



**HAL**  
open science

# Modélisation et simulation de systèmes à structure dynamique basés sur des équations différentielles avec DEVS statique

Aya Attia, Clément Foucher, Luiz Fernando

## ► To cite this version:

Aya Attia, Clément Foucher, Luiz Fernando. Modélisation et simulation de systèmes à structure dynamique basés sur des équations différentielles avec DEVS statique. Journées Francophones de la Modélisation et de la Simulation, Nov 2024, Cargese (Corse), France. hal-04769752

**HAL Id: hal-04769752**

**<https://laas.hal.science/hal-04769752v1>**

Submitted on 6 Nov 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Modélisation et simulation de systèmes à structure dynamique basés sur des équations différentielles avec DEVS statique

Aya Attia  
aya.attia@laas.fr

Clément Foucher  
clement.foucher@laas.fr

Luiz Fernando Lavado Villa  
lflavado@laas.fr

LAAS-CNRS, Université de Toulouse, UPS, Toulouse, France

## Résumé

*La modélisation des systèmes à structure dynamique est un sujet de débat dans de nombreux domaines en raison de sa complexité et la difficulté de la représentation des changements structurels. Pour répondre à ce défi, nous proposons une approche intégrée qui combine les formalismes DEVS et DTSS pour simuler des systèmes basés sur des équations différentielles et capable de représenter les changements structurels de manière efficace et précise. Notre approche est conçue de manière générique pour garantir son adaptabilité à différents domaines. Afin de tester son efficacité, nous l'avons appliquée à un cas concret, celui de l'analyse de l'équilibre des charges dans un réseau électrique.*

**Mots-clés :** Modélisation et simulation, Systèmes à structure dynamique, DEVS, DTSS, Calcul symbolique

## Abstract

*Modeling dynamic structure systems is a subject of discussion across diverse domains due to its complexity and the challenges of accurately representing structural changes. For this goal, we introduce an integrated approach that combines DEVS and DTSS formalisms in order to simulate systems based on differential equations and give an accurate representation of structural changes. Our approach is designed to be generic in order to ensure its adaptability to various domains. To validate this approach, we applied it to a specific use case: load flow analysis for power systems.*

**Keywords:** Modeling and simulation, Dynamic structure systems, DEVS, DTSS, Symbolic computation

## 1 Introduction

La modélisation et la simulation des systèmes complexes représentent un défi majeur dans di-

vers domaines. Ces systèmes présentent souvent des comportements dynamiques difficiles à décrire formellement. Pour relever ce défi, les chercheurs et les ingénieurs se tournent de plus en plus vers des outils de modélisation et de simulation avancés, notamment les formalismes et simulateurs basés sur les approches DEVS (Discrete Event System Specification), DTSS (Discrete Time System Specification) et DESS (Differential Equation System Specification).

Dans cet article, nous proposons une méthodologie de modélisation basée sur certains de ces formalismes pour représenter les évolutions de systèmes à structure dynamique. Un système à structure dynamique est un système dans lequel la structure et la composition varient au fil du temps. En combinant les approches DTSS et DEVS, nous sommes en mesure de saisir à la fois les aspects discrets et continus de systèmes représentés par des équation différentielles, ce qui offre une méthodologie efficace pour leur modélisation et leur simulation. Enfin, nous mettons en œuvre cette méthodologie pour analyser l'équilibre de puissance dans un réseau électrique, démontrant son utilité dans des applications concrètes.

## 2 Formalismes de modélisation et simulateurs

Dans cette section, nous introduisons les formalismes de base de la théorie de la modélisation et de la simulation ainsi que les simulateurs correspondants.

### 2.1 Formalismes

Les formalismes DESS, DTSS et DEVS sont des outils puissants utilisés pour modéliser et simuler une grande variété de systèmes complexes [1]. Chacun de ces formalismes offre des avantages et des fonctionnalités uniques, ce qui

les rend adaptés aux différents types de systèmes et facilite les tâches de modélisation.

