Fast calculation of electrical transients in power systems after a change of topology

Thomas, Romain

DOI
10.4233/uuid:40d94dfa-75f8-4b72-b313-3e72c782b9f9

Publication date
2017

Document Version
Final published version

Citation (APA)
Thomas, R. (2017). *Fast calculation of electrical transients in power systems after a change of topology*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:40d94dfa-75f8-4b72-b313-3e72c782b9f9

FAST CALCULATION OF ELECTRICAL TRANSIENTS IN POWER SYSTEMS AFTER A CHANGE OF TOPOLOGY

FAST CALCULATION OF ELECTRICAL TRANSIENTS IN POWER SYSTEMS AFTER A CHANGE OF TOPOLOGY

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op donderdag 30 november 2017 om 10:00 uur

door

Romain THOMAS

Diplôme d'Ingenieur en Génie Electrique et Automatique
Institut National Polytechnique de Toulouse
Ecole Nationale Supérieur d'Electrotechnique,
d'Electronique, d'Hydraulique et des Télécomucations,
Toulouse, France,
geboren te Quimper, Frankrijk

Dit proefschrift is goedgekeurd door de

Promotor: Prof. ir. L. van der Sluis
Promotor: Prof. dr. ir. C. Vuik
Copromotor: Dr. D.J.P. Lahaye

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. ir. L. van der Sluis,	Technische Universiteit Delft
Prof. dr. ir. C. Vuik,	Technische Universiteit Delft
Dr. D.J.P. Lahaye,	Technische Universiteit Delft

Onafhankelijke leden:

Prof. dr. S. Schöps,	Technische Universität Darmstadt
Prof. dr. ing. A.J.M. Pemen,	Technische Universiteit Eindhoven
Prof. dr. ir. C.W. Oosterlee,	CWI & Technische Universiteit Delft
Prof. dr. ir. H.X. Lin,	Technische Universiteit Delft
Prof. dr. P. Palensky,	Technische Universiteit Delft, reservelid

Keywords: Power system, Elctrical transient, Modeling nethods, Ordinary differential equations, Integration methods, Runge-Kutta methods, linear solvers

Printed by: Ridderprint, the Netherlands

Front & Back: *Tout au bout du fil*, Anne et Florence Ramos & *A venir*, Florence Ramos

ISBN 978-94-6299-791-2

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

Without mathematics, there's nothing you can do.
Everything around you is mathematics.
Everything around you is numbers.

Shakuntala Devi

Contents

SUMMARY

A Power system is composed of various components such as generators, transformers, transmission lines, switching devices and loads. They have their mathematical model and graphical representation. Sometimes, a power system's change of topology occurs due to events like short circuits, lightning striking a transformer, or a reconfiguration of the transmission system.

In this thesis, a new way of simulating large scale power systems is presented from the modeling point of view. In the literature, a lot of modeling methods and mathematical tools are available to tackle this subject. However, this thesis mainly focuses on the time domain simulation of large scale power systems - and in particular, transients which appear after a change of topology.

A change of topology in electrical networks impact time domain simulations on two levels. The first impact is that it is necessary to update or re-compute the set of equations. The computation time of this action on the topology can be significant - especially for large scale power systems. The second impact of this change of topology is the transient that will occur. Usually, this change will impose to numerically compute fast oscillations in currents and voltages until they reach a new steady state.

The novelty of this thesis is that it tackles the effect of switching action in terms of the modeling approach and the mathematical tools that can be used during the time integration method. For these reasons, the thesis focuses on two issues. The first is about the modeling approach of electrical networks, when switching devices are included. The second aspect examines the numerical solver for large power systems. The goal of this thesis is to compute the overall time domain simulation of a large scale power system faster.

The modeling approaches and the mathematical tools, which are needed to understand the contributions of the thesis, are described in Chapters 2 and 3, respectively. Chapter 2 demonstrates the basics of the electrical engineering theorems and modeling approaches of electrical networks. In Chapter 3, mathematical definitions and tools are defined. This literature review helps circumvent the problem of numerical simulation of electrical powers when a switching action occurs.

In Chapter 4, a new approach for modeling power system is shown, called the block modeling method. The methodology is based on the fact that mathematical descriptions of every power system components are known. The connection between them is made by a combination of switching devices.

Two physical properties of power systems need to be taken into account for this approach. The first property is that power system components have strayed capacitors between a phase and the ground. They then help to connect the power system components mathematically. However, Kirchoff's voltage law makes the parallel connection of capacitors a computational problem. For this reason, the switching devices are non-ideal and they are represented by a resistance of small value when they are in a closed position.

Finally, these properties makes the computation of the system of equations of the electrical network possible. The voltage at the terminal of each power system component is known,as is the topology of switching devices. Then, the computation of the current through all switching devices can be effectuated. Those currents are then used to calculate the derivative of the voltage across all strayed capacitors.

This chapter offers an introduction to non-linear elements in power systems. Non-linear elements considered in this thesis are a change in function in time of a value of a lumped element, such as an inductor whose inductance varies over time. This evolution in time of a lumped element value does not introduce new differential variables. Moreover, the substitution of switching devices by arc models is shown too. This change adds new non-linear differential equations. This section of the chapter subsequently focuses on the computation of the Jacobian matrix.

Chapter 5 describes and discusses the utilization of the block modeling approach on several linear and non-linear electrical networks. In the first instance, linear electrical systems of different sizes, including several switching actions, are studied and compared with an electrical power system software used for fast transient simulations. In the second instance, non-linear electrical networks, where several arc models replace some switching devices, are investigated. In particular, the effect on the overall computation time with the way of calculating the Jacobian matrix is also investigated.

Usually, power systems are stiff due to the difference between the small and large time constants due to the topology and the value of the different lumped elements. Then, the implicit integration method is recommended. This fact then implies the solving of systems of equations where the computation time increases factorially with their sizes. Chapter 6 presents the numerical study of the utilization of inexact solvers, instead of direct or iterative solvers for the simulation of large stiff power systems.

Chapter 7 presents the different electrical networks of various sizes and stiffnesses. These test cases allow the study of the effect of the sizes, stiffness, and time-stepping strategy when inexact solvers are employed. This study focuses on the computation of the first transient of the power system when no energies are involved, to the next steady state. For this reason, there is no switching action included in the studies. However, the block modeling method is applied to compute the system of differential equations.

Finally, Chapter 8 offers the conclusions and recommendations for future work.

SAMENVATTING

ELEKTRICITEITSNETTEN bestaan uit verschillende componenten, zoals generatoren, transformatoren, transmissielijnen, schakelapparaten en belastingen. Ze hebben elk hun wiskundige model en grafische weergave. Soms kan de topologie van een elektriciteitsnet veranderen door gebeurtenissen zoals kortsluiting, blikseminslag in een transformator of een her-configuratie van het transmissie en distributie systeem.

In dit proefschrift wordt een nieuwe manier gepresenteerd van het simuleren van grootschalige elektriciteitsnetten vanuit een modelleringsbenadering. In de literatuur zijn er veel modellen en wiskundige instrumenten beschikbaar om dit onderwerp aan te pakken. Dit proefschrift richt zich echter vooral op de simulatie van tijdsdomeinen van grootschalige elektriciteitsnetten - en in het bijzonder transiënten die verschijnen na een verandering van topologie.

Een verandering van topologie in elektrische netwerken beïnvloedt tijdsdomeinsimulaties op twee niveaus. De eerste impact is dat het nodig is om de set vergelijkingen bij te werken of opnieuw te berekenen. De berekeningstijd van deze actie op de topologie kan significant zijn - vooral voor grootschalige elektriciteitsnetten. De tweede impact van deze verandering van topologie is de transient die zal optreden. Gewoonlijk zal deze wijziging het noodzakelijk maken om snelle oscillaties in stromen en spanningen numeriek te berekenen tot ze een nieuwe stabiele toestand bereiken.

Het vernieuwende van dit proefschrift is dat de thesis het effect van wisselwerking aanpakt tussen de modelleringsbenadering en de wiskundige hulpmiddelen die tijdens de tijdintegratiemethode kunnen worden gebruikt. Daarom richt het proefschrift zich op twee problemen. De eerste gaat over de modelleringsbenadering van elektrische netwerken wanneer schakelaars worden meegenomen. Het tweede aspect onderzoekt de numerieke oplosser voor grote netten. Het doel van dit proefschrift is om de algehele tijddomein simulatie van een grootschalige elektriciteitsnetten sneller te berekenen.

De modelbenaderingen en de wiskundige hulpmiddelen die nodig zijn om de bijdragen van het proefschrift te begrijpen, worden beschreven in respectievelijk hoofdstuk 2 en 3. Hoofdstuk 2 toont de basis van de elektrotechnische theorieën en modelleringsbenaderingen van elektrische netwerken. In hoofdstuk 3 worden wiskundige begrippen en gereedschappen gedefinieerd.

In hoofdstuk 4 wordt een nieuwe benadering voor het modelleren van het elektriciteitsnetten getoond, genaamd de blokmodelleringsmethode. De methodologie is gebaseerd op het feit dat wiskundige beschrijvingen van alle componenten van het elektriciteitsnet bekend zijn. De verbinding tussen de blokken wordt gemaakt door een combinatie van schakelaars.

Voor deze aanpak moet rekening worden gehouden met twee fysieke eigenschappen van het elektriciteitsnet. De eerste eigenschap is dat componenten van het elektriciteitsnet altijd een zekere capaciteit hebben tussen een fase en de grond. Zij helpen dan mathematisch de componenten te verbinden. De spanningswet van Kirchhoff verbiedt

parallelle aansluiting van condensatoren. Om deze reden worden de schakelaars niet als ideaal gezien en worden ze gerepresenteerd door een weerstand van kleine waarde wanneer ze in een gesloten positie staan.

Tenslotte maken deze eigenschappen de berekening van het stelsel van vergelijkingen van het elektrische netwerk mogelijk. De spanning aan de klemmen van elke netwerk component is bekend, evenals de topologie van schakelaars. Vervolgens kan de berekening van de stroom door alle schakelapparaten worden bewerkstelligd. Deze stromen worden dan gebruikt om de afgeleide van de spanning te berekenen over alle capaciteiten.

Dit hoofdstuk geeft voorts een inleiding tot niet-lineaire elementen in elektriciteitsnetten. Niet-lineaire elementen die in dit proefschrift worden beschouwd zijn een verandering van waarde in de tijd van een netwerk element, zoals een spoel waarvan de inductiviteit in de loop van de tijd varieert. Verder wordt ook de vervanging van schakelaars door boogmodellenbesproken. Deze voegen nieuwe niet-lineaire differentiaalvergelijkingen toe. Dit gedeelte van het hoofdstuk richt zich vervolgens op de berekening van de Jacobiaanse matrix.

Hoofdstuk 5 beschrijft het gebruik van de blokmodelleringsbenadering op verschillende lineaire en niet-lineaire elektrische netwerken. Op de eerste plaats worden lineaire elektrische netwerken van verschillende grootte en diverse schakelacties bestudeerd en vergeleken bestaande software die wordt gebruikt voor snelle transiënte simulaties. Op de tweede plaats worden niet-lineaire elektrische netwerken onderzocht, waar verschillende boogmodellen de schakelaars vervangen. In het bijzonder wordt ook de invloed op de totale berekeningstijd van de berekeningsmethode voor de Jacobiaanse matrix onderzocht.

Meestal zijn elektriciteitsnetten wiskundig beschouwd stijf vanwege het verschil tussen de kleine en grote tijdsconstanten die bepaald worden door het netwerk en de waarde van de verschillende Netwerkelementen. Dan wordt de impliciete integratie methode aanbevolen. Dit feit betekent dat het oplossen van systemen van vergelijkingen waar de berekeningstijd een factor met hun grootten toeneemt. Hoofdstuk 6 presenteert de numerieke studie van het gebruik van niet exacte oplossers, in plaats van directe of iteratieve oplossers voor de simulatie van grote stijve elektriciteitsnetwerken.

Hoofdstuk 7 presenteert de verschillende elektrische netwerken van verschillende groote en stijfheid. Deze testcases maken het mogelijk om het effect op de grootte stijfheid en tijdstapstrategie te bestuderen wanneer inexacte oplossers worden gebruikt.

Ten slotte geeft hoofdstuk 8 de conclusies en aanbevelingen voor toekomstig onderzoek op het terrein van de berekening van snelle verschijnselen in hoogspanningsnetten

1

INTRODUCTION

1.1. MOTIVATION

ON September 4, 1882, in the South of Manhattan, New York City, United States of America, the inhabitants and workers illuminated their households and offices via electrical lamps for the first time [1, 2]. This huge project was led by T. Edison. At the beginning of the project, a study had shown two main advantages of using electricity for lighting. The first one was economical because an electrical lamp costed lesser than candles or gas to illuminate a room. The second advantage was the brightness - electrical lights had a higher luminosity than flames, and they significantly increased the comfort for the eyes.

The power plant to produce the electrical power imagined by T. Edison was initially composed of a single generator. Then later, with the expansion of the electrical network and more interconnected households and buildings, five new generators were added. On the inauguration day, T. Edison described the lifetime of the generators as: "*They will go on forever unless stopped by an earthquake*" [3]. At that time, the scientific community believed that electricity might impact the creation of earthquakes [4].

Since that day, power systems permit the production, transportation and supply of electrical energy [5, 6]. For example, in T. Edison's power system design, the generators provided the electrical power. Then, this energy was transported via an electrical network to the different light bulbs, and finally, these lamps transformed this energy into light. This design is still applied in today's power systems, and are composed of power plants that produce electrical power. This electrical power is distributed to the consumers via an interconnection of overhead transmission lines and underground cables.

In order to produce electrical power, power plants need a primary source of energy - such as a dam (hydro power plants), gas, oil and coal (thermal power plants), or uranium (nuclear power plants). They respectively contributed 48.8%, 12.8%, and 28.8% of the European Union's production in 2013, which was 3.10 million Gigawatt-hours (GWh). Finally, the remaining 9.6% electric power produced was due to wind, solar, and geothermal primary energies (renewable energy)[7].

The energy is transported and distributed to the consumers via overhead transmission lines, underground cables, and transformers [8]. The distance between power plants and consumers, and the power to be consumed, determines the design of the transmission and distribution system. The transmission system corresponds to the network between the power plants and the electrical sub-stations near the cities. Then, the distribution system corresponds to the network between these sub-stations and the households. The difference between the transmission and distribution system is the level of voltage needed to reduce the losses during the long distance transportation of energy. Transformers are used to change the voltage level between both systems.

There are around 552 million consumers in Europe. In 2013, the total consumption of electric power in the European Union was 2.77 million GWh [9]. The European Union defined three groups of users: industrial organizations, which consumed 1 million GWh, the transportation systems (e.g. trains), which consumed 0.66 million GWh, and third, private households, administrations, and commerce - which used 1.22 million GWh that year.

In general, a power system can be divided into two separate entities: the electric utility, which provides the electrical power and usually owns a distribution network, and the transmission system operator, which manages the transmission system. It coordinates the production and the demand for electric energy with the different electric utilities and other transmission system operators in real-time, and also actively participates in the electrical market. Finally, a power system operator conducts a multitude of projects to ensure the transportation of energy for the future.

In order to play its role, the transmission system operator needs a lot of information. First, it requires a complete and accurate model of its electrical network. Then, it needs to collect data to predict the consumption and the production of energy along the power system. With all this information, it uses a supervisory control and data acquisition which monitors and manages the whole power system in real-time [10]. Besides, it utilizes this information for its research for its future transmission system. Finally, the transmission system operator works on different time scales, as shown in Figure 1.1.

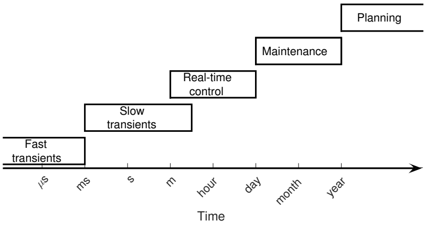


Figure 1.1: The power system events that occur, and their time frame

According to Figure 1.1, various actions that could occur on the electrical network have different time scales (from microseconds to several years). The definitions of these acts are:

- Planning: Consists of designing future electrical networks. Usually, the design is defined in collaboration with the electric utilities and their view on the future consumption of electrical energy;

- Maintenance: Permits the change of components that malfunction, or have reached the end of their life, for a better sustainability of the power system;
- Real-time control: Allows the control and operation of the network, from the data along the network, in order to reduce losses and increase the efficiency of the system;
- Slow transients: For example, after a short-circuit, the recovery of the optimal frequency and voltages can be slow according to the power system's topology;
- Fast transients: For instance, just after clearing a short-circuit via a circuit breaker, a very rapid increase of voltage can occur between the extremities of the circuit breaker.

The study of fast transients is necessary for the sustainability of the power system, and it is effectuated during the planning stage. For example, the design of circuit-breakers, transmission lines, and cables depends on the instantaneous maximal currents and voltages that they can support [11]. Then, the evolution in time of currents and voltages can be obtained by a time domain simulation for a better sustainability of the electrical network. For these reasons, this thesis focuses on those phenomena.

For the most accurate time domain simulation, the mathematical model of the power system should be as precise as possible. In order to obtain the mathematical expression, Maxwell's equations can be applied [12]. However, this method requires a comprehensive understanding of the physics involved, and a significant computational power. For this reason, the circuit theory has been developed for de-complexing Maxwell's equations and predicting the electrical behavior of an electrical circuit [13].

When the circuit theory is applied to a power system, the simplest model consists of using lumped elements (e.g. resistances, inductances, capacitances, and sources). Additionally, every lumped element has its own graphical representation and mathematical description [14]. Their electrical behavior depends on a relation between the current through them and the voltage across them [15]. Besides, this relationship can be linear or non-linear. For example, the utilization of an arc model to represent the interruption of current in a circuit breaker introduces non-linear equations [16].

The circuit theory consists of a multitude of laws and theorems [17–20]. The most useful rules are Kirchhoff's laws [17]. Kirchhoff's first law concerns the current interactions of different, lumped elements that are connected. Kirchhoff's second law represents the voltage interaction between lumped elements. Then, when they are applied, the network equations are found. The Norton-Thevenin theorem [18, 19] or Millman theorem [20] can be used to either simplify the graph or find the voltage at a single point.

From these laws and theories, several methods are available to model any power systems based on circuit theory. Traditional modeling approaches were developed 50 years ago for the time domain simulation. These methods are the nodal analysis [21], the modified nodal analysis [22], and the cut set methods [23]. A brief description of these methods is given below:

- After applying the nodal analysis method, the network equations are found and they can be solved easily. This method is based on Kirchhoff's current law. Moreover, this approach imposes a fixed time-step. Finally, simulations of large scale power systems use this algorithm;

- The modified nodal analysis method uses Kirchhoff's current and voltage laws to obtain a set of differential algebraic equations. Then, this method allows the splitting of the time interval with various time-steps. However, this approach is restricted in terms of an algorithm to obtain the time domain solution due to its mathematical formulation;

- A set of ordinary differential equations is obtained by the use of the cut-set method. Unlike the modified nodal analysis, it allows the use of any algorithm in order to obtain the time domain solution. However, with this approach, the methodology to get the mathematical representation of a power system grows faster than its size .

Finally after a change of topology, these modeling methods require the updation of the whole system of equations.

For example, a change of topology can happen when a switching action occurs [24]. Then, the system of equations needs to be updated. Also, this change of topology may create very fast oscillations in voltages and currents across the power system [11]. These oscillations can have a frequency of up to 10MHz, and decrease over time to reach a new steady state. Consequently, a switching action requires a lot of computational resources - first, to update the system of equations, and second, to compute the time signals of the high-frequency oscillations.

Now, the power system equations are known and the time domain simulation is required. For the modified nodal analysis method and the cut set method, two methods are available to compute the time evolution of the current and voltage across the electrical network. The first method is the analytical solution; however, this method only works on linear power systems. The second method is the numerical approach, and it can be utilized for any power system.

The analytical solution of a set of linear differential equations can be calculated by computing the time constant or by using the Laplace transform and inverse Laplace transform [25–27]. These two methods are computationally expensive for large sets of linear equations. As a consequence, they are not recommended for the study of large power systems.

Numerical integration methods - such as the Euler methods, the Trapezoidal rule, or the Runge-Kutta methods - are applied to obtain the time domain solution [28–30]. These methods are usually less time consuming than analytical approaches, especially for large power systems. However, the solution is an approximation of the analytical solution and is based on the Taylor expansion [31]. Finally, this estimation depends on the properties of the numerical integration method applied.

The time domain solution of the numerical integration method depends mainly on:

- The time-step used:

The choice of the time-step is important for the numerical solution. On the one hand, it influences the accuracy of the simulation, and on the other , it is the dominant factor for the computation time, since the computation time of an overall simulation depends on the number of steps. Then, when the time-step is small, the computation time is larger than when a bigger time-step is applied;

- The type of the numerical integration method:

  There are two types of methods - explicit and implicit. Explicit methods have a quick process of calculation, while implicit methods require more computational power per time-step. In the case of a stiff system of equations, an implicit method - in general - allows a larger time-step and can be more accurate than an explicit method;

- The order of the numerical integration method:

  It corresponds to the higher derivative of the Taylor expansion. Then, the order also plays a role in the error at each time-step, which is proportional to the time-step power of the order of the method.

From the characteristics of a power system, the choice of a numerical integration method can be made. First, it should not have a high order, because it requires a large computation time and not a small order due to the error made at each time-step - therefore, second-, third-, or fourth-order methods are often chosen. Second, power systems are stiff. This results in high-frequency oscillations created after a switching action, and consequently, implicit methods are recommended. Third, some numerical integration methods allow adaptive time-stepping algorithms[29]. They change the time-step according to the change in stiffness of the variables to integrate.

Implicit methods require more computation time per time-step than explicit methods. This increase in complexity is due to the linear systems of equations to solve in each time-step[30]. In the literature, two types of solvers are used: linear solvers for linear systems of equations, and non-linear solvers for non-linear systems of equations [32, 33]. Unless they require a large computational power, they always converge to the steady-state for any time-step applied.

The category of the linear solver is composed of two subcategories - direct or iterative methods [34]. The direct methods are derived from the Gaussian elimination, such as LU or Cholesky factorizations. Small systems of equations usually require such solvers. Otherwise, iterative solvers are applied, like GMRES [35]- which is a Krylov subspace method [32].

Non-linear solvers are mainly based on the Newton-Raphson iteration process[33]. They permit a convergence to the solution after several iterations when, during the process, a linear system of equations has to be solved. Therefore, linear solvers are used.

Also, some features have to be taken into account to speed up the computation. The first feature is to write the system of equations in a matrix form. This form contains many zeros; then, the sparse notation can be employed [36]. Folowing this, only non-zero elements and their indexes are stored. Finally, the numerical arithmetic is faster. The second useful feature is to use a re-ordering method such as the Approximate Minimum

Degree (AMD) method, for example[37]. It permits a reduction in the number of non-zero elements during the factorization. As a consequence, it decreases the computation time needed to solve the system of equations.

Presently, most new studies are looking into hardware or co-simulation [38, 39] to overcome the computational challenge. Present-day hardware studies delve into the use of graphical processor units (GPUs). Co-simulation studies instead consider using a multitude of computers to perform a single simulation.

The utilization of GPU typically gives a relative speed-up for large power systems under consideration, because arithmetic operations such as matrix-vector operations can be easily parallelized[38]. However, there usually are one or several linear systems of equations to solve at each time-step. Furthermore, it is used for very large power systems.

The co-simulation of power systems could also give impressive results by using a multitude of computers[39]. However, the work of matching results at the interface between computers can lead to a numerical instability and inaccuracy of the time domain solution.

The computation time of large power systems is a challenge. The computation time is related to their size. Besides, their size increases every day. A lot of studies are performed on this topic because faster simulations imply that more simulations can be made [40–42]. Then, new developments can be achieved quickly.

The computational challenge is not posed only to use of numerical integration method and the solver. It is a combination of the previous facts as well as the method to compute the set of equations. Today, with the increased of power electronics due to the utilization of renewable energy - and therefore - switching actions, new modeling methods which tackle these problems have to be developed.

1.2. PROBLEM DEFINITION

As explained in the previous section, modeling methods rely on algorithms developed 50 years ago. After a switching action, it becomes computationally expensive due to the change of topology. As a consequence, the first research question of this thesis to be answered is:

Is it possible to design a new methodology to model a power system which allows the use of adaptive time-stepping strategy with a relatively small computation time for re-computing or updating the network equations in the case of a switching event within an ordinary differential equation formulation?

The second research question is related to the association of the linear solver and the numerical integration method. Usually, during the integration process, solving the system of equations takes the most computation time. As a consequence, the time domain solution computation depends on the size, stiffness, tolerances, and topology of the power system under consideration. Then, the second research question of this thesis to be answered is:

How strong is the interaction between the time loop of a numerical, semi-explicit, diagonally implicit integration method of an ordinary differential equation and the solver for obtaining the time domain solution of a power system?

1.3. Research objectives

The research objectives divide the previous research questions into two categories - the modeling approach of a power system, and the numerical integration aspect of the time domain solution. Finally, the following sections describe the two research objectives.

1.3.1. Objectives for the modeling approach of a power system

- To review the state-of-the-art modeling methods for power system simulations, with a focus on traditional approaches;
- To develop a new approach to view a power system, especially in terms of ordinary differential equations when a switching action occurs;
- To be able to replace switching devices by non-linear elements such as arc models, for which the Jacobian matrix is easy to assemble.

1.3.2. Objectives for the numerical approach

- To review the state-of-the-art numerical tools, such as integration methods and linear solvers, which could be used for the simulation of power systems;
- To analyze the use of inaccurate linear solvers for the implicit numerical integration of ordinary differential equations for power system simulations.

1.4. Research approach

To satisfy the research problems and objectives, the following research approach has been used. In the beginning, a literature study on the modeling approaches and numerical analysis methods is performed. This survey reveals the actual advantages and drawbacks of the various modeling methods and mathematical tools being used.

Then from the literature review, a new approach to model power systems is developed. This process benefits from the different methods - without their inconveniences. Then, the implementation of non-linear elements, such as arc models, is considered for this method. Finally, for testing purposes, various power systems are studied.

As explained previously, a large number of numerical integration methods are available. As a consequence, the relation between solvers and numerical integration method is investigated. Then, this relationship provides the possibility of using special linear solver, instead of regular solvers, for the study of large linear power systems without compromising the accuracy of the time domain solution.

1.5. Outline of the thesis

The thesis chapters are defined as follows:

- Chapter 2 - This chapter contains the literature review of the state-of-the-art, from the circuit theory to the principal modeling method (the nodal analysis, the modified nodal analysis, the cut-set method). Moreover, it also contains some power system components.

- Chapter 3 - This chapter presents the different notions of the mathematical analysis of this thesis. A brief introduction to the ordinary differential equations is given. Afterwards, basic integration methods are illustrated. Finally, introductions to solve systems of equations are shown.
- Chapter 4 - A new approach to model power systems is presented in this chapter, which is later refered to as the block modeling method. First, a detailed demonstration is given, and then, a simple example is studied. Finally, the utilization of arc models is described for the analytical Jacobian matrix computation.
- Chapter 5 - Various different power systems resulting from the block modeling method are studied. Both linear and non-linear power systems are discuted.
- Chapter 6 - The use of inaccurate linear solvers during the numerical integration of power systems is investigated in this chapter. Then, advantages and consequences are drawn.
- Chapter 7- Here, several power systems of different sizes and stiffness are investigated when approximate solvers are used during the integration process. The study employs a fixed and adaptive time-step.
- Chapter 8 - The main conclusions, contributions, and recommendations for further research are offered in this chapter.

REFERENCES

[1] *Edison's light turned on, downtown building supplied from the sation in Pearl street,* The Sun , New York edition (Sep 5, 1882).

[2] *The Edison electric light at last,* The Brooklyn Daily Eagle (Sep 5, 1882).

[3] *Miscellaneous city news; Edison's electric light. the times building illuminated by electricity.* The New York Times (Sep 5, 1882).

[4] *Earthquakes,* The New York Times (Oct 2, 1882).

[5] P. Schavemaker and L. van der Sluis, *Electrical Power System Essentials* (John Wiley and Sons, Chichester, 2008).

[6] C. C. Antonio Gomez-Exposito, Antonio J . Conejo, *Electric Energy Systems, Analysis and Operation* (CRC Press city, Boca Raton, 2008).

[7] Eurostat Statistics Explained, *Electricity production, consumption and market overview,* (2016).

[8] S. A. Nasar and F. Trutt, *Electric Power Systems* (CRC Press, Boca Raton, 1998).

[9] Eurostat, *Electricity consumption by industry, transport activities and households/services (gwh),* (2009-2014).

[10] B. Theraja and A. Theraja, *Text Book of Electrical Technology: Volume 3: Transmission, Distribution and Utilization* (S. Chand & Company Ltd, New Delhi, 2013).

[11] L. van der Sluis, *Transients in Power Systems* (John Wiley & Sons, Chichester, 2001).

[12] J. C. Maxwell, *A dynamical theory of the electromagnetic field,* Philosophical Transactions of the Royal Society of London **155**, 459 (1865).

[13] R. DeCarlo and P.-M. Lin, *Linear Circuit Analysis,* 2^{nd} ed. (OXFORD UNIVERSITY PRESS, New York, 2001).

[14] L. Chua, C. Deseor, and E. Kuh, *Linear and nonlinear circuits* (McGraw-Hill Book Company, New York, 1987).

[15] B. Theraja and A. Theraja, *Text Book of Electrical Technology: Volume 1: Basic Electrical Engineering in S.I. Units* (S. Chand & Company Ltd, New Delhi, 2004).

[16] M. Kapetanovic, *High voltage circuit breakers* (Falculty of Electrotechnical Engineering, Sarajevo, Sarajevo, 2011).

[17] G. Kirchhoff, *Ueber den durchgang eines elektrischen stromes durch eine ebene, insbesondere durch eine kreisförmige,* Annalen der Physik **140**, 497 (1845).

[18] E.L.Norton, *Design of finite networks for uniform frequency characteristic.Technical Report TM26-0-1860*, Tech. Rep. (Bell Laboratories, 1926).

[19] L. Thévenin, *Sur un nouveau théoreme d'électricité dynamique [on a new theorem of dynamic electricity],* CR des Séances de l'Académie des Sciences **97**, 159 (1883).

[20] J. Millman, *A useful network theorem,* proceeding of the IRE **28**, 413 (1940).

[21] H. W. Dommel, *Digital computer solution of electromagnetic transients in single-and multiphase networks,* Power Apparatus and Systems, IEEE Transactions on **PAS-88**, 388 (1969).

[22] C. W. Ho, A. E. Ruehli, and P. A. Brennan, *The modified nodal approach to network analysis,* Circuits and Systems, IEEE Transactions on **22**, 504 (1975).

[23] E. Kuh and R. Rohrer, *The state-variable approach to network analysis,* Proceedings of the IEEE **53**, 672 (1965).

[24] R. Smeets, L. Van der Sluis, M. Kapetanovic, D. F. Peelo, and A. Janssen, *Switching in electrical transmission and distribution systems* (John Wiley & Sons city, Chichester, 2014).

[25] J. Vlach and K. Singhal, *Computer methods for circuit analysis and design* (Van Nostrand Reinhold, New York, 1983).

[26] P. Dyke, *Convolution and the solution of ordinary differential equations,* in *An Introduction to Laplace Transforms and Fourier Series* (Springer city, 2014) pp. 39–82.

[27] J. Stewart, *Calculus: early transcendentals* (Cengage Learning, Belmont, 2015).

[28] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, 1st ed. (Society for Industrial and Applied Mathematics, Philadelphia, 1998).

[29] W. Hundsdorfer and J. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations* (Springer, New York, 2003).

[30] K. Brenan, S. Campbell, and L. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equation* (North-holland, Amsterdam, 1989).

[31] P. D. Lax and M. S. Terrell, *Calculus with applications* (Springer, New York, 2014).

[32] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. (Society for Industrial and Applied Mathematics, Philadelphia, 2003).

[33] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Vol. 30 (Society for Industrial and Applied Mathematics, New York, 1970).

