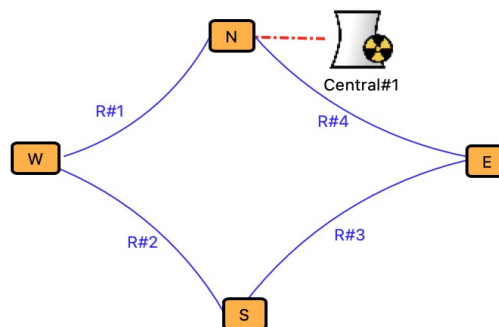


Extension 1 : réseau de distribution électrique

Dans cette extension, vous prendrez en compte la localisation géographique des points d'injection et des points de livraison, et les pertes d'énergie sur les lignes électriques qui les relient.

Principes

- Chaque point d'injection et point de livraison est localisé sur un plan (dimension 2) ; on utilisera un système de coordonnées cartésiennes.
- Pour simplifier le réseau électrique, on considère une « clusterisation » des points d'injection et point de livraison ; dans un cluster chaque point possède les mêmes coordonnées ; pour illustrer ce principe, un cluster peut être une ville ou un village.
- Le réseau électrique est un graphe dont les nœuds sont les clusters et les arcs des routes d'approvisionnement d'énergie. Voici un exemple très simple avec 4 clusters et 4 routes ; sur ce schéma, j'ai également placé un producteur dans le cluster N.



- Chaque arc a un poids qui est égal à la distance euclidienne entre les deux clusters reliés par cet arc.
- Tous les consommateurs d'un même cluster sont alimentés par un seul et même producteur. Pour simplifier le sujet, on ne considère que les producteurs de masse (les centrales).
- Chemin d'approvisionnement : quand un cluster est alimenté par un producteur qui n'est pas associé à ce cluster, on définit un *chemin d'approvisionnement*. Ce chemin est constant pour cette extension. Sur l'exemple ci-dessus, pour alimenter S, deux chemins sont possibles : je décide a priori et pour toute la simulation d'un chemin unique, par exemple R#1 + R#2.
- Le poids d'un chemin est la somme du poids des arcs qui le composent.
- On ne traitera que des réseaux connexes (il existe au moins un chemin possible entre 2 clusters quelconques).
- Le calcul des pertes est limité aux pertes Joules que l'on considère simplement proportionnelles à la distance entre producteur et consommateur. Deux cas possibles :
 - Soit le producteur et le consommateur sont dans le même cluster : dans ce cas, la perte est nulle.

- Soit le producteur et le consommateur sont dans 2 clusters distincts ; dans ce cas, la perte est proportionnelle à la distance entre les deux clusters, c'est à dire proportionnelle au poids du chemin d'approvisionnement entre les deux clusters.

Compléments demandés

- Compléter la conception afin d'introduire les classes et les relations utiles.
- Compléter le modèle de calcul afin de prendre en compte les pertes liées au réseau.
- L'affichage CSV quotidien ou annuel demandé pour le projet individuel devra être complété avec une colonne supplémentaire indiquant l'énergie perdue sur l'ensemble des lignes utilisées.
- Un affichage CSV quotidien ou annuel spécifique à un cluster donné devra aussi être fourni : dans ce cas, la colonne Pertes comprendra les pertes accumulées sur toutes les lignes sortant du cluster.
- Une notice de quelques lignes décrivant précisément votre format de sortie.

Extension 2 : paramétrage de la simulation par fichier

Dans cette extension, vous configurerez votre simulation en lisant les paramètres donnés dans un ou plusieurs fichiers de configuration au format CSV. Ce ou ces fichiers décriront :

