

Travaux Tutorés Génie Logiciel

TT à rendre pour le **Vendredi 8 avril 2022 18h00**

L'objectif de ce TT est de se **familiariser à la modélisation objet au travers la construction de classes** et de manipuler les conteneurs de la STL comme `list`, `vector`, `map`, ... , et de mettre en oeuvre le **polymorphisme** en C++.

Il est réalisé en équipe de 4 élèves ayant chacun une responsabilité de management dans le projet :

1. Un responsable de la gestion de la répartition de l'écriture du code et de son déroulé.
2. Un responsable de l'élaboration des tests unitaires de validation du/des codes produits et du respect du principe d'encapsulation.
3. Un responsable de la documentation du code (<https://www.doxygen.nl/manual/docblocks.html> avec des balises doxygen) et de l'élaboration et du suivi d'une chartre d'écriture.
4. Un responsable du rapport latex et de la gestion des commits sur la forge.

L'élève responsable d'une activité recevra un bonus de 25% de la note obtenue dans cette activité.

Une revue de mi-étape sera réalisée lors des TT du lundi 21 Mars 2022.

Le rendu consiste en les codes développés (avec un makefile permettant de compiler et de lancer l'exécutable en dehors de tout IDE) et le rapport décrivant les choix d'implémentation (avec des insertions de listing) ainsi qu'une partie rédigée par chacun des responsables illustrant comment il a mis en oeuvre sa responsabilité auprès de ces partenaires de projet tout au long de ce dernier.

1 Structuration des données pour le réseau électrique

On souhaite disposer d'un outil de simulation numérique permettant de simuler les transitoires électromagnétiques dans les réseaux électriques. Pour cela nous modélisons ce réseau comme un circuit constitué d'une **liste de composants** électriques élémentaires comme des sources de courants I_s , des sources de tension V_s , des résistances R , des capacités C , des inductances L ,...

Chaque composant peut être vu de manière générique comme un objet de type **nports** ayant des caractéristiques comme son type (ou fonctionnalité) de type **string** qui dépend d'un nombre fini de paramètres de type **complexe**, et qui comprend un nombre fini de ports ou noeuds de type **entier** permettant d'insérer ce composant dans le réseau.

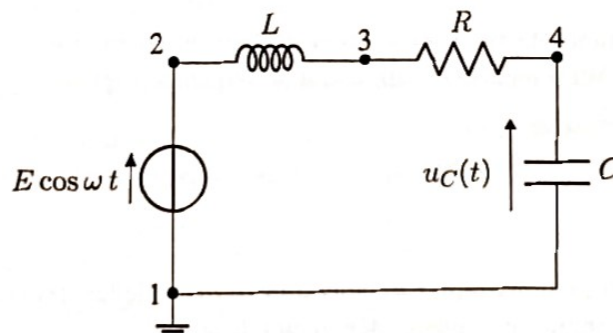
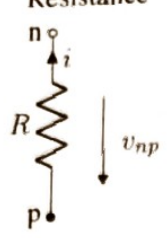
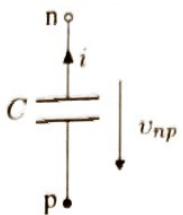
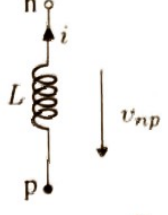
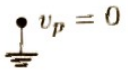
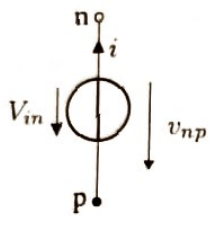
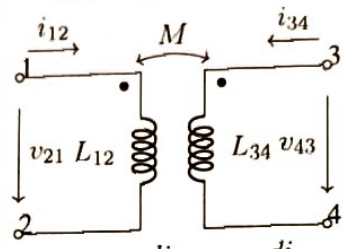
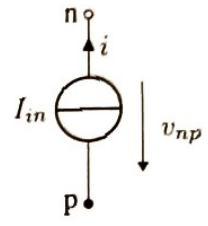
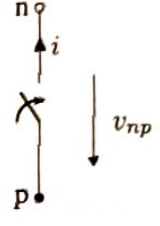