[34] W. Hackbusch, *Iterative solution of large sparse systems of equations*, Vol. 95 (Springer Science & Business Media, New York, 2012).

[35] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,* Society for Industrial and Applied Mathematics: Journal on Scientific and Statistical Computing **7**, 856 (1986).

[36] Å. Björck, *Numerical methods in matrix computations* (Springer, Cham, 2015).

[37] P. R. Amestoy, T. A. Davis, and I. S. Duff, *An approximate minimum degree ordering algorithm,* SIAM Journal on Matrix Analysis and Applications **17**, 886 (1996).

[38] Z. Zhou and V. Dinavahi, *Parallel massive-thread electromagnetic transient simulation on gpu,* IEEE Transactions on Power Delivery **29**, 1045 (2014).

[39] B. Zhang, W. Deng, L. Ruan, T. Wang, J. Quan, Q. Cao, Y. Teng, W. Wang, Y. Yuan, and L. Li, *Circuit cosimulation of 500-kv transformers in ac/dc hybrid power grid,* IEEE Transactions on Applied Superconductivity **26**, 1 (2016).

[40] F. Wang and M. Yang, *Fast electromagnetic transient simulation for over-voltages of transmission line by high order radau method and v-transformation,* IET Generation, Transmission Distribution **10**, 3639 (2016).

[41] S. Chiniforoosh, H. Atighechi, and J. Jatskevich, *A generalized methodology for dynamic average modeling of high-pulse-count rectifiers in transient simulation programs,* IEEE Transactions on Energy Conversion **31**, 228 (2016).

[42] A. A. van der Meer, M. Gibescu, M. A. M. M. van der Meijden, W. L. Kling, and J. A. Ferreira, *Advanced hybrid transient stability and emt simulation for vsc-hvdc systems,* IEEE Transactions on Power Delivery **30**, 1057 (2015).

2

Electrical background

2.1. Introduction

Electrical engineering is a complex and broad subject. This field discusses topics ranging from the interaction between magnetic and electromagnetic fields to the control of power systems. It also incorporates the understanding of electronic components and computer sciences. As specified in the previous chapter, this thesis focuses on the time domain solution of power systems. Thus, only the basics of electrical engineering are necessary.

To simulate a power system, its equations need to be computed. As a consequence, the fundamental idea of modeling power systems is to apply the circuit theory that simplifies the Maxwell equations [1, 2]. Next according to the circuit theory, several approaches to model power systems are available in the literature. These approaches are the nodal analysis method [3], the modified nodal analysis method [4] and, the cut-set method [5]. The studies of large power systems usually use the nodal analysis method, with the two other modeling methods being used less commonly for such computation. Other methods, such as co-simulation methods, are available [6]. However, they are not studied in this thesis.

The methodology applied to simplify the Maxwell equations is shown in Section 2.2. Then, a brief description of the various lumped elements is given in Section 2.3. Next, in Section 2.4, Kirchhoff's laws are demonstrated under some assumptions. Then, Norton's theorem and an introduction to the graph theory applied to circuit theory are treated in Section 2.5 and Section 2.6. Finally, the various approaches to model power systems are described in Section 2.7, along with some power system elements in Section 2.8. Section 2.9 provides information about the drawbacks, and the advantages of the modeling approaches.

2.2. Maxwell equations

The goal of the Maxwell equations, which were developed by J.C. Maxwell, is to predict the electrical behavior of voltages and currents of an electrical circuit [1]. The displacement of electrical charges creates voltages and currents, and by consequence, electrical and magnetic fields. This interaction between them is quantified by the Maxwell equations, as shown in Table 2.1.

	Integral equations	Differential equations
Gauss' law	$\oiint_{\partial\Omega} DdS = \iiint \rho dV$	$\nabla \cdot D = \rho$
Gauss' law for magnetism	$\oiint_{\partial\Omega} BdS = 0$	$\nabla \cdot B = 0$
Maxwell-Faraday equation	$\oint_{\partial\Sigma} Edl = -\frac{d}{dt}\iint_{\Sigma} BdS$	$\nabla \times E = -\frac{\partial B}{\partial t}$
Ampere's circuital law	$\oint_{\partial\Sigma} Hdl = \iint_{\Sigma} JdS + \frac{d}{dt}\iint_{\Sigma} DdS$	$\nabla \times H = J + \frac{\partial D}{\partial t}$

Table 2.1: Maxwell equations

Equations of Table 2.1 are related to each other by additional mathematical formulations about the constitution of the different media, as summarized in Table 2.2.

Dielectric medium	Magnetic medium	conductor
$D = \epsilon E$	$B = \mu H$	$E = \rho_e J$
		$J = \sigma E$

Table 2.2: Media's constitutive equations

Then, Figure 2.1 illustrates the schematic representation of the Maxwell equations from Tables 2.1 and 2.2.

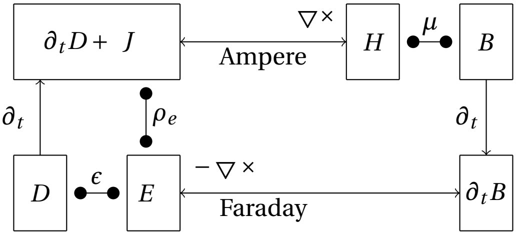


Figure 2.1: Representation of the Maxwell's equations

The wavelength (λ) of a periodical electrical wave (periodic signal) is defined such as [7]:

$$\lambda = \frac{v}{f}, \tag{2.1}$$

where v is the speed of light ($v \approx 300.000.000 m/s$) and f the frequency of the signal [Hz].

The wavelength sheds light on the possible utilization of lumped elements (e.g. resistances) to model the electrical circuit instead of the Maxwell equations (Table 2.1). In order to use lumped elements, the size of the network studied strictly needs to be smaller than λ. Consequently, only the left side and right side of Figure 2.1 are taken into account with lumped elements, without interaction between them.

2.3. MAIN LUMPED ELEMENTS

Lumped elements used in this thesis have two terminals. The current entering through one terminal is the same as the current leaving the other terminal [2]. Their physical dimensions are small compared to the wavelength of the frequencies that excite them. A mathematical relation exists between the voltage at their terminals and the current through the lumped element. These relations can be extrapolated from the Maxwell equations (Table 2.1) and the medium constitution (Table 2.2). Each type of lumped element has its own graphical representation. Then, when several lumped elements connected constitute an electrical diagram, the interconnection point of their terminals is called a node. Table 2.3 shows the most common lumped elements.

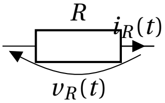
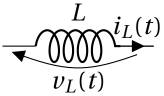
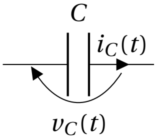
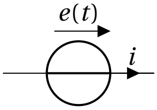
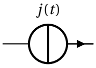
Lumped element	**Symbol**	**Time domain equations**
Resistance		$v_R(t) = Ri_R(t)$
Inductance		$v_L(t) = L\frac{di_L}{dt}$
Capacitance		$i_C(t) = C\frac{dv_C}{dt}$
Voltage source		$e(t)$
Current source		$j(t)$

Table 2.3: Lumped elements characteristics

Finally, another important lumped element is the ground (Figure 2.2). It symbolizes the 0V reference. An electrical network can have several ground symbols. Moreover, they are all connected together.

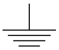


Figure 2.2: Ground symbol

2.4. KIRCHHOFF'S LAWS

The Kirchhoff current and voltage laws were formulated by G. Kirchhoff in 1845 [8]. They govern the interaction between voltages and currents of an electrical circuit. To satisfy the Maxwell equations (Table 2.1), the wavelength of current and voltage need to be much bigger than the size of the different lumped elements. As a consequence, the electrical charges are transported almost instantaneously through the various components, according to these laws.

2.4.1. KIRCHHOFF'S CURRENT LAW

Kirchhoff's current law states that the algebraic sum of the currents entering a node is equal to the algebraic sum of the currents leaving the node, as shown in Eq. 2.2.

$$\sum_{currents\ entering} i_e = \sum_{currents\ leaving} i_l \tag{2.2}$$

This law forbids the series connection of current sources. For example, when two or more current sources, which have different values of current injection, are in series, they do not satisfy Kirchhoff's current law because the current sum at each node is not zero.

2.4.2. KIRCHHOFF'S VOLTAGE LAW

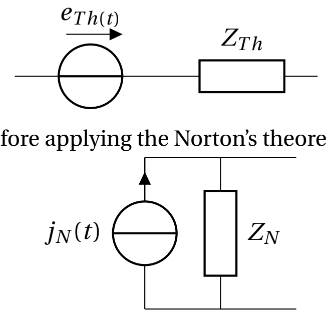
Kirchhoff's voltage law states that the algebraic sum of the voltage drop of all lumped elements of a closed loop is equal to 0V, as shown in Eq. 2.3.

$$\sum_{number\ of\ lumped\ elements\ of\ a\ close\ loop} u_j = 0 \tag{2.3}$$

This law disallows the parallel connection of two voltage sources. For example, when two or more voltage sources, which impose several potentials, are in parallel, they do not satisfy Kirchhoff's voltqge law because the sum of the voltage of a loop is not zero.

2.5. NORTON'S THEOREM

E.L. Norton formulated this theorem in 1926 [9]. It transforms an arbitrary source, composed of lumped elements, into a non-ideal current source. For example, the Norton theorem is applied to Figure 2.3a, which represents a two-port terminal, non-ideal voltage source.



(a) Non-ideal voltgae source (before applying the Norton's theorem)

(b) Non-ideal current source (After applying the Norton's theorem)

Figure 2.3: Application of the Norton's Transformation

Then after the transformation, Figure 2.3b is found where $j_N(t) = \frac{e_{Th(t)}}{Z_{Th}}$ and $Z_N = Z_{Th}$. Moreover, the dual of this theorem is called the Thevenin theorem [10].

2.6. GRAPH THEORY

Graph theory is a useful tool for computations with the Kirchhoff laws [11, 12]. First, an electrical circuit is composed of nodes and lumped elements. Second, a graph consists of nodes (n) and branches (b). In fact, an electrical node corresponds to a graph node, and a lumped element to a branch. In a graph, the branches are oriented, and their orientations show an arbitrary current flow for the computation of the Kirchhoff laws.

Then, the incidence matrix ($K_{CL} \in \mathbb{Z}^{b\times n-1}$) can be obtained by applying Kirchhoff's current law at each node of a graph, except to the ground node, usually denoted node 0. Thus, Kirchhoff's current law can be written as:

$$K_{CL} i = 0, \tag{2.4}$$

where the vector $i = [i_1 \; i_2 \; ... \; i_b]^T$ corresponds to the current through each branch and

$$K_{CL_{kj}} = \begin{cases} +1 & \text{if branch } i_j \text{ is connected and oriented to the node } k \\ -1 & \text{if branch } i_j \text{ is connected and not oriented to the node } k \\ 0 & \text{if the branch } i_j \text{ is not connected to the node } k \end{cases} .$$

In addition, a graph can be denoted, such as $G(V,E)$, where the vertices V represent the node, and the edges E represent the node connection of lumped elements. On $G(V,E)$, a spanning tree T is constructed. This tree includes, by definition, all the elements of the graph. The set of edges of $G(V,E)$ not belonging to T defines the co-tree denoted by L.

For example, the following electrical network is studied for the notions of edges, vertices, tree, and co-tree.

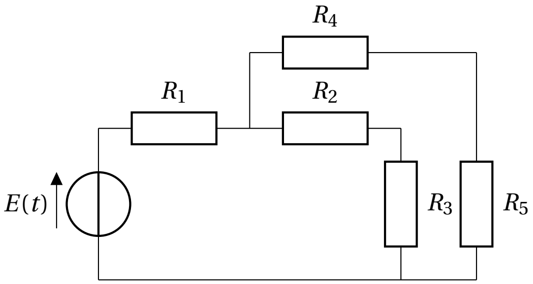


Figure 2.4: Electrical network example

The electrical network, shown in Figure 2.4, is composed of six lumped elements and five nodes. The graph representation of the previous electrical network is presented in Figure 2.5.

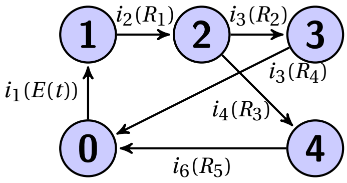


Figure 2.5: Graph of the Figure 2.4

From Figure 2.5, the mathematical formulation of V and E are:

$$V = \{0,\ 1,\ 2,\ 3,\ 4\}, \tag{2.5}$$

$$E = \{(0,\ 1),\ (1,\ 2),\ (2,\ 3),\ (2,\ 4),\ (3,\ 0),\ (4,\ 0)\}. \tag{2.6}$$

In addition, another vector $Name_E$ can be defined by naming the edges of the graph from the lumped representation, such as:

$$Name_E = \{E(t),\ R_1,\ R_2,\ R_3,\ R_4,\ R_5\}. \tag{2.7}$$

In addition, the current flows from the first node to the second node of each edge. From this notation, the incidence matrix of Figure 2.5 can be computed, and is:

$$K_{CL} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \tag{2.8}$$

Now, the spanning tree is obtained by splitting all the loops of the graph. Then, the co-tree is obtained by the removing eges of the graph for obtaining the spanning tree. The spanning tree and co-tree of the graph, shown in Figure 2.6, is given in the following figure.

Using the same methodology as the graph, the spanning tree $T(V_T, E_T)$ of Figure 2.6a can be defined by:

$$V_T = \{0,\ 1,\ 2,\ 3,\ 4\},0 \tag{2.9}$$

$$E_T = \{(0,\ 1),\ (1,\ 2),\ (2,\ 3),\ (4,\ 0)\}, \tag{2.10}$$

$$Name_{E_T} = \{E(t),\ R_1,\ R_2,\ R_5\}. \tag{2.11}$$

Then, the co-tree definition is:

$$V_L = \{0,\ 1,\ 2,\ 3,\ 4\}, \tag{2.12}$$

$$E_L = \{(2,\ 4),\ (3,0)\}, \tag{2.13}$$

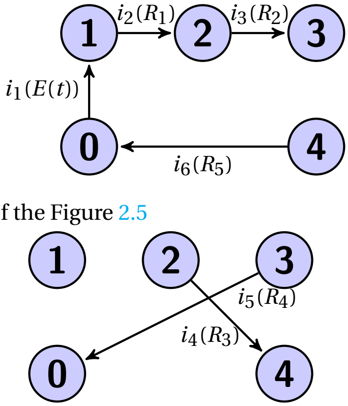
$$Name_{E_L} = \{R_3,\ R_4\}. \tag{2.14}$$

Finally, a graph can be defined as:

$$V = V_T\ \cup\ V_L, \tag{2.15}$$

$$E = E_T\ \cup\ E_L, \tag{2.16}$$

$$Name_E = Name_{E_T}\ \cup\ Name_{E_L}. \tag{2.17}$$



(a) Spanning tree of the graph of the Figure 2.5

(b) Co-tree of the graph of the Figure 2.5

Figure 2.6: Tree and co-tree of the graph of the Figure 2.5

2.7. Modeling methods

This section offers a brief introduction to the three most important modeling methods:

- The nodal analysis method;
- The modified nodal analysis method;
- Cut-set method.

2.7.1. Nodal analysis method

The nodal analysis method was created by H.W. Dommel, who started to work on it at the Munich Institute of Technology (Germany) in the 1960s [3]. He continued his work at BPA (Bonneville Power Administration, USA) with Scott-Meyer, his collaborator in those days on the Electromagnetic Transients Program (EMTP). They published the source code in the public domain - which explains the popularity of this method.

Mathematical expression

A power system composed of n nodes and b branches (lumped elements) is under consideration. Then, the mathematical expression at each time-step of the time domain simulation of the nodal analysis method is:

$$Yu = i, \tag{2.18}$$

where the matrix $Y \in \mathbb{R}^{n-1\times n-1}$ is the admittance matrix, $i \in \mathbb{R}^{n-1}$ represents the excitation of the power system under consideration at each node, and $u \in \mathbb{R}^{n-1}$ represents the voltage between the nodes and the ground (from u_{10} to $u_{(n-1)0}$ where $u_{10} = u_1 - u_0$). The vector u is called the nodal voltage vector. Moreover, the admittance matrix needs to be updated when the power system's topology changes.

According to Eq. 2.18, the admittance matrix represents the topology of the power system under consideration. The vector j describes the current to inject at each node. For this reason, all voltage sources have to be converted into non-ideal current sources by means of Norton's theorem (Section 2.5). As a consequence, the vector u needs to be computed by solving the linear system of equations (Section 3.4.1). Two examples will demonstrate the simplicity and drawbacks of the nodal analysis method. The first example is an arbitrary sample electrical diagram composed only of a current source, a voltage source, and three resistors. The second example shows the mathematical expression of the nodal analysis method applied to an RLC circuit.

Example 1
The first example consists of the following electrical circuit [13].

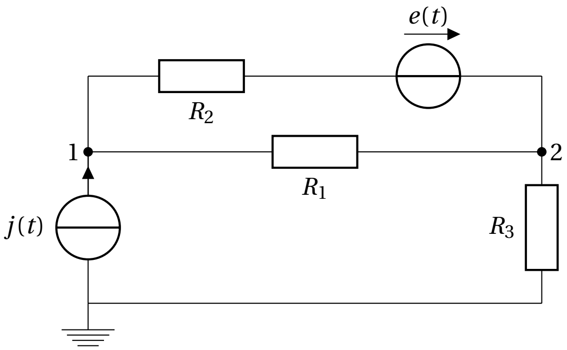


Figure 2.7: Electrical diagram 1

In Figure 2.7, a voltage source $e(t)$ is used. As explained previously in Section 2.5, the non-ideal voltage source ($e(t)$ and R_2 in series) needs be replaced by its non-ideal current source equivalent ($\frac{e(t)}{R_2}$ and R_2 in parallel) according to Norton's theorem. Then, the transformed electrical circuit is:

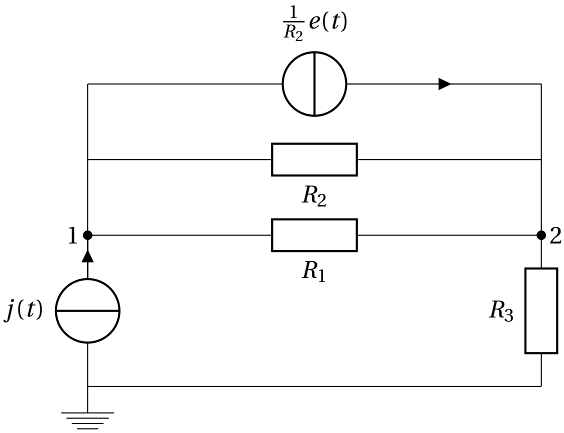


Figure 2.8: Norton's theorem transformation of Figure 2.7

The sample electrical diagram of Figure 2.8 is only composed of resistances and current sources. Thus, Kirchhoff's current law is applied at each node. For node number one, Kirchhoff's current law reads as:

$$-j(t)+\frac{1}{R_1}(u_{10}-u_{20})+\frac{1}{R_2}(u_{10}-u_{20})+\frac{1}{R_2}e(t)=0. \tag{2.19}$$

A similar equation for node number two gives:

$$\frac{1}{R_3}u_{20}+\frac{1}{R_1}(u_{20}-u_{10})+G_2(u_{20}-u_{10})-\frac{1}{R_2}e(t)=0. \tag{2.20}$$

Equations 2.19 and 2.20 can be combined in the matrix form of the Eq. 2.18, such as:

$$\begin{bmatrix} \frac{1}{R_1}+\frac{1}{R_2} & -\frac{1}{R_1}-\frac{1}{R_2} \\ -\frac{1}{R_1}-\frac{1}{R_2} & \frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3} \end{bmatrix}\begin{bmatrix} u_{10} \\ u_{20} \end{bmatrix}=\begin{bmatrix} j(t)-\frac{1}{R_2}e(t) \\ \frac{1}{R_2}e(t) \end{bmatrix} \tag{2.21}$$

Example 2

Now, a sample RLC network is studied (Figure 2.9) [14].

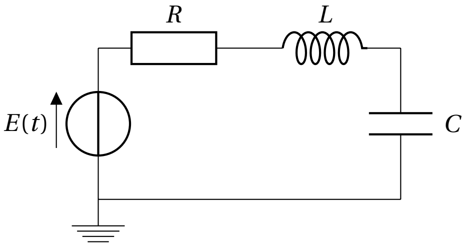


Figure 2.9: Sample RLC network

In fact, the electrical diagram of Figure 2.9 gives a problem because inductances and capacitances impose differential equations (Table 2.3), and so, the time integration method needs to be used. Then, the Trapezoidal rule has been chosen. This integration method is used to convert differential equations into algebraic equations (Section 3.3.2). Therefore, inductances and capacitances can be represented by a control current source with a parallel resistance, as shown in Figure 2.10, according to Eq. 3.19 to 3.22.

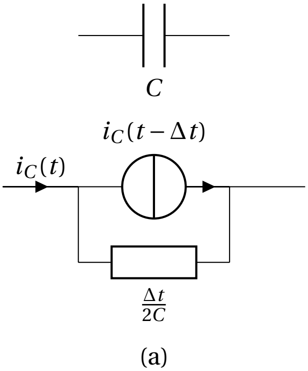


Figure 2.10: Capacitance (a) and inductance (b) transformation

According to Figure 2.10, inductance and capacitance models are related to the time-step. Furthermore, it is necessary to constantly compute the current through all inductances and capacitances (called historical current) to initialize their current sources for the next time-step [15].

Now, the first step is to replace the voltage source, the inductance, and the capacitance by their equivalent models, as shown in Figure 2.11.

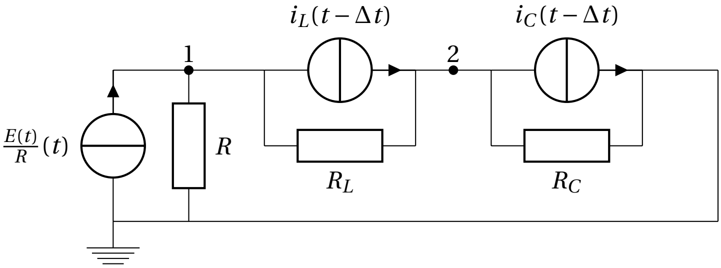


Figure 2.11: Sample RLC network after transformation

Then, the following set of equations is found after applying the nodal analysis method to Figure 2.11.

$$\begin{bmatrix} \frac{1}{R}+\frac{\Delta t}{2L} & -\frac{\Delta t}{2L} \\ -\frac{\Delta t}{2L} & \frac{\Delta t}{2L}+\frac{2C}{\Delta t} \end{bmatrix} \begin{bmatrix} u_{10} \\ u_{20} \end{bmatrix} = \begin{bmatrix} \frac{E(t)}{R}-i_L(t-\Delta t) \\ -i_c(t-\Delta t)+i_L(t-\Delta t) \end{bmatrix} \tag{2.22}$$

GRAPH THEORY APPLIED TO THE NODAL ANALYSIS METHOD

Graph theory is a useful tool to obtain the mathematical expression of the nodal analysis method. As shown in Section 2.6, it is possible to write Kirchhoff's current law in a matrix form. However, a branch represents the parallel associations of a current source and resistance. According to Figure 2.11, Figure 2.12 shows its graph only where Norton's theorem has been employed.

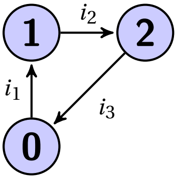


Figure 2.12: Graph of the Figure 2.11

In Figure 2.12, branch i_1 represents the current source and its parallel resistance. Then, branch i_2 describes the current source of the inductance and its parallel resistance. Finally, branch i_3 represents the current source of the capacitance and its parallel resistance. Thus, Kirchhoff's current law is applied to Figure 2.12, and it is defined as:

$$\begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = 0, \tag{2.23}$$

where, the matrix $K_{CL} \in \mathbb{R}^{n\times b}$ is:

$$K_{CL} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}. \tag{2.24}$$

Moreover, to obtain Eq. 2.18, it is necessary to compute matrix G and vector i_{source} such as:

$$K_{CL} G K_{CL}^T u = -K_{CL} i_{source} \tag{2.25}$$

where

$$G_{jj} = \begin{cases} G_j \text{ if branch } i_j \text{ represents a resistance} \\ 0 \text{ else} \end{cases},$$

and

$$i_{source_k} = \begin{cases} j_k \text{ if the branch } i_k \text{ represents a current source} \\ 0 \text{ else} \end{cases}.$$

Matrix G and vector i_{source} corresponding to Figure 2.11 are:

- $G = \text{diag}(\begin{bmatrix} \frac{1}{R} & \frac{\Delta t}{2L} & \frac{2C}{\Delta t} \end{bmatrix});$
- $i_{source} = \begin{bmatrix} \frac{E(t)}{R} & i_L(t-\Delta t) & i_c(t-\Delta t) \end{bmatrix}^T.$

From this information, Eq. 2.22 is found.

2.7.2. MODIFIED NODAL ANALYSIS METHOD

The modified nodal analysis method was developed by C.W. Ho, A.E. Ruehli, and P.A. Brennan [4]. This approach is mainly applied to simulate transient stabilities of power systems or power electronics, and it is based on the Kirchhoff's laws. The mathematical expression of this method can be obtained by using graph theory as well. Unlike the nodal analysis method, this formulation allows the use of a time-stepping strategy with respect to the nodal analysis. Consequently, computation time may decrease. The modified nodal analysis method can have at least two forms, index 1 and 2 (Section 3.2). To simulate the time domain solution for both indexes, an implicit numerical integration method can be used (Section 3.2).

MATHEMATICAL EXPRESSION

A power system composed of n nodes, b branches, n_e voltage sources, n_L inductances, and n_C capacitances is under consideration. Thus, the mathematical expression of the modified nodal analysis method is:

$$M\frac{dx(t)}{dt} + Ax(t) + g(t) = 0, \tag{2.26}$$

where matrix $M \in \mathbb{R}^{(n-1+n_L+n_e)\times(n-1+n_L+n_e)}$ is the mass matrix. Matrices M and $A \in \mathbb{R}^{(n-1+n_L+n_e)\times(n-1+n_L+n_e)}$ are related to the topology of the power system. Vector $x \in \mathbb{R}^{n-1+n_L+n_e}$ contains algebraic and differential variables. Also, vector $g(t) \in \mathbb{R}^{n-1+n_L+n_e}$ represents the excitation of the power system. Moreover after a change of topology, it is necessary to update matrix A.

According to Figure 2.9, five equations can be found. The first three equations are related to Kirchhoff's current law (Eq. 2.27 to 2.29). Then, the fourth equation is related to Kirchhoff's voltage law (Eq. 2.30). The last equation imposes the potential across u_{10} due to the voltage source because it is located between node 1 and the ground (Eq. 2.31)[14].

$$0 = i_E(t) - \frac{u_{10} - u_{20}}{R} \tag{2.27}$$

$$0 = \frac{u_{10} - u_{20}}{R} - i_L \tag{2.28}$$

$$0 = i_L - i_C = i_L - C\frac{du_{30}}{dt} \tag{2.29}$$

$$0 = E(t) - (u_{10} - u_{20}) - v_L - v_C = u_{20} - L\frac{di_L}{dt} - u_{30} \tag{2.30}$$

$$0 = E(t) - u_{10} \tag{2.31}$$

Thus, it is possible to write the previous equations into a matrix form, such as:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -C & 0 & 0 \\ 0 & 0 & 0 & -L & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{u_{10}} \\ \dot{u_{20}} \\ \dot{u_{30}} \\ \dot{i_L} \\ \dot{i_E} \end{bmatrix} + \begin{bmatrix} -\frac{1}{R} & \frac{1}{R} & 0 & 0 & 1 \\ \frac{1}{R} & -\frac{1}{R} & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_{10} \\ u_{20} \\ u_{30} \\ i_L \\ i_E \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ E(t) \end{bmatrix} = 0 \tag{2.32}$$

Graph theory applied to the modified nodal analysis method

As for the nodal analysis methods, a matrix form can be obtained from the graph theory [16]. Firstly, matrices L_{list} and C_{list}, which are necessary for the computation of the modified nodal analysis set of equations from the graph theory, are developed - such as matrix G of the nodal analysis. Then, vector e_{source} is computed according to vector i_{source} definition of the nodal analysis too. Finally, the modified nodal analysis set of equations can be divided into three subsets of equations. The first subset is related to Kirchhoff's current law. The second subset is linked to Kirchhoff's voltage law, and the last subset is tied to the voltage source node relation. Then, Figure 2.13 shows the graph used for the modified nodal analysis of the electrical diagram of Figure 2.9.

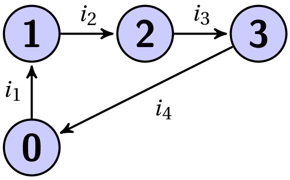


Figure 2.13: Graph of the Figure 2.9

The first subset of equations can be written as:

$$K_{CL}C_{list}K_{CL}^T\dot{u} + K_{CL}GK_{CL}^T u + (K_{CL}L_{list})i_L + (K_{CL}E_{list})i_e + K_{CL}i_{source}(t) = 0, \tag{2.33}$$

where vector $u \in \mathbb{R}^{n-1}$ is composed of voltages u_{10} to $u_{(n-1)0}$, vector $i_L \in \mathbb{R}^{n-1}$ represents the current through the different inductances, and vector $i_e \in \mathbb{R}^{n-1}$ represents the current through the different voltage sources. The second subset of equations is:

$$-L_{list}\dot{i_L} + (K_{CL}L_{list})^T u = 0. \tag{2.34}$$

Finally, the last subset of equations is:

$$(K_{CL}E_{list})^T u - K_{CL}e_{source}(t) = 0. \tag{2.35}$$

The previous methodology is now applied to Figure 2.9 to find the differential algebraic equations. As a consequence, the different matrices and vectors for the computation of the system of equations are:

- $K_{CL} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$;
- $G = \text{diag}\begin{bmatrix} 0 & \frac{1}{R} & 0 & 0 \end{bmatrix}$;
- $L_{list} = \text{diag}\begin{bmatrix} 0 & 0 & L & 0 \end{bmatrix}$;
- $C_{list} = \text{diag}\begin{bmatrix} 0 & 0 & 0 & C \end{bmatrix}$;
- $E_{list} = \text{diag}\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$;
- $e_{source}(t) = \begin{bmatrix} E(t) & 0 & 0 & 0 \end{bmatrix}^T$;
- $i_{source}(t) == \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$.

When all matrices and vectors found previously are put together in a matrix form, Eq. 2.32 is found. In fact, it would be the same only after performing the following transformation. If all elements of the i^{th} row of Eq. 2.26 are equal to 0, the i^{th} column and i^{th} row of the matrices M and A can be deleted, along with the i^{th} row of the vector $g(t)$.

2.7.3. CUT-SET METHOD

The cut-set method [17] was popularized by E.S. Kuh and R.A. Rohrer in 1965 [5]. This method computes the ordinary differential equations of a power system, is based only on graph theory, and it allows for time-stepping algorithms without restriction (Section 3.3). Nevertheless, this approach is the most complex and computationally expensive modeling method to arrive at the mathematical expression.

MATHEMATICAL EXPRESSION

A power system composed of n_s sources, n_L inductances, and n_C capacitances is under consideration. Thus, the mathematical expression of the cut-set method is:

$$\frac{dx(t)}{dt} = \dot{x} = Ax + Bg(t), \tag{2.36}$$

where matrix $A \in \mathbb{R}^{(n_L+n_C)\times(n_L+n_C)}$ is the state matrix, matrix $B \in \mathbb{R}^{(n_L+n_C)\times(n_s)}$ is the input matrix, vector $x \in \mathbb{R}^{n_L+n_C}$ is the state vector, and vector $g(t) \in \mathbb{R}^{n_s)}$ is the time input vector.

RLC CIRCUIT EXAMPLE

By applying the methodology developed in [5] to Figure 2.9, the following ordinary differential equations are found:

$$\begin{bmatrix} \dot{u}_c \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & -\frac{1}{RL} \end{bmatrix} \begin{bmatrix} u_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} e(t) \tag{2.37}$$

Equation 2.37 corresponds well to the ordinary differential equations of Figure 2.9, according to [14].