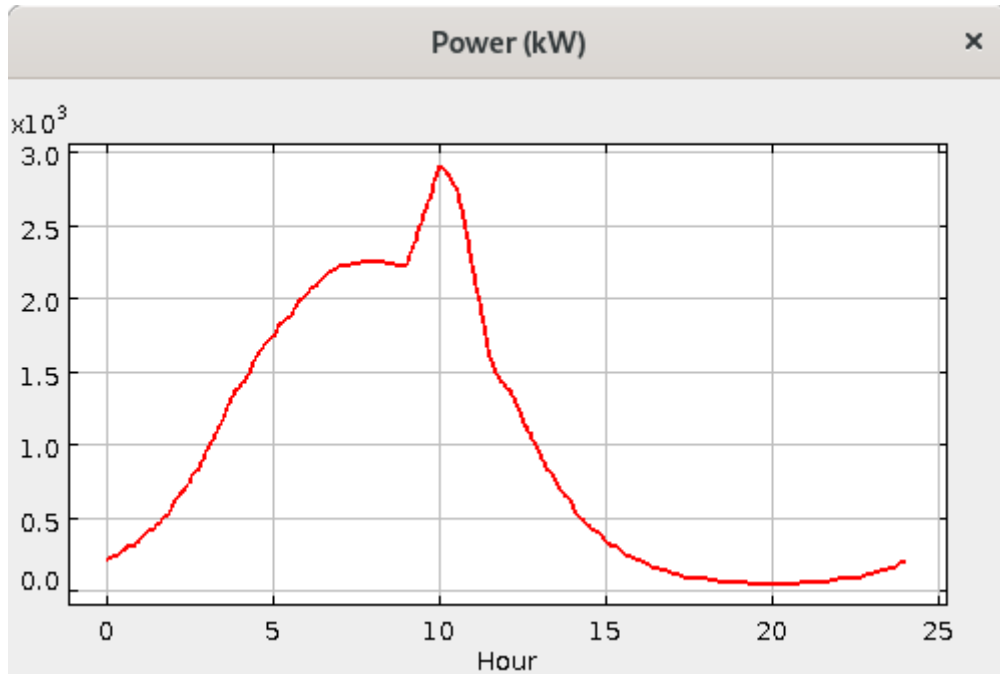
- Les producteurs d'énergie, y compris les paramètres permettant de paramétrer les modèles de production utilisés
- Les consommateurs d'énergie, y compris les paramètres permettant de paramétrer les modèles de consommation utilisés
- Les informations utiles à l'extension 1 : clusters, lignes, coefficients de pertes Joule sur les lignes, etc.

Compléments demandés

- Décrire le format de chacun de vos fichiers de configuration dans une notice complète comportant des exemples.
- Compléter la conception afin d'introduire les classes et les relations utiles. Vous pourrez vous inspirer du PersonReader de la séance M6, et choisir soigneusement les collections standard Java pour stocker les différents objets créés à la lecture du fichier de configuration.
- Écrire les fichiers de configuration nécessaires à la création du scénario d'intégration de projet individuel, ou à la création d'un nouveau scénario d'intégration exploitant l'extension 1.

Extension 3 : interface utilisateur graphique Swing

Cette extension consiste à ajouter une interface utilisateur graphique qui permette de sélectionner des producteurs ou consommateurs dont on souhaite visualiser la courbe de production ou consommation électrique. Les courbes de production ou consommation seront affichées au moyen de la bibliothèque Ptpplot déjà utilisée dans les activités du cours.



Exemple de graphique sur une journée.

NB : vous pouvez mettre plusieurs courbes sur un même graphique.

Pour la sélection des consommateurs ou producteurs, nous proposons de rendre visible les différentes agrégations qui ont pu être faites lors de la construction des consommateurs, par exemple en construisant un foyer à partir de plusieurs appareils électriques, ou un quartier/village/ville à partir de plusieurs foyers. De cette façon, les différents niveaux de granularité des consommateurs pourront être visualisés sous une forme hiérarchique, et ainsi la sélection d'un consommateur pourra être faite à la souris, que le consommateur soit un appareil électrique élémentaire ou une ville entière.



Le widget JTree de la bibliothèque Swing permet d'afficher un tel contenu hiérarchique ; vous concevrez donc une interface utilisateur Swing pour sélectionner dynamiquement le(s) producteur(s) ou consommateur(s) dont on souhaite visualiser la courbe de puissance électrique.