Le formalisme DESS est utilisé pour modéliser les systèmes dynamiques continus, où les évolutions des variables d'état sont décrites par des équations différentielles. Une variable d'état dans DESS est décrite par l'équation 1 :

$$\frac{dq}{dt} = a \times q + b \times x \quad (1)$$

Les modèles DESS [2] sont particulièrement adaptés pour représenter des phénomènes physiques, chimiques ou biologiques dans lesquels les changements se produisent de manière continue dans le temps. Ils permettent une modélisation détaillée et précise des dynamiques du système à travers des équations différentielles. Toutefois, en pratique, la simulation de systèmes à temps continu n'est pas faisable. C'est pourquoi il est souvent nécessaire de discrétiser le temps pour accomplir la modélisation de manière efficace. C'est précisément là que le DTSS trouve son utilité.

Le formalisme DTSS divise le temps en intervalles discrets et modélise les changements de l'état du système à chaque intervalle de temps. Une variable d'état dans DTSS est décrite par l'équation 2 :

$$q(t + 1) = a \times q(t) + b \times x(t) \quad (2)$$

DTSS est adapté pour modéliser des systèmes à évolution continue, mais dont les états sont observés à des intervalles de temps discrets. Ces modèles permettent une simulation précise des systèmes à temps discret, en considérant les changements temporels à chaque pas de simulation. En revanche, l'utilisation de DTSS engendre parfois la simulation de plusieurs pas de temps qui ne correspondent pas nécessairement à des changements significatifs du système, ce qui peut conduire à une surcharge calculatoire. C'est là qu'intervient DEVS, offrant une approche efficace pour gérer les événements discrets et les transitions d'état.

Le formalisme DEVS se concentre sur la modélisation des systèmes à événements discrets. Dans cette approche, la représentation d'un système continu passe par la quantisation de l'état. Dans un tel système, un événement discret symbolise donc un changement de l'état dont la valeur dépasse celle du quantum. Il existe deux catégories de modèles DEVS [3] :

**Modèles atomiques :** ces modèles incarnent des composants qui représentent les comportements individuels. En réaction aux événements entrants, ils produisent des sorties et mettent à jour leur état interne.

**Modèles couplés :** ces modèles décrivent la manière dont les composants sont connectés et interagissent entre eux. Ils spécifient les règles de couplage et de synchronisation entre ces composants, ce qui permet de représenter la structure et le comportement global du système.

La figure 1 montre l'évolution d'un système avec les différents formalismes discutés.

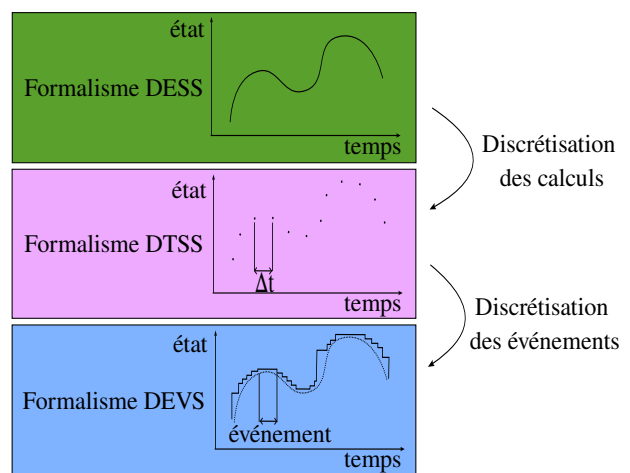


FIGURE 1 – Evolution d'un système avec DESS, DTSS et DEVS

DEVS ne prend pas en charge les événements parallèles, ce qui limite sa capacité à représenter des systèmes complexes distribués. Pour surmonter cette limitation, DEVS parallèle (PDEVS) [4] a été développé en tant qu'extension de DEVS, permettant ainsi la modélisation et la simulation de systèmes à événements discrets avec une structure parallèle et distribuée.

## 2.2 Simulateurs

Chaque formalisme décrit précédemment est associé à une méthode de simulation.

Une méthode de simulation, ou un simulateur, est un outil informatique qui permet d'exécuter des modèles de systèmes et de produire des résultats simulés. Il permet de reproduire le comportement d'un système réel dans un environnement virtuel, en utilisant des modèles, des données d'entrée et des paramètres spécifiques. Un simulateur est associé à un formalisme de modélisation.

Le simulateur DESS [2] est spécialisé dans la modélisation des calculs différentiels. Ce simulateur gère les équations différentielles et contribue à la résolution des problèmes complexes de calcul dans le système.

Le simulateur DTSS se concentre sur la simulation des systèmes dynamiques temporels discrets et s'occupe de discrétiser les données d'entrée. Ce type de simulateur est notamment apprécié dans le contexte de la simulation multi-agents [5].