GRAPH THEORY

For example, an electrical network is represented by an interconnection of resistors, inductors, capacitors, voltage sources, and current sources. The difference from the other methods is that the graph is decomposed according to the spanning tree and co-tree, as developed in Section 2.6. The tree/co-tree decomposition of $G(V,E)$ is made such that capacitors and voltage sources are in the tree while the inductors and current sources are in the co-tree. Resistors can appear in both the tree and co-tree. Schematically, the decomposition is made as:

$$\begin{aligned} T &= \{\text{voltage sources}\} \cup \{\text{capacitors}\} \cup \{\text{ resistors}\} \qquad (2.38) \\ L &= \{\text{ resistors}\} \cup \{\text{inductors}\} \cup \{\text{current sources}\}\,. \qquad (2.39) \end{aligned}$$

For brevity, the graph $G(V,E)$ allows a tree/co-tree decomposition, as prescribed by Eq. 2.38 and Eq. 2.39. Vectors $i(t)$ and $v(t)$ are the time-dependent current and voltage variables associated to the edges E. They can be partitioned into sub-vectors related to the tree/co-tree decomposition and the variable associated to the edges E, such as:

$$i = \begin{bmatrix} i_t \\ i_\ell \end{bmatrix} = \begin{bmatrix} i_e \\ i_{C_t} \\ i_{R_t} \\ i_{R_\ell} \\ i_{L_\ell} \\ i_J \end{bmatrix}, \qquad (2.40)$$

and

$$v = \begin{bmatrix} v_t \\ v_\ell \end{bmatrix} = \begin{bmatrix} v_e \\ v_{C_t} \\ v_{R_t} \\ v_{R_\ell} \\ v_{L_\ell} \\ v_J \end{bmatrix}. \qquad (2.41)$$

Given the spanning tree T, a maximal number of linearly independent fundamental loops and cut-sets can be constructed. A fundamental loop consists of a subset of tree edges and a single co-tree edge. A fundamental cut-set consists of the co-tree edges and a single tree edge. Let K_{VL} denote the incidence matrix between the edges and the fundamental cut-sets, and let $K_{VL} = [K_{VL_t}\ K_{VL_\ell}] = [K_{VL_t}\ I]$ be a tree/co-tree partitioning of the columns of K_{VL}. A maximal number of linearly independent Kirchhoff's voltage laws can then be expressed as $K_{VL}v = \mathbf{0}$, or equivalently, as $K_{VL_t}v_t = -v_\ell$. Let Q denote the incidence matrix between the edges and the fundamental cut-sets, and let $Q = [Q_t\ Q_\ell] = [I\ Q_\ell]$ be a tree/co-tree partitioning of the columns of Q. A maximal number of linearly independent Kirchhoff current laws can then be expressed as $Q\ i = \mathbf{0}$, or equivalently, $i_t = -Q_\ell i_\ell(t)$. The columns of Q and K_{VL} are orthogonal to each other, and therefore $Q^T K_{VL} = [K_{VLt}\ I]^T [I\ Q_\ell] = 0$, from which it follows that $K_{VL_t} = -Q_\ell^T$. This equality can be employed to write a set of linearly independent Kirchhoff's voltage and current laws, as

$$v_\ell(t) = Q_\ell^T v_t \text{ and } i_t = -Q_\ell i_\ell(t)\,, \text{ respectively.} \qquad (2.42)$$

By decomposing the tree and co-tree vectors further into their respective components, it is possible to write:

$$\begin{pmatrix} v_{R_\ell} \\ v_{L_\ell} \\ v_j \end{pmatrix} = Q_\ell^T \begin{pmatrix} v_e \\ v_{C_t} \\ v_{R_t} \end{pmatrix} \text{ and } \begin{pmatrix} i_e \\ i_{C_t} \\ i_{R_t} \end{pmatrix} = -Q_\ell \begin{pmatrix} i_{R_\ell} \\ i_{L_\ell} \\ i_J \end{pmatrix} \text{respectively.} \tag{2.43}$$

The first component on the left hand side of both equations can be eliminated using the constitutive equations Table 2.3, which is re-formulated as:

$$v_{L_\ell} = L_\ell \frac{d i_{L_\ell}(t)}{dt} = L_\ell \dot{i_{L_\ell}} \text{ and } i_{C_t} = C_\ell \frac{d v_{C_t}(t)}{dt} = C_\ell \dot{v_{C_t}}, \tag{2.44}$$

where L_ℓ and C_t denotes the inductance of the ℓ-th branch and capacitance of the t-th branch, respectively. The second component on the right hand side of both equations can be rewritten using Ohm's Law (Table 2.3), which is re-formulated as:

$$v_{R_t} = R_t i_{R_t} \text{ and } i_{R_\ell} = G_\ell v_{R_\ell}, \tag{2.45}$$

With these changes the two vector equations in Eq. 2.43 become:

$$\begin{pmatrix} v_{R_\ell} \\ L_\ell \dot{i_{L_\ell}} \\ v_J \end{pmatrix} = Q_\ell^T \begin{pmatrix} V_e \\ v_{C_t} \\ R_t i_{R_t} \end{pmatrix} \text{ and } \begin{pmatrix} i_e \\ C_t \dot{v_{C_t}} \\ i_{R_t} \end{pmatrix} = -Q_\ell \begin{pmatrix} G_\ell v_{R_\ell} \\ i_{L_\ell} \\ v_{R_\ell} \end{pmatrix} \text{respectively.} \tag{2.46}$$

A system of differential-algebraic equations for the currents through the inductors i_{L_ℓ} and the voltage across the capacitors v_{C_t} is thus obtained. By eliminating the algebraic equations from Eq. 2.46, this system can be reduced to a system of ordinary differential equations. The voltages associated with the co-tree resistance v_{R_ℓ} and the currents associated with the tree resistances $v i_{R_t}(t)$, for instance, can be written in terms of the state variables i_{L_ℓ} and v_{C_t} and source term variables by using the second set of equations of (2.46) and subsequently eliminated. The system of ordinary differential equations that describes the evolution of the state variables as:

$$x(t) = \begin{pmatrix} v_{C_t}(t) \\ i_{L_\ell}(t) \end{pmatrix} \tag{2.47}$$

can then be written as:

$$\dot{x(t)} = f(x(t), t) = Ax(t) + Bg(t), \tag{2.48}$$

where matrix $A \in \mathbb{R}^{(n_L+n_C)\times(n_L+n_C)}$, matrix $B \in \mathbb{R}^{(n_L+n_C)\times(n_e+n_i)}$ and these matrices are related to the topology of the power system. Vector $x \in \mathbb{R}^{n_L+n_C}$ contains only differential variables, and vector $g(t) \in \mathbb{R}^{n_e+n_i}$ represents the excitation of the power system.

2.8. POWER SYSTEM ELEMENTS

In this section, some power system elements used in this thesis are presented. These elements are:

- Switches;

- Generator model;
- Load model;
- PI-section model;
- Double PI-section model;
- Arc models.

2.8.1. SWITCHES

A switch can interrupt the current between two nodes. It has two positions, open and closed. Ideal switches are not recommended due to Kirchhoff's voltage law, because a switching device may permit the parallel connection of voltage sources of different amplitudes. As a consequence, the following model is used.

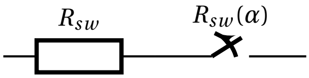


Figure 2.14: Non-ideal switch

The resistance R_{sw} represents the internal resistance and α the switch position. In the case of a closed switch, α is equal to 1. Otherwise, α is equal to 0.

2.8.2. GENERATOR MODEL

Several generator models are available in the literature. For electromagnetic transients, the following model is adequate (1-phase model).

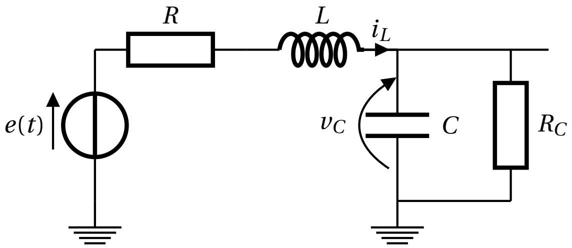


Figure 2.15: Generator diagram

The state space representation of Figure 2.15 is:

$$\dot{x} = A_G x + B_G g(t) = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{R_C C} \end{bmatrix} \begin{bmatrix} i_L \\ v_C \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} e(t). \tag{2.49}$$

where the generator model's parameters are R, R_C, L, C,and $e(t)$ is the voltage waveform of the voltage source. This generator is usually used for the simulation of transient recovery voltage. The resistance R_C is employed for damping voltage v_C.

2.8.3. LOAD MODEL

The load model is represented in the following figure. This model offers the possibility of tuning the load for being resistive, inductive, or capacitive.

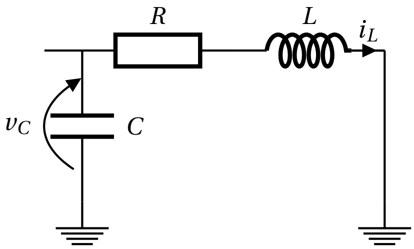


Figure 2.16: Load diagram

The state space representation of Figure 2.16 is:

$$\dot{x} = A_L x \begin{bmatrix} 0 & -\frac{1}{C} \\ \frac{1}{L} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} v_C \\ i_L \end{bmatrix}, \tag{2.50}$$

where R, L and C are the load model's parameters.

2.8.4. PI-SECTION MODEL

PI-section models lines and cables from two different points of the network. The PI-section network symbol is depicted in the following figure.

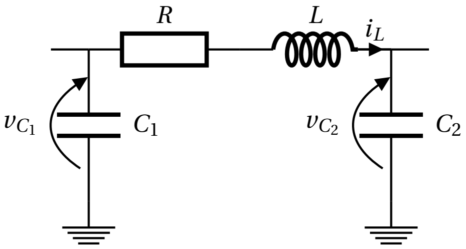


Figure 2.17: PI-section diagram

The state space representation of Figure 2.17 is:

$$\dot{x} = A_{PI} x = \begin{bmatrix} 0 & -\frac{1}{C_1} & 0 \\ \frac{1}{L} & -\frac{R}{L} & -\frac{1}{L} \\ 0 & \frac{1}{C_2} & 0 \end{bmatrix} \begin{bmatrix} v_{C_1} \\ i_L \\ v_{C_2} \end{bmatrix}, \tag{2.51}$$

where R, L, C_1 and C_2 are the PI-section model's parameters.

2.8.5. DOUBLE PI-SECTION MODEL

PII-section models consist of two PI-sections in series. It permits the increase of length in the transmission line, and the accuracy of the model. Its representation is illustrated in the following electrical diagram.

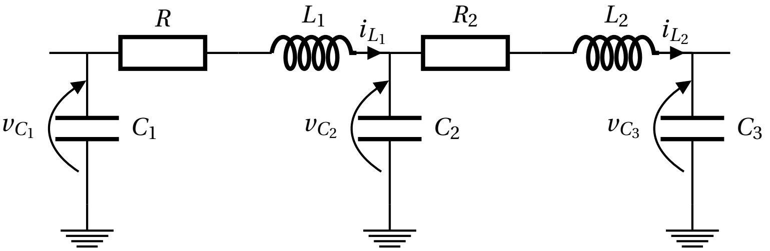


Figure 2.18: Double PI-section diagram

The state space representation of Figure 2.18 is:

$$\dot{x} = A_{PII}x = \begin{bmatrix} 0 & -\frac{1}{C_1} & 0 & 0 & 0 \\ \frac{1}{L_1} & -\frac{R_1}{L_1} & -\frac{1}{L_1} & 0 & 0 \\ 0 & \frac{1}{C_2} & 0 & -\frac{1}{C_2} & 0 \\ 0 & 0 & \frac{1}{L_2} & -\frac{R_2}{L_2} & -\frac{1}{L_2} \\ 0 & 0 & 0 & \frac{1}{C_3} & 0 \end{bmatrix} \begin{bmatrix} v_{C_1} \\ i_{L_1} \\ v_{C_2} \\ i_{L_2} \\ v_{C_3} \end{bmatrix}, \tag{2.52}$$

where R_1, R_2, L_1, L_2, C_1, C_2 and C_3 are the double PI-section model's parameters.

2.8.6. ARC MODELS

An arc model can replace a circuit breaker, a type of switch (Section 2.8.1). It models the transition from a closed to an open position of a switch by using non-linear equations. These equations describe the internal conductivity of the arc. The representation of the arc model is shown in the following figure.

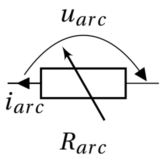


Figure 2.19: Lumped arc model

The resistance R_{arc} represents the arc resistance, i_{arc} is the current through the arc, and u_{arc} is the voltage across the arc. In the literature, three classical models are available (Cassie, Mayr, and Habedank models [18]). The following differential equation performs the change of conductivity described by Cassie's model:

$$\frac{dg_c}{dt} = \frac{g_c}{\tau_c}\left(\frac{u_{arc}^2}{U_c^2} - 1\right), \tag{2.53}$$

while the Mayr equation is described by the following equation:

$$\frac{dg_m}{dt} = \frac{g_m}{\tau_m}\left(\frac{u_{arc}i_{arc}}{P_m} - 1\right), \tag{2.54}$$

where g_c and g_m represent the conductivity of Cassie's and Mayr's models, respectively. Also, τ_c and τ_m are the time constant of the Cassie and Mayr model. Furthermore, U_c is the steady-state voltage and P_m is the steady-state arc power loss.

The series connection of the Cassie and Mayr models is considered as the Habedank model. As the result, the resistivity of the electrical arc in this model is given by:

$$R_{arc} = \frac{1}{g_{arc}} = \frac{1}{\frac{1}{g_c} + \frac{1}{g_m}}, \tag{2.55}$$

where i_{arc} is defined as:

$$i_{arc} = \frac{u_{arc}}{R_{arc}} = u_{arc}g_{arc}. \tag{2.56}$$

2.9. CONCLUSION

In this chapter, a brief introduction to circuit theory has been given, and power system elements are given. The traditional modeling methods are the nodal analysis, the modified nodal analysis, and the cut-set method. The nodal analysis is mainly used for the simulation of large-scale power systems. This method is based on the nodal admittance matrix and a numerical integration method (the Trapezoidal rule). The main disadvantage of this approach is the extensive computation time of an overall simulation due to the fixed time-step.

The modified nodal analysis applied to the power system under consideration gives a set of differential algebraic equations. This method can use an adaptive time-stepping integration method. However, it allows only implicit numerical integration methods, and it is complicated to solve differential algebraic equations as shown in Chapter 3.

The cut-set method gives the state-space representation of the power system under consideration. The methodology to compute the set of equations is its primary drawback. In fact, this method is used for the investigation of new components. This approach allows all types of numerical integration methods.

Then, from the various modeling methods, simulations of large-scale power systems become slow for two reasons:

- They impose a type of numerical integration method;
- They are computationally expensive after a switching action.

REFERENCES

[1] J. C. Maxwell, *A dynamical theory of the electromagnetic field,* Philosophical Transactions of the Royal Society of London **155**, 459 (1865).

[2] R. DeCarlo and P.-M. Lin, *Linear Circuit Analysis,* 2^{nd} ed. (OXFORD UNIVERSITY PRESS, New York, 2001).

[3] H. W. Dommel, *Digital computer solution of electromagnetic transients in single-and multiphase networks,* Power Apparatus and Systems, IEEE Transactions on **PAS-88**, 388 (1969).

[4] C. W. Ho, A. E. Ruehli, and P. A. Brennan, *The modified nodal approach to network analysis,* Circuits and Systems, IEEE Transactions on **22**, 504 (1975).

[5] E. Kuh and R. Rohrer, *The state-variable approach to network analysis,* Proceedings of the IEEE **53**, 672 (1965).

[6] P. R. Amestoy, T. A. Davis, and I. S. Duff, *An approximate minimum degree ordering algorithm,* SIAM Journal on Matrix Analysis and Applications **17**, 886 (1996).

[7] L. Chua, C. Deseor, and E. Kuh, *Linear and nonlinear circuits* (McGraw-Hill Book Company, New York, 1987).

[8] G. Kirchhoff, *Ueber den durchgang eines elektrischen stromes durch eine ebene, insbesondere durch eine kreisförmige,* Annalen der Physik **140**, 497 (1845).

[9] E.L.Norton, *Design of finite networks for uniform frequency characteristic.Technical Report TM26-0-1860,* Tech. Rep. (Bell Laboratories, 1926).

[10] L. Thévenin, *Sur un nouveau théoreme d'électricité dynamique [on a new theorem of dynamic electricity],* CR des Séances de l'Académie des Sciences **97**, 159 (1883).

[11] A. Ioinovici, *Computer-Aided Analysis of Active Circuits* (MARCELL DEKKER, inc, New York, 1990).

[12] J. Vlach and K. Singhal, *Computer methods for circuit analysis and design* (Van Nostrand Reinhold, New York, 1983).

[13] R. Smeets, L. Van der Sluis, M. Kapetanovic, D. F. Peelo, and A. Janssen, *Switching in electrical transmission and distribution systems* (John Wiley & Sons city, Chichester, 2014).

[14] L. van der Sluis, *Transients in Power Systems* (John Wiley & Sons, Chichester, 2001).

[15] N. Watson and J. Arrillaga, *Power systems electromagnetic transients simulation,* Vol. 39 (Institution of Electrical Engineers, London, 2003).

[16] S. Schöps, *Multiscale modeling and multirate time-integration of field/circuit coupled problems,* Ph.D. thesis, Universitätsbibliothek Wuppertal, Wuppertal (2011).

[17] T. Bashkow, *The A matrix, new network description,* IRE transactions on circuit Theory **4**, 117 (1957).

[18] M. Kapetanovic, *High voltage circuit breakers* (Sarajevo, 2011).

3

MATHEMATICAL BACKGROUND

3.1. INTRODUCTION

TODAY, mathematics helps to solve a significant amount of problems [1]. This observation is also true with respect to the simulations of power systems. For example, it helps find network equations, differential or not (Chapter 2). It also permits the obtaining of the time domain solution, as will be shown in this chapter.

When differential equations model a power system, two possible types of differential equations exist - differential algebraic and ordinary[2]. For solving them, numerical integration methods are applied, such as the Euler or Runge-Kutta methods [3–5]. Due to the stiffness of power systems [6], implicit integration methods are recommended. This recommendation leads to solving systems of equations during the integration process.

A significant amount of linear and non-linear solvers are available in the literature[7, 8]. One can distinguish direct and iterative linear solvers. In the case of a small linear system of equations, direct solvers are recommended. Otherwise, iterative methods are advised. Iterative methods are typically faster to converge if they are used in combination with a pre-conditioner. As a consequence, a speed-up can be observed by choosing the appropriate solver. Finally, to solve a set of non-linear equations, the Newton-Raphson method is applied.

This chapter is organized as follows: Section 3.2 shows the differences and links between differential algebraic and ordinary differential equations, and Section 3.3 introduces basic integration methods. A brief description on linear and non-linear solvers is given in Section 3.4, and finally, Section 3.5 offers the conclusions.

3.2. ORDINARY DIFFERENTIAL EQUATIONS AND DIFFERENTIAL ALGEBRAIC EQUATIONS

When the modified nodal analysis method (Section 2.7.2) or the cut-set method (Section 2.7.3) models a power system, differential equations are found. It exists in two forms of differential equations:

- Ordinary Differential Equations (ODEs);
- Differential Algebraic Equations (DAEs).

The general notation of an ODE is:

$$\dot{x} = \frac{dx}{dt} = Ax + Bg(t), \tag{3.1}$$

whereas the form of a DEA is:

$$M\dot{x} + Ax + Bg(t) = 0, \tag{3.2}$$

where the matrix M is the mass matrix. The matrices M, A and B are affected by the network topology. The vector x contains only the differential variables in the case of an ODE. In the case of DAEs, the vector x contains algebraic-variables as well. Finally, the vector $g(t)$ excites the power system.

When the matrix M is diagonal, the DAE is an ODE. There are several types of DAEs that are classified by the so-called differentiation index or simply, index. The index represents the number of reductions needed to get an ODE. In the example of the modified nodal analysis method, the index is two for Eq. 2.32. In order to get an ODE, two stages are necessary. The first stage is to write the voltage across the capacitances in terms of current through the inductances and algebraic variables. The second stage is to include the algebraic equations into the differential equation. For example, the demonstration of this methodology is shown below.

- stage 1 The capacitor is placed between the nodes 3 and 0. As a consequence, u_{30} can be renamed u_C. Then, the DAE of order 1 can be obtained after reordering Eq. 2.32 as:

$$\begin{bmatrix} -C & 0 & 0 & 0 & 0 \\ 0 & -L & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{u}_c \\ \dot{i}_L \\ \dot{u}_{10} \\ \dot{u}_{20} \\ \dot{i}_E \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{R} & -\frac{1}{R} & 1 \\ 0 & -1 & -\frac{1}{R} & \frac{1}{R} & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_c \\ i_L \\ u_{10} \\ u_{20} \\ i_E \end{bmatrix}$$
$$+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ E(t) \end{bmatrix} = 0 \tag{3.3}$$

- stage 2 Now, the DAE can be formulated as block matrices of the following form:

$$\begin{bmatrix} M_{11} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + \begin{bmatrix} A_{11} & A_{11} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ g_2(t) \end{bmatrix} = 0, \tag{3.4}$$

Then, the ODE representation is:

$$\begin{bmatrix} M_{11} \end{bmatrix} \begin{bmatrix} \dot{x} \end{bmatrix} + \begin{bmatrix} A_{11} + A_{11}A_{22}^{-1}A_{21} \end{bmatrix} \begin{bmatrix} x \end{bmatrix} + \begin{bmatrix} A_{11}A_{22}^{-}1A_{21}g_2(t) \end{bmatrix} = 0. \tag{3.5}$$

Finally, by putting $M_{11}\dot{x}$ at the right hand side of Eq. 3.5, and then, multiplying by M_{11} on the left side of the equation, the ODE form according to Eq. 3.1, Eq. 2.37 is found.

Another particularity of DAEs is that it is possible to increase their index. Finally, the utilization of DAEs implies the use of a fully implicit numerical integration method [9].

3.3. NUMERICAL INTEGRATION METHODS

In this section, the Euler methods and the Trapezoidal rule are presented first [2, 3], followed by the Runge-Kutta methods [10]. To study the effect of these various numerical integration methods, the following electrical diagram is used (Figure 3.1) [11].

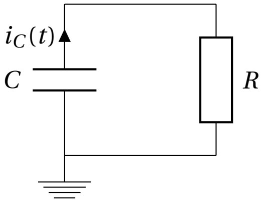


Figure 3.1: RC circuit where $C = 1F$ and $R = 0.5\Omega$

According to Figure 3.1, the differential equation of this electrical circuit is:

$$\dot{v_C} = f(v_C) = -\frac{1}{RC}v_c. \tag{3.6}$$

Eq. 3.6 is defined as an initial value problem. In case of $v_c(t_0 = 0) = 0$, the solution of this equation is $v_c(t) = 0$ for $t \geq 0$. As a consequence, Eq. 3.6 needs to be initialized with a value $v_c(t_0 = 0)$ different of zero. For this problem, v_c is initialized as:

$$v_C(t_0 = 0) = 10V. \tag{3.7}$$

Moreover, Eq. 3.8 presents the analytical solution of Eq. 3.6:

$$v_C(t) = v_C(t_0)e^{-\frac{t}{RC}} = v_C(t_0)e^{-\frac{t}{\tau}} = v_C(t_0)e^{-\lambda t}, \tag{3.8}$$

where the time constant τ or eigenvalue λ of the electrical circuit depends on the values of the resistance and of the capacitance. Thus, Figure 3.2 shows the time evolution of the voltage across the capacitor, according to Eq. 3.8.

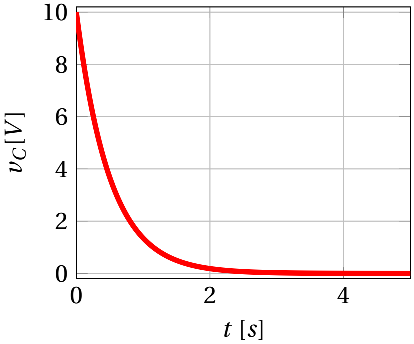


Figure 3.2: Representation of the time domain solution of Eq. 3.6, where the time constant is $\tau = 0.5s$

In general, numerical integration methods are based on the Taylor series of an exponential (Eq. 3.9). Then, the order (p^{th}) of the numerical integration method corresponds to the higher derivative used in this equation. Thus, the local truncation error (error at each time-step) is calculated from the following equation:

$$u_{n+1} = \sum_{i=0}^{p} \frac{\Delta t^i}{i!} A^i u_n + \mathcal{O}(\Delta t)^{p+1} \tag{3.9}$$

where the time step is denoted by Δt, and the time t_n at the time iteration n is defined as $t_0 + n\Delta t$ for a fixed time-step where t_0 is the initial time. Finally, u_{n+1} represents an approximation of the solution of the differential variables at the time t_{n+1} such that $x(t_{n+1}) = u_{n+1} + \mathcal{O}(\Delta t)^{p+1}$, where $\mathcal{O}(\Delta t)^{p+1}$ represents the local truncation error.

3.3.1. EULER METHODS

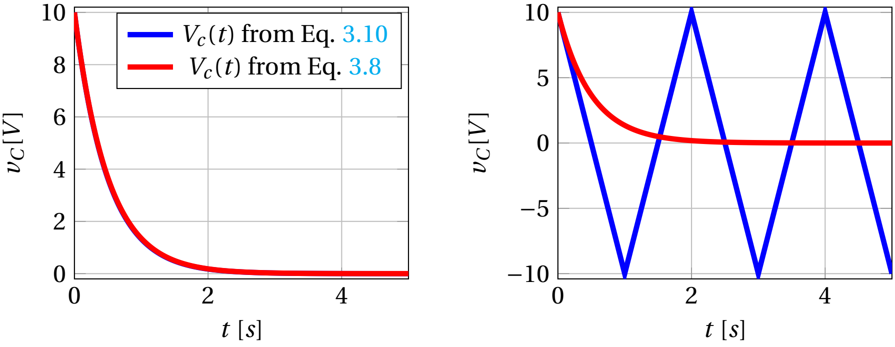
L. Euler published methods to solve ordinary differential equations numerically [3]. The explicit and implicit methods are called Euler Forward and Euler Backward respectively. Furthermore, these two approaches are first order ($p = 1$). The explicit method uses only information from the past to arrive at the solution of the next step. On the contrary, the implicit method also needs the solution to compute it. As a consequence, this method requires more computations per time-step than explicit methods because a set of equations has to be solved at each time-step.

EULER FORWARD METHOD

The numerical formulation of the Euler forward method is:

$$x_{n+1} = x_n + \Delta t f(t_n, x_n). \tag{3.10}$$

Then, this integration method integrates Eq. 3.6 with time-steps of 10ms and 1s. The numerical solution with these different time-steps is shown in Figure 3.3.



(a) $\Delta t = 10ms$

(b) $\Delta t = 1s$

Figure 3.3: Numerical experiments of Euler forward method

Figure 3.3a represents an acceptable approximation of the result according to the analytical solution (Figure 3.2) because the two curves can be superposed. However, the curve of Figure 3.3b shows oscillations because the time-step chosen was just at the stability limit: a larger time-step would have made the time solution divergent. To study the stability, the test equation used is:

$$\dot{x} = \lambda x = f(t, x), \tag{3.11}$$

So, the convergence is obtained only when:

$$|1 + \Delta t \lambda| \leq 1, \tag{3.12}$$

which yields to the stability region of the Euler forward method. For instance, Δt is positive, so λ should be negative to satisfy the following equation:

$$\Delta t \leq \frac{2}{-\lambda}. \tag{3.13}$$

In Figure 3.3b, the requirement of Eq. 3.13 was just at the stability limit because $\lambda = -2$. As a consequence, the numerical integration method was oscillating around the solution. The stability region could then be drawn in the complex plane (Figure 3.4).

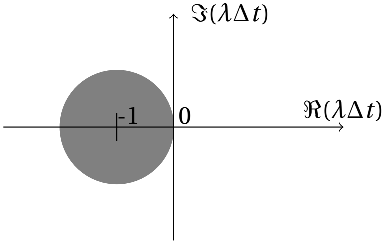


Figure 3.4: Stability region: Stable inside of the circle

For example, if an RLC circuit is studied, this circuit will have two eigenvalues, and they could be complex conjugate with the real part negative. The local truncation error equation (Eq. 3.14) for this numerical method is derived from Eq. 3.9 and 3.10, and it is defined as:

$$e_{n+1} = \frac{\Delta t^2}{2}\frac{d^2x(\tau_n)}{dt^2}, \tag{3.14}$$

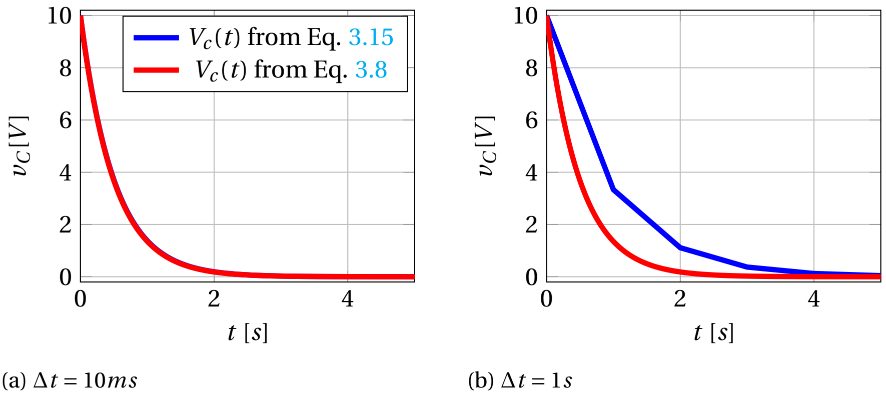
where τ_n is comprised between t_n and $t_n + \Delta t$, and it is arbitrarily chosen.

EULER BACKWARD METHOD

The formulation of Euler backward method is:

$$x_{n+1} = x_n + \Delta t f(\Delta t, x_{n+1}). \tag{3.15}$$

Similar to the Euler forward, time-steps of 10ms, and then, of 1s are applied with this method to Eq. 3.6. Thus, Figure 3.5 shows the time domain representations.



(a) $\Delta t = 10ms$

(b) $\Delta t = 1s$

Figure 3.5: Numerical experiments of Euler backward

According to Figure 3.5, the time domain solution of Eq. 3.6 converges for a time-step of 1s. To understand this difference with regard to the previous method, the region of convergence is studied (Eq. 3.16).

$$\frac{1}{|1-\Delta t\lambda|} \leq 1. \tag{3.16}$$

From Eq. 3.16, the stability region can be plotted as seen in Figure 3.6.

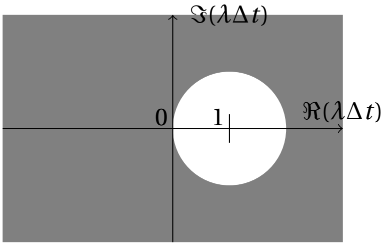


Figure 3.6: Stability region: Stable outside of the circle

In the case of Euler backward, there is convergence when $\Re(\lambda) \le 0$ for any Δt (no mathematical restriction on the time-step).For this reason, although the Euler forward method diverged to a suitable solution with a large time-step, the Euler backward method converged to 0V. However, it requires more computational power than the Euler forward, but can allow more significant time-steps and still converge. The local truncation error equation of this approach is the same as that for Euler forward (Eq. 3.14).