Alternativement, vous pouvez sélectionner plus « grossièrement » les producteurs/consommateurs suivant les catégories (classes) que vous aurez définies dans le simulateur : centrales photovoltaïques/éoliennes/hydrauliques/etc, appareils électriques... Pour cela, intégrez dans votre interface graphique des boutons à cocher (check boxes) correspondant à ces catégories. À chaque fois que l'utilisateur aura terminé de cocher sa sélection, un graphique Ptpplot sera présenté (la validation de la sélection pourra faire appel à un bouton supplémentaire).

Vous pouvez aussi combiner ces deux outils de sélection ou en proposer d'autres.

Pour l'intégration des quatre extensions, l'interface graphique devra aussi permettre d'exploiter les données ou traitements des autres extensions : charger un fichier de configuration, lancer l'optimisation, etc.

Compléments demandés

- Compléter la conception afin d'introduire les classes et les relations utiles. Vous utiliserez le patron de conception « Modèle-Vue-Contrôleur » (MVC) en vous inspirant de la documentation ci-dessous.
- Développer l'interface graphique décrite ci-dessus, d'abord sans sélection de producteurs/consommateurs, puis avec sélection.

Documentation

Voici quelques références pour construire une interface graphique avec Swing :

- La classe RNGPlotter que vous avez utilisée pendant M4 et M5
- Le design pattern MVC (<https://en.wikipedia.org/wiki/Model-view-controller>)
- Une archive SwingExamples.zip sera mise à disposition sur le LMS : elle contient quelques exemples de code utilisant les composants graphiques listés ci-après, en particulier des éléments utilisant le design pattern MVC.
- [The Java Tutorial, Trail: Creating a GUI With JFC/Swing](#), notamment :
 - How to use labels
 - How to use buttons, check boxes and radio buttons
 - How to use HTML in Swing components
 - Introduction to event listeners
 - How to write an action listener (ce qui personnalise la réaction au clic sur un bouton)
 - How to use panels
 - How to use tabbed panes
 - How to use scroll panes
 - How to use trees

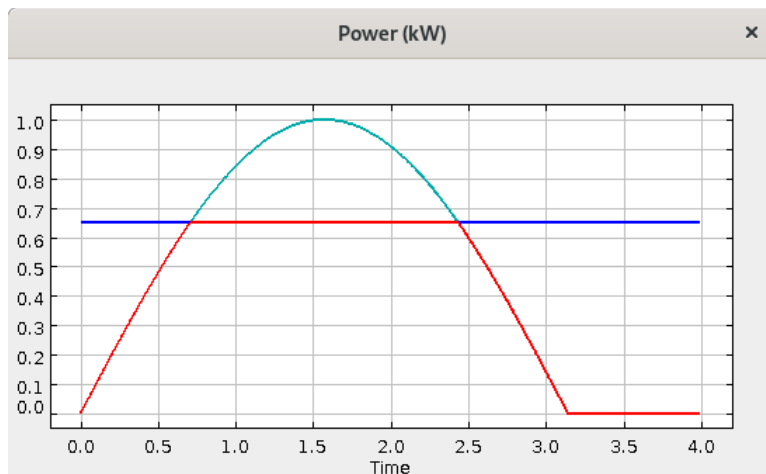
- How to use GridBagLayout
- How to use file choosers
- Les classes de l'API Java 8 : JTabbedPane, JPanel, JLabel, JButton, JCheckBox, JTree, GridBagLayout, JFileChooser, etc.
- Les classes de l'API Ptpplot : Plot, etc.

Extension 4 : construction de modèles

Dans cette extension, on cherche à fournir des outils pour construire des modèles de production/consommation riches et complexes. Des applications possibles seraient par exemple de pouvoir construire un modèle de puissance de panneaux photovoltaïques en combinant un modèle des spécifications constructeurs du panneau solaire avec un modèle d'ensoleillement et un modèle de couverture nuageuse, ou un modèle de puissance d'éolienne à partir d'une courbe de puissance nominale de l'éolienne et d'un modèle de vent.

On commencera avec des modèles élémentaires de production ou de consommation, comme des constantes, des fonctions affines, des sinusoides, des « créneaux » positifs ou négatifs, du bruit blanc gaussien... Ces modèles doivent être paramétrables (valeur de la constante, fréquence de la sinusoïde, etc.).