Le simulateur DEVS [6] est dédié à la simulation des systèmes caractérisés par des événements discrets. Pour simuler des modèles basés sur DEVS, il existe deux types de simulateurs [3] : ceux qui simulent les modèles atomiques, appelés simplement *simulateurs*, et ceux qui simulent les modèles couplés, appelés *coordinateurs*. Ces outils de simulation sont utilisés pour simuler des systèmes à événements discrets en traitant les événements entrants et en mettant à jour l'état interne du système en fonction de ces événements.

Nous avons tiré parti des avantages de ces formalismes et simulateurs pour modéliser et simuler des systèmes à structure dynamique, couvrant tout le processus de modélisation, de la génération des équations aux calculs nécessaires pour simuler le système étudié.

Notons bien que dans le reste de cet article, nous adopterons une méthode basée sur les équations différentielles mais distincte de DESS. En effet, notre contribution implique l'utilisation d'équations différentielles qui ne dépendent pas du temps, mais plutôt d'autres variables.

### 3 Motivations et objectifs

De notre point de vue, il est impératif de concevoir une approche généralisée capable de modéliser les changements structurels de manière efficace, dynamique et adaptable à divers domaines d'application, et ce pour plusieurs raisons. Tout d'abord, il est parfois difficile d'utiliser les approches existantes à cause de leur complexité et de la difficulté à les appliquer pour certains cas réels. De plus, les systèmes modernes deviennent de plus en plus complexes et interdépendants. Il est donc nécessaire de posséder des modèles simples et génériques afin de s'adapter rapidement aux changements structurels.

L'exploitation des calculs symboliques [7] as-

sure l'élaboration d'une approche générique. En effet, cette technique facilite la création d'équations et garantit la simplification des expressions, la résolution d'équations, la dérivation et l'intégration. Les calculs d'équations symboliques impliquent des équations différentielles pour capturer les variations continues des grandeurs physiques ainsi que d'équations à temps discret et à événements discret.

Nous proposons un simulateur basé sur les formalismes DEVS, DTSS et les calculs symboliques. Ce simulateur permet de représenter un système continu à structure dynamique basé sur des équations différentielles.

Les ajustements continus de ces équations tout au long de la simulation reflètent les transformations du système au fil du temps simulé, permettant ainsi une représentation fluide et précise des comportements individuels et des interactions dynamiques du système. En développant cette approche, nous sommes en mesure de décrire les interactions entre les composants et donner une représentation détaillée de l'évolution du système à chaque instant de la simulation.

## 4 Méthodologie

Dans cette section, nous présentons l'algorithme, l'implémentation et l'architecture du système utilisé pour créer le prototype de modélisation des systèmes à structure dynamique proposé. Nous détaillons la méthodologie employée pour développer ce prototype, en mettant l'accent sur les étapes clés de la conception et de la mise en œuvre. Nous décrivons également l'architecture du système et expliquons comment les différents composants interagissent entre eux pour fournir une modélisation efficace de système à structure dynamique.

### 4.1 Algorithme de modélisation

La méthodologie proposée s'appuie sur l'application d'un algorithme illustré dans la figure 2, conçu pour diriger le processus de modélisation des systèmes à structure dynamique.

Dans cet algorithme, nous identifions des 'invariants', c'est-à-dire des calculs qui doivent être effectués dans la plupart des modèles du type ciblé. Ces calculs peuvent inclure la création de variables symboliques, la spécification des conditions initiales, la définition des contraintes du système, la formulation d'équations et la

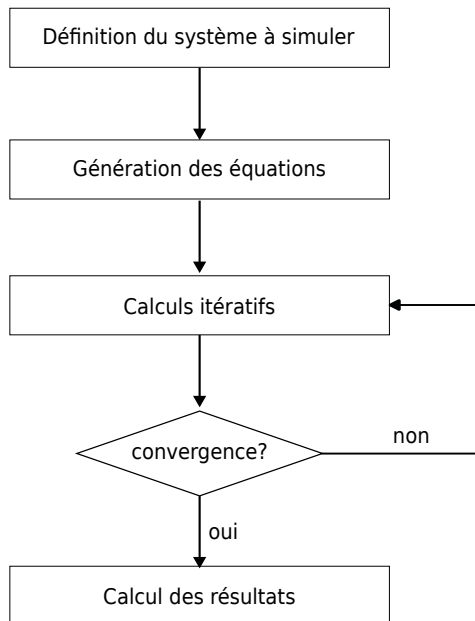


FIGURE 2 – Algorithme de modélisation

résolution de ces équations. Ces invariants sont des éléments clés de la modélisation et sont souvent nécessaires pour décrire les interactions entre les différents composants du système.