3.3.2. TRAPEZOIDAL RULE

The trapezoidal rule averages the Euler forward and backward methods [2]. The mathematical formulation of this numerical integration method is:

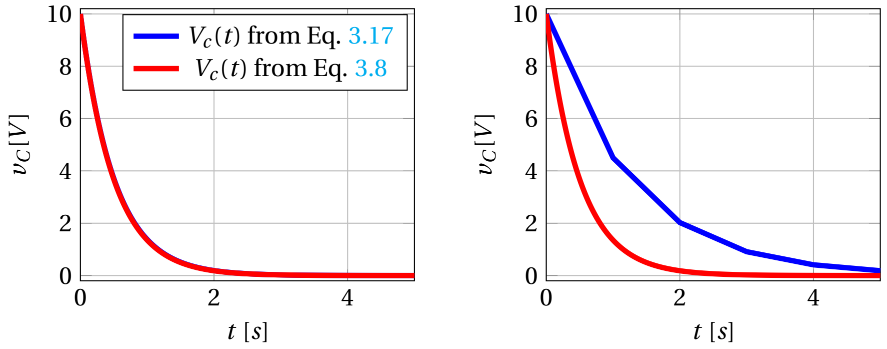
$$x_{n+1} = x_n + \frac{\Delta t}{2}\left(f(t, x_n) + f(t+\Delta t, x_{n+1})\right). \tag{3.17}$$

Also, this method is more accurate than the Euler methods because it is a second order integration method ($p = 2$), and is implicit too. That means that the local truncation error is smaller than with the Euler methods, and it is defined as:

$$e_n = \frac{\Delta t^2}{12}\frac{d^3x(\tau_n)}{dt^3}, \tag{3.18}$$

where τ_n is located between t_n and $t_n + \Delta t$.

Figure 3.7 shows the time domain solution of Eq. 3.6 when this equation is integrated by the aim of the trapezoidal rule, with time-steps of 10ms and then of 1s.



(a) $\Delta t = 10ms$ (b) $\Delta t = 1s$

Figure 3.7: Numerical experiments of the Trapezoidal rule

As expected, the time domain solutions converge for these time-steps. This convergence can also be seen in Figure 3.8, which shows the stability region.

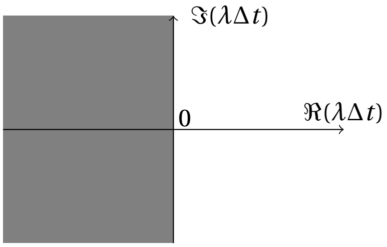


Figure 3.8: Stability region: Stable in the left half plane

According to this figure, the trapezoidal rule allows any time-step when $\Re(\lambda) \leq 0$. Moreover, the trapezoidal rule is used during the nodal analysis method (Section 2.7.1). For example, an inductance L is placed between a node k and m. According to the differential equation of an inductance (Table 2.3), the current injection formulation is:

$$i_{k,m}(t+\Delta t) = \frac{\Delta t}{2L}(v_k(t+\Delta t) - v_m(t+\Delta t)) + i_{k,m}(t), \tag{3.19}$$

where:

$$i_{k,m}(t) = i_{k,m}(t) + \frac{\Delta t}{2L}(v_k(t) - v_m(t)), \tag{3.20}$$

When the inductance L is replaced by a capacitance C, it holds:

$$i_{k,m}(t+\Delta t) = \frac{2C}{\Delta t}(v_k(t+\Delta t) - v_m(t+\Delta t)) + i_{k,m}(t), \tag{3.21}$$

where:

$$i_{k,m}(t) = -i_{k,m}(t) - \frac{2C}{\Delta t}(v_k(t) - v_m(t)), \tag{3.22}$$

where $i_{k,m}$ represents the historical current. The equivalent representations of Eq. 3.19 to 3.22 are shown in Figure 2.10.

3.3.3. RUNGE-KUTTA METHODS

One of the most famous numerical integration method schemes is the Runge-Kutta methods (RK methods). They had been developed a century ago, and are the results of research by C. Runge and M. Kutta. The standard mathematical formulation of this method is:

$$\begin{cases} k_i = f(t + c_i \Delta t, x_n + \Delta t \sum_{j=1}^{s} a_{ij} k_j) & \text{for } 1 \le i \le s \\ x_{n+1} = x_n + \Delta t \sum_{i=1}^{s} b_i k_i) \end{cases}, \tag{3.23}$$

where s represents the number of stages, and a_{ij}, b_i and c_i are the coefficients of the method according to the Butcher tableau (Table 3.1) [12]. The Butcher coefficients a_{ij} and b_i are computed according to the Taylor expansion in Eq. 3.9.

c	a
	b

Table 3.1: Butcher tableau

For example, the following tables represent different the Butcher tableau with three stages. From these representations, explicit and implicit methods can be extracted.

0	0	0	0
c_2	a_{21}	0	0
c_3	a_{31}	a_{32}	0
	b_1	b_2	b_3

(a) Explicit Runge-Kutta

c_1	a_{11}	0	0
c_2	a_{21}	a_{22}	0
c_3	a_{31}	a_{32}	a_{32}
	b_1	b_2	b_3

(b) Diagonally Implicit Runge-Kutta

c_1	a_{11}	a_{12}	a_{13}
c_2	a_{21}	a_{22}	a_{23}
c_3	a_{31}	a_{32}	a_{32}
	b_1	b_2	b_3

(c) Implicit Runge-Kutta

Table 3.2: Rung-Kutta Butcher tableau type

For example, Table 3.2a represents an explicit method or Explicit Runge-Kutta (ERK) because the matrix formed by the entries a_{ij} is strictly lower diagonal. In addition, the vector k_i is only dependent on the other vector k with an index inferior to i. Meanwhile, Table 3.2c represents an implicit method or Implicit Rung-Kutta (IRK) because all vectors k_i are needed to compute all of them. Therefore, a linear system of equations has to be solved with a size proportional to the size of the matrix A and the number of stages of the method. The last category of Runge-Kutta method is represented by Table 3.2b. This method is also an implicit Runge-Kutta method; however, it a Diagonally Implicit Runge-Kutta (DIRK) [13]. The difference with the IRK methods is that there is now s linear system of equations of size of the matrix A. A further reduction in computational cost is provided by single diagonal implicit methods (SDIRK) characterized by a lower diagonal matrix a_{ij} with a constant diagonal. The value of the diagonal will be denoted by γ. Finally, Explicit Single Diagonal Implicit Runge-Kutta method (ESDIRK) is a sub-method of a SDIRK where $a_{11} = 0$ or the vector k_1 is found explicitly.

For example, Table 3.3 shows the Butcher tableau of the Euler methods and the Trapezoidal rule.

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

(a) Euler forward

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

(b) Euler backward

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 0 & 1 \\ \hline & 0.5 & 0.5 \end{array}$$

(c) Trapezoidal rule

Table 3.3: Butcher tableau of basic numerical integration methods

The RK method formulation (Eq. 3.23) is now applied to solve Eq. 3.6, when the Butcher tableau of the trapezoidal rule, which is an ESDIRK method, is used as shown in Eq. 3.24.

$$\begin{cases} k_1 = \lambda u_C \\ (1 - \Delta t \lambda) k_2 = \lambda (u_C) \\ u_{Cn+1} = u_{Cn} + \Delta t \,[0.5 k_1 + 0.5 k_2] \end{cases} . \tag{3.24}$$

According to Eq. 3.24, a linear system of equation has to be solved, and it concords with the implicit definition of the method.

In addition, some Runge-Kutta methods allow the adaptive change of the time-step from the local truncation error. The Butcher representation of adaptive ESDIRK methods is

$$\begin{array}{c|cccc} 0 & 0 & 0 & \cdots & 0 \\ c_2 & a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ c_6 & a_{61} & a_{62} & \cdots & a_{66} \\ \hline & b_1 & b_2 & \cdots & b_6 \\ & \hat{b}_1 & \hat{b}_2 & \cdots & \hat{b}_6 \end{array}$$

Table 3.4: Butcher tableau of adaptive ESDIRK methods

Table 3.4 allows an adaptive time-stepping strategy based on an approximation of the local truncation error e_{n+1} because the $\hat{b}$ coefficients correspond to solution of the order $p-1$. Then, e_{n+1} is defined as:

$$e_{n+1} = \Delta t \sum_{i=1}^{s} \left(b_i - \hat{b}_i\right) k_i \, . \tag{3.25}$$

The new time Δt_{new} is then computed as either an increase or a decrease of the old one Δt_{old}, according to the rule [14]:

$$\Delta t_{new} = \sqrt[p]{\frac{\text{Tol}}{\frac{||e_{n+1}||_\infty}{||u_{n+1}||_\infty}}}\,\theta\Delta t_{old}, \tag{3.26}$$

where $\theta \leq 1$ is a safety factor and Tol is a user-specified absolute tolerance.

In this thesis, the implicit part of the ARK4(3)6L[2]SA-ESDIRK ($p = 4$, $s = 6$) is used [15]. This method is an ESDIRK method, and is depicted as follows.

0	0	0	0	0	0	0
0.50	0.25	0.25	0	0	0	0
0.33	0.14	−0.06	0.25	0	0	0
0.62	0.14	−0.22	0.45	0.25	0	0
0.85	0.10	−0.59	0.81	0.28	0.25	0
1	0.16	0	0.19	0.68	−0.28	0.25
	0.16	0	0.19	0.68	−0.28	0.25
	0.15	0	0.19	0.70	−0.32	0.27

Table 3.5: Butcher tableau of ARK4 method

As a consequence, only one factorization is necessary for each time-step. Finally, according to Eq. 3.23 and Table 3.5, the system of equations for each stage at each time-step is defined as solving for Eq. 3.1:

$$(I - \gamma\Delta t A)k_i = f\left(u_n + \sum_{j=1}^{i-1} a_{ij}\Delta t k_j, t_n + c_i\Delta t\right) \quad \text{for } 2 \leq i \leq 6, \tag{3.27}$$

where the matrix I is the identity matrix.

This ESDIRK method has been chosen for several reasons. This method is A-stable, stiffly accurate, and has a stage order equal to two. The stage order strongly influences the accuracy of the method when applied to stiff problems. Besides, this approach is L-stable, which is beneficial when handling the stiffness of the problem. Finally, adaptive time-stepping strategies can be applied.

3.4. SOLVERS

As developed before, systems of equations have to be solved to obtain the time domain solution. In the literature, two types of systems of equations exist:

- linear;
- non-linear.

Linear systems of equations are easy to solve, and many solvers are available[7, 8]. However, non-linear systems of equations require more computations than linear systems. In general, the Newton-Raphson method is applied in such cases.

3.4.1. LINEAR SOLVERS

There are two types of linear solvers - direct and iterative. Direct methods are mostly used for small sets of equations, and iterative ones are used for larger systems of equations. In this section, the following equation is considered:

$$Ax = b, \tag{3.28}$$

where the matrix A represents the coefficient of the set of equations, the vector b is the input vector, and the vector x is the unknown vector. Also, re-ordering methods can be applied to the matrix A to reduce the number of non-zero elements of the decomposition.

DIRECT METHODS

The first direct solver studied in this thesis is the LU factorization or Gaussian elimination [16]. This method decomposes the matrix A into a lower and upper triangular matrices, L and U. The process to solve the Eq. 3.28 by the LU factorization is:

- Computing L and U such as:
$$LU = A; \tag{3.29}$$
- Solving y from: $Ly = b$ (Forward substitution);
- Solving x from: $Ux = y$ (Backward substitution).

In the case of the Cholesky factorization, the principle is to compute the decomposition of a matrix A which should be symmetric and definite positive, such as:

$$LL^T = A. \tag{3.30}$$

It appears that the Cholesky decomposition is more efficient regarding memory usage and CPU time by a factor of two. However, the forward and backward substitution computation times remain the same as that of the LU factorization. For example, the following matrix A is symmetric definite positive:

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \tag{3.31}$$

Then, after applying the LU factorization to matrix A, the matrices L and U are:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0 & \frac{-2}{3} & 1 \end{bmatrix} \quad U = \begin{bmatrix} 2 & -1 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & 0 & \frac{4}{3} \end{bmatrix}. \tag{3.32}$$

In the case of Cholesky factorization, the matrix L is:

$$L = \begin{bmatrix} \sqrt{2} & 0 & 0 \\ \frac{1}{\sqrt{2}} & \sqrt{\frac{3}{2}} & 0 \\ 0 & \sqrt{\frac{2}{3}} & \frac{2}{\sqrt{3}} \end{bmatrix} \tag{3.33}$$

Another factorization used for time domain simulations of a power system is the KLU factorization [17]. Its principle is to decompose the matrix A in matrices L and U. The difference with the LU factorization is that a re-ordering method is applied to obtain a block triangular form. This action on the matrix, if block triangular blocks are found, permits the use of the Schur complement. As a consequence, it might decrease the computation time if triangular blocks are found during the factorization process. The Cholesky and KLU factorizations are mainly used to solve the set of equations of the nodal analysis method (Section 2.7.1).

ITERATIVE METHODS

A basic iterative method for solving Eq.3.28 is defined as follows [7]:

$$x_{k+1} = x_k + M^{-1} r_k, \tag{3.34}$$

where $M \approx A$ and the vector r is the residual vector, and it is defined as:

$$r_k = b - Ax_k. \tag{3.35}$$

Basic iterative methods efficiently converge to a suitable solution if:

- the matrix M is sufficiently close to the matrix A;
- $M^{-1} r_k$ is easy to compute.

The simplest method is the Richardson iteration method[7]. This approach defines the matrix M as the identity matrix. To approximate the speed of convergence, the spectral radius is defined by:

$$\rho = max|eig\left(I - M^{-1}A\right)|, \tag{3.36}$$

Then, ρ measures the speed of convergence. When $\rho << 1$, the convergence is fast. When $\rho > 1$, there is no convergence.

Advanced iterative methods, such as GMRES or Bi-CGSTAB [18, 19], are based on the Krylov Subspace, defined as:

$$K_i(A, r_0) = \text{span}(r_0, Ar_0, ..., A^{i-1} r_0), \tag{3.37}$$

Thus, it is possible to generate the iterate such as:

$$x_i \in x_0 + K_i(A, r_i). \tag{3.38}$$

To terminate the iteration process, stopping criteria should be applied. These criteria are the maximum number of iterations, or when the residual error becomes insignificant. Furthermore, the choice of the starting point (vector x_0) of the iteration impacts the convergence. For example, when x_0 is close to the solution, then the convergence is usually fast.

Advanced iterative methods use a preconditioner in their algorithm. The preconditioner typically accelerates convergence. For example, the incomplete LU factorization (iLU(ℓ))[7] is used as a preconditioner for them. The variable ℓ defines the level of fill-in of the decomposition ($\ell > 0$ and when $\ell = 0$, there is no fill-in). The principle of this factorization is to obtain a good approximation of the matrix A of the form:

$$A = LU + R = M + R, \tag{3.39}$$

where the matrix R is the residual matrix. Thus, the forward and backward substitutions are used to solve a system of equations as with the LU factorization. However, the computed solution is an approximation.

RE-ORDERING METHODS

It is typically useful to re-order the matrix A to obtain less non-zero elements during the decomposition, and then to be faster during the forward and backward substitutions. In the literature, several re-ordering methods are available - such as the Approximate Minimum Degree (AMD), the Quotient Minimum Degree (QMD), and the Nested Dissection (ND) [20–22]. The idea is to solve the following system of equations:

$$PAP^T Px = Pb, \tag{3.40}$$

where the matrix P is the permutation matrix, and is the result of the re-ordering method. Moreover, the matrix P is defined as

$$PP^T = I. \tag{3.41}$$

As a result, the solution of the system of equation is:

$$x = P^T (PAP^T)^{-1} Pb. \tag{3.42}$$

Reordering methods are useful for reducing the fill-in of the iLU(l) factorization and the number of non-zero elements of any factorization.

3.4.2. NEWTON-RAPHSON METHOD

In the case of a non-linear system of equations, the Newton-Raphson method solves the following problem:

$$f(x) = 0, \tag{3.43}$$

where $f : R \to R$ and $x \in R$. To obtain the solution to this problem, the following equation is iterated as:

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}, \tag{3.44}$$

where the function $f'(x) = \frac{df(x)}{dx}$. For converging, it is necessary to take into account several factors. The primary factor for a fast convergence is to start close to the solution. Another factor is the value of the derivative. When the derivative is large, the convergence is faster. However, if the derivation is null, another starting point is chosen to re-start the iterative process.

For example, the function $f(x)$ is defined by:

$$f(x) = e^x - 4. \tag{3.45}$$

Then, the Newton-Raphson iteration is applied to obtain the solution of Eq. 3.45 when $x_0 = 3$ and $x_0 = 30$. The results after several iterations are shown in Table 3.6.

$x_0 = 3$	$x_0 = 30$
$x_1 = 2.1991$	$x_1 = 29$
$x_2 = 1.6427$	$x_2 = 28$
$x_3 = 1.416$	$x_3 = 27$
$\vdots$	$\vdots$
$x_{10} = 1.3863$	$x_{10} = 20$

Table 3.6: Solution of the iterate for two starting points

The solution of this problem is $x = ln(4) \approx 1.386$. As shown in Table 3.6, the starting point has an influence on the speed of convergence, and so, on the computation time.

To compute the solution of a non-linear system of equations, the following equation is considered:

$$f(x) = 0, \tag{3.46}$$

where $f : R^n \to R^n$ and $x \in R^n$. In order to solve a set of equations, the iterative process of Eq. 3.44 becomes:

$$x_n = x_{n-1} - J(x)^{-1} f(x), \tag{3.47}$$

where the matrix $J(x)$ is defined as:

$$J(x) = J = \frac{\partial f(x)}{\partial x}, \tag{3.48}$$

and it is the Jacobian matrix. The same problems as Eq. 3.45 can be encountered, and so a nearby starting point for the solution is advised. In the case of a singular Jacobian matrix, a new starting point is also recommended. To solve the following linear system of equations:

$$J(x)y = f(x), \tag{3.49}$$

a linear system of equations needs to be solved. Either a direct or iterative solution method can be employed.

In the case of non-linear power systems, the same principle for solving non-linear equations is applied during the time integration method. Moreover, according to the Butcher tableaux of an ESDIRK, for example (Table 3.4), Eq. 3.50 shows the ESDIRK formulation derived from the Newton-Raphson iteration method at each stage if non-linear equations are used:

$$\begin{cases} (I - \Delta t \gamma J)\delta_i^k = k_i^k - f(t + c_i \Delta t, x_n + \Delta t \sum_{j=1}^{i-1} a_{ij} k_j) \\ k_i^{k+1} = k_i^k - \delta_i^k \end{cases} \quad for\ 2 \leq i \leq s, \tag{3.50}$$

where the vector δ_i^k represents the error on the iterate. One of the stopping criteria of this methodology is to stop it when $\|\delta_i^k\|$ becomes smaller than a specified defined tolerance. Also, the Jacobian matrix is usually computed once at each time-step.

3.5. Conclusion

This chapter demonstrates that the mathematical properties of the power system under consideration depend on the modeling method used (Section 2.7) and the user specifications. For these reasons, short introductions to differential equations, numerical integration methods, and solvers are offered.

A power system is usually stiff. Then, an implicit numerical method is advised. In this thesis, the ESDIRK ARK4 is mainly used to integrate the ODE expression of the power system under consideration. Then, the focus of this thesis in Chapters 6 and 7 is about the use of linear solvers - especially when the power system becomes large and its interaction with the numerical integration method is used.

References

[1] P. Tipler, *Physics for Scientists and Engineers* (W. H. Freeman Worth Publishers, New York, 1999).

[2] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, 1st ed. (Society for Industrial and Applied Mathematics, Philadelphia, 1998).

[3] L. Euler, *Institutionum calculi integralis*, 3rd ed., Institutionum calculi integralis (Impensis Academiae Imperialis Scientiarum Petropolitanae, Petropoli, 1824).

[4] C. Runge, *Uber die numerische auflosing von differentialgleichungen,* **46**, 167 (1895).

[5] W. Kutta, *Beitrag zur naherungsweisen integration totaler differentialgleichungen,* **46**, 435 (1901).

[6] L. van der Sluis, *Transients in Power Systems* (John Wiley & Sons, Chichester, 2001).

[7] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. (Society for Industrial and Applied Mathematics, Philadelphia, 2003).

[8] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Vol. 30 (Society for Industrial and Applied Mathematics, New York, 1970).

[9] K. Brenan, S. Campbell, and L. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equation* (North-holland, Amsterdam, 1989).

[10] J. Butcher, *A history of Runge-Kutta methods,* Applied Numerical Mathematics **20**, 247 (1996).

[11] R. DeCarlo and P.-M. Lin, *Linear Circuit Analysis*, 2^{nd} ed. (OXFORD UNIVERSITY PRESS, New York, 2001).

[12] J. Butcher, *On the implementation of implicit Runge-Kutta methods,* BIT Numerical Mathematics **16**, 237 (1976).

[13] C. A. Kennedy and M. H. Carpenter, *Diagonally implicit Runge-Kutta methods for ordinary differential equations. a review,* (2016).

[14] W. Hundsdorfer and J. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations* (Springer, New York, 2003).

[15] C. A. Kennedy and M. H. Carpenter, *Additive Runge-Kutta schemes for convection-diffusion-reaction equations,* Applied Numerical Mathematics **44**, 139 (2003).

[16] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, Vol. 6 (MIT press Cambridge, Cambridge, 2001).

[17] T. A. Davis and E. Palamadai Natarajan, *Algorithm 907: KLU, a direct sparse solver for circuit simulation problems,* ACM Transactions on Mathematical Software (TOMS) **37**, 36 (2010).

[18] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,* SIAM Journal on scientific and statistical computing **7**, 856 (1986).

[19] H. A. Van der Vorst, *Bi-CGSTAB: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems,* SIAM Journal on scientific and Statistical Computing **13**, 631 (1992).

[20] P. R. Amestoy, T. A. Davis, and I. S. Duff, *An approximate minimum degree ordering algorithm,* SIAM Journal on Matrix Analysis and Applications **17**, 886 (1996).

[21] A. George and J. W. Liu, *A fast implementation of the minimum degree algorithm using quotient graphs,* ACM Transactions on Mathematical Software (TOMS) **6**, 337 (1980).

[22] B. Hendrickson and E. Rothberg, *Improving the run time and quality of nested dissection ordering,* SIAM Journal on Scientific Computing **20**, 468 (1998).

4

Switching action network calculation

4.1. Introduction

A Basic power system is composed of generators, transmission lines and loads - as explained in Chapter 1. They all have their mathematical representation independent of each other. Also, switching devices connect these power system components for the configuration or the reliability of the electrical network, as shown in Section 2.8.1.

Switching action can occur - for exmaple - due to a change of topology, a short-circuit, or when switching a transmission line off. Then, fast transients are visible [1, 2]. These transients are composed of high-frequency oscillations until the power system reaches a new steady state value.

The literature on transient calculations is vast [3–5]. A significant amount of computer programs is also available [6–10]. However, their main drawback is the full re-computation of the power system equations or the introduction of non-linearities in the power system equations. In both cases, the overall computation of the time domain simulation could be less expensive if an optimization method is applied to re-compute the set of equations after a switching action.

The idea behind this new approach is to only update the necessary elements of the differential equations after a switching action. Also, this approach should not introduce unnecessary non-linearities [10]. Finally, the power system should be modeled via a state-space representation or Ordinary Differential Equations (ODEs) [11], and the use of non-ideal switches [2, 4, 12].

The choice of a state-space representation has been made for several reasons. The first one is that there are a significant number of methods to obtain the time domain solution of ODEs numerically, as shown in Chapter 3 [11, 13–16]. Secondly, it is possible to have an adaptive time-stepping algorithm to allow a small time-step during transients and a large one during steady-states [16]. Finally, the studies of power system components are usually made in a state-space representation [17–19].

In the literature review of Chapter 2, three main modeling methods are available. The first one is the nodal analysis method [20]. H. Dommel proposed this approach in the late 1960s. This method is appreciated, for example, for the simulation of large-scale power systems [21]. The second method is the modified nodal analysis method [22]. Chung-Wen Ho et *al.* published this method in 1975. Several studies of large-scale power systems are made with this modeling method [23]. The final one is the cut-set method [24]. E. S. Kuh and R. A. Rohrer popularized this in 1969 [25]. However, this modeling method is not popular for the simulation of large-scale power systems.

As discussed previously, the nodal analysis method is mainly applied to simulate large-scale power systems, and it is based on Kirchhoff's current law. It relies on the nodal admittance matrix in combination with a numerical integration method (the Trapezoidal rule) [2, 3]. The main problems of this modeling approach are the fixed time-steps imposed by the numerical integration method, and also the difficulty in introducing non-linear elements. Finally, this approach is not applicable if an ODE representation is used. However, the fast calculation of the admittance matrix is the main advantage of this method - especially if only resistances are taken into account. Besides, the admittance matrix does not vary with the time-step chosen [26].

The modified nodal analysis is based on Kirchhoff's current and voltage laws. This modeling approach then gives a Differential Algebraic Equation (DAE). One of the major computer programs for this method is the Simulation Program with Integrated Circuit Emphasis (SPICE) [8, 27]. It is used for the simulation of electronic devices or sometimes, power electronic devices. In the literature, people studied the switching of capacitors with this modeling approach [28, 29]. Also, these studies are mostly for electronic and signal applications.

The cut-set method is rarely used for the simulation of large-scale power systems. However, this approach gives a state space representation. The reason for this is its methodology to compute the set of equations, from the cut-set of its graph representation of its electrical diagram [4]. After a switching action, the re-computation of the set of equations necessitates the restarting of the cut-set method. Another approach is to consider a switch as a controlled voltage source [10]. However, this approach includes non-necessary non-linear equations.

The ODE representation can be obtained from the DAE formulation, as shown in Chapter 3 [30, 31]. However, after a switching action, this transformation is necessary and that made this method expensive. Converter studies have shown that it is possible to write their state space representation according to the position of the switch [32–34]. However, for large-scale power systems with n switches, it is useless to compute and store n^2 state space representations. In paper [19], the authors studied a methodology for coupling state representation of a power system. They couple the differential equations via the nodal voltages and currents at each component terminal.

As shown before, the studies of power system components are usually made in a state-space representation. Then, the idea is to combine these mathematical descriptions together in a larger state space representation, which represents the power system. Finally, the link to couple those differential equations is to use non-ideal switches to connect the different power system components.

These switching devices have stray capacitances to ground, as it is the case in practice [1]. Then, differential variables are introduced, and they represent the voltage at each terminal of each power system component in the state space representation. After that, the computation of the switching admittance matrix (which was derived from the admittance matrix of a resistive network) helps to calculate the current leaving each component terminal. The global mathematical description can then be easily updated after a switching action via the new switching admittance matrix. The difference with [19] is that this methodology updates the state space representation of only a few equations according to the switch topology for interconnected power system components.

From the review of problems and advantages of the different modeling methods, this approach overcomes the considerable computer time required for building the set of equations needed by the cut-set method. In addition, it is as simple as establishing the nodal admittance matrix after a switching action. At the same time, this approach gives a state space representation. The novelty is the methodology being used to connect different components of a power system by using the switches' position and updating only necessary equations. Also, this method can easily handle non-linear power system elements such as arc models or variable lumped elements.

This chapter is organized as follows. Section 4.2 provides a first methodology to use this new approach. For more complex connection, the methodology is demonstrated in Section 4.3. Section 4.4 shows the application of the block modeling methodology with an analytical example. Then, Section 4.5 shows the utilization of time-dependent and non-linear electrical power systems. Finally, Section 4.6 offers the conclusions.

4.2. BLOCK MODELING METHOD METHODOLOGY

4.2.1. INTRODUCTION

A power system contains various types of components - like generators, transformers, loads, transmission lines, and cables (Section 2.8) [35]. Each of them can be represented by a network of lumped elements (e.g. resistors, inductors, capacitors, etc.) and a set of differential equations can then be derived (Table 2.3).

Other important devices, which are essential for power systems operation and sustainability, are switching devices [2]. In general, each terminal of a power system component is connected to one or more switches. A switch has two terminals, and it can be either open or closed. The open position isolates its terminals. When the switching device is closed, the connection between the two terminals is made. In this chapter, a closed one is represented by a resistor with a small value of resistance. Finally, every switch terminal has a parasitic capacitance.

The idea is now to connect the power system components, called block models, with each other via their terminals. The connection between them is made via an interconnection of switching devices. After a switching action, traditional methods require the re-computation of the complete set of equations. This approach updates only a few numbers of differential equations. To use the method, the block models' conditions are:

- Each block model has its own ODE representation when it is uncoupled from other blocks;
- Each block model (except when the block represents a voltage source or a capacitor) consists of a loop of lumped elements;

- Each block model has at least one terminal;
- Each terminal has a capacitance (real or parasitic from switches) called a link capacitor or a voltage source connected in parallel.

According to these conditions, the voltage at each terminal, called link voltage, is present as a differential or input variable. Then, these voltages assist in the computation of the current flowing between block models by taking into account the switches' position. Finally, the updation of the equations is done from the calculation of the admittance matrix between block models.

4.2.2. MATHEMATICAL FORMULATION

Now, a power system with n block models is considered as an example. The mathematical formulation of the general state space representation is:

$$\frac{dx}{dt} = \dot{x} = Ax + Bg(t). \tag{4.1}$$

Then, it is beneficial to write Eq. 4.1 as:

$$\dot{x} = (\hat{A} + \tilde{A})x + (\hat{B} + \tilde{B})g(t), \tag{4.2}$$

where matrices $\hat{A}$ and $\hat{B}$ are constant during the simulation, and matrices $\tilde{A}$ and $\tilde{B}$ are related to switching actions. As a consequence, some elements of matrices $\tilde{A}$ and $\tilde{B}$ will change after a switching action in the electrical network.

In addition, matrices $\hat{A}$ and $\hat{B}$ are block diagonal. These block diagonal matrices are composed of the various $\hat{A}_*$ and $\hat{B}_*$ block matrices of the different block models that form the electrical network. Also, matrices $\hat{A}$ and $\hat{B}$, and vectors x and $g(t)$ of an electrical diagram of n block models can be expressed as:

$$\hat{A} = \begin{bmatrix} \hat{A}_{1,p_1\times p_1} & 0_{p_1\times p_2} & \cdots & 0_{p_1\times p_n} \\ 0_{p_2\times p_1} & \hat{A}_{2,p_2\times p_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{p_{n-1}\times p_n} \\ 0_{p^n\times p_1} & \cdots & 0_{p^n\times p_{n-1}} & \hat{A}_{n,p_n\times p_n} \end{bmatrix}, x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix},$$

$$\hat{B} = \begin{bmatrix} \hat{B}_{1,p_1\times s_1} & 0_{p_1\times s_2} & \cdots & 0_{p_1\times s_n} \\ 0_{p_2\times s_1} & \hat{B}_{2,p_2\times s_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_{p_{n-1}\times s_n} \\ 0_{p_n\times s_1} & \cdots & 0_{p_n\times s_{n-1}} & \hat{B}_{n,,p_n\times s_n} \end{bmatrix}, \text{ and } g(t) = \begin{bmatrix} g_1(t) \\ \vdots \\ g_n(t) \end{bmatrix}.$$

The state space representation after applying the cut-set method is computationally expensive, especially for large power systems. This occurs because large linear systems of equations need to be built during the computational process when the topology of the power system changes. For example, an electrical network composed of E voltage sources, R_l resistors, R_t resistors, C capacitors, and L inductors requires to solve $5C$ linear systems of equations of size C, $5L$ systems of equations of size L, $C + R_l + E$ linear systems of equations of size R_l and $L + R_t + E$ linear systems of equations of size R_t. This task is rather time-consuming.

The same approach can also be applied for the computation of the ODE representation of the network from a DAE representation of index one. A power system composed of E voltage sources, R resistors, C capacitors, and L inductors requires to solve one linear system of equations of size R by applying the Schur complement [36]. This approach is less time-consuming than the cut-set method.

The block modeling method gives the same state space representation as the cut-set method or the DAE transformation. The difference is that the block modeling approach only requires the differential equations of the block models and how they are connected with each other. Thus, it is computationally cheap because it needs to solve only small set of equations in some cases, as explained in Section 4.3. Meanwhile, in the case of unknown block equations, these two approaches can be applied and also be perfomed in parallel because each block is independent of each other.

When a switching action takes place in the electrical network, the whole process has to be repeated for the cut-set method or DAE transformation, bus this is not necessary for the block modeling method. After a switching action, only few differential equations change in general. As a result, a limited number of elements of the matrices $\tilde{A}$ or $\tilde{B}$ have to be updated.

