Sciences pour l'ingénieur »

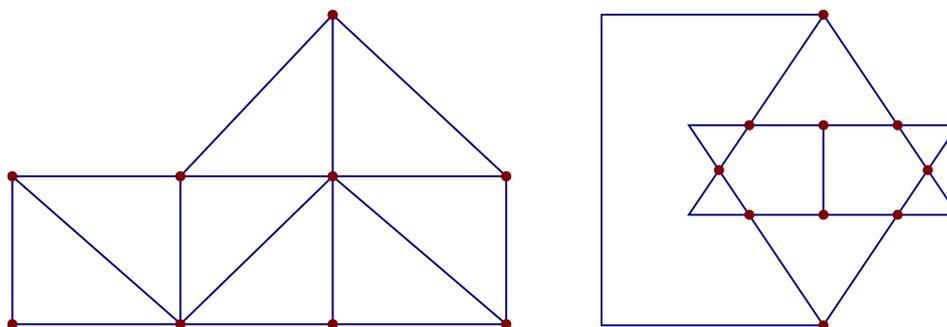
### DS 2.3 - Énergétique et circuits électriques

#### TD 4 : Méthodes d'analyse des circuits

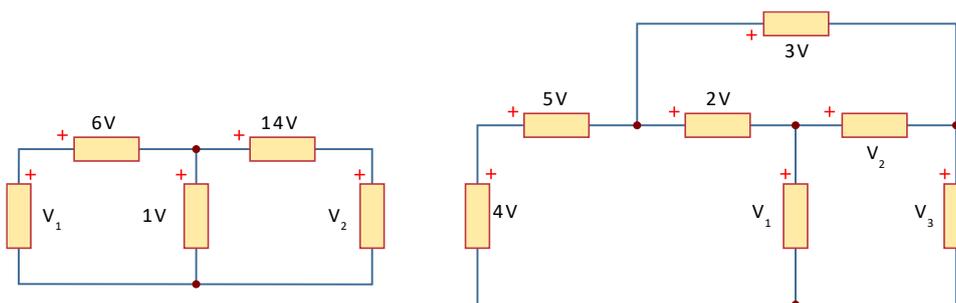
Franco FERRUCCI

[franco.ferrucci@upf.pf](mailto:franco.ferrucci@upf.pf)

**Exercice 4.1** (inspiré des ex. 2.5 et 2.6, réf. [1]). Pour les graphes de réseau dans la figure ci-dessous, déterminer le nombre de nœuds, de branches et de mailles indépendantes :

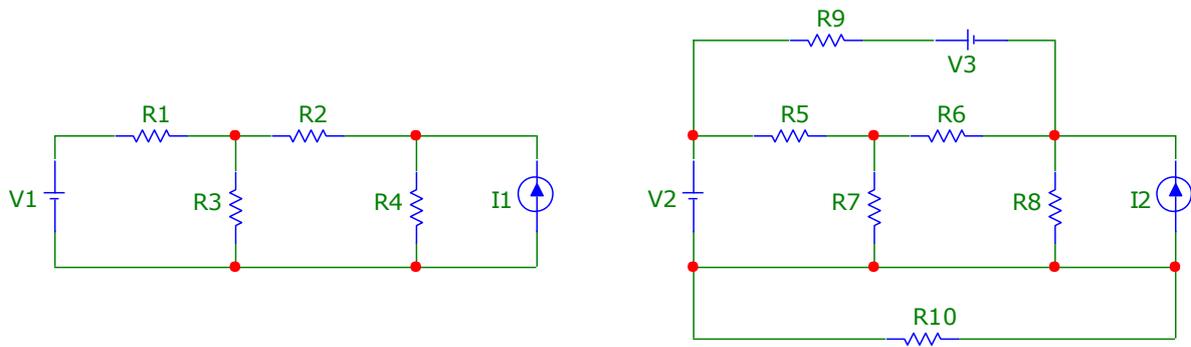


**Exercice 4.2** (inspiré des ex. 2.11 et 2.12, réf. [1]). **Application de la LKT.** Calculer les tensions inconnues dans les circuits suivants.