L'algorithme comprend les étapes suivantes :

1. **Définition du système à simuler** : cette étape implique la collecte des données essentielles à la modélisation du système, y compris les valeurs initiales, le nombre et le type des composants ainsi que leurs relations. De plus, elle comprend la configuration du système, au cours de laquelle nous définissons les paramètres et les conditions initiales requises pour les calculs ultérieurs.
2. **Génération des équations** : cette étape consiste à générer des équations symboliques à calculer dans les itérations et pour les résultats finaux. Pour ce faire nous exploitons la notion de calcul symbolique en créant des équations représentant les relations entre les différentes variables du système. Ces équations utilisent des symboles mathématiques et représentent des variables du système.
3. **Calculs itératifs** : une fois les équations générées, nous entamons une série d'itérations pour ajuster les inconnues du système et converger à une solution.
4. **Test de convergence** : à la fin de chaque itération, nous évaluons si le système a convergé vers une solution satisfaisante. Si ce n'est pas le cas, nous revenons aux calculs itératifs pour continuer le processus.

5. **Calculs des résultats** : une fois que les inconnues sont calculées, nous effectuons les calculs finaux pour obtenir les résultats de l'analyse.

#### 4.2 Implémentation de l'algorithme et architecture du simulateur

Nous avons donc découpé chacune des phases de l'algorithme présenté précédemment en noyaux de calculs et implémenté chacun de ces noyaux dans un composant DEVS indépendant. Ces composants sont ensuite couplés entre eux de manière à ce qu'ils s'échangent la représentation du système modélisé (sous la forme d'équations, de variables, etc.). On obtient ainsi un simulateur, basé sur une implémentation PDEVS statique, propre à une classe de systèmes, capable de représenter n'importe quel système de cette classe, y compris si ces systèmes sont à structure dynamique.

L'implémentation de cet algorithme conduit à la mise en place d'un simulateur qui adopte l'architecture détaillée dans la figure 3.

Cette implémentation conduit à la mise en place des composants suivants :

- **Horloge** : ce composant permet de générer les pas de simulation nécessaires à une simulation DTSS. L'horloge fournit un signal temporel discret qui définit le rythme de progression du temps dans le simulateur DTSS.
- **Initialisation** : ce composant a pour rôle de fournir les valeurs initiales et les configurations nécessaires au démarrage de la simulation du système étudié.
- **Génération des équations** : ce composant produit des formules mathématiques sous forme d'équations qui représentent les interactions entre les diverses variables du système, facilitant ainsi la prédiction de son comportement dans différentes conditions. Ces équations peuvent être des équations différentielles ou discrètes, en fonction de la nature dynamique du système et des modèles utilisés pour le représenter.
- **Calculs des équations et observation de convergence** : ce composant garantit l'évaluation des équations en fonction des valeurs d'entrée recueillies et détermine s'il est nécessaire de les recalculer pour obtenir les valeurs conduisant à la convergence. Une fois la convergence atteinte, les résultats sont obtenus. Si la convergence n'est pas atteinte, le processus se termine après que les résultats divergent.