4.2.3. BLOCK MODEL REPRESENTATION AND PARAMETERS

In this section, the block model representation of linear power system elements - which are presented in Section 2.8 - are developed according to the block modeling mathematical approach in Table sec-BMM:MF:tab:PSC and Table 4.1. In addition, a voltage source and capacitor representation are given too.

Power system components	Number of terminals	Link voltages	Link capacitors
Generator	1	V_c	C
Load	1	V_c	C
PI section	2	V_{c_1}, V_{c_2}	C_1, C_2
Double PI section	2	V_{c_1}, V_{c_2}	C_1, C_2
Voltage source	1	$e(t)$	
Capacitance	1	V_c	C

Table 4.1: Block model link voltage and link capacitor representation

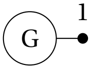
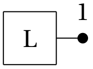
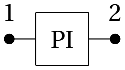
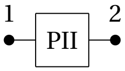
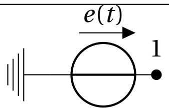
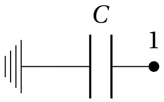
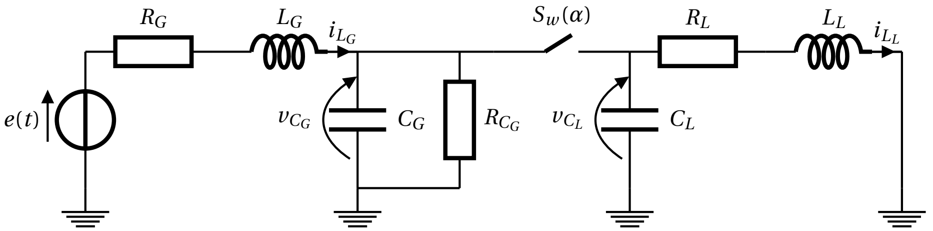
Power system components	**Symbol**	$\hat{A}$	$\hat{B}$	**x**	**g(t)**
Generator		$\hat{A}_G = A_G$	$\hat{B}_G = B_G$	$\hat{x}_G = x_G$	$\hat{g}_G(t) = g_G(t)$
Load		$\hat{A}_L = A_L$	$\hat{B}_L^{2\times 0}$	$\hat{x}_L = x_L$	$\hat{g}_L(t)^{0\times 1}$
PI section		$\hat{A}_{PI} = A_{PI}$	$\hat{B}_{PI}^{3\times 0}$	$\hat{x}_{PI} = x_{PI}$	$\hat{g}_{PI}(t)^{0\times 1}$
Double PI section		$\hat{A}_{PII} = A_{PII}$	$\hat{B}_{PII}^{5\times 0}$	$\hat{x}_{PII} = x_{PII}$	$\hat{g}_{PII}(t)^{0\times 1}$
Voltage source		$\hat{A}_V^{0\times 0}$	$\hat{B}_V^{5\times 0}$	$\hat{x}_V^{0\times 1}$	$\hat{g}_V(t) = e(t)$
Capacitance		$\hat{A}_C = 0$	$\hat{B}_C^{1\times 0}$	$\hat{x}_C = V_C$	$\hat{g}_C(t)^{0\times 1}$

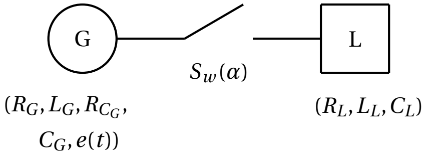
Table 4.2: Block model symbol and state space representation

4.2.4. FIRST EXAMPLE

The following sample network diagram and block diagram of Figure 4.1 illustrates the block modelling method.



(a) Lumped representation of a generator (left) connected to a load (right) via a switching device (S_w)



(b) Block representation of a generator (left) connected to a load (right) via switching device (S_w)

Figure 4.1: Lumped elements and a block model representation of a generator (left) connected to a load (right) via a switching device (S_w)

The state space representation of this network, when the switch is open, matrices A and B are equal to $\hat{A}$ and $\hat{B}$, respectively, and given by:

$$\dot{x} = \begin{bmatrix} \hat{A}_{G,2\times 2} & 0_{2\times 2} \\ 0_{2\times 2} & \hat{A}_{L,2\times 2} \end{bmatrix} \begin{bmatrix} x_G \\ x_L \end{bmatrix} + \begin{bmatrix} \hat{B}_{G,2\times 1} & 0_{2\times 0} \\ 0_{2\times 1} & \hat{B}_{L,2\times 0} \end{bmatrix} \begin{bmatrix} g_G(t) \\ g_L(t) \end{bmatrix}, \tag{4.3}$$

which leads to:

$$\dot{x} = \begin{bmatrix} -\frac{R_G}{L_G} & -\frac{1}{L_G} & 0 & 0 \\ \frac{1}{C_G} & -\frac{1}{R_{C_G}C_G} & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{C_L} \\ 0 & 0 & \frac{1}{L_L} & -\frac{R_L}{L_L} \end{bmatrix} \begin{bmatrix} i_{L_G} \\ v_{C_G} \\ v_{C_L} \\ i_{L_L} \end{bmatrix} + \begin{bmatrix} \frac{1}{L_G} \\ 0 \\ 0 \\ 0 \end{bmatrix} e(t). \tag{4.4}$$

Now, the interconnection of the generator and the load is made via a resistor R_{sw} which represents the closed switching device. Then, a current flows from the generator to the load, and the differential equations of the link capacitors C_g and C_L change. Besides, the current flow is a function of R_{sw} and of the link voltages v_{C_G} and v_{C_L}. The differential variable equations of the generator then become:

$$\dot{i}_{LG} = -\frac{R_G}{L_G} i_{LG} - \frac{1}{L_G} v_{CG} + \frac{1}{L_G} e(t), \tag{4.5}$$

$$\dot{v}_{CG} = \frac{1}{C_G} i_{LG} - \frac{1}{R_{C_G}C_G} v_{CG} - \frac{\left(v_{CG} - v_{C_L}\right)}{R_{sw}C_G}, \tag{4.6}$$

and the load block differential variable equations are now:

$$\dot{v}_{C_L} = \frac{\left(v_{CG} - v_{C_L}\right)}{R_{sw}C_G} - \frac{1}{C_L} i_{L_L}, \tag{4.7}$$

$$\dot{i}_{LG} = \frac{1}{L_L} v_{C_L} - \frac{R_L}{L_L} i_{L_L}. \tag{4.8}$$

Then, the new state space representation of the block model representation, when the switch is closed, becomes:

$$\dot{x} = \begin{bmatrix} -\frac{R_G}{L_G} & -\frac{1}{L_G} & 0 & 0 \\ \frac{1}{C_G} & -\frac{1}{R_{C_G}C_G} - \frac{1}{R_{sw}C_G} & \frac{1}{R_{sw}C_G} & 0 \\ 0 & \frac{1}{R_{sw}C_L} & -\frac{1}{R_{sw}C_L} & -\frac{1}{C_L} \\ 0 & 0 & \frac{1}{L_L} & -\frac{R_L}{L_L} \end{bmatrix} \begin{bmatrix} i_{L_G} \\ v_{C_G} \\ v_{C_L} \\ i_{L_L} \end{bmatrix} + \begin{bmatrix} \frac{1}{L_G} \\ 0 \\ 0 \\ 0 \end{bmatrix} e(t). \tag{4.9}$$

By subtracting Eq. 4.10 to 4.4, the matrices $\tilde{A}$ and $\tilde{B}$ of this block model diagram are:

$$\tilde{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{\alpha}{R_{sw}C_G} & \frac{\alpha}{R_{sw}C_G} & 0 \\ 0 & \frac{\alpha}{R_{sw}C_L} & -\frac{\alpha}{R_{sw}C_L} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } \tilde{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

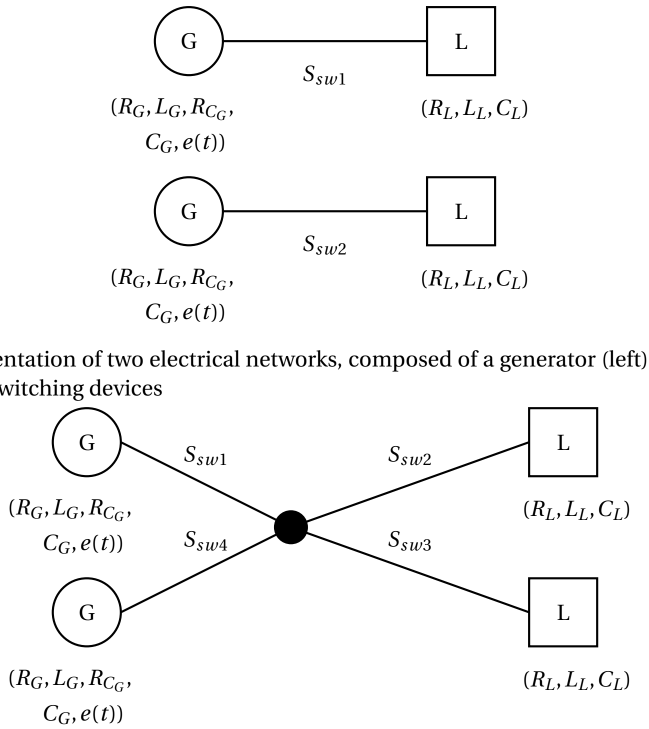
where α represents the position of the switch. When $\alpha = 1$, the switching device is closed. Otherwise, $\alpha = 0$ and the switch is open. As mentioned before, the calculation of the two states does not require solving a linear system of equations. From the matrix $\hat{A}$, a pseudo admittance matrix of the connection between the two block models can be observed. The nodal equations of the connection between the block models are:

$$-\begin{bmatrix} \frac{\alpha}{R_{sw}} & -\frac{\alpha}{R_{sw}} \\ -\frac{\alpha}{R_{sw}} & \frac{\alpha}{R_{sw}} \end{bmatrix} \begin{bmatrix} v_{C_G} \\ v_{C_L} \end{bmatrix} = \begin{bmatrix} i_{C_G} \\ i_{C_L} \end{bmatrix}. \tag{4.10}$$

In addition, the value and the position in the state variable of the link capacitors are necessary.

4.3. CONNECTING BLOCK MODEL MATRICES

The following two block representations are now studied.



(a) Block representation of two electrical networks, composed of a generator (left) connected to a load (right) via switching devices

(b) Block representation of one network, composed of two generators (left) connected to a loads (right) via an interconnection of the switching devices S_{sw1} to S_{sw4} to an additional electrical node

Figure 4.2: Two examples of possible connections between block models

The state space representation of Figure 4.2a is:

$$\dot{x} =$$

$$\begin{bmatrix} -\frac{R_G}{L_G} & -\frac{1}{L_G} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{C_G} & A_{2,2} & \frac{1}{R_{sw}C_G} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{R_{sw}C_L} & -\frac{1}{R_{sw}C_L} & -\frac{1}{C_L} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{L_L} & -\frac{R_L}{L_L} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{R_G}{L_G} & -\frac{1}{L_G} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{C_G} & A_{6,6} & \frac{1}{R_{sw}C_G} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{R_{sw}C_L} & -\frac{1}{R_{sw}C_L} & -\frac{1}{C_L} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{L_L} & -\frac{R_L}{L_L} \end{bmatrix} \begin{bmatrix} i_{L_G} \\ v_{C_G} \\ v_{C_L} \\ i_{L_L} \\ i_{L_G} \\ v_{C_G} \\ v_{C_L} \\ i_{L_L} \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{1}{L_G} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{L_G} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e(t) \\ e(t) \end{bmatrix}, \tag{4.11}$$

where $A_{2,2} = A_{4,4} = -\frac{1}{R_{C_G}C_G} - \frac{1}{R_{sw}C_G}$ and the ODE representation of Figure 4.2b is:

$$\dot{x} =$$

$$\begin{bmatrix} -\frac{R_G}{L_G} & -\frac{1}{L_G} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{C_G} & A_{2,2} & \frac{(G_1*G_2)}{G_{tot}C_G} & 0 & 0 & -\frac{(G_1*G_3)}{G_{tot}C_G} & -\frac{(G_1*G_4)}{G_{tot}C_G} & 0 \\ 0 & \frac{(G_1*G_2)}{G_{tot}C_L} & -\frac{G_2-G_2^2}{G_{tot}C_L} & -\frac{1}{C_L} & 0 & -\frac{G2*G3}{G_{tot}C_G} & -\frac{G2*G4}{G_{tot}C_G} & 0 \\ 0 & 0 & \frac{1}{L_L} & -\frac{R_L}{L_L} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{R_G}{L_G} & -\frac{1}{L_G} & 0 & 0 \\ 0 & -\frac{G_1*G_3}{R_{sw}C_G} & -\frac{G_2*G_3}{R_{sw}C_G} & 0 & \frac{1}{C_G} & A_{6,6} & \frac{G_3*G_4}{G_{tot}C_G} & 0 \\ 0 & -\frac{G_1*G_4}{G_{tot}C_G} & -\frac{G_2*G_4}{G_{tot}C_G} & 0 & 0 & \frac{G_3*G_4}{G_{tot}C_L} & -\frac{G_4-G_4^2}{G_{tot}C_L} & -\frac{1}{C_L} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{L_L} & -\frac{R_L}{L_L} \end{bmatrix} \begin{bmatrix} i_{L_G} \\ v_{C_G} \\ v_{C_L} \\ i_{L_L} \\ i_{L_G} \\ v_{C_G} \\ v_{C_L} \\ i_{L_L} \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{1}{L_G} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{L_G} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e(t) \\ e(t) \end{bmatrix}, \tag{4.12}$$

where $\begin{cases} G_{tot} = G_1 + G_2 + G_3 + G_4 & \text{if } \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \neq 0, \\ G_{tot} = 1 & \text{else} \end{cases}$, $A_{2,2} = -\frac{1}{R_{C_G}C_G} - \frac{G_1-G_1^2}{G_{tot}C_G}$ and $A_{6,6} = -\frac{1}{R_{C_G}C_G} - \frac{G_3-G_3^2}{G_{tot}C_G}$.

From Eq. 4.11 and Figure 4.2a, there is no connection possible between the two sub-networks. Then, when one of the switches changes position, it will not affect the differential equations of the other sub-network. For this reason, there are two switch blocks. On Figure 4.2b, there is an extra-node to connect the block models. In Eq. 4.12, a connection exists between all block models. Then, there is one switch block. In addition, such cases with an extra-node require the solving of a linear set of equations of size one.

From the previous electrical networks, a switch block is an interconnection of nbs switches between two or more block models. Each switch block has nbn nodes that correspond to the nbb block models connected via the switch block, and to k additional nodes from the topology of switching elements ($nbn = nbb + k$). Finally, a complete block diagram may contain more than one switch block.

Then, the incidence matrix K_{CL} of every switch block is defined as:

$$K_{CL}(m,j) = \begin{cases} +1 & \text{if the current through the switch } j \\ & \text{is connected to and oriented towards the node } m \\ -1 & \text{if the current through the switch } j \\ & \text{is connected and not oriented towards the node } m \\ 0 & \text{if switch } j \\ & \text{is not connected to node } m \end{cases},$$

to satisfy the following equation:

$$K_{CL} i_s = -i, \tag{4.13}$$

where $i_s = [i_{s_1} \cdots i_{s_{nbs}}]^T$ and $i = [i_t \ 0]^T$. The vector i_t corresponds with the current leaving each block model terminal connected via the switch block. Also, the vector i_t is organized such that the first components of the vector correspond to the currents through the nbs link capacitors, and next, the current from the nbe voltage sources.

During the initialization, the incidence matrix of each block switch is computed and stored. These matrices will be used for the initial state space representation, and reused when one of their switching devices changes position. Also, each block switch is associated with a diagonal conductance matrix G, where each element of the diagonal is composed of the admittance of corresponding switching devices ($S_{w_n} - > G(n,n) = \alpha_n / R_{sw_n} = G_n$ for example).

The incidence matrix and the conductivity of the switches are the inputs to calculate the admittance matrix by using the same rules as those to compute the admittance matrix of the nodal analysis method. Then, from this matrix it is possible to update the matrices $\tilde{A}$ and $\tilde{B}$. However, a block model can be connected to only one additional extra node via switches, but it can also be connected with many block models. Finally, three steps are used for each switch block during the initialization, and when a switching action takes place in this particular switch block. These steps are specified in the following subsections.

4.3.1. FIRST STEP: FULL ADMITTANCE MATRIX

The full admittance matrix (Y_{full}) is computed as:

$$Y_{full} v = K_{CL} G K_{CL}^T v = -i, \tag{4.14}$$

where $G = diag([G_{sw_1}(\alpha_1) \cdots G_{sw_{nbs}}(\alpha_{nbs})])$ and $v = [v_t \ v_n]^T$.

4.3.2. SECOND STEP: REDUCTION OF THE FULL ADMITTANCE MATRIX

This step is not necessary if the number of nodes of the block switch is equal to the number of terminals ($nbb = nbn$). In that case, the admittance matrix Y is simply defined as:

$$Y = -Y_{full}. \tag{4.15}$$

Now, when the number of nodes of the block switch is different from the number of terminals ($nbb \neg nbn$), the Schur complement method [36] is applied to compute the matrix Y as:

$$Y_{full} v = \left[\begin{array}{cc} Y_{full_1} & Y_{full_2} \\ Y_{full_3} & Y_{full_4} \end{array} \right] \left[\begin{array}{c} v_t \\ v_n \end{array} \right] = - \left[\begin{array}{c} i_t \\ 0 \end{array} \right], \tag{4.16}$$

which gives:

$$Y v_t = -i_t \tag{4.17}$$

where:

$$Y = -Y_{full_1} + Y_{full_2} Y_{full_4}^{-1} Y_{full_3}. \tag{4.18}$$

It is, however, not possible to form the Schur complement when:

- A switch is open between a terminal and an extra node, or between two additional nodes. Then, zeros may appear on the diagonal of the matrix Y_{full_4};

- A switching device is closed between two extra nodes, and neither of these additional nodes is connected via a closed switch to a terminal. Then, a singularity may appear in the matrix Y_{full_4}.

To circumvent these problems, the following actions are taken:

- For the first problem, zeros on the diagonal of the matrix Y_{full_4} can be replaced by ones. This action will not affect the resulting matrix Y, and it is then possible to form the Schur complement.

- For the second problem, this particular switch is then considered open because there is no current flowing by it. Then, it does not play a role for the computation of the differential equations, and the first issue is present, and has already been tackled.

Another way to overcome these problems is to connect a small capacitance to each extra node, and the Schur complement does not have to be formed any more ($Y_{full} = Y$). However, with this approach, the number of differential variables increases and the matrix A can be bad conditioning and lead to problems during the integration process. The Schur complement is only used when the number of nodes is larger than the number of block models connected through a switch block. Finally, when a switching action takes place, the Schur complement is applied on a small scale because only the switch block containing the switching action is taken into account.

4.3.3. THIRD STEP: UPDATING THE CONNECTING BLOCK MATRICES

From the admittance matrix Y, the matrices $\tilde{A}$ and $\tilde{B}$ can be updated where nbb block models are connected via the switch block - which includes nbe voltage source block models like:

$$\tilde{A}_{I_1(m,l)I_1(m,j)} = \frac{1}{C(m,l)} Y_{lj}, \tag{4.19}$$

for $1 \leq l, j \leq nbb - nbe$ where $I_1(m,l)$ gives the position of the link capacitor of the block model l, at the switch block m, in state vector x, and $C(m,l)$ contains the value of the link capacitor of block model l at the switch block m.

$$\tilde{B}_{I_1(m,l)I_2(m,j)} = \frac{1}{C(m,l)} Y_{l(j+nbb-1)} \tag{4.20}$$

for $nbb - nbe \leq j \leq nbb$ where $I_2(m,j)$ gives the position of the voltage source in the time-input vector, at the switch block m.

4.4. EXAMPLE OF AN ANALYTICAL SOLUTION OF A SWITCH BLOCK

In this section, a block model diagram composed of one switch block is investigated. The results achieved by the block modeling method are identical to the results from Kirchhoff's laws. As explained before, all switches are non-ideal. For this demonstration, the block diagram of Figure (4.3) is considered.

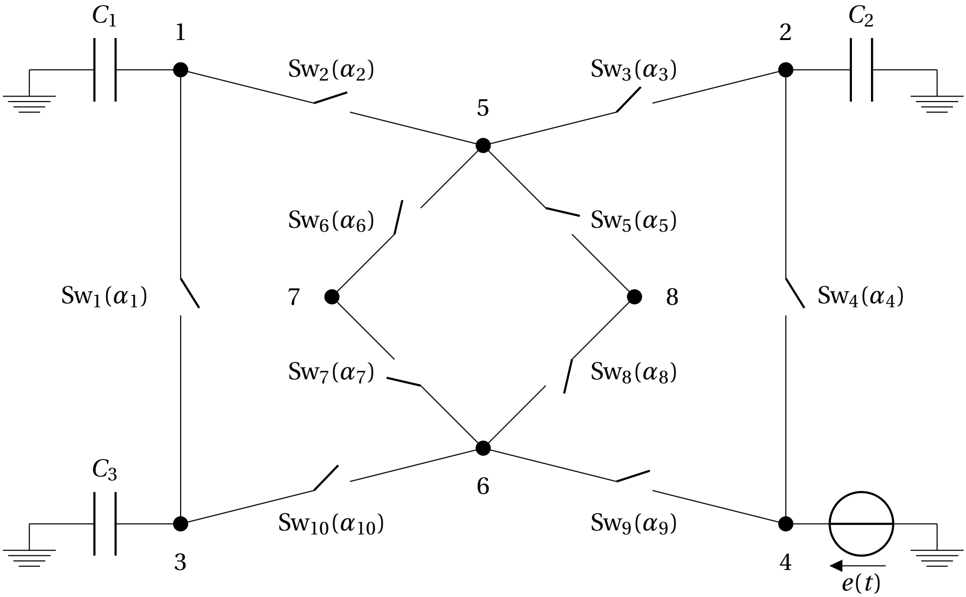


Figure 4.3: Electrical block diagram of a switch block with eight switching devices

4.4.1. ANALYTICAL SOLUTION

From the block diagram of Figure (4.3), it is possible to write $x = [v_{C_1}\ v_{C_2}\ v_{C_3}]^T$, $\hat{A} = 0^{3\times3}$, $\hat{B} = 0^{3\times3}$ and $g(t) = e(t)$. The analytical solution for the current through each capacitor is related to the position (closed or open) of each switching device. This results in a rather complicated formula. For the demonstration, $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_5 = \alpha_7 = 1$ is considered and the other values of α_i are zero. The solution with Kirchhoff's laws is:

$$\left\{ \begin{array}{ll} \dot{v}_{C_1} & = \left(G_1(v_{C_3} - v_{C_1}) - G_2 v_{C_1} + G_2 \frac{G_2 V_{C_1} + G_3 V_{C_2}}{G_2 + G_3} \right) / C_1 \\ \dot{v}_{C_2} & = \left(-G_3 v_{C_2} + G_3 \frac{G_2 V_{C_1} + G_3 V_{C_2}}{G_2 + G_3} \right) / C_2 \\ \dot{v}_{C_3} & = \left(G_1(v_{C_1} - v_{C_3}) \right) / C_3 \end{array} \right. . \tag{4.21}$$

4.4.2. BLOCK-MODELING SOLUTION

According to Figure (4.3), the block switch parameters are $nbb = 4$, $nbe = 1$, $nbs = 10$, $nbn = 8$, and $m = 1$. In addition, the state vector employed defines $I_1(1,1) = 1$, $I_1(1,2) = 2$, $I_1(1,3) = 3$, $I_2(1,1) = 1$, $C(1,1) = C_1$, $C(1,2) = C_2$ and $C(1,3) = C_3$. Initially, the matrix K_{CL} is computed, as defined in Section 4.3, as:

$$K_{CL} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}, \tag{4.22}$$

and the matrix $G(\alpha)$ is defined as:

$$G(\alpha) = \operatorname{diag}\left(\begin{bmatrix} G_1 & G_2 & G_3 & G_4 & G_5 & G_6 & G_7 & G_8 & G_9 & G_{10} \end{bmatrix} \right). \tag{4.23}$$

In addition, the following vector is considered $i_t = \begin{bmatrix} i_{c_1} & i_{c_2} & i_{c_3} & i_e \end{bmatrix}^T$. Now, the three steps necessary to obtain the updated matrices $\tilde{A}$ and $\tilde{B}$ according to Section 4.3 are applied as shown below.

FIRST STEP: FULL ADMITTANCE MATRIX

By applying Eq. 4.14, Eq.4.24 is found:

$$Y_{full} = \begin{bmatrix} G_1 + G_2 & 0 & -G_1 & 0 & -G_2 & 0 & 0 & 0 \\ 0 & G_3 & 0 & 0 & -G_3 & 0 & 0 & 0 \\ -G_1 & 0 & G_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -G_2 & -G_3 & 0 & 0 & G_2 + G_3 + G_5 & 0 & 0 & -G_5 \\ 0 & 0 & 0 & 0 & 0 & G_7 & -G_7 & 0 \\ 0 & 0 & 0 & 0 & 0 & -G_7 & G_7 & 0 \\ 0 & 0 & 0 & 0 & -G_5 & 0 & 0 & G_5 \end{bmatrix} \tag{4.24}$$

SECOND STEP: REDUCTION OF THE FULL ADMITTANCE MATRIX

In this case, nbn is different from nbb. Then, the Schur complement is used according to the following sub-matrices and to Eq. 4.16:

$$Y_{full_1} = \begin{bmatrix} G_1+G_2 & 0 & -G_1 & 0 \\ 0 & G_3 & 0 & 0 \\ -G_1 & 0 & G_1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.25}$$

$$Y_{full_3} = Y_{full_2}^T = \begin{bmatrix} -G_2 & -G_3 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{4.26}$$

$$Y_{full_4} = \begin{bmatrix} G_2+G_3+G_5 & 0 & 0 & -G_5 \\ 0 & G_7 & -G_7 & 0 \\ 0 & -G_7 & G_7 & 0 \\ -G_5 & 0 & 0 & G_5 \end{bmatrix}. \tag{4.27}$$

The matrix Y_{full_4} is not invertible. Its determinant equals zero because the switch labelled Sw_7 in Figure 4.3 is closed. Since this switch is connected to two extra nodes and none of these nodes are linked to a terminal via a closed switch, it can be considered as an open switch. Finally, Y_{full_4} becomes:

$$Y_{full_4} = \begin{bmatrix} G_2+G_3+G_5 & 0 & 0 & -G_5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -G_5 & 0 & 0 & G_5 \end{bmatrix}. \tag{4.28}$$

Now, there are several zeros on the diagonal. In line with the methodology, the zeros are replaced by the number one:

$$Y_{full_4} = \begin{bmatrix} G_2+G_3+G_5 & 0 & 0 & -G_5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -G_5 & 0 & 0 & G_5 \end{bmatrix}. \tag{4.29}$$

Finally by applying Eq. 4.18, the following admittance matrix Y is found:

$$Y = \begin{bmatrix} \frac{G_2^2}{G_2+G_3} - G_2 - G_1 & \frac{G_2G_3}{G_2+G_3} & G_1 & 0 \\ \frac{G_2G_3}{G_2+G_3} & \frac{G_3^2}{G_2+G_3} - G_3 & 0 & 0 \\ G_1 & 0 & -G_1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4.30}$$

THIRD STEP: UPDATING THE CONNECTING BLOCK MATRICES

Now by combining Eq. 4.19 and Eq. 4.20, the following state space representation is found:

$$\dot{x} = \begin{bmatrix} \frac{\frac{G_2^2}{G_2+G_3}-G_2-G_1}{C_1} & \frac{\frac{G_2G_3}{G_2+G_3}}{C_1} & \frac{G_1}{C_1} \\ \frac{\frac{G_2G_3}{G_2+G_3}}{C_2} & \frac{\frac{G_3^2}{G_2+G_3}-G_3}{C_2} & 0 \\ \frac{G_1}{C_3} & 0 & \frac{-G_1}{C_3} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_{C_1} \\ v_{C_2} \\ v_{C_3} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} e(t). \tag{4.31}$$

The state space representation corresponds to the analytical solution found from Kirchhoff's laws (Eq. 4.21).

4.5. TIME-DEPENDENT AND NON-LINEAR ELECTRICAL NETWORKS

4.5.1. MATHEMATICAL FORMULATION

Power systems are non-linear. For this reason, Eq. 4.2 can be developed, for example, as:

$$\dot{x} = (\hat{A}+\tilde{A})x + A_{non}(x)x + (\hat{B}+\tilde{B})g(t) + v(x), \tag{4.32}$$

where the time dependent lumped element and non-linear functions are included via the functions $A_{non}(x)$ and $v(x)$.

4.5.2. INTRODUCTION TO SOME TIME-DEPENDENT AND NON-LINEAR ELECTRICAL NETWORKS

For example, the inductance of the load of Figure 4.1 is time-dependent ($L_L = l(t)$). Then, the mathematical equations of the load block model are now:

$$\dot{x_L} = \hat{A}_L x_L + \hat{B}_L g_L(t) \tag{4.33}$$

$$= \begin{bmatrix} 0 & -\frac{1}{C} \\ \frac{1}{l(t)} & -\frac{R}{l(t)} - \frac{dl(t)}{l(t)dt} \end{bmatrix} \begin{bmatrix} v_{C_L} \\ i_{L_L} \end{bmatrix}, \tag{4.34}$$

and then, the new state space representation is:

$$\dot{x} = \left[\begin{bmatrix} \hat{A}_G & 0 \\ 0 & \hat{A}_L \end{bmatrix} + \tilde{A} \right] \begin{bmatrix} x_G \\ x_L \end{bmatrix} + \left[\begin{bmatrix} \hat{B}_G & 0 \\ 0 & \hat{B}_L \end{bmatrix} + \tilde{B} \right] \begin{bmatrix} g_G(t) \\ g_L(t) \end{bmatrix} \tag{4.35}$$

$$= \left[\begin{bmatrix} -\frac{R_G}{L_G} & -\frac{1}{L_G} & 0 & 0 \\ \frac{1}{C_G} & -\frac{1}{R_{C_G}C_G} & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{C_L} \\ 0 & 0 & \frac{1}{l(t)} & -\frac{R_L}{l(t)} - \frac{dl(t)}{l(t)dt} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{\alpha}{R_{sw}C_G} & \frac{\alpha}{R_{sw}C_G} & 0 \\ 0 & \frac{\alpha}{R_{sw}C_L} & -\frac{\alpha}{R_{sw}C_L} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right] \begin{bmatrix} i_{L_G} \\ v_{C_G} \\ v_{C_L} \\ i_{L_L} \end{bmatrix}$$

$$+ \begin{bmatrix} \frac{1}{L_G} \\ 0 \\ 0 \\ 0 \end{bmatrix} e(t). \tag{4.36}$$

In the second example, the switching device is now replaced by an arc model called the Habedank model (Section 2.8.6) [37]. This arc model simulates the interruption of the current through it by changing its conductivity. Its mathematical expressions are shown from Eq. 2.53 to Eq. 2.56. From these equations, the mathematical representation of this electrical network is:

$$
\begin{aligned}
\begin{bmatrix} \dot{i}_{L_G} \\ \dot{v}_{C_G} \\ \dot{v}_{C_L} \\ \dot{i}_{L_L} \\ \dot{g}_c \\ \dot{g}_m \end{bmatrix}
&= \begin{bmatrix}
-\frac{R_G}{L_G} & -\frac{1}{L_G} & 0 & 0 & 0 & 0 \\
\frac{1}{C_G} & -\frac{1}{R_{C_G} C_G} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{1}{C_L} & 0 & 0 \\
0 & 0 & \frac{1}{L_L} & -\frac{R_L}{L_L} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} i_{L_G} \\ v_{C_G} \\ v_{C_L} \\ i_{L_L} \\ g_c \\ g_m \end{bmatrix}
+ \begin{bmatrix} \frac{1}{L_G} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} e(t) \\
&+ \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{g_{arc}}{C1} & \frac{g_{arc}}{C1} & 0 & 0 & 0 \\
0 & \frac{g_{arc}}{C2} & -\frac{g_{arc}}{C2} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix} i_{L_G} \\ v_{C_G} \\ v_{C_L} \\ i_{L_L} \\ g_c \\ g_m \end{bmatrix}
+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{\alpha}{\tau_c}\left(\frac{(g_{arc} u_{arc})^2}{U_c^2 g_c} - g_c\right) \\ \frac{\alpha}{\tau_m}\left(\frac{(g_{arc} u_{arc})^2}{P_m} - g_m\right) \end{bmatrix},
\end{aligned}
\tag{4.37}
$$

where α represents the state of the arc model. When $\alpha = 0$, the arc model is inactive, else, it is activated.