Pour combiner ces modèles, on souhaite disposer d'opérateurs simples qui permettent d'obtenir de nouveaux modèles à partir de modèles existants, comme la somme de deux puissances électriques ou l'écrêtage d'une production par une fonction, soit respectivement $P_1(t) + P_2(t)$ et $\min(P_1(t), f(t))$.



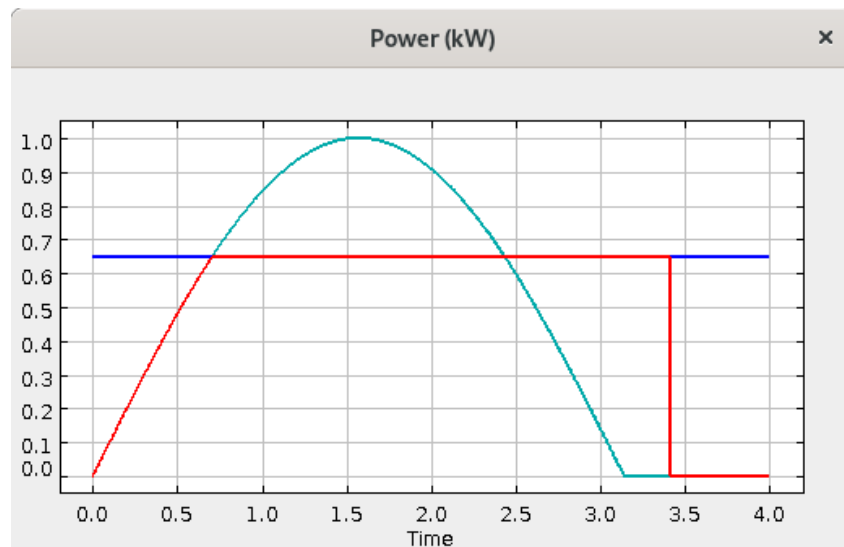
Exemple d'écrêtage par une fonction constante (courbe résultante en rouge)

De même, il peut être utile de s'intéresser à une consommation minimum (consommation de veille d'un appareil par exemple).

On peut aussi s'intéresser à la composition de fonctions ; dans ce cas il faut être vigilant sur les domaines et co-domaines de ces fonctions. Par exemple, la courbe de puissance d'une éolienne est généralement donnée en fonction de la vitesse du vent, il faudra donc pouvoir composer cette fonction avec celle d'un modèle qui donnerait la vitesse du vent au cours du temps. Autre exemple plus simple : la composition d'une fonction de retard (latence) avec les modèles élémentaires de puissance permet de modéliser une inertie dans les modèles, $P(t) = P_1(\text{delay}(t))$.

On peut également s'intéresser à des fonctions plus élaborées, qui permettraient par exemple d'éviter un écrêtage en adjoignant un accumulateur (batterie) à une centrale de production : l'énergie qui aurait été écrêtée est alors emmagasinée dans un accumulateur (de capacité finie !) tant que la production est supérieure à la fonction d'écrêtage, puis restituée quand la production est inférieure. On peut ainsi parler de mise en forme d'un modèle par un autre ou *shaping*, et on notera

donc que certains modèles peuvent être des modèles à mémoire (ici, la quantité d'énergie qui a été stockée dans l'accumulateur).



Exemple de shaping d'une sinusoïde supérieure (en vert) par une fonction constante (en bleu).
La surface écrêtée en haut de la sinusoïde est reportée à droite (courbe rouge).

Compléments demandés

- Complétez la conception afin d'introduire les classes et les relations utiles pour représenter des modèles de production ou de consommation élémentaires, ou obtenus par les opérations définies permettant de combiner des modèles élémentaires.
- Rédigez une documentation concise mais complète de ces modèles et opérateurs.
- Fournissez un scénario unitaire pour chacun des modèles de productions/consommations élémentaires, et un scénario d'intégration pour une combinaison suffisamment riche de ces modèles. Les sorties pourront se faire au format CSV dans la console comme pour le projet individuel.
- Une illustration graphique de ces scénarios (unitaires + intégration), soit sous forme de fichiers PNG obtenus par exploitation des sorties CSV dans un tableur, soit directement dans l'IHM de l'extension 3.