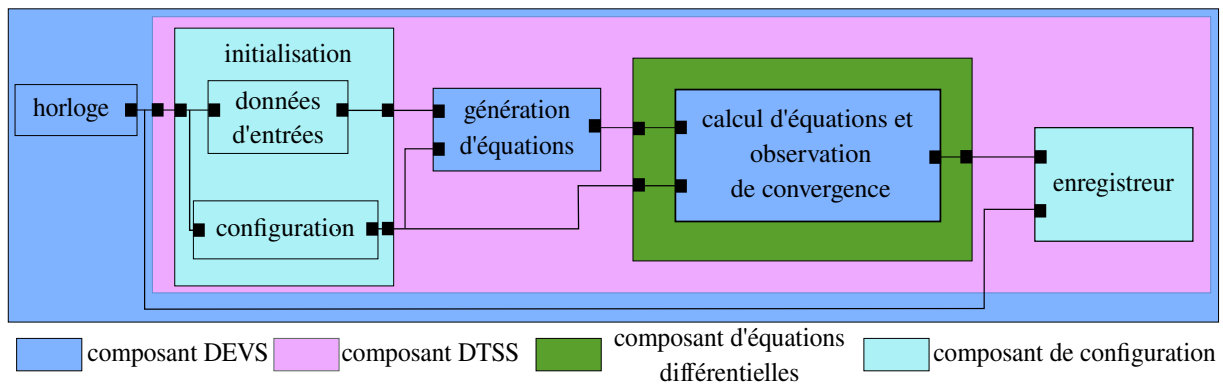


FIGURE 3 – Architecture générée par l'algorithme proposé

- **Enregistrement des résultats** : ici, nous enregistrons une trace des résultats obtenus pour une analyse ultérieure.

## 5 Cas d'application

Nous avons exploité la méthodologie décrite dans la section 4 pour modéliser et simuler l'analyse de l'équilibre de puissance dans un réseau électrique. Un réseau de type AC est caractérisé par un courant alternatif, un réseau DC par un courant continu, et un réseau hybride par la combinaison de ces deux types de courants. Le réseau simulé peut être de n'importe laquelle de ces architectures.

### 5.1 Contexte

L'analyse de l'équilibre de puissance [8] ou analyse de flux de charge, est une technique utilisée dans le domaine de l'ingénierie électrique pour évaluer et analyser le comportement des réseaux électriques. Cette analyse vise à déterminer les conditions de fonctionnement du réseau électrique en termes de tensions, de courants et de puissances, afin d'assurer un fonctionnement stable et fiable du système.

Plus précisément, l'analyse de flux de charge permet de calculer les paramètres du réseau électriques. Pour un réseau AC les paramètres sont la matrice d'admittance  $[Y]$ , les impédances  $Z$ , les tensions  $V$ , les angles de phase  $\delta$ , les puissances actives  $P$  et réactives  $Q$ . Ces paramètres varient sous différentes conditions de charge et de configuration du réseau [9].

Ces calculs sont essentiels pour garantir que le réseau fonctionne dans des limites acceptables de tension et de puissance, tout en assurant

l'équilibre entre la production et la consommation d'énergie électrique.

Les résultats de l'analyse de flux de charge sont utilisés pour prendre des décisions importantes dans la planification, la conception et l'exploitation des réseaux électriques, notamment pour dimensionner les équipements, optimiser les flux de puissance et évaluer la stabilité du système face aux variations de charge et aux perturbations externes.

### 5.2 Simulation de l'analyse de l'équilibre de puissance

Le système d'analyse de l'équilibre de puissance représente un exemple des systèmes qui subissent des changements structurels au cours de leur évolution.

Pour modéliser ce type de système, nous commençons par définir le système à simuler. Ce système se compose de différents composants, chacun étant doté de son propre bus. Les bus sont reliés entre eux par des lignes. De plus, il est nécessaire de fournir les valeurs initiales et les configurations nécessaires pour démarrer la simulation.

La figure 4 offre une présentation simplifiée d'un exemple de réseau électrique.

Ensuite nous passons à l'étape de génération des équations :

- La matrice d'admittance qui sont définies par :

$$Y_{i,j} = \begin{cases} -1 \times admittance_{i,j} & \text{si } i \neq j \\ \sum_{k=1}^n admittance_{i,k} & \text{si } i = j \end{cases}$$

où  $admittance_{i,j}$  représente l'admittance entre les nœuds  $i$  et  $j$ .

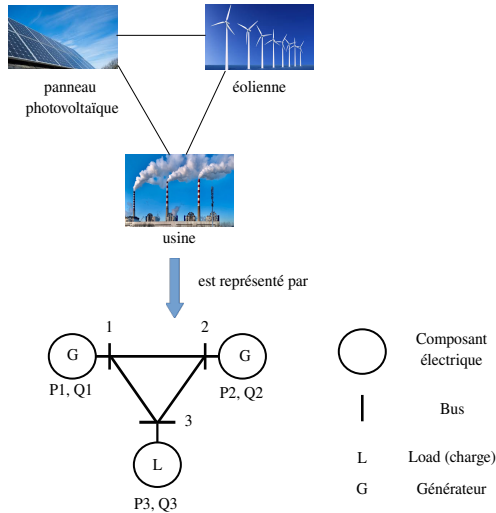


FIGURE 4 – Représentation simplifiée d'un réseau électrique

- P et Q, à utiliser dans les itérations et dans les résultats, qui sont définies par :