During the integration process, non-linear lumped elements will lead to time-dependent or non-linear system of equations. Then, the Jacobian matrix is computed to solve them with a Newton-Raphson iteration. The definition of the Jacobian matrix is:

$$
J(x) = J = \frac{\partial f(x)}{\partial x} \tag{4.38}
$$

$$
\begin{aligned}
&= A + \begin{bmatrix}
\frac{\partial \sum_{j=1}^{p} A_{non}(1,j) x_j}{\partial x_1} & \cdots & \frac{\partial \sum_{j=1}^{p} A_{non}(1,j) x_j}{\partial x_p} \\
\vdots & \ddots & \vdots \\
\frac{\partial \sum_{j=1}^{p} A_{non}(p,j) x_j}{\partial x_p} & \cdots & \frac{\partial \sum_{j=1}^{p} A_{non}(p,j) x_j}{\partial x_p}
\end{bmatrix}
+ \begin{bmatrix}
\frac{\partial v(1)}{\partial x_1} & \cdots & \frac{\partial v(1)}{\partial x_p} \\
\vdots & \ddots & \vdots \\
\frac{\partial v(p)}{\partial x_p} & \cdots & \frac{\partial v(p)}{\partial x_p}
\end{bmatrix} \\
&= A + A_{non} + J_{non} + J_v.
\end{aligned}
\tag{4.39}
$$

At each time-step, the Jacobian matrix has to be computed. Two methods are available to compute the equations - the numerical method and the analytical method. The numerical method will require a larger amount of computation time than the analytical one. For example, the Jacobian matrix of Figure 4.1, when the non-linear inductance is used, is:

$$J = \left[\begin{bmatrix} -\frac{R_G}{L_G} & -\frac{1}{L_G} & 0 & 0 \\ \frac{1}{C_G} & -\frac{1}{R_{C_G} C_G} & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{C_L} \\ 0 & 0 & \frac{1}{l(t)} & -\frac{R_L}{l(t)} - \frac{dl(t)}{l(t)dt} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{\alpha}{R_{sw} C_G} & \frac{\alpha}{R_{sw} C_G} & 0 \\ 0 & \frac{\alpha}{R_{sw} C_L} & -\frac{\alpha}{R_{sw} C_L} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\right]. \tag{4.40}$$

In addition, the Jacobian matrix, when the arc model is present, is:

$$\begin{aligned} J = & \begin{bmatrix} -\frac{R_G}{L_G} & -\frac{1}{L_G} & 0 & 0 & 0 & 0 \\ \frac{1}{C_G} & -\frac{1}{R_{C_G} C_G} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{C_L} & 0 & 0 \\ 0 & 0 & \frac{1}{L_L} & -\frac{R_L}{L_L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ & + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\alpha}{R_{arc} C_G} & \frac{\alpha}{R_{arc} C_G} & 0 & -\frac{\alpha u_{arc} g^2}{C_G (1,1) g_c^2} & -\frac{\alpha u_{arc} g^2}{C_G g_m^2} \\ 0 & \frac{\alpha}{R_{arc} C_L} & -\frac{\alpha}{R_{arc} C_L} & 0 & \frac{\alpha u_{arc} g^2}{C_L g_c^2} & \frac{\alpha u_{arc} g^2}{C_L (1,2) g_m^2} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ & + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha \frac{2g^2 u_a rc}{\tau_c U_c^2 g_c} & -\alpha \frac{2g_h^2 u_{arc}}{\tau_c U_c^2 g_h} & 0 & \frac{\alpha}{\tau_c}\left(\frac{u_a rc^2 g_h}{U_c^2 g_c^2}\left(1 - \frac{2g}{g_m}\right) - 1\right) & \frac{\alpha 2 u_{arc}^2 g_h^3}{\tau_c U_c^2 g_c g_m^2} \\ 0 & \alpha \frac{2g_h^2 u_{arc}}{\tau_m P_m} & -\alpha \frac{2g_h^2 u_{arc}}{\tau_m P_m} & 0 & \frac{\alpha 2 u_{arc}^2 g_h^3}{\tau_m P_m g_c^2} & -\frac{\alpha}{\tau_m}\left(\frac{2u_{arc}^2 g_h^3}{P_m g_m^2} - 1\right) \end{bmatrix} \end{aligned} \tag{4.41}$$

4.5.3. DISCUSSION

Now, the illustration of time-dependent or non-linear elements has been shown in this section. Also, the Jacobian computation is possible for such simple electrical networks. However, when the lumped element value changes in time, the Jacobian matrix is the same as the state space representation. In the case of the arc models, the state space representation can be computed easily from the same information as that for a switching device. Meanwhile, it is recommended to put the arc model between two block terminals. Then, a methodology can be extrapolated from the previous example to compute the state space representation and the Jacobian matrix. In fact, in order to compute the Jacobian matrix, and the place of the state variables (which represent the block model terminal), are necessary along with the position of the differential variables, which represent the different conductivities of the arc models.

4.6. Conclusion

In this chapter, a new approach to model power systems containing switching devices is studied. The method couples the differential equations of the various components of a power system via switches. This results in local updates of the system of equations when switching actions occur. Besides, this approach allows for the parallelization of computationnal effort of the cut-set method or the modified nodal analysis method, by using these modeling approaches only on power system components, in oder to obtain an ODE representation when block models are not known. Then, this methodology can be applied to couple these block models. The conclusions of this approach are:

1. The method can be applied for most configurations of switches among power system components;
2. This approach can be used to model power electronic converters;
3. Non-linear power systems can be handled.

Finally, on a supercomputer, this modeling method can also be implemented to increase the size of the power system under consideration and to decrease the computation time.

References

[1] R. Smeets, L. Van der Sluis, M. Kapetanovic, D. F. Peelo, and A. Janssen, *Switching in electrical transmission and distribution systems* (John Wiley & Sons city, Chichester, 2014).

[2] L. van der Sluis, *Transients in Power Systems* (John Wiley & Sons, Chichester, 2001).

[3] J. Vlach and K. Singhal, *Computer methods for circuit analysis and design* (Van Nostrand Reinhold, New York, 1983).

[4] N. Watson and J. Arrillaga, *Power systems electromagnetic transients simulation*, Vol. 39 (Institution of Electrical Engineers, London, 2003).

[5] P. R. Adby, *Applied circuit theory: matrix and computer methods*, JOHN WILEY & SONS, N. Y., 1980, 250 (1980).

[6] *PSCAD/EMTDC: Electromagnetic transients program including dc systems* (1994).

[7] *Electromagnetic Transient Program EMTP RV* (2007).

[8] A. Vladimirescu, *The SPICE book* (John Wiley & Sons, Inc., New York, 1994).

[9] N. Bijl and L. D. Van Sluis, *New approach to the calculation of electrical transients, part i: Theory,* European transactions on electrical power **8**, 175 (1998).

[10] *MATLAB SimPowerSystems for Use with Simulink User's Guide, Version 4.1.1.*

[11] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, 1st ed. (Society for Industrial and Applied Mathematics, Philadelphia, 1998).

[12] R. DeCarlo and P.-M. Lin, *Linear Circuit Analysis*, 2^{nd} ed. (OXFORD UNIVERSITY PRESS, New York, 2001).

[13] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical solution of initial-value problems in differential-algebraic equations*, Vol. 14 (Society for Industrial and Applied Mathematics, Philadelphia, 1996).

[14] G. Wanner and E. Hairer, *Solving ordinary differential equations II : Stiff and Differential-Algebraic Problems*, Vol. 1 (Springer Berlin Heidelberg, Berlin Heidelberg, 2010).

[15] C. A. Kennedy and M. H. Carpenter, *Diagonally implicit Runge-Kutta methods for ordinary differential equations. a review,* (2016).

[16] W. Hundsdorfer and J. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations* (Springer, New York, 2003).

[17] M. Vaezi and A. Izadian, *Piecewise affine system identification of a hydraulic wind power transfer system,* IEEE Transactions on Control Systems Technology **23**, 2077 (2015).

[18] A. Luna, J. Rocabert, J. I. Candela, J. R. Hermoso, R. Teodorescu, F. Blaabjerg, and P. Rodríguez, *Grid voltage synchronization for distributed generation systems under grid fault conditions,* IEEE Transactions on Industry Applications **51**, 3414 (2015).

[19] M. Brucoli, F. Torelli, and M. Trovato, *State space representation of interconnected power systems for dynamic interaction studies,* Electric Power Systems Research **5**, 315 (1982).

[20] H. W. Dommel, *Digital computer solution of electromagnetic transients in single-and multiphase networks,* Power Apparatus and Systems, IEEE Transactions on **PAS-88**, 388 (1969).

[21] J. K. Debnath, A. M. Gole, and W.-K. Fung, *Graphics processing unit based acceleration of electromagnetic transients simulation,* (2015).

[22] C. W. Ho, A. E. Ruehli, and P. A. Brennan, *The modified nodal approach to network analysis,* Circuits and Systems, IEEE Transactions on **22**, 504 (1975).

[23] Y. Oshima, H. Oguma, N. Ishihara, and K. Masu, *Simulation and evaluation of pv power generation for energy management system using spice,* in *2016 International Conference on Information and Communication Technology Convergence (ICTC)* (2016) pp. 74–77.

[24] T. Bashkow, *The A matrix, new network description,* IRE transactions on circuit Theory **4**, 117 (1957).

[25] E. Kuh and R. Rohrer, *The state-variable approach to network analysis,* Proceedings of the IEEE **53**, 672 (1965).

[26] A. Ioinovici, *Computer-Aided Analysis of Active Circuits* (MARCELL DEKKER, inc, New York, 1990).

[27] F. Bizzarri, A. Brambilla, G. S. Gajani, and S. Banerjee, *Simulation of real world circuits: Extending conventional analysis methods to circuits described by heterogeneous languages,* IEEE Circuits and Systems Magazine **14**, 51 (2014).

[28] J. Vandewalle, H. De Man, and J. Rabaey, *Time, frequency, and z-domain modified nodal analysis of switched-capacitor networks,* IEEE Transactions on circuits and systems **28**, 186 (1981).

[29] J. Xu and J. Yu, *Equivalent circuit models of switches for SPICE simulation,* Electronics letters **24**, 437 (1988).

[30] G. Verghese, B. Lévy, and T. Kailath, *A generalized state-space for singular systems,* IEEE Transactions on Automatic Control **26**, 811 (1981).

[31] G. M. Huang, K. Men, and X. Song, *A new remodeling technique for power system dynamic analysis,* in *2005 IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific* (IEEE, 2005) pp. 1–6.

[32] B. De Kelper, L. A. Dessaint, K. Al-Haddad, and H. Nakra, *A comprehensive approach to fixed-step simulation of switched circuits,* IEEE Transactions on Power Electronics **17**, 216 (2002).

[33] R. J. Dirkman, *Generalized state space averaging,* in *Power Electronics Specialists Conference, 1983 IEEE* (IEEE, 1983) pp. 283–294.

[34] P. C.-K. Luk, S. Aldhaher, W. Fei, and J. F. Whidborne, *State-space modeling of a class converter for inductive links,* IEEE Transactions on Power Electronics **30**, 3242 (2015).

[35] P. Schavemaker and L. van der Sluis, *Electrical Power System Essentials* (John Wiley and Sons, Chichester, 2008).

[36] F. Zhang, *The Schur complement and its applications*, Vol. 4 (Springer Science & Business Media, 2006).

[37] M. Kapetanovic, *High voltage circuit breakers* (Falculty of Electrotechnical Engineering, Sarajevo, Sarajevo, 2011).

5

NUMERICAL ANALYSIS OF THE BLOCK MODELING METHOD

5.1. INTRODUCTION

IN this chapter, various networks are studied to demonstrate the block modeling method. As presented in Chapter 4, two type of power systems are considered - linear and non-linear. Then, several sub-cases of different sizes are investigated as well. To compare the results of the block modeling method in terms of speed-up and accuracy, the software PSCAD is used for linear networks and Matlab/SimPowerSystem is used for non-linear systems [1, 2]. In the case of a linear network, the simulation of these test cases with the block modeling method is implemented in C/C++ with the help of the toolbox PETSc [3, 4]. To illustrate the non-linear network, this modeling method has been performed in Matlab. Finally, for the same precision in the larger test cases, a speed improvement factor 11 is achieved for the linear case, and factor 20 for the non-linear system.

All computations were performed on a machine with an Intel(R) Core(TM)2 Duo 3GHz CPU and 4GB memory. Two cores were used to obtain the results with PSCAD and Matlab simulation. Other simulations have been computed on only one computer core.

This chapter is organized as follows. Section 5.2 presents and discusses the different linear networks. Section 5.3 gives the numerical results when arc models are used instead of switching devices. Finally, Section 5.4 offers the conclusion of the utilization of the block modeling method.

5.2. LINEAR POWER SYSTEMS

5.2.1. DESCRIPTION OF TEST CASES

Two networks, modeled with the nodal analysis method and with the block modeling method, are considered. The computation time of a complete simulation (from the calculation of the first set of equations till the end of the simulation) and the accuracy of the time-domain solution are taken for comparison.

The first test network consists of 40 block models (five generators, eleven loads, fourteen PI-sections and nine double PI-sections) and sixteen switch blocks (sixty-three switches), as depicted in Figure 5.1. The block models' parameters in accordance with Table 4.2 are:

- G: $R = 0.1\Omega$, $L = 0.2mH$, $C = 1\mu F$ and $R_C = 100\Omega$;
- L: $R = 1\Omega$, $L = 0.4mH$ and $C = 1\mu F$;
- PI-section: $R = 0.1m\Omega$, $L = 0.2mH$ and $C_1 = C_2 = 0.1\mu F$;
- Double PI-section: $R_1 = R_2 = 0.1m\Omega$, $L_1 = L_2 = 0.2mH$ and $C_1 = C_2 = C_3 = 0.1\mu F$;
- Switch: $R_{sw} = 0.1\Omega$.

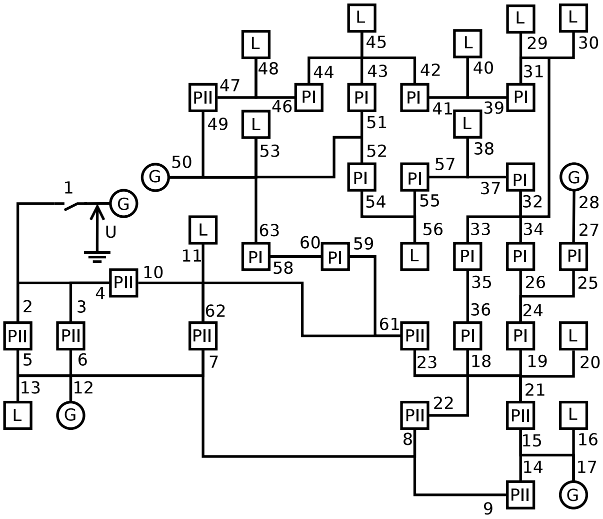


Figure 5.1: Block diagram representation of the network composed of 40 block-models and 16 switch blocks

The initial conditions of the network are $x(t_0) = 0$. All switches are in the closed position, except for the switch with label 1 in Figure 5.1 that is being closed at $t = 0.06s$.

The second test network is created by copying the block diagram of Figure 5.1 five times. Different switching times are applied to illustrate the effect of updating the set of equations several times. For this reason, there are no links among the five networks.

The systems are simulated with the nodal analysis program PSCAD and the block modeling approach, with the Trapezoidal rule as integration method. A time step of $1\mu s$ was chosen to capture the different transient oscillations because of the initial conditions and switching actions. To get a better understanding of the method, the ARK4(3)6L[2]SA-ESDIRK [5] shown in Section 3.3.3 is employed as a numerical integration method to replace the Trapezoidal rule. This approach and the various numerical integration processes are implemented in C/C++ by using the PETSc toolbox.

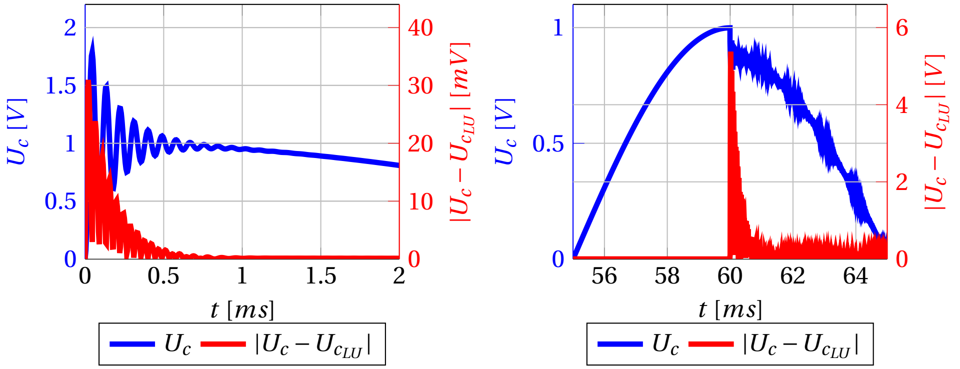
5.2.2. RESULTS AND DISCUSSIONS

From the PSCAD software and this new approach, results are given in Table 5.1 and illustrated by Figures 5.2 to 5.4. Table 5.1 shows the computation time for different integration methods. Then, Figures 5.2 to 5.4 present the time domain solution of the voltage U (close to switch 1; see Figure 5.1) for the initialization and around the time of a switching event. The error between the computed and the analytical solution is also shown in these figures. In these figures, only the smaller network is considered.

Modeling method	Integration method	Network	
		1	2
PSCAD	TR (1μs)	25.4	139
BMM	TR (1μs)	30	48
BMM	ARK4 (tol=0.001)	5.71	12

Table 5.1: Computation times in seconds of the Nodal Analysis program PSCAD and the Block Modeling Method (BMM) with a time-step of 1μs (Trapezoidal rule), and the ARK4 method with a tolerance of 0.001

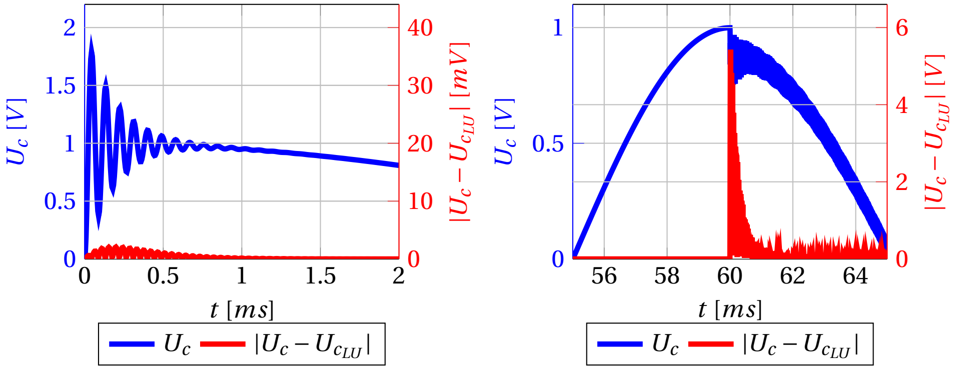
Table 5.1 shows the computation time, in seconds, of the overall simulation of the various test networks. For the larger test case, PSCAD was almost 11 times slower than the block modeling method with adaptive time-stepping. The speedup was of the order 22 when both were running on a single core.



(a) Initialization

(b) Around the switching action

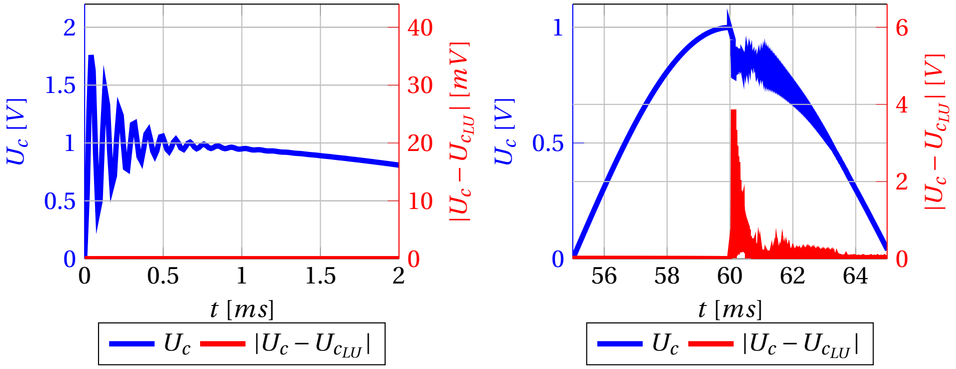
Figure 5.2: Computation of signal U and its error with the sofware PSCAD, and a time-step of 1μs



(a) Initialization

(b) Around the switching action

Figure 5.3: Computation of signal U and its error with the block modeling method, the Trapezoidal rule as integration method, and a time-step of $1 \mu s$



(a) Initialization

(b) Around the switching action

Figure 5.4: Computation of signal U and its error with the block modeling method, the ARK4(3)6L[2]SA-ESDIRK method as integration method, and an adaptive time-stepping algorithm

From Figures 5.2 to 5.4, the time domain solutions of the signal U were almost identical regarding the waveforms at the initialization, and around the switching action with the various methods. Also, error waveforms were similar with the different methods. Finally when the ARK4(3)6L[2]SA-ESDIRK methd is used, the error waveform was smaller than the waveforms for other methods around the switching action, because the order of the ARK4 method is higher than the order of the Trapezoidal rule, and it has a time-stepping algorithm based on the error.

5.3. NON-LINEAR POWER SYSTEMS

In this part, two power systems were considered. They were both modeled via the block modeling method implemented in Matlab for the different arc models. The integration method chosen is the numerical integration called ode23tb with a relative tolerance of 10^{-4}.

5.3.1. TEST CASES PRESENTATION

In this part, two test cases are studied. The first one is composed of 22 block models, 10 switch blocks, 22 switches and two Habedank models - as shown in Figure 5.5.

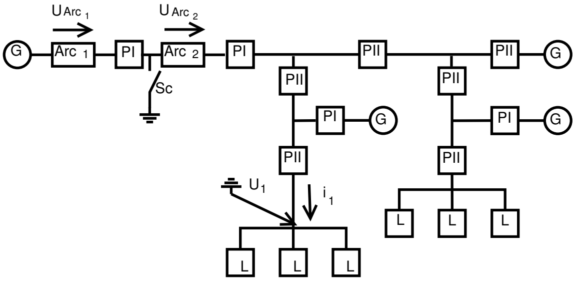


Figure 5.5: Block representation of the synthetic network 2, composed of 22 block models, 10 node-blocks and including 2 Habedank models

This network consists of 66 differential variables. Then, the second test case under consideration is a 3-phase power system where each phase is composed of two arc models, 36 block models, 16 switch blocks and 42 switches. Figure 5.6 represents a one-line representation of the network under consideration.

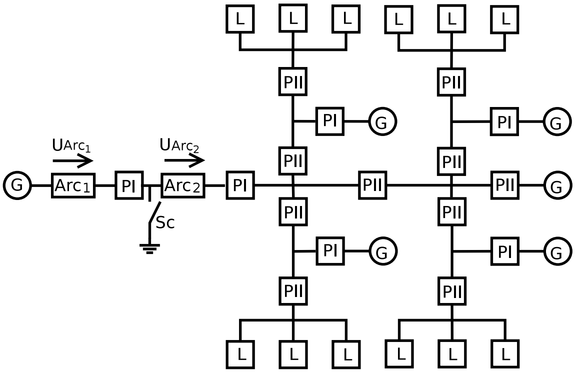


Figure 5.6: Block representation of one phase of the synthetic network 3, composed of 36 block models, 16 node blocks and two Habedank models

The total number of differential variables was 324 for this network. The parameters for each block model type of the previous different block diagrams are:

- G: $R = 0.001\Omega$, $L = 3.52mH$, $C = 1.98\mu F$, $R_C = 100\Omega$, and
$$e(t) = \begin{cases} 591960t\cos(100\pi t) & \text{if t<0.1} \\ 59196\cos(100\pi t) & \text{else} \end{cases};$$

- L: $R = 1\Omega$, $L = 35.2mH$ and $C = 1.98\mu F$;

- PI-section: $R = 10\mu\Omega$, $L = 20\mu H$ and $C_1 = C_2 = 20\mu F$;

- Double PII-section: $R_1 = R_2 = 10\mu\Omega$, $L_1 = L_2 = 20\mu H$ and $C_1 = C_2 = C_3 = 20\mu F$;

- Arc: $\tau_c = 12\mu s$, $U_c = 5kV$, $\tau_m = 4\mu s$ and $P_m = 2MW$.

- Switch: $R_{sw} = 0.1m\Omega$.

The scenario of these test cases is the following:

- From $t = 0s$ to $t < 0.345s$: All currents and voltages of the block model diagrams reach their steady state;

- At $t = 0.345s$: A short-circuit is initiated by closing the switching device SC;

- At $t = 0.5s$: Arc models are activated;

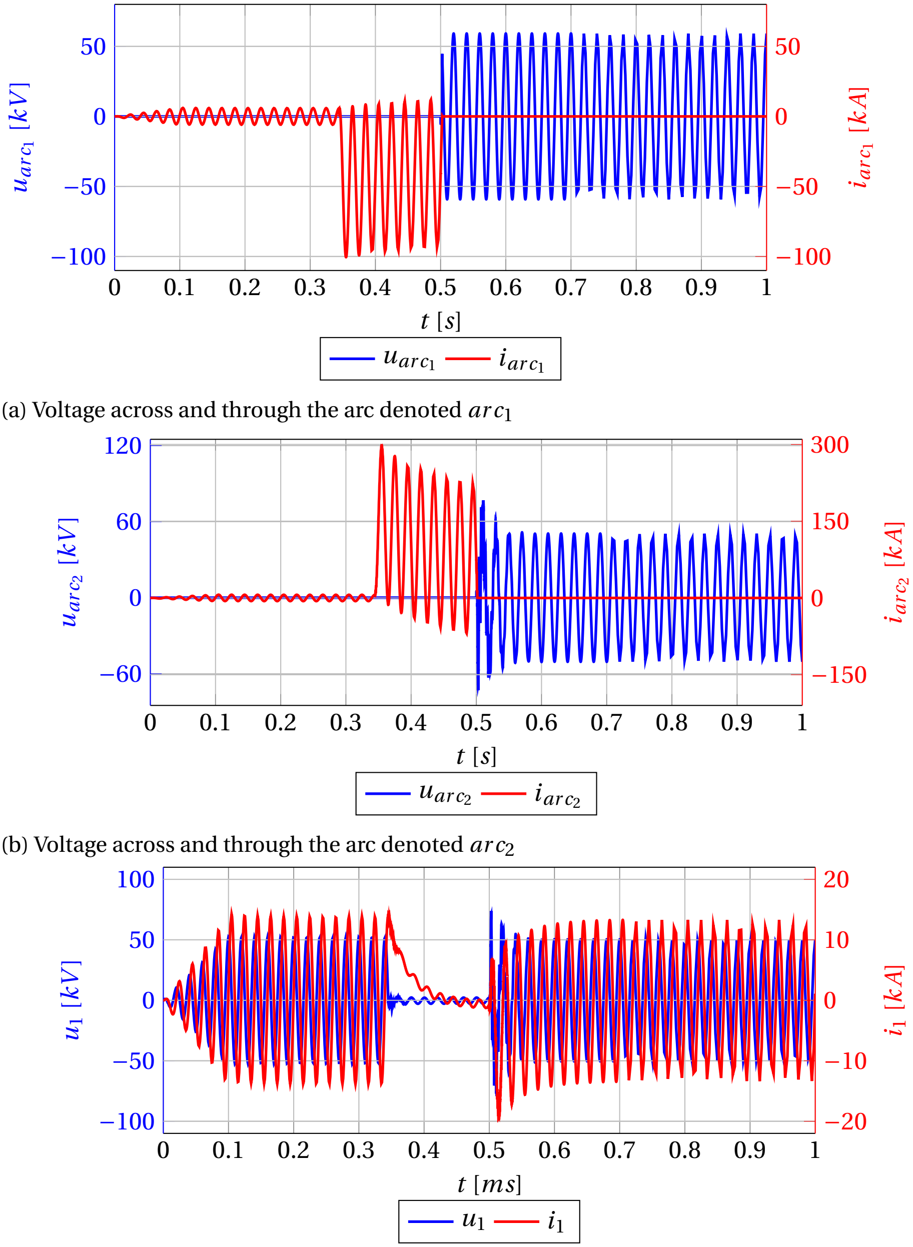
- From $t > 0.5s$ to $t = 1s$: Arc models either interrupt or do not interrupt the short-circuit current in accordance with the parameters of the block diagrams.

In the first instance, the computation time of the previous scenario is studied for the network of Figure 5.5. The numerical and the analytical Jacobian during the simulation have then been investigated.

Secondly, the block diagram shown in Figure 5.6 is examined. Now, the short-circuit is only present in the first phase. Then, the computation time of the overall simulation is investigated when the analytical and numerical Jacobian are used.

5.3.2. PRESENTATION OF THE RESULTS AND DISCUSSION

Figure 5.7a and 5.7b present the time domain solution of the current and voltage related to the arc model, denoted 1 and 2 in Figure 5.5. Then, the voltage and current denoted U and I are drawn in Figure 5.7c. Finally, Table 5.2 shows the computation of the overall simulation for the analytical and numerical calculation of the Jacobian matrix.



(a) Voltage across and through the arc denoted arc_1

(b) Voltage across and through the arc denoted arc_2

(c) Voltage at the node 1 and the current flowing through it

According to Figures 5.7a and 5.7b, the interruption of the current to both sides of the PI-section, which contains the short-circuit, was achieved. Although, once the arc model number two interrupted the short-circuit current, the voltage and the current of the node 1 on Figure 5.7c reached a new steady state because the topology of the network changed.

Numerical Jacobian	Analytical Jacobian
90.3	20.3

Table 5.2: Computation time in seconds of the network shown in Figure 5.5

According to Table 5.2, the computation time was four times faster with the analytical Jacobian than with the numerical Jacobian. This result was expected because usually, it is computationally less expensive to compute the analytical Jacobian instead of the numerical Jacobian.

Now, the results of the second part are shown. Table 5.3 gives an indication of the computation time for the overall simulation.

Numerical Jacobian	Analytical Jacobian
>3600	211

Table 5.3: Computation time in second of the network shown in Figure 5.6

Table 5.3 shows that for such a relatively small 3-phase network, the application of several arc models made it necessary to compute the Jacobian matrix analytically to obtain an acceptable computation time.

5.4. CONCLUSION

In this chapter, the speed-up in getting the time domain solution by using the block modeling method is shown. The simulation of linear networks was 11 times faster when using a combination of this new approach to model power systems and advanced numerical integration methods. Then, the simulation of an electrical network including several arc models can be sped up by giving the mathematical expression of the Jacobian matrix.

Finally, the mathematical expression of a network when a switching action appears has been improved by using this modeling approach, but the computation of the time domain solution can still be improved. The numerical integration methods applied were implicit, and the overall computation time can be improved, as explained and shown in the next chapters.

REFERENCES

[1] *PSCAD/EMTDC: Electromagnetic transients program including dc systems* (1994).

[2] *MATLAB SimPowerSystems for Use with Simulink User's Guide, Version 4.1.1.*

[3] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F.

Smith, S. Zampini, H. Zhang, and H. Zhang, *PETSc Users Manual*, Tech. Rep. ANL-95/11 - Revision 3.7 (Argonne National Laboratory, 2016).

[4] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *Efficient management of parallelism in object oriented numerical software libraries,* in *Modern Software Tools in Scientific Computing,* edited by E. Arge, A. M. Bruaset, and H. P. Langtangen (Birkhäuser Press, 1997) pp. 163–202.