FIGURE 1 – Circuit RLC

1. La première étape de ce projet consiste à construire l'environnement permettant de créer un circuit électrique à partir d'un fichier de donnée de type SPICE dont une ligne générique est :

```
typecomposant noeud0 noeud1 ... noeud m-1 parameter0 ... parameter p-1
```

- (a) Ecrire les classes sous forme canonique permettant de lire, de stocker, d'ajouter un composant, d'enlever un composant, et d'afficher un circuit électrique
- (b) Le circuit devra pouvoir contenir les composants suivants :

Résistance  $v_{np} = Ri$	Condensateur  $C \frac{dv_{np}}{dt} = i$	Bobine d'inductance  $v_{np} = L \frac{di}{dt}$	mise à la terre  $v_p = 0$
Source de voltage  $v_{np} = V_{in}$	Inductance mutuelle  $v_{21} = L_{12} \frac{di_{12}}{dt} + M \frac{di_{34}}{dt}$ $v_{43} = M \frac{di_{12}}{dt} + L_{34} \frac{di_{34}}{dt}$	source de courant  $i = I_{in}$	switch  $i = 0$ si ouvert

2 Simulation numérique du réseau électrique

2.1 Les lois de Kirshoff

D'un point de vue physique les équations du réseau électrique sont régies par les lois de Kirshoff :

1. La loi des mailles (ensembles de dipôles connectés bout en bout et constituant un circuit fermé) : La somme algébrique des tensions [ou différences de potentiel (d.d.p.)] le long d'une maille est nulle.
2. la loi des noeuds : La somme algébrique des courants (comptés positivement (resp. négativement) quand ils arrivent (resp. sortent) à un noeud est nulle.

2.2 Analyse nodale augmentée AC du réseau électrique en modélisation phaseur

Dans l'analyse en courant alternatif les sources de tension ou les sources de courant oscillent avec une fréquence f (généralement 50Hz ou 60Hz) définissant la fréquence angulaire $\omega = 2\pi f$. On a une représentation dite phaseur avec $j^2 = -1$:

$$i(t) = \bar{i}e^{j\omega t}, \bar{i} \in \mathbb{C},$$

$$v(t) = \bar{v}e^{j\omega t}, \bar{v} \in \mathbb{C}.$$

Cette représentation permet de transformer les équations différentielles des composants en des relations algébriques dans le corps des nombres complexes dépendant de ω :

$$v_p(t) - v_n(t) = Ri(t) \Rightarrow (\bar{v}_p - \bar{v}_n)e^{j\omega t} = \bar{i}e^{j\omega t} \Rightarrow \bar{i} = \bar{v}_p - \bar{v}_n$$

$$v_p(t) - v_n(t) = L \frac{di(t)}{dt} \Rightarrow (\bar{v}_p - \bar{v}_n)e^{j\omega t} = Lj\omega \bar{i}e^{j\omega t} \Rightarrow \bar{i} = \frac{1}{Lj\omega}(\bar{v}_p - \bar{v}_n)$$

$$i(t) = C \frac{d}{dt}(v_p(t) - v_n(t)) \Rightarrow \bar{i}e^{j\omega t} = Cj\omega(\bar{v}_p - \bar{v}_n)e^{j\omega t} \Rightarrow \bar{i} = Cj\omega(\bar{v}_p - \bar{v}_n)$$