$$P_i = \sum_{j=1}^n |V_i||V_j||Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j)$$

$$Q_i = \sum_{j=1}^n |V_i||V_j||Y_{ij}| \sin(\theta_{ij} - \delta_i + \delta_j)$$

où  $V_i$  et  $V_j$  sont les tensions aux nœuds i et j,  $\delta_i$  et  $\delta_j$  sont les angles de phase aux nœuds i et j,  $Y_{ij}$  et  $\theta_{ij}$  représentent la norme d'admittance et son angle entre les nœuds i et j.

- La matrice Jacobienne  $J$  qui sera utilisée pour résoudre les équations de flux de charge par la méthode de Newton-Raphson. Elle est définie par :

$$J = \begin{pmatrix} \frac{\partial P}{\partial \delta} & \frac{\partial P}{\partial V} \\ \frac{\partial Q}{\partial \delta} & \frac{\partial Q}{\partial V} \end{pmatrix}$$

où les éléments de la matrice jacobienne sont les dérivées partielles des puissances actives et réactives par rapport aux angles de phase et aux magnitudes des tensions.

Puis nous utilisons des techniques de simulation à temps discret pour résoudre itérativement les équations de P et Q, tout en ajustant les variables inconnues telles que l'angle de déphasage  $\delta$  et la tension V du système.

Ceci est réalisé par la résolution du système d'équations défini par :

$$[\Delta_{PQ\_calculés}] = J \times [\Delta_{inconnus}]$$

Périodiquement, nous évaluons les événements discrets pour tester la convergence du système. Une fois que la convergence est atteinte, nous calculons les valeurs de P et Q pour les résultats finaux et les enregistrons.

Une description de différentes étapes pour la modélisation de ce système est présenté dans la figure 5.

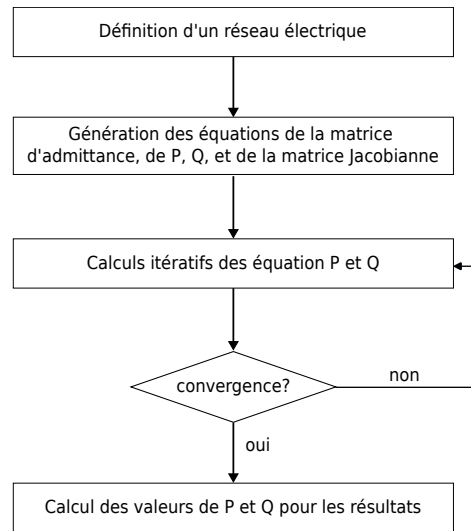


FIGURE 5 – Algorithme de modélisation de l'analyse de l'équilibre de puissance

### 5.3 Implémentation

Pour modéliser et simuler l'évaluation de l'équilibre des charges dans un réseau électrique, nous avons spécifié les modèles décrits dans la section 4.2 :

- Génération des équations :
  - Équations de la matrice d'admittance
  - Équations des P et Q
  - Équations de la matrice Jacobienne
- Calcul des équations :
  - Équations de P et Q
  - Équations de la matrice Jacobienne
  - Équations de  $\Delta PQ$
  - Équations à inconnues

Une illustration des connexions entre les différents composants est présentée dans la figure 6. Les modèles *générateur d'équations* et *solveur* représentent des modèles couplés, tandis que les modèles d'*équations de la matrice d'admittance* et *calcul de PQn* sont des modèles atomiques. La simulation des modèles *générateur d'équations* et *solveur* crée des coordinateurs, tandis que la simulation des modèles d'*équations*



de la matrice d'admittance et calcul de PQn créé des simulateurs.

En résumé, l'implémentation de notre système combine différents formalismes pour modéliser et simuler efficacement les aspects discrets et continus du système, tout en utilisant les fonctionnalités spécifiques de chaque formalisme pour répondre aux besoins de notre application.

## 6 Résultats

Nous avons appliqué cette implémentation sur des cas réels. La figure 7 montre l'évolution structurelle du réseau étudié.