**Exercice 4.2.** À partir des circuits ci-dessous :

- Déterminer le nombre de branches et de nœuds ( $b$  et  $N$ ).
- En utilisant le théorème fondamental de la topologie, calculer le nombre de boucles indépendantes ( $L$ ), à l'aide de la formule  $b = N + L - 1$ .
- Indiquer le nombre d'inconnues.
- Déterminer le nombre d'équations indépendantes issues de la LKC ( $\sum i_k = 0$ ) et de la LKT ( $\sum v_k = 0$ ), nécessaires pour obtenir un système d'équations résoluble.



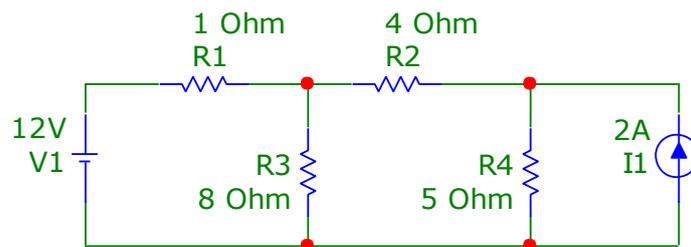
**Exercice 4.3** (Inspiré de l'ex. 2.7, réf. [1]).

Plantez le système d'équations (sans le résoudre) à l'aide de la méthode par sélection d'équations indépendantes de Kirchhoff (méthode dite « **force brute améliorée** » dans notre cours).

Ensuite, résolvez numériquement le circuit en utilisant cette fois la **méthode des nœuds**, en calculant les tensions et courants de tous les composants.

**Remarque** : n'oubliez pas de choisir à l'avance le sens des courants dans les branches, ce qui fixe par conséquent le signe « + » sur les composants, en suivant :

- la convention du dipôle générateur pour les sources,
- la convention du dipôle passif pour les résistances.

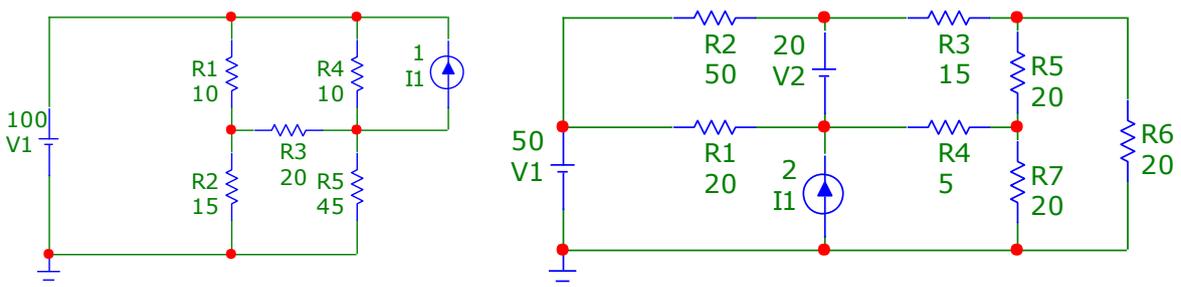


**Exercice 4.4.** Résoudre les circuits en utilisant la **méthode de nœuds**. Puis, à partir des tensions de nœuds obtenues, calculer :

- Pour le circuit à gauche : la tension aux bornes de la source de courant
- Pour le circuit à droite : la tension aux bornes de la résistance R5.