[5] C. A. Kennedy and M. H. Carpenter, *Additive Runge-Kutta schemes for convection-diffusion-reaction equations,* Applied Numerical Mathematics **44**, 139 (2003).

6

Inaccurate solver for the simulation of power systems

6.1. Introduction

The design and daily operation of electrical networks are currently undergoing rapid changes. New projects of HVDC link, large offshore wind farms, and the interconnection of power systems make the electrical network increasingly larger and complex. In addition, all these changes rely on the use of mathematical modeling and numerical simulation tools. The development of new numerical techniques is, therefore, mandatory.

In this chapter, the efficiency of computing the time variation of the current and voltage in an electrical network is studied [1–3]. In addition, the block modeling method is applied to arrive at the state space representation of the electrical network, with the properties shown in Chapter 4. The resulting system of ordinary differential equations is typically stiff due to the parameters of the different network elements. Also, only the linear block model shown in Table 4.1 is employed. The network topology is assumed to be fixed throughout the simulation. Moreover, to simplify the studies, the exclusion of switching action is made during the simulation. Additionally, zero initial conditions are imposed. These initial conditions give rise to a set of initial transients to appear. Once these transients have disappeared, the system oscillates in its steady state regime.

The resulting system of ordinary differential equations can be integrated, as shown in Chapter 3 [4, 5]. The stiffness present in the problem motivates the use of implicit methods. The requirement of solving systems of linear equations renders these methods computationally expensive. Two directions of research to cope with this issue have been developed in the literature. The first resulted in diagonally implicit Runge-Kutta that retains good stability properties while being more efficient than their fully implicit counterpart. The second gives rise to a body of research showing how to efficiently deploy iterative linear and non-linear solvers in the time-step loop. The development of fast linear and non-linear solvers embedded in backward difference methods was considered, for example, [6–9]. For fully implicit Runge-Kutta methods, preconditioners exploiting the Kronecker product were considered in [10, 11]. Domain decomposition iterative solvers for transients in power systems have recently been studied in [12].

The ARK4(3)6L[2]SA-ESDIRK [13] method shown in Section 3.3.3 is used for this study. In addition, this integration method can handle either a fixed or adaptively constructed time-step. This method requires solving a sequence of five linear systems at each time-step. The solution to each of these systems provides an intermediate stage in updating the current numerical approximation to the next time-step. The size of the linear system is equal to the number of differential equations. To alleviate the computational bottleneck of the linear system's solving process, iterative methods can be used. Both stationary and non-stationary iterative methods for linear systems exist [14–16]. An incomplete LU (iLU) factorization is used with partial fill-in after re-ordering to split the coefficient matrix. This splitting is then used to construct a stationary iterative method. Also, a limited number of iterations are employed, and the result obtained in this way is denoted as an inexact linear solver. The accuracy of this solver can be varied by altering the amount of fill-in or by combining it with an outer GMRES iteration. The inexact linear solver is embedded as an inner iteration in a Runge-Kutta time integration method that acts as an outer iteration. The speed of convergence of the inner iteration then depends on factors such as the size of the time-step and the stiffness of the problem. The interest of this chapter is to theoretically and computationally investigate the efficiency of the inner-outer scheme.

This chapter is organized as follows. Section 6.2 states the problem's definition. Then, a definition of approximate solvers is given in Section 6.3. It is followed by the conclusion in Section 6.4.

6.2. Definition of problem

In this chapter, the numerical integration method used is the ARK4(3)6L[2]SA-ESDIRK method shown in Section 3.3.3. This numerical approach proposes to solve five dependent linear systems of equations (Eq. 3.27), which can be summarized as:

$$\mathscr{A}(\Delta t)k_i = q_i \quad \text{for } 2 \le i \le 6, \tag{6.1}$$

where $\mathscr{A}(\Delta t) = I - \gamma\Delta t A$ and $q_i = f\left(u_n + \sum_{j=1}^{i-1} a_{ij}\Delta t k_j, t_n + c_i\Delta t\right)$.

For calculating the k_i vectors, the LU factorization or the GMRES method, with the incomplete LU factorization (iLU) as preconditioner, are usually used to solve Eq. 6.1. In this chapter, more efficent alternatives are considered .

6.3. Approximate linear solver definition

In this section, instead of using an LU factorization or GMRES method, inaccurate solvers are studied. In particular, this section studies the interaction between the inaccurate solutions, the stages of the numerical integration method, and the time-stepping algorithm.

First, the qualities of an approximate linear solver are:

- Provides a good approximation of the solution of the system of equations;

- Faster than Gaussian elimination.

Second, a Basic Iterative Method (BIM) - where the stopping criterion has been modified - could be an inexact linear solver. Although the solution is approximated, it is only one component of the time domain solution. Here, one iteration of iLU(ℓ) factorization as BIM is used as an inexact linear solver. Finally to construct an iterative scheme, the decomposition of the matrix $\mathscr{A}(\Delta t)$ can be made as:

$$\mathscr{A}(\Delta t) = \mathscr{M}(\Delta t) - \mathscr{N}(\Delta t), \tag{6.2}$$

Then, the iLU factorization of the matrix $\mathscr{A}(\Delta t)$ of Eq. 3.27 can be defined as:

$$\mathscr{A}(\Delta t) = L(\Delta t)U(\Delta t) - R(\Delta t), \tag{6.3}$$

where the matrix $L(\Delta t)$ is the lower triangular matrix and the matrix $U(\Delta t)$ is the upper triangular matrix of the iLU factorization. Moreover, the matrix $R(\Delta t)$ is the residual matrix of the iLU factorization. This matrix is null in the case of the LU factorization. In addition, at each time-step, only one factorization is effectuated due to the definition of the matrix $\mathscr{A}(\Delta t)$ (Eq. 6.1). Moreover, according to Eq. 6.2, matrices $\mathscr{M}(\Delta t)$ and $\mathscr{N}(\Delta t)$ are defined as:

$$\mathscr{M}(\Delta t) = L(\Delta t)U(\Delta t), \tag{6.4}$$

$$\mathscr{N}(\Delta t) = R(\Delta t). \tag{6.5}$$

The relation between the residual vector of the stage i at the iteration m (r_i^m) is defined as:

$$r_i^m = q_i - \mathscr{A}(\Delta t)\tilde{k}_i^m, \tag{6.6}$$

and the error vector is expressed as:

$$\epsilon_i^m = k_i - \tilde{k}_i^m, \tag{6.7}$$

where $\tilde{k}_i^m$ is the approximate solution of the stage i after m iterations. The norm of r_i^m informs the quality of the approximate linear solver. The residual vector is usually easier to compute than the error vector because the vector k_i is unknown. In fact, if the norm of the residual error becomes small, the norm of the error vector is also small. Finally, the normalized residual at each stage (z_i) is defined as:

$$z_i = \frac{\|r_i^m\|_2}{\|q_i\|_2}, \tag{6.8}$$

for $2 \leq i \leq s$. In order to judge the inexact solver, the spectral radius can be employed as:

$$\rho(I - \mathscr{M}(\Delta t)^{-1}\mathscr{A}(\Delta t)) \tag{6.9}$$

From the definition of the spectral radius, the solution of the linear system of equations converges only when ρ is less than one. Moreover, an asymptotic convergence or divergence can be observed after some iterations. Finally, a super linear convergence can be seen after one iteration, when the residual error values are plotted in function of the number of iterations.

Here, one iteration is employed. Then, ρ needs to be smaller than one. Besides, as shown in Eq. 6.1, the time-step plays an important role in the converge of the BIM. Also, two possible strategies to control the time-step can be applied: a fixed time-step method and an adaptive time-step. In addition, as shown in Eq. 6.3, the level of fill-in ℓ affects the spectral radius.

6.4. CONCLUSION

For a fixed level of fill-in ℓ of the iLU factorization and a fixed time-step, several cases are possible. In the case where ρ is strictly superior to one, the time domain solution will diverge and the simulation will stop due to the accumulation of the error after each time-step, because the inexact solver diverges. Then, for being able to simulate the electrical network under consideration, two actions can be taken. The first one is to decrease the time-step. The system of equations will then become mainly diagonally dominant, and the spectral radius will be reduced. The second action is to increase the level of fill-in ℓ of the iLU factorization. Then, the matrix decomposition will be closer to the LU factorization. Finally, when the level of fill-in becomes too large, the iLU factorization becomes the LU factorization and it will not lead to a speed-up.

For a fixed level of fill-in ℓ of the iLU factorization and an adaptive time-stepping algorithm, the time-step selection procedure will reduce or increase as a function of error between the order p and p-1 of the numerical integration method. For example, if the inaccurate solutions of the k_i vectors have large residual errors, then the error between between the two orders of the method will be considerable. Then, the time-step will be rejected and a smaller time-step will be employed. In this way, the non-convergence is avoided at the expensive of a larger number of time-steps.

Finally during a simulation, the variation of the level of fill-in ℓ is not considered. This non-variation comes from the fact that a symbolic factorization and numerical factorization is employed, because it will speed-up the computation time. Then, if there is a variation over time of the level of fill-in ℓ of the iLU factorization, a new symbolic factorization would have to be computed, and will slow down the required computation time.

REFERENCES

[1] L. van der Sluis, *Transients in Power Systems* (John Wiley & Sons, Chichester, 2001).

[2] J. Grainger and W. Stevenson, *Power System Analysis*, Electrical Engineering Series (McGraw-Hill, New York, 1994).

[3] N. Watson and J. Arrillaga, *Power systems electromagnetic transients simulation*, Vol. 39 (Institution of Electrical Engineers, London, 2003).

[4] G. Wanner and E. Hairer, *Solving ordinary differential equations II : Stiff and Differential-Algebraic Problems*, Vol. 1 (Springer Berlin Heidelberg, Berlin Heidelberg, 2010).

[5] W. Hundsdorfer and J. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations* (Springer, New York, 2003).

[6] T. F. Chan and K. R. Jackson, *The use of iterative linear-equation solvers in codes for large systems of stiff IVPs for ODEs,* SIAM journal on scientific and statistical computing **7**, 378 (1986).

[7] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold, *Using Krylov methods in the solution of large-scale differential-algebraic systems,* SIAM Journal on Scientific Computing **15**, 1467 (1994).

[8] F. Perini, E. Galligani, and R. D. Reitz, *A study of direct and Krylov iterative sparse solver techniques to approach linear scaling of the integration of chemical kinetics with detailed combustion mechanisms,* Combustion and Flame **161**, 1180 (2014).

[9] J. E. Pessanha, A. A. Paz, R. Prada, and C. P. Poma, *Making use of BDF-GMRES methods for solving short and long-term dynamics in power systems,* International Journal of Electrical Power & Energy Systems **45**, 293 (2013).

[10] L. O. Jay, *Inexact simplified Newton iterations for implicit Runge-Kutta methods,* SIAM Journal on Numerical Analysis **38**, 1369 (2000).

[11] K. Dekker, *Partitioned Krylov subspace iteration in implicit Runge-Kutta methods,* Linear Algebra and its Applications **431**, 488 (2009).

[12] P. Aristidou, S. Lebeau, and T. Van Cutsem, *Power system dynamic simulations using a parallel two-level Schur-complement decomposition,* IEEE Transactions on Power Systems **31**, 3984 (2016).

[13] C. A. Kennedy and M. H. Carpenter, *Additive Runge-Kutta schemes for convection-diffusion-reaction equations,* Applied Numerical Mathematics **44**, 139 (2003).

[14] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Frontiers in Applied Mathematics, Vol. 17 (SIAM, Philadelphia, 1997).

[15] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. (Society for Industrial and Applied Mathematics, Philadelphia, 2003).

[16] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, Cambridge Monographs on Applied and Computational Mathematics, Vol. 13 (Cambridge University Press, New York, 2003).

7

Approximate solver for the simulations of power system

7.1. Introduction

In this chapter, various power systems are studied to demostrate the use of approximate linear solvers during the numerical integration process. As presented in Chapter 6, two sizes of power systems are considered - small and large. In order to describe the relationship between approximate solvers, time-step, size of the power system, and stiffness, this chapter focuses first on the use of a fixed time-step, and then second, on the utilization of an adaptive time-stepping algorithm. Finally, for the same precision for the larger test cases, a speed-up of five can be noticed in the case of the stiffer large power system.

The PETSc software library is employed for the implementation. At each time-step, a linear solver [1, 2], which is computed by the Krylov Subspace (KSP) and the preconditioner (PC) methods of this library, is used. Simulations are performed on a Linux system with an Intel (R) Core(TM) 2 Duo 2.7GHz CPU processor, with 4GB RAM.

This chapter is organized as follows: Section 7.2 presents and discusses the various power systems and experiments; Section 7.3 offers the results and explains them; and, Section 7.4 states the conclusion of the utilization of approximate solvers.

7.2. Test case presentation

In this section, details of the test cases are given, in particular, on the time-stepping procedures and the linear solution methods used in the numerical experiments.

The construction of four synthetic power systems is made so that there are two small and two large ones. The small and large systems contain $d = 1,034$ and $d = 20,999$ differential variables, respectively. Each system is created by interconnecting - via transmission lines - a set of generators and loads. The block modeling method is applied to generate the matrices A and B in the linear state space representation Eq. 3.1 of the power system (Chapter 4). The smallest time constant, and therefore the stiffness of the system of ODEs, is determined by the largest eigenvalue of the matrix A. The eigenvalues of the matrix A are determined by the topology of the network and the lumped elements properties of the different components. These properties are the resistance, capacitance and inductance values. The construction is made such that the spectral radius of the matrix A is more sensitive to the changes in the values of the lumped elements than to the changes in the topology of the network. This fact allows for the increase in stiffness of a power system, by changing the range of the lumped elements values without changing the topology of the system - and therefore - the non-zero structure of the matrices A and B. It also allows for the increase in size of the power system, while simultaneously preserving its smallest time constant. The ratio between the largest and smallest eigenvalue is 3×10^8 for the non-stiff power systems, and 7×10^{12} and 1×10^{13} for the small and large stiff power system, respectively. In all four cases, the input function $g(t)$ has a frequency of 50 Hz.

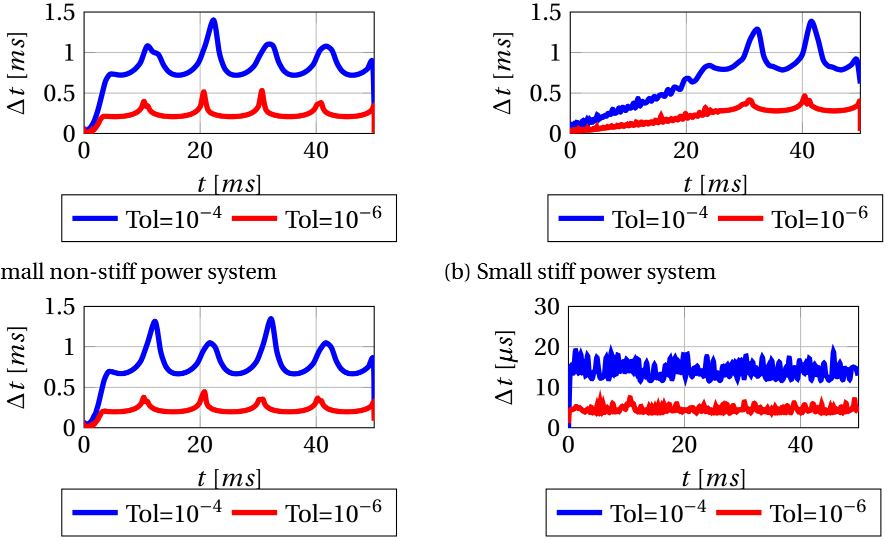
The system of ODEs (Eq. 3.1) is integrated with zero initial conditions ($x_0 = 0$) from $t = 0$ to $t = 50$ ms using the fourth order SDIRK (ARK4) method with either a fixed (Alg. 1) or adaptive time-step (Alg. 2). In the former case, various time steps are used. In the latter case, two absolute tolerance values will be imposed. To illustrate the adaptive selection of the time-step, Figure 7.1 shows the evolution of the time-step over time in integrating the four different power systems. Also, two absolute tolerances Tol are used, and they are defined in Eq. 3.26, namely $\mathrm{Tol} = 10^{-4}$ and $\mathrm{Tol} = 10^{-6}$.

Algorithm 1: Fixed time-step

```
Data: $A$, $B$, $g(t)$, $x(0)$, $t_{final}$ and $\Delta t$
begin
    $t_0 = 0$
    $u_0 = x(0)$
    $n = 0$
    AMD re-ordering method on $\mathscr{A}(\Delta t)$
    Factorization of the re-ordered matrix $\mathscr{A}(\Delta t)$
    while $t < t_{final}$ do
        Solve the linear systems of equations
        Update $u_{n+1}$
        $t_{n+1} = t_n + \Delta t$
        $n = n + 1$
```

Algorithm 2: Adaptive time-step

```
Data: $A$, $B$, $g(t)$, $x(0)$, $t_{final}$ and $Tol$
begin
    $t_n = 0$
    $u_0 = x(0)$
    $n = 0$
    $\Delta t = 10^{-3}$
    AMD re-ordering method on $\mathscr{A}(\Delta t)$
    Symbolic factorization of the re-ordered matrix $\mathscr{A}(\Delta t)$
    while $t < t_{final}$ do
        Numerical factorization
        Solve the linear systems of equations
        Calculate $e_{n+1}$
        if $e_{n+1} \leq Tol$ then
            Update $u_{n+1}$
            $t_{n+1} = t_n + \Delta t$
            $n = n + 1$
        Calculate new $\Delta t$
```



(a) Small non-stiff power system

(b) Small stiff power system

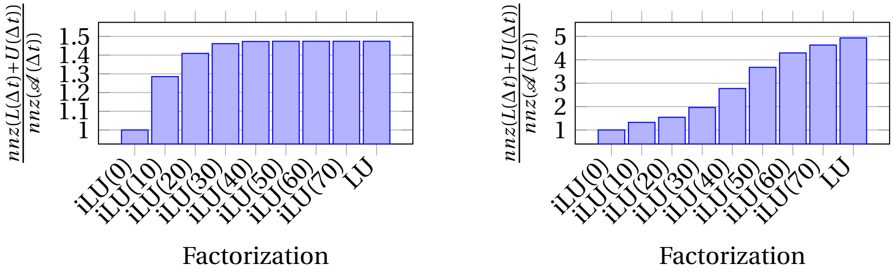
(c) Large non-stiff power system

(d) Large stiff power system

Figure 7.1: Time evolution of the time-step in cases where adaptive time-stepping with a tolerance $Tol = 10^{-4}$ and $Tol = 10^{-6}$ is used with a direct solver for the four power systems.

Figure 7.1 shows that - as expected - the more stringent tolerance results in a smaller time-step. The figure also indicates that for the small and large non-stiff system, the time-step initially grows to oscillate subsequently between a minimum and maximum value. As the stiffness of the small and large systems is approximately the same, the minimum and maximum values of the time-step are roughly equal. This oscillation indicates that the system is in a steady state. The consecutive peaks in the time-steps are separately by approximately 10 ms, the half period of the 50 Hz excitation. For the small stiff power system, the growth of time-step and reaching the steady state requires a longer period. Once the steady state is obtained, the time-step oscillates between a minimum and maximum value, as in the small non-stiff system. For the large stiff system, the time-step remains uniformly inadequate in time. To obtain a larger time-step, the simulation needs a final time superior of 9.6 seconds with a tolerance of Tol $= 10^{-4}$. The ARK4 time-integration method involves six stages and requires the solution of five linear systems of size $d \times d$ with the same coefficient matrix $\mathscr{A}(\Delta t) = I - \Delta t \gamma A$ - as shown in Section 6.2. The results are obtained by both iterative and direct solution methods. In the case of a fixed time-step, the matrix $\mathscr{A}(\Delta t)$ is AMD reordered and LU-factored before the time-stepping loop (Alg. 1). In the case of a variable time-step, only the reordering, and the symbolic factorization can be performed before the time-stepping loop (Alg. 2). As iterative solution method, iLU(ℓ) as a BIM and as a preconditioner for GMRES for various values of ℓ is used. By using iLU(ℓ) as a BIM, a single iteration is always employed. The accuracy of this approximate solver is increased by increasing ℓ. By using iLU(ℓ) as a preconditioner for GMRES, one or five iterations are performed. As different stages have no immediate relation, a zero initial guess is made.

The efficiency of iLU(ℓ) is determined by the balance of its computational cost and its speed of convergence. The computational cost is determined by the number of non-zeros in the $L(\Delta t)$ and $U(\Delta t)$ factors, as shown in Figure 7.2.

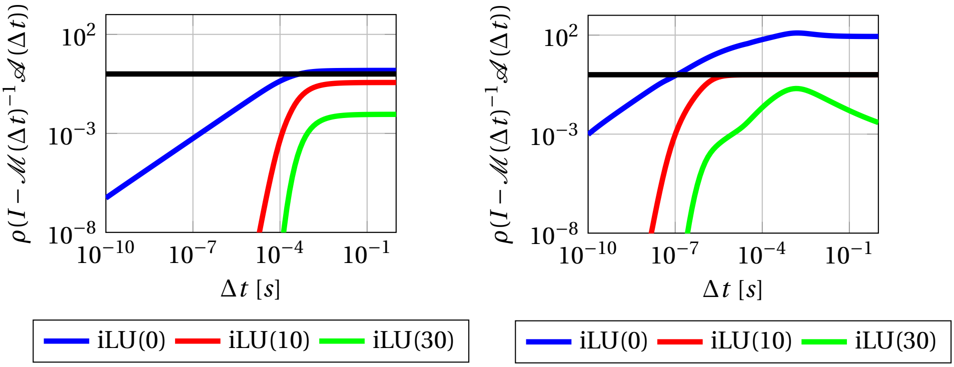


(a) Small power system

(b) Large power system

Figure 7.2: Normalized number of non-zeros in the incomplete factorization of $\mathscr{A}(\Delta t)$ for various levels of fill-in, and the LU factorization of $\mathscr{A}(\Delta t)$ for the small (a) and large (b) power systems.

Figure 7.2 shows a number scaled by the number of non-zeros in the matrix A for various levels of fill-in ℓ after AMD reordering for both the small and large power system. The number of non-zeros of $I - \Delta t \gamma A$ is equal to the number of non-zeros of $\mathscr{A}(\Delta t)$ for all possible values of Δt, and does not vary with the stiffness of the power system. Figure 7.2a shows that, for the small system, the fill-in $\ell = 20$ results in many non-zeros of $L(\Delta t)$ and $U(\Delta t)$ that is almost as large as the number of non-zeros of $\mathscr{A}(\Delta t)$. For the large system, the fill-in $\ell = 40$ results in a number of non-zeros of $L(\Delta t)$ and $U(\Delta t)$ - is roughly half the number of non-zeros of the LU decomposition of $\mathscr{A}(\Delta t)$. Now, the speed of converge of iLU(ℓ) is determined by the spectral radius of the error propagation matrix (Eq. 6.9), and it is shown in Figure 7.3 for the small power systems.



(a) Small non-stiff power system

(b) Small stiff power system

Figure 7.3: Spectral radius of the incomplete factorization used as a BIM for various levels of fill-in as a function of the time-step for the small non-stiff (a) and stiff (b) power systems.

Figure 7.3 shows the spectral radius as a function of the time-step Δt for the small non-stiff and stiff power system for various levels of fill-in ℓ. This figure shows a decrease of the spectral radius for decreasing Δt. By reducing Δt, the matrix $\mathscr{A}(\Delta t)$ increasingly approximates the identity matrix better, and the iLU approximation becomes closer to the complete LU factorization that - in the limit of $\Delta t \to 0$ - results in a spectral radius equal to zero. A larger level of fill-in leads to a smaller spectral radius. A smaller one is easier to obtain for the non-stiff power system. In the case of the stiff power system, a low level of fill-in and a large time-step results in a non-convergent solver.

7.3. RESULTS AND DISCUSSIONS

In this section, numerical results obtained are shown in two subsections. While the first one discusses the fixed time-step, the second one investigates the adaptive time-step.

7.3.1. FIXED TIME-STEP

In this section, the results obtained using a fixed time-step for the small non-stiff and the small stiff power system are discussed first. Then, the results obtained using a fixed time-step again for the large non-stiff, and large stiff power systems are investigated. This order displays the relationship between the stiffness and parameters in the approximate linear solver.

Tables 7.1 and 7.2 show the computation time of Alg. 1 with various linear solvers. The "NC" character represents the non-convergence of a simulation. The non-convergence can appear for two reasons: first, the simulation may have ended before the final time was reached, due to the accumulation of error; second, the output may have given an incorrect result, such as an impossible voltage or current, and this simulation could have stopped after a few more time-steps due to the accumulation of error.

Linear solver	Small non-stiff				Small stiff			
Δt	$1ms$	$0.1ms$	$10\mu s$	$1\mu s$	$1ms$	$0.1ms$	$10\mu s$	$1\mu s$
LU	0.039	0.194	1.752	17.287	0.039	0.194	1.743	17.280
ILU(0)	*NC*	*NC*	1.425	14.071	*NC*	*NC*	*NC*	*NC*
ILU(10)	0.036	0.168	1.495	14.716	*NC*	*NC*	*NC*	*NC*
ILU(30)	0.036	0.170	1.521	14.979	*NC*	0.171	1.518	15.090
GMRES ILU(0) 1it	*NC*	*NC*	2.510	24.973	*NC*	*NC*	*NC*	*NC*
GMRES ILU(10) 1it	*NC*	0.284	2.655	26.257	*NC*	*NC*	*NC*	*NC*
GMRES ILU(30) 1it	0.048	0.290	2.699	26.828	*NC*	0.289	2.697	26.783
GMRES ILU(0) 5it	*NC*	0.665	4.373	34.133	*NC*	*NC*	*NC*	*NC*
GMRES ILU(10) 5it	0.090	0.385	2.653	26.439	*NC*	*NC*	*NC*	68.471
GMRES ILU(30) 5it	0.058	0.290	2.719	26.834	0.069	0.402	3.705	26.999

Table 7.1: CPU time in seconds for the small non-stiff and the small stiff power systems, for the direct and various iterative linear solvers, in case of fixed time-steps with various values of Δt

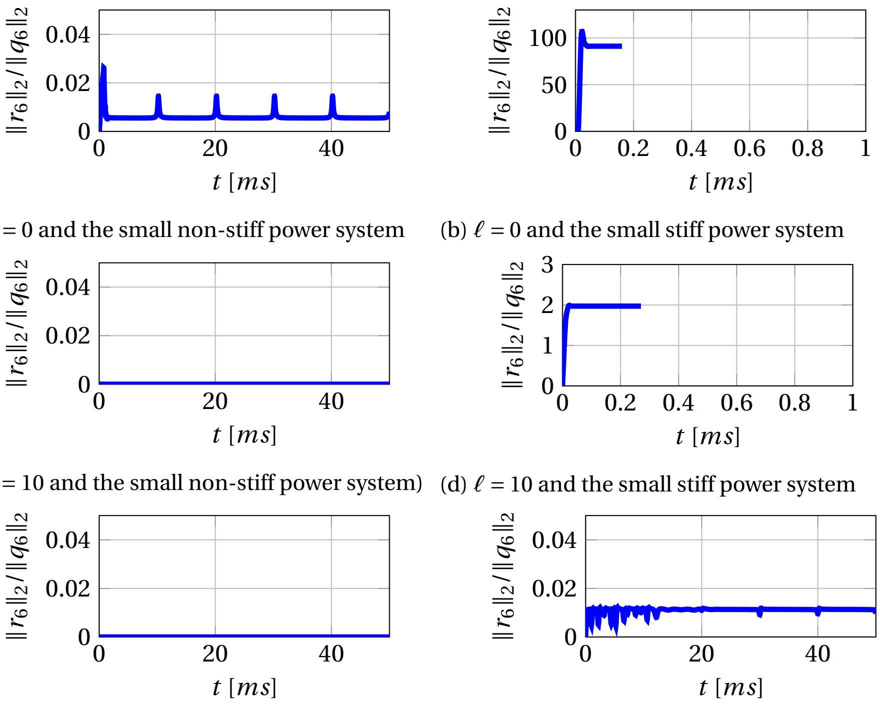
Linear solver	lLarge non-stiff				Large stiff			
Δt	$1ms$	$0.1ms$	$10\mu s$	$1\mu s$	$1ms$	$0.1ms$	$10\mu s$	$1\mu s$
LU	20.17	26.97	94.33	770.30	20.70	27.01	95.24	765.45
ILU(0)	*NC*	*NC*	45.44	276.81	*NC*	*NC*	*NC*	*NC*
ILU(10)	20.01	22.18	47.32	293.06	*NC*	*NC*	*NC*	*NC*
ILU(30)	19.74	22.22	49.12	311.25	*NC*	*NC*	*NC*	310.03
ILU(40)	20.16	23.02	50.84	332.45	*NC*	*NC*	51.47	330.91
GMRES ILU(0)L 1it	*NC*	*NC*	68.23	503.15	*NC*	*NC*	*NC*	*NC*
GMRES ILU(10) 1it	*NC*	25.06	72.39	539.88	*NC*	*NC*	*NC*	*NC*
GMRES ILU(30) 1it	*NC*	25.06	72.39	539.88	*NC*	*NC*	*NC*	590.15
GMRES ILU(40) 1it	20.29	26.24	82.18	643.08	*NC*	*NC*	83.72	646.49
GMRES ILU(0) 5it	*NC*	33.00	107.48	700.00	*NC*	*NC*	*NC*	*NC*
GMRES ILU(10) 5it	21.37	27.10	72.83	572.19	*NC*	*NC*	*NC*	1436.65
GMRES ILU(30) 5it	21.63	25.99	88.47	592.11	*NC*	*NC*	155.89	784.61
GMRES ILU(40) 5it	20.60	26.15	82.47	645.35	22.88	36.19	117.75	653.57

Table 7.2: Total CPU time (in seconds) for the large non-stiff and the large stiff power systems, for both direct and various iterative linear solvers, in case of fixed time-steps with various values of Δt ($\tau = 10ms$)

Tables 7.1 and 7.2 show that when the Δt decreases, the number of time-steps increases, and then the computation time increases. The computation effort for computing u_n is the same for any Δt when LU or one iteration of iLU(ℓ) factorization is applied. Although the solution converged for the non-stiff power systems, the solution diverged for the stiff power systems for the same ℓ and Δt. This remark is expected from Figure 7.3 because the spectral radius of these stiff power systems for those simulations is almost one for such Δt. When the size of the power systems increases, the level of fill-in required increases, or a smaller Δt has to be employed. These increases are understandable from the fact that the iLUℓ) factorization depends on the time-step, and the size is also important.

In terms of computation time, the use of an inexact linear solver for the small system permits a gain of 2s with $\ell = 30$ and $\Delta t = 1\mu s$ for a simulation of 50ms. However, for the larger test cases, a speed-up of a factor two can be observed with $\ell = 30$ and $\Delta t = 1\mu s$. This speed-up almost corresponds to the difference of the ratio of the number of non-zero elements of the factorization, divided by the number of non-zero elements of $\mathscr{A}(\Delta\sqcup)$ (Figure 7.2). Finally, the use of GMRES with iLU(ℓ) as preconditioner requires more computation time than LU or one iteration of iLU(ℓ) factorization. Meanwhile, for $\Delta t = 1ms$ and the large stiff power system, the time domain solution converges after five iterations. However, the utilization of such a time-step ($\Delta t = 1ms$) is not recommended in case of transient simulations because it cannot capture the fast transient oscillations. Finally, the use of an inexact solver can improve the computation time.

Figures 7.4 and 7.5 represent the normalized residual of the sixth stage, which is characteristic of the other stages, of ARK4(3)6L[2]SA-ESDIRK method for various level of fill-in ℓ, stiffnesses, and power system sizes.