Tout d'abord, nous avons commencé avec un réseau composé de deux composants. Ensuite ce réseau évolue en ajoutant un troisième composant. Puis ce dernier évolue encore pour qu'il soit composé par cinq composants. Nous avons comparé nos résultats avec les données fournies dans [10] [11] [12]. Les valeurs utilisées pour simuler ce réseau sont fournies dans les tableaux 1 et 2.

**NB :**

- Nous avons fait plusieurs cas d'étude mais pour simplifier nous présentons ici une version réduite.

t	du_bus	vers_bus	résistance	réactance
1	1	2	0	0.1
2	1	2	0.02	0.04
2	2	3	0.0125	0.025
2	1	3	0.01	0.03
3	1	2	0.02	0.1
3	1	5	0.05	0.25
3	2	3	0.04	0.2
3	2	5	0.05	0.25
3	3	4	0.05	0.25
3	3	5	0.08	0.4
3	4	5	0.1	0.5

TABLE 1 – Résistance et réactance des lignes

Après avoir saisi ces valeurs en tant que données d'entrée, nous avons utilisé les modèles présentés dans la figure 6 afin de générer un simulateur d'analyse d'équilibre de charge. Le tableau 3 montre les résultats.

Nous avons comparé les résultats obtenus avec ceux existants dans les références des exemples et nous avons constaté que nous avons obtenu approximativement les mêmes résultats avec 0.2% d'erreur.

t	type	bus	P	Q	V	$\delta$	inc
1	S	1			1	0	
1	C	2	-2	-1	1	0	$\delta;V$
2	S	1			1.05	0	
2	C	2	-4	-2.5	1	0	$\delta;V$
2	G	3	2		1.04	0	$\delta$
3	S	1			1.05	0	
3	C	2	-0.96	-0.62	1	0	$\delta;V$
3	C	3	-0.35	-0.14	1	0	$\delta;V$
3	C	4	-0.16	-0.08	1	0	$\delta;V$
3	G	5	0.24		1.02	0	$\delta$

TABLE 2 – Caractéristiques énergétiques des composants

t	résultats
1	P1 : 1.9999, Q1 : 1.6828
2	P1 : 2.1841, Q1 : 1.4085, Q3 : 1.4617
3	P1 : 1.2701, Q1 : 0.7213, Q5 : 0.3196

TABLE 3 – Résultats pour l'analyse du flux des charges avec les modèles créés

## 7 Conclusion

En conclusion, nous avons proposé une approche intégrée et générique pour la modélisation de systèmes à structure dynamique. Notre approche combine les formalismes DEVS et DTSS pour modéliser les systèmes à équations différentielles et représenter les changements structurels. Cette approche combine les formalismes DEVS et DTSS pour représenter ces systèmes, tout en permettant de rendre compte des changements structurels. En utilisant des calculs symboliques pour décrire les interactions entre les composants, notre méthodologie offre une solution flexible et adaptable aux différents domaines. En outre, la simulation de différents systèmes conduit à la création de simulateurs spécifiques à chaque cas d'utilisation.

Nous avons présenté l'implémentation et l'architecture des systèmes qui appliquent cette méthodologie, ainsi que l'interaction entre les différents composants qui coexistent. Cette approche a été appliquée avec succès à l'analyse de flux de charges dans un réseau électrique, démontrant ainsi son efficacité dans la représentation adéquate de l'évolution du système à chaque instant de la simulation. Les prochaines étapes de notre travail visent à automatiser la création des simulateurs sur la base des équations propres à une classe de système. Pour ce faire, nous envisageons d'utiliser l'approche du langage spécifique à un domaine et les