En écrivant les équations en modélisation phaseur de chaque composant et en utilisant la loi des noeuds de Kirshoff pour le réseau RLC de la figure 1, on obtient :

$$\begin{aligned}
\bar{v}_1 &= 0 & (1) \\
\bar{v}_2 - \bar{v}_1 + Z_s \bar{i}_{12} &= E & (2) \\
\bar{v}_3 - \bar{v}_2 - Lj\omega \bar{i}_{23} &= 0 & (3) \\
\bar{v}_4 - \bar{v}_3 - R \bar{i}_{34} &= 0 & (4) \\
Cj\omega(\bar{v}_1 - \bar{v}_4) - \bar{i}_{41} &= 0 & (5)
\end{aligned}
\qquad
\begin{aligned}
\bar{i}_{12} - \bar{i}_{23} &= 0 & (6) \\
\bar{i}_{23} - \bar{i}_{34} &= 0 & (7) \\
\bar{i}_{34} - \bar{i}_{41} &= 0 & (8) \\
\bar{i}_{41} - \bar{i}_{12} &= 0 & (8)
\end{aligned}$$

On obtient ainsi la matrice d'admittance \mathbb{Y} complexe reliant les phaseurs courants et potentiels :

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & Z_s & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & -Lj\omega & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & -R & 0 \\ Cj\omega & 0 & 0 & -Cj\omega & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}}_{\mathbb{Y}} \underbrace{\begin{pmatrix} \bar{v}_1 \\ \bar{v}_2 \\ \bar{v}_3 \\ \bar{v}_4 \\ \bar{i}_{12} \\ \bar{i}_{23} \\ \bar{i}_{34} \\ \bar{i}_{41} \end{pmatrix}}_x = \underbrace{\begin{pmatrix} 0 \\ E \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_b \quad (9)$$

Remarques :

1. On a approximé par le complexe $Ee^{j\omega t}$ la source de voltage pour représenter la partie réelle $E \cos(\omega t)$
2. Les voltages et les courants seront les $Re(\bar{v}e^{j\omega t})$ et $Re(\bar{i}e^{j\omega t})$

3 Cahier des charges des fonctionnalités

On veillera à implémenter à minima les fonctionnalités suivantes :

1. Ajouter un attribut **system** à la classe **circuit** correspondant à la matrice d'admittance complexe \mathbb{Y} . Cette matrice étant particulièrement creuse, on la stockera dans un conteneur **map<ij, complexe<double>>** où la clé **ij** sera une classe contenant l'indice de ligne et l'indice de colonne et possédant une relation d'ordre.
2. Développer une méthode **circuit : :buildSystem(double omega)** qui construit la matrice \mathbb{Y} par l'analyse nodale modifiée augmentée (MANA).
3. Ajouter un attribut **rhs** de type **vector<complexe<double>** à la classe **circuit** correspondant au second membre de $\mathbb{Y}\mathbb{V} = \mathbb{I}$ et une méthode **circuit : :setRhs()** qui définit ce second membre par la MANA.
4. Ajouter une méthode **circuit : :solveSystem()** qui résout le problème $\mathbb{Y}x = b$ sur des matrices stockées **map<ij, complexe<double>>**
5. Proposer des **tests unitaires** permettant de valider chacune des méthodes.
6. Proposer un **cas d'application** contenant des inductances, des résistances, des capacités et des sources de Voltage qui soit **conséquent en terme de noeuds et de mailles**.

On essaiera d'implémenter la fonctionnalités suivante (question ouverte) :

bonus Connaissant un circuit proposer et implémenter une méthode permettant de donner l'ensemble des mailles d'un circuit.

4 Résultats attendus

Le TT devra être git-er fréquemment dans le projet forge associé. Vous ne devez giter que les fichiers nécessaires (.h, .cpp, fichiers de données, makefile, rapport latex) (pas les .o ni les executables ni les fichiers de compilation de latex). Les commentaires des push doivent être explicites sur l'apport du push. Le rapport latex veillera à être écrit dans un français syntaxiquement et grammaticalement correct :

1. Expliquer les notions de C++ que vous utilisez en détaillant certaines parties de votre code. (L'exercice consiste à me montrer que vous avez compris et mis en oeuvre certaines notions du C++ (utiliser le package listings, le package circuitikz permet de faire des circuits en latex).
2. Contenir un main qui vérifie que le code est bien implémenté en proposant des tests unitaires de chacune des méthodes demandées.