(a) $\ell = 0$ and the small non-stiff power system

(b) $\ell = 0$ and the small stiff power system

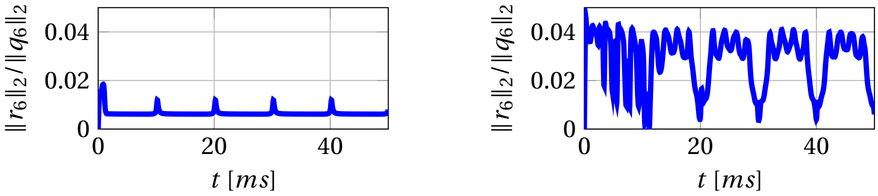
(c) $\ell = 10$ and the small non-stiff power system)

(d) $\ell = 10$ and the small stiff power system

(e) $\ell = 30$ and the small non-stiff power system

(f) $\ell = 30$ and the small stiff power system

Figure 7.4: Scaled residual norm after the sixth stage of the ARK4(3)6L[2]SA-ESDIRK method, using ILU(ℓ) as a BIM with $\ell = 0$ (a and b), $\ell = 10$ (c and d) and $\ell = 30$ (e and f) in case of a fixed time-step of $\Delta t = 10\mu s$. The left and right columns correspond to the small non-stiff and small stiff power system, respectively



(a) $\ell = 0$ and the large non-stiff system

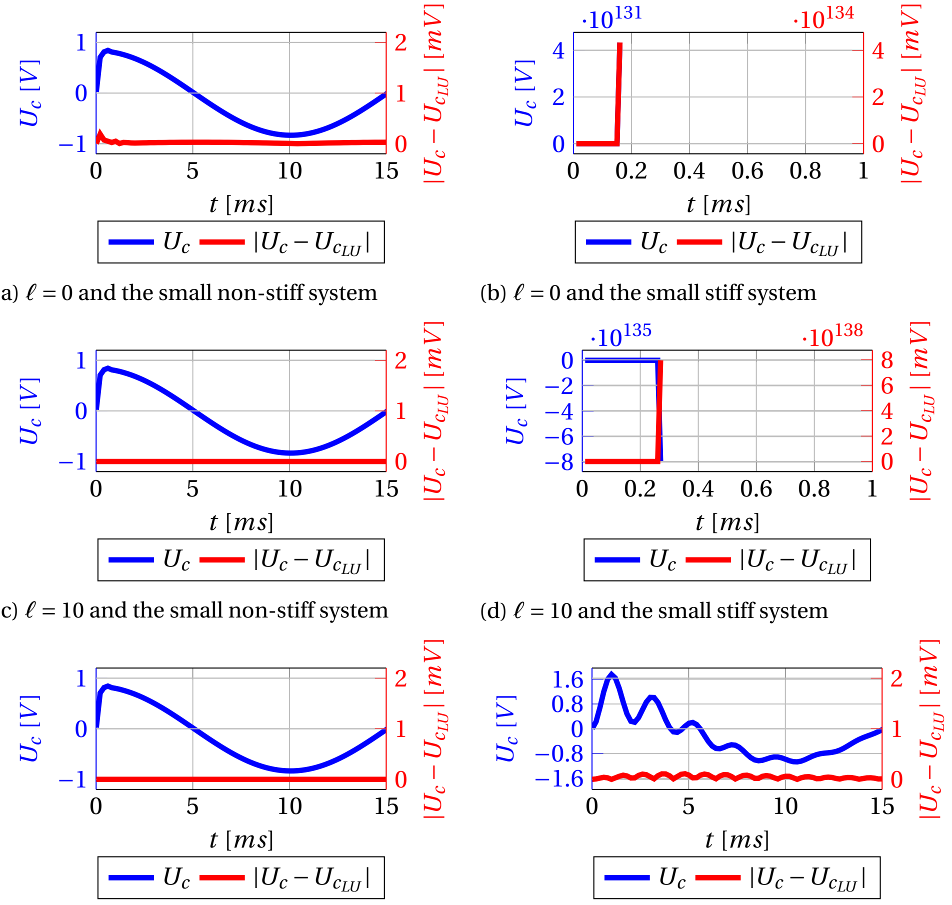
(b) $\ell = 40$ and the large stiff system

Figure 7.5: Scaled residual norm after the sixth stage of the ARK4(3)6L[2]SA-ESDIRK method, using ILU(ℓ) as a BIM with $\ell = 0$ for the large non-stiff power system (a) and $\ell = 40$ for the large stiff power system (b) in case of a fixed time-step of $\Delta t = 10\mu s$

Figure 7.4 shows that the non-stiff system was able to go to the end of the simulation with a maximal normalized residual of 0.045 for iLU(0). Also, as expected, it decreases when the level of fill-in ℓ increases. For iLU(0), the first peak appears due to the transient oscillations when the power system passed from the initial condition to the steady-state. Then every 10ms, a pick occurred when the time domain solution of the different signal crossed zero. The zero-crossing implies, in this case, the higher derivative of the system. When the stiffness increased, the simulation stopped after few time-steps if the level of fill-in ℓ was too small. In fact, the normalized error was significant in such cases. The simulation stopped because the error realized at one time-step impacts the succeeding time-steps. For this reason, the simulation stopped at an earlier time with a level of fill-in $\ell = 0$ than with $\ell = 10$. The normalized residual was 100 on average for $\ell = 0$, and 2 on average for $\ell = 10$. The non-convergence to a small residual can be extrapolated from Figure 7.3 because $\rho > 1$ for these levels of fill-in ℓ. Also, these simulations stopped because the accumulation of the error at each time-step finally approached a higher number than the computer could handle. Finally, with $\ell = 30$, the same remark as for the non-stiff power system can be concluded. That time, the duration of the transient part was almost 10ms. Although the peaks occur every 10ms for the non-stiff power systems, for the stiff power systems, a depression appeared every 10ms.

For the large power system, the same observations can be made. For this reason, the non-stiff power system with $\ell = 0$ and the stiff power system with $\ell = 40$ (Figure 7.5) is shown. The normalized residual has the same shape as that of the small non-stiff system, and same observations can be made between both non-stiff systems. For the stiff system, the maximal normalized residual is 0.5. With such a value, an error could be expected in the time domain solution. Meanwhile, it was supposed to have a lower normalized residual based on the results for the non-stiff power system.

Figures 7.6 and 7.7 represent the time domain solution of a random variable U_c for various level of fill-in ℓ, stiffnesses, and power system sizes. Also, the error between this solution and the solution obtained, when the LU factorization is used as a solver, is shown as well.



(a) $\ell = 0$ and the small non-stiff system

(b) $\ell = 0$ and the small stiff system

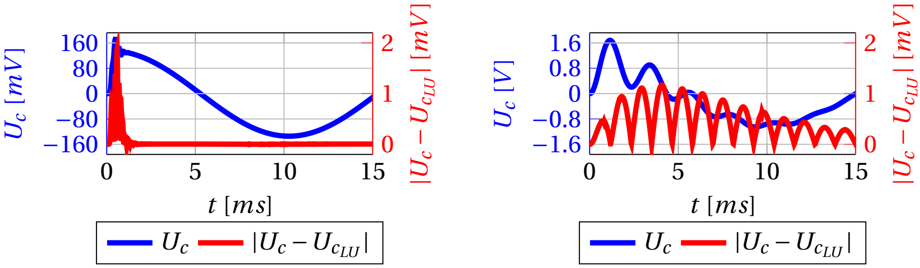
(c) $\ell = 10$ and the small non-stiff system

(d) $\ell = 10$ and the small stiff system

(e) $\ell = 30$ and the small non-stiff system

(f) $\ell = 30$ and the small stiff system

Figure 7.6: Component solution of the voltage signal through one of the capacitors $U_c(t)$ using an approximate linear solver and the absolute value of the difference between this solution and the solution $U_{c_{LU}}(t)$, computed using a direct solver in case of a fixed time-step of $\Delta t = 10\mu s$. As an approximate linear solver, $iLU(\ell)$ is used as a BIM with $\ell = 0$ (a and b), $\ell = 10$ (middle row) and $\ell = 30$ (bottom row). The left and right columns of subfigures correspond to the small non-stiff and small stiff power systems, respectively



(a) $\ell = 0$ and the large non-stiff power system

(b) $\ell = 40$ and the large stiff power system

Figure 7.7: Component solution of the voltage signal through one of the capacitors $U_c(t)$ using an approximate linear solver and the absolute value of the difference between this solution and the solution $U_{c_{LU}}(t)$, computed using a direct solver in case of a fixed time-step of $\Delta t = 10\mu s$. As an approximate linear solver, $ILU(\ell)$ is used as a BIM with $\ell = 0$ for the large non-stiff problem (a) and $\ell = 40$ for the large stiff problem (b)

Figure 7.6 reveals that the errors between the approximate linear solvers and LU on the signal U_c were equal to zero. These results are expected according to Figure 7.4 for $\ell = 10$ and $\ell = 30$. For $\ell = 0$, an error is observed. Meanwhile, this error is small (less than 1mV for a signal of 1V). This error is understandable because the choice of this Δt multiplies the residual error, which helps to get an excellent approximation of the time-domain solution. For the stiff case, the signal U_c reached an impossible voltage and then stopped suddenly. In fact, the simulation ended because some signals were higher than the maximal number accepted by the computer. Finally, U_c for a fill-in of $\ell = 30$ for the stiff power system had almost an error of 0mV. The same remarks can be applied to the non-stiff with $\ell = 0$.

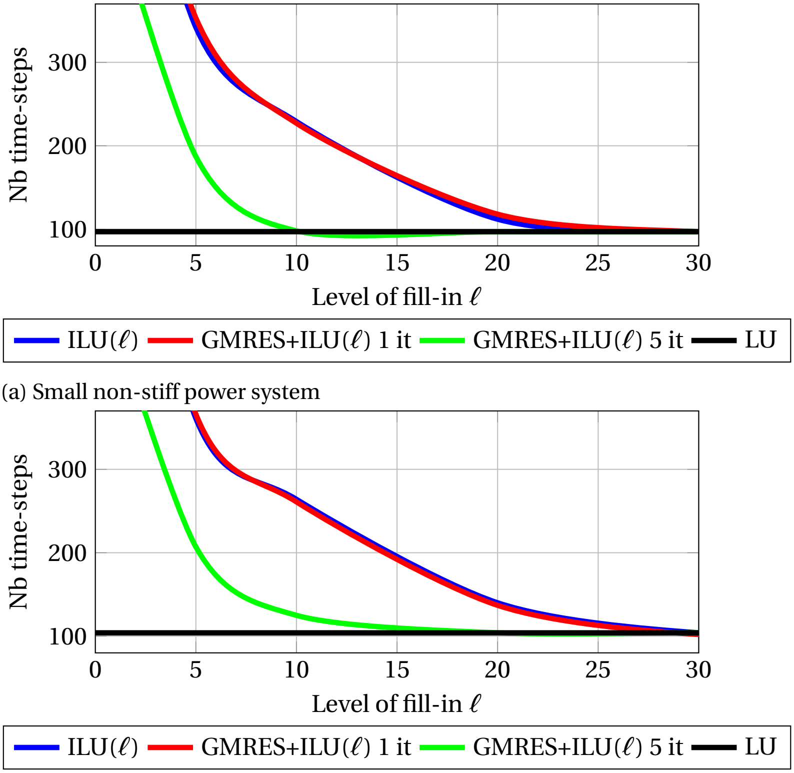
For the large power system, the same observations can be done. For this reason, the non-stiff power system with $\ell = 0$ and the stiff power system with $\ell = 40$ are studied (Figure 7.7). A peak of error occurred during the transient part for the non-stiff system. However, the error (2mV) is relatively small in comparison with the amplitude of U_c (160mV). Meanwhile, the computation time is 100s faster with the iLU(ℓ) factorization than with the LU factorization. In the case of the large stiff power system, the error is less important than with the non-stiff case. Finally, one iteration of iLU(40) was 1.63 times faster than using the LU factorization for solving the systems of equations.

The fixed time-step simulations offer an understanding of the properties of inexact solvers. According to Tables 7.1 and 7.2, there is a strong connection between the level of fill-in ℓ and the time-step for performing a simulation. Also from Figures 7.4 and 7.5, the level of fill-in depends on the size of the system of equations and the stiffness, as well, to get a small normalized residual. Moreover from Figures 7.6 and 7.7, when the time domain solution of the power system converges with the utilization of approximate solvers, the error is almost negligible. Finally, the choice of approximate solvers depends on the size of the system of equations, and its stiffness according to these results.

7.3.2. ADAPTIVE TIME-STEPPING STRATEGY

In this section, results obtained using an adaptive time-step algorithm (Alg. 1) are shown and discussed. Firstly, the presentation of the results of the non-stiff power system is made. Secondly, stiff power systems are studied. This ordering differs from the one in the previous section. It allows for the highlighting of the effects of problem size on the performance of the approximate linear solvers. The results mainly show the number of time-steps, and the overall CPU time varies as a function of the level of accuracy of the various approximate solvers, the tolerance imposed on the time-stepping procedure, and the stiffness of the problem. .

Figure 7.8 shows the evolution of the number of time-steps as a function of the level of fill-in ℓ of iLU(ℓ) as BIM, or as preconditioner for GMRES for $0 \leq t \leq 50\,\mu$s with an imposed absolute tolerance Tol $= 10^{-4}$ as a function of the level of fill-in ℓ using various approximate linear solvers. The number of time-steps using the direct solver is clearly independent of ℓ, and is represented as a vertical line as a reference.

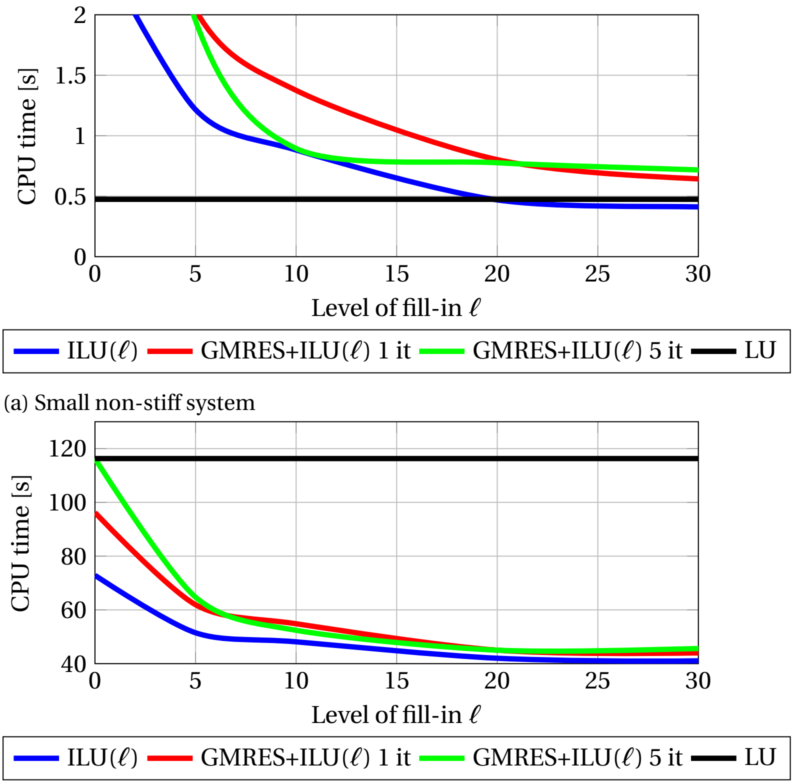


(a) Small non-stiff power system

(b) Large non-stiff power system

Figure 7.8: Number of time-steps for three approximate linear solvers and a direct solver versus the level of fill-in time integrating the small (a) and large (b) non-stiff power system with an adaptive time-step with a tolerance of 10^{-4}

Figure 7.8 shows that the number of time steps using the direct solver is - therefore - the same for both the small non-stiff and large non-stiff power systems because they have the same smallest time constant. Then, the number of iterations decreases with the increase of the level of fill-in ℓ of the iLU factorization used as BIM or as preconditioner for GMRES. In addition, several iterations of GMRES with iLU(ℓ) as preconditioner helps to reduce the number of time-steps with regard to one iteration of iLU(ℓ) or of GMRES with the same preconditioner. Also, five iterations of GMRES reach the number of time-steps required by the LU factorization faster than the other solvers. Then, when the number of time-steps are the same, increasing the level of fill-in ℓ of the iLU factorization will only increase the computation time. The same remark is valid when the other solvers increase their level of fill-in ℓ, whereas they have already reached the same number of time-steps of the direct solver. For example, a significant amount of time-steps is required for iLU(0) in comparison with LU. In fact, this remark is expected from the fixed time-step results (Section 7.3.2) and the Δt evolution (Figure 7.1). The iLU(0) does not converge with relatively large Δt ($\Delta t < 10\mu s$). However, the number of time-steps might positively impact the overall computation time. For this reason, Figure 7.9 shows the computation time of the overall simulation of the non-stiff power systems and various solvers, according to Alg. 2. In addition, the value for the direct solver is now different for small and large non-stiff systems, and is included in the figures as a reference.



(a) Small non-stiff system

(b) Large non-stiff system

Figure 7.9: Overall CPU time in seconds for three approximate linear solvers versus the level of fill-in and a direct solver in time integrating the small (a) and large (b) non-stiff power system with a tolerance of 10^{-4}

From Figure 7.9, a distinction between the small and large non-stiff power systems can be observed. In the case of the former power systems, the computation decreased when the level of fill-in ℓ of the factorization increased to reach the same computation time as when the LU factorization was used. For this reason, direct solvers are recommended for the computation of a small power system. In the case of the large non-stiff power systems, the computation time is always timed faster for the different levels of fill-in ℓ than with the LU factorization. Although the number of time-steps is more significant with iLU(0) than with LU, the computation is already faster with iLU(0) than with LU. This time, the size of the power system starts to be significant, and then, the number of non-zero elements is two times smaller than with LU. Besides, the numerical factorization consumes less computation time because there are fewer factors to compute. Also, the smallest computation time was observed when the number of time-steps is higher than with the direct solver. Furthermore, a small increase just after the minimum computation time is seen, and when the fill-in increases, the computation time will reach the same computation time as the direct solver. Finally, the utilization of one or five iterations of GMRES with iLU(ℓ) was slower than one iteration of iLU(ℓ) as BIM.

Table 7.3 and Table 8.1 show the overall computation time of Alg. 2 for the stiff power systems. Also, the number of time-steps accepted and rejected are shown too. As seen in the previous results of this section, the same time domain accuracy is found due to the adaptive method.

Linear solver	Tol = 10^{-4}	Tol = 10^{-6}
	CPU in secs. (accepted/rejected)	
LU	0.10 (116 / 2)	0.27 (368 / 6)
iLU(30)	0.28 (492 / 12)	1.21 (2235 / 99)
iLU(35)	0.13 (202 / 3)	0.47 (850 / 20)
iLU(40)	0.08 (116 / 2)	0.21 (368 / 6)
GMRES iLU(30) 1 it	0.55 (619 / 88)	1.75 (2240 / 71)
GMRES iLU(35) 1 it	0.23 (239 / 17)	0.69 (866 / 27)
GMRES iLU(40) 1 it	0.11 (116 / 2)	0.30 (368 / 6)
GMRES iLU(30) 5 it	0.15 (116 / 2)	0.41 (368 / 6)
GMRES iLU(35) 5 it	0.14 (116 / 2)	0.37 (368 / 6)
GMRES iLU(40) 5 it	0.11 (116 / 2)	0.30 (368 / 6)

Table 7.3: CPU time (in seconds) for the small stiff power system, for various iterative solvers and the direct solver and two absolute tolerances in case of an adaptive time-step selection. The first number inside the brakes is the total number of factorizations and the second one is the number of time-steps accepted

Linear solver	Tol = 10^{-4}	Tol = 10^{-6}
	CPU in secs. (accepted/rejected)	
LU	945.70 (3817 / 267)	2599.47 (11804 / 115)
iLU(30)	326.88 (23729 / 207)	563.46 (41938 / 771)
iLU(35)	172.69 (9328 / 15)	412.46 (23181 / 79)
iLU(40)	**142.32** (5804 / 5)	416.75 (18767 / 5)
GMRES iLU(30) 1it	455.86 (23684 / 230)	793.66 (41839 / 759)
GMRES iLU(35) 1it	230.00 (9672 / 10)	546.47 (24402 / 86)
GMRES iLU(40) 1it	178.20 (5799 / 8)	533.13 (18774 / 6)
GMRES iLU(30) 5it	171.96 (3817 / 267)	350.79 (11803 / 81)
GMRES iLU(35) 5it	145.62 (3817 / 267)	**333.92** (11803 / 111)
GMRES iLU(40) 5it	165.69 (3817 / 267)	415.21 (11804 / 111)

Table 7.4: CPU time (in seconds) for the large stiff power system, for various iterative solvers and the direct solver and two absolute tolerances in case of an adaptive time-step selection. The first number inside the brakes is the total number of factorizations, and the second one is the number of time-steps accepted.

From Table 7.3, the same remarks for the small stiff power systems can be made for the small non-stiff power systems too. A direct solver can be recommended. However, the results for the large power systems are a bit different. GMRES method is seen to reduce the number of time-steps, but does not bring any gain in CPU time. Table 8.1 shows that for the large stiff system, the use of iLU(40) as a BIM results in a more-than-five fold reduction of the CPU-time employed by the direct solver, for both tolerances employed. This gain is achieved despite the fact that ILU(40) requires roughly one and half times more time-steps. The GMRES-based solvers are seen to be more competitive than the direct solver in terms of CPU time. For both tolerances, the use of five outer iterations makes these solvers accurate to the degree that the number of time-steps required equals the one used by the direct solver. The CPU time required, however, is not less than ILU(40) method as a BIM.

From all these results, the choice of a linear solver during the integration process depends on three factors - the size and the stiffness of the power system under consideration, and the accuracy desired. The size of the linear system of equations affects the time to resolve it. For small systems of equations, LU factorization is recommended. When the size increases, a new factor is to be taken into account - the stiffness. The stiffness plays a major role in solving the linear system of equations using iterative method, due to the eigenvalues of the matrix A. For example, a medium size power system can be simulated with LU factorization for non-stiff systems, and with adequate basic iterative methods for stiff systems. The last factor is the accuracy for a time-stepping algorithm. When the required accuracy is high, it may impose a very small time-step. For stiff large power systems, GMRES method with a suitable preconditioner is recommended.

7.4. CONCLUSIONS

The sequence of linear systems for the stage vectors at each time-step were solved inaccurately using an incomplete factorization, with various levels of fill-in. A limited number of iterations of this solver, either as a basic iterative method or as preconditioner for

GMRES, are used. This inner solver was combined with an outer time-stepping loop, with either a fixed or an adaptively selected time-step. In the case of a fixed time-step, the time-stepping procedure fails to converge in case the linear solver is not sufficiently accurate. This non-convergence is more likely to occur in cases where the power system modeled is stiff. With the adaptive time-step strategy, the non-convergence is compensated by reverting to a smaller time-step, with linear systems that are easier to solve. Numerical results show a speed-up of computations that increases with the problem size. For the largest problem considered in this chapter, a speed-up by a factor of more than five was observed.

REFERENCES

[1] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, *PETSc Users Manual*, Tech. Rep. ANL-95/11 - Revision 3.7 (Argonne National Laboratory, 2016).

[2] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *Efficient management of parallelism in object oriented numerical software libraries,* in *Modern Software Tools in Scientific Computing,* edited by E. Arge, A. M. Bruaset, and H. P. Langtangen (Birkhäuser Press, 1997) pp. 163–202.

8

Conclusion

8.1. Introduction

Chapter 1 defined several research questions and some research objectives. Then, in Chapters 2 and 3, a short introduction to the different parts of the thesis was given for the electrical and mathematical backgrounds. Chapters 4 and 5, and Chapter 6 and 7 display the contributions to electrical engineering and numerical mathematics, respectively. Finally, this concluding chapter summarizes the answers to the research questions, and the main contributions of the presented research.

This chapter is organized as follows: Section 8.2 gives the answers to the research questions; the main contributions are given in Section 8.3; and, Section 8.4 provides some recommendations for future research.

8.2. Answer to the research question

Let first consider the first research question, stated in Section 1.2:

Is it possible to design a new methodology to model a power system, which allows for the use of adaptive time-stepping strategy with a relatively small computation time for re-computing or updating the network equations in the case of a switching event within an ordinary differential equation formulation?

This can be answered in three parts:

- As shown in Chapter 4, the block modeling method permits the achievement of power system equations in a state space representation. As defined in Chapter 3, this mathematical formulation allows for the utilization of an adaptive time-steping algorithm during the numerical integration;

- After a switching action, the re-computation of the network equations - in some particular cases - requires the utilization of the Schur complement (Chapter 4). However, the Schur complement is usually applied on a small system of equations. Also, the user could implement the change of topology analytically: for example, for the utilization of power electronics. Finally, the utilization of capacitance on an extra node allows one to avoid the Schur complement. This approach increases the number of differential variables. However, the matrix A can then be bad conditioning; this can lead to iterative methods requiring more iterations to solve the systems of equations during the integration process (Chapter 3);

- In Chapter 5, the use of arc models was shown with this new approach to model power systems. The use of arc models results in non-linear differential equations. The Jacobian matrix can be computed analytically. The use of numerical differentiation is thus avoided. This is a distinct advantage of the block modeling method for transient simulations.

The second research question stated in Section 1.2 is:

How strong is the interaction between the time loop of the numerical implicit integration method of an ordinary differential equation and the solver for obtaining the time domain solution of a power system?

The interaction between the time loop and the solver of the systems of equations is hard to quantify. This interaction depends on the size of the power system, the stiffness of the model, the time-step, and the time-stepping strategy used. Then, the answer to this question can be split into three segments:

- As shown in Chapters 6 and 7, the use of an inexact solver can reduce the computation time. However, the accuracy of the time domain solution is affected. Meanwhile, two others parameters can affect the accuracy of the solution: the electrical network models and the integration method. Finally, a compromise has to be reached between computation time and accuracy of the results;

- As shown in Chapter 7, the use of the adaptive time-stepping strategy adds several advantages. The first one is that by reducing the time-step, the approximate solver converges to a sufficiently accurate solution. However, the number of time-steps increases. The second one is that several iterations may not help decrease the computation of the time domain solution. The last advantage is the relative speed-up observed when an appropriate approximate solver is chosen;

- From the results of Chapter 7, the following table summarizes the type of linear solvers to use, according to the size and the stiffness of the power system, and the accuracy required during the numerical integration method.

	nb 1000		
	Low accuracy		
Non-stiff	LU	iLU(ℓ)	iLU(ℓ)
Stiff	LU	iLU(ℓ)	GMRES+iLU(ℓ)
	High accuracy		
Non-stiff	LU	iLU(ℓ)	GMRES+iLU(ℓ)
Stiff	LU	GMRES+iLU(ℓ)	GMRES+iLU(ℓ)

Table 8.1: Summary of the type of linear solver to use, according the number of differential variables nb and the stiffness of the power system, and the accuracy required

8.3. CONTRIBUTION

The first contribution of this thesis is to give a new view on power system modeling. Instead of considering it as a connection of lumped elements, it is possible to see each element that comprises it as a set of equations. Those components are then connected - and therefore, those sets of equations have to be interconnected too. This new approach to viewing power systems permits the linking of different sets of equationsusing the properties of the power system by itself.

The second contribution of this thesis is the study of inaccurate solvers to reduce the computing time during the numerical integration method. As seen in the previous chapter, the computation time can be reduced by at least half, in the case of a large power system.

8.4. FUTURE WORK

Based on this study, three conjoint directions can be considered for future research studies in the electrical engineering or numerical analysis fields.

- The use of power electronics in power system

  Today, the utilization of converters in power systems is increasing. According to the results of this thesis - and particularly, the results of the block modeling method - the simulation of power electronics in a large electrical network could be improved. For example, considering a converter as a block model simular to the arc model, where the conductivity could change, could be one such simulation.

- Modeling transformers

  Another optic for improving the block modeling method would be the ability to model transformers. Transformers are usually non-linear because of the relationship between magnetic and current excitation. Then, if the non-linearity would be handled, similiar to how the arc models are handled, especially for the calculation of the Jacobian matrix, a significant speed-up may appear.

- Inexact Newton methods for transients in power systems

  Approximate solvers allow a gain of relative computation time for the large power system cases in this thesis. Also, approximate solvers gave good results because of the time-step employed. Besides, when a non-linearity appears, a small time-steps in necessary; then, the use of approximate solvers could help reduce the computation time without compromising the quality of the time domain solution.

ACKNOWLEDGEMENTS

First of all, I thank Prof. L. van der Sluis and Prof. C. Vuik for offering me the opportunity to work on the simulation of electrical networks within the Intelligent Electrical Power Grids and the Numerical Analysis groups at the faculty of Electrical Engineering, Mathematics and Computer Science of the Delft University of Technology, The Netherlands., and the initiative Power Web. Also, I thank Dr. D. Lahaye my daily supervisor.

I would like to thank Prof. L. van der Sluis for his expertise in the history and his overlook of electrical engineering. I also liked our conversations, which is as important as our field of study, baking cake. Finally, I am thankful for his help to visit RTDS Technologies Inc. for 3 months.

I would like to acknowledge Prof. C. Vuik for his support, wisdom and way to manage the communication between the electrical engineers and mathematicians. Also, I am thankful for allowing me to go for 3 months in Canada.

I would like to be grateful to Dr. D. Lahaye for all discussions we had during the last 4 years at least one every week. I learned with him the importance of being clear. Also, I would like to thank you for the 4 years of co-operation. I would like to thank for having challenging me and make me aware of how scientific world works.

I express gratitude to Paul Forsyth for accepting to supervise me during 3 months at RTDS Technologies Inc., Winnipeg, Canada. I had a nice time there where I discovered what I wanted to after studying. I met amazing persons there, Eshan, Gregory, Melanie, Christian and all people of the company.

I am thankful to IEPG department where I worked. I would like first to thanks all my officemate Laura, Freek, Gerben, Shahab, Arun, Claudio, Rishabh and Zong. Without all our conversations, studying with you all, would not have passed so fast. Also, I don't forget Matija, Ilya, Peter, Mart, Kaikai, Jose, Hossein, Lian, Marjan, Ellen, Mart, Martjin, Sharmila, Arcadio and Arjen for our scientific and unscientific talks. I would like to acknowledge Mario, Bart and Swasti for the losing hair equation and all talks we had about that primordial equation.

Ann-Sophie and Heiko, I am so happy that we built a tower of glasses. Without you, my 4 years in Delft would have not been the same.

3 years ago, I started to play Ultimate Frisbee and that was legend... wait for it ... dary and still it is. I met a lot of exceptional people around the world, thanks Rienk, Mario, Baline, Alex, Michiel, Annika, ... and Spike.

I would like to thank my new colleagues and DigSILENT Gmbh for their support.

Finally I would like to thank my family and friends from France who supported me along this amazing experience.

22nd October 2017, Nehren, Germany,
Romain Thomas

Curriculum Vitæ

Romain Thomas

09-04-1989	Born in Quimper, France.
2004–2007	Hight school Lycée Yves-Thépot, Quimper, France
2007–2009	DUT Génie électrique et informatique industrielle Universitée de Bretagne Occidental, Brest, France
2009–2012	Master of Engineering in Électronique, Énergie Électrique & Automatique INP-ENSEEIHT, Toulouse, France
2012–2016	PhD. in Electrical Engineering and Numerical mathematics Technische Universiteit Delft, Delft, The Netherlands
2017	Software developer DIgSILENT, Gomaringen, Germany

List of Publications

List of Journal Publications

- R. Thomas, L. van der Sluis, D. Lahaye and C. Vuik, Network Calculations with Switching Devices, under review for IET Electric Power Application.

List of Conference Publications

- R. Thomas, D. Lahaye, C. Vuik and L. van der Sluis, *Inexact Linear Solvers for Transients in Power System Networks*, Proceedings of NUMELEC 2017, Paris, France. (In preparation, abstract accepted)

- R. Thomas, D. Lahaye, C. Vuik and L. van der Sluis, *Simulation of Arc Models with the Block Modelling Method*, Proceedings of IPST 2015, Cavtat, Croatia.

List of Book contributions

- R. Smeets and al. , *Switching in electrical transmission and distribution systems*, John Wiley & Sons city, Chichester, 2014.

- P. Schavemaker and L. van der Sluis, *Electrical Power System Essentials*,John Wiley & Sons city, Chichester, 2014.