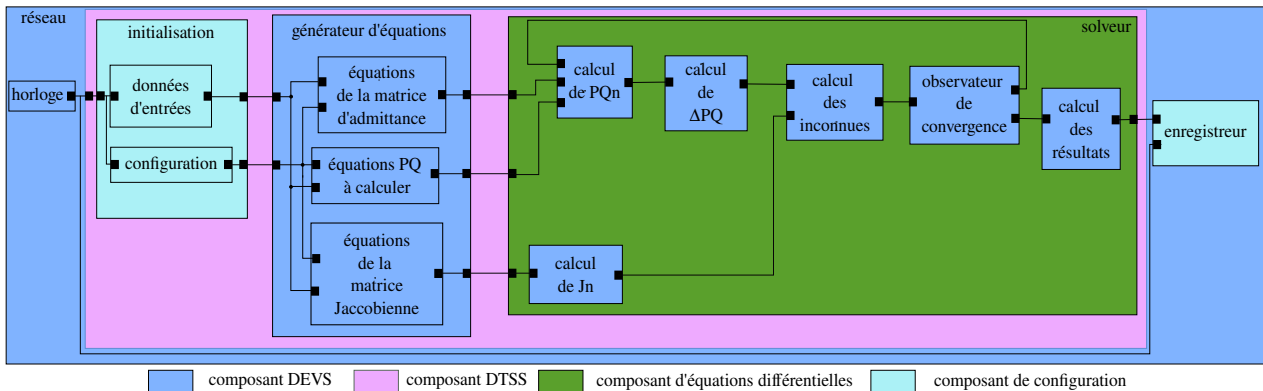


FIGURE 6 – Simulateur pour l'analyse de l'équilibre de puissance

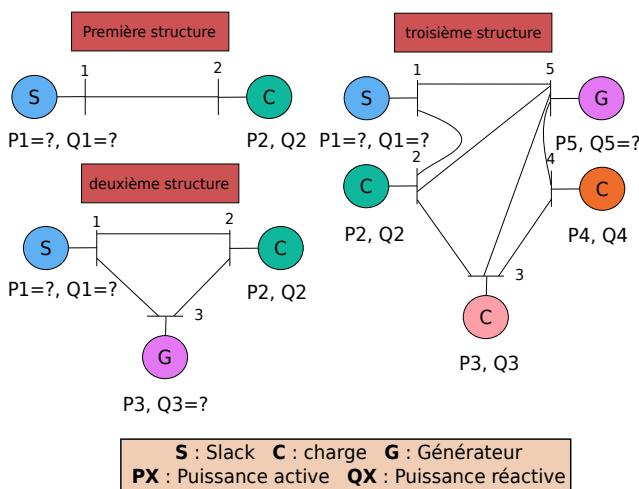


FIGURE 7 – Evolution structurelle du réseau étudié

graphes. Ces pistes nous permettront de concevoir une méthode pour générer des simulateurs adaptés à des domaines particuliers.

## Références

[1] Bernard Zeigler, Alexandre Muzy, and Ernesto Kofman. *Theory of Modeling and Simulation : Discrete Event & Iterative System Computational Foundations*. Academic press, December 2018.

[2] François E Cellier and Ernesto Kofman. *Continuous system simulation*. Springer Science & Business Media, 2006.

[3] Bernard P Zeigler. Hierarchical, modular discrete-event modelling in an object-oriented environment. *Simulation*, 49(5) :219–230, 1987.

[4] Alex Chung Hen Chow and Bernard P Zeigler. Parallel devs : A parallel, hierarchical, modular modeling formalism. In *Proceedings of Winter Simulation Conference*, pages 716–722. IEEE, 1994.

[5] Fabien Michel. *Formalisme, outils et éléments méthodologiques pour la modélisation et la simulation multi-agents*. PhD thesis, Montpellier II, 2004.

[6] Hans Vangheluwe. The discrete event system specification (devs) formalism. *Course Notes, Course : Modeling and Simulation (COMP522A)*, McGill University, Montreal Canada, 13, 2001.

[7] Debidas Ghosh. *Introduction To Theory Of Automata, Formal Languages, And Computation*. PHI Learning Pvt. Ltd., 2013.

[8] Turan Gonen. *Electric power distribution engineering*. CRC press, 2015.

[9] Mohammed Albadi and K Volkov. Power flow analysis. *Computational Models in Engineering*, pages 67–88, 2020.

[10] Tom Overbye. Lecture 13 newton-raphson power flow department of electrical and computer engineering ECE 476 POWER SYSTEM ANALYSIS. <https://slideplayer.com/slide/8858669/>.

[11] Arindam Ghosh. Power system analysis. [https://archive.nptel.ac.in/content/storage2/courses/108104051/chapter\\_4/4\\_4.html](https://archive.nptel.ac.in/content/storage2/courses/108104051/chapter_4/4_4.html).

[12] Universidad de León España. STEPS TO SOLVE POWER FLOW ANALYSIS : FOR DUMMIES. <https://www.studocu.com/es/document/universidad-de-leon-espana/auditoria/new-raph-helpful/35325538>.