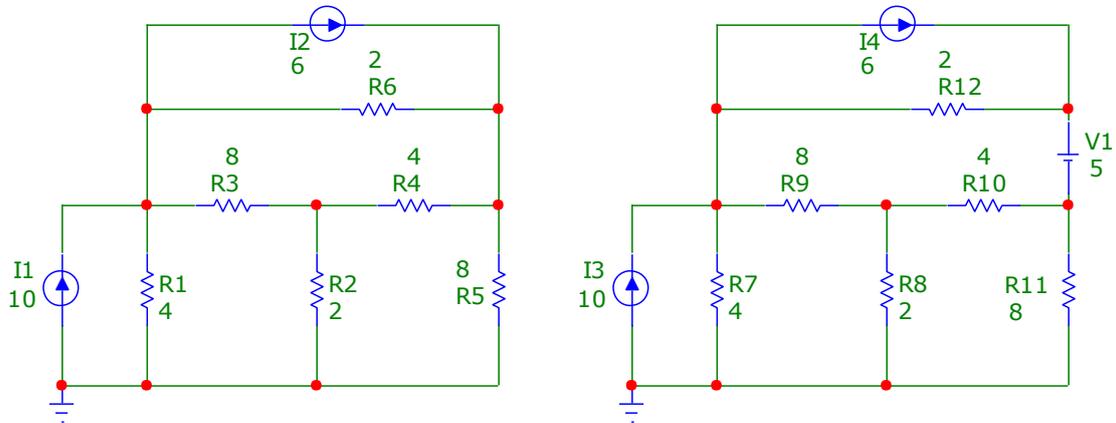
Pour calculer le vecteur de solution  $\mathbf{x}$  de l'équation matricielle  $\mathbf{A}\cdot\mathbf{x}=\mathbf{b}$ , vous pouvez utiliser le code Python disponible à la fin de la section. Enfin, utilisez Microcap pour vérifier vos résultats

**Remarque** : ces circuits ont été créé avec Microcap. Comme dans la plupart des logiciels de type SPICE, les valeurs des composants sont indiquées sans unités dans le schéma. Par exemple, pour le circuit à gauche, la source de tension est de 100 V, et la résistance R5 vaut 45  $\Omega$ .



**Exercice 4.4.** Résoudre les circuits en utilisant la **méthode de nœuds**. Pour calculer le vecteur de solution  $\mathbf{x}$  de l'équation matricielle  $\mathbf{A}\cdot\mathbf{x}=\mathbf{b}$ , vous pouvez utiliser le code Python disponible à la fin de la section. Enfin, utilisez Microcap pour vérifier vos résultats.

**Remarque :** ces circuits ont été créé avec Microcap, donc, les valeurs des composants sont indiquées sans unités dans le schéma.



**Extra extra !** Ci-dessous, vous trouverez un code Python qui permet de résoudre un système du type  $\mathbf{A}\cdot\mathbf{x}=\mathbf{b}$ , obtenu à partir de l'application de la méthode des nœuds. Modifiez la matrice  $\mathbf{A}$  et le vecteur  $\mathbf{b}$  selon les équations que vous avez établies pour le circuit donné, afin d'obtenir les tensions aux nœuds ! Vous pouvez exécuter ce code de différentes manières :

- **Google Colab** – en ligne, (nécessite un compte Google) : <https://colab.research.google.com/>
- **PythonAnywhere** – simple, sans inscription : <https://www.pythonanywhere.com/try-ipython/>
- Sur votre ordinateur – vous pouvez exécuter ce code directement dans **Spyder**.

```
import numpy as np

# -----
# Méthode des nœuds : résolution de A·x = b
# -----

# L'étudiant doit définir ici :
# - la matrice A (matrice des coefficients)
# - le vecteur b (second membre)

# Décommenter et remplir avec les valeurs du circuit
# A = np.array([
#     [val11, val12, ...],
#     [val21, val22, ...],
#     ...
# ])
# b = np.array([val1, val2, ...])

# Résolution du système linéaire
try:
    x = np.linalg.solve(A, b)

    # Affichage des résultats avec 4 chiffres après la virgule
    print("Tensions des nœuds (en Volts) :")
    for i, tension in enumerate(x, start=1):
        print(f"v{i} = {tension:.4f} V")

except np.linalg.LinAlgError:
    print("X Erreur : la matrice A est singulière ou le système est mal défini.")
```

**Exercices complémentaires.** Afin de renforcer votre compréhension des concepts étudiés, je vous suggère de travailler les exercices suivants, extraits de « Analyse des circuits électriques » Charles K Alexander - Matthew NO Sadiku (disponible à la Bibliothèque Universitaire).

3.2	3.5	3.18
3.3	3.11	3.19
3.4	3.15	3.25