TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Antero Marjamäki

Coupling reduced order dynamic electromagnetic models with circuit simulators.

Master of Science Thesis

Examiner: Assist. Professor Paavo Rasilo
The examiner and the topic of the thesis
were approved on 9 August 2017

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master of Science Degree Programme in Electrical Engineering
Marjamäki Antero: Coupling reduced order dynamic electromagnetic models with circuit simulators.
Master of Science Thesis, 54 pages
August 2017
Major: Scientific modelling
Examiner: Assist. Professor Paavo Rasilo
Keywords: Discrete empirical interpolation method, dynamic, model reduction, nonlinear, proper orthogonal decomposition

The progress made in power electronics has raised new challenges concerning devices which contain magnetic components. It is crucial to be able to model the electromagnetic phenomena inside a device and also the behaviour of the device when it works as a part of a circuit. The first case is usually dealt with using finite element analysis and the second case by using circuit simulators. One goal of this thesis is to allow the results of the detailed analysis to be utilized also in the behavioural study conducted using circuit simulators.

In this thesis we firstly introduce some background of electromagnetic modelling. Next two promising methods, proper orthogonal decomposition (POD) and discrete empirical interpolation method (DEIM), are studied with detail and they are applied as an example to a single-phase transformer. The main emphasis is to show how these methods are applied to a dynamic nonlinear electromagnetic model. First a finite element model of the transformer is constructed and reduced. The reduced order model is attached to a circuit simulator Simscape and a simple example circuit is solved to obtain numerical results.

The results show that POD and DEIM methods decrease the computational work of the original model and the results remain feasibly accurate. The dimension of the equation system reduces 99% from the original. We also see a 75 % decrease in stepwise computational time and a 44% decrease in the computational time of the circuit simulator run. However in this case the performance of the circuit simulator is limited and there is a lot of overhead involved. The reduction is expected to be better if these techniques are applied to larger 3-D problems and if the performance of the circuit simulator coupling is improved. In conclusion these methods can be applied to a general class of dynamic nonlinear electromagnetic problems. They could be used to link finite element models to circuit simulators. It could be possible to develop a software module which creates a circuit model automatically based on some finite element model. The techniques can also be used to form homogenized material models of materials which have a fine microstructure.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Sähkötekniikan koulutusohjelma
Marjamäki, Antero: Redusoitujen dynaamisten sähkömagneettisten mallien kytkeminen piirisimulaattoriin.

Diplomityö, 54 sivua
Elokuu 2017
Pääaine: Teknis-tieteellinen mallintaminen
Tarkastaja: Assist. Professor Paavo Rasilo
Avainsanat: Discrete empirical interpolation, dynaaminen, epälineaarinen, Malliasteen pudotus, proper orthogonal decomposition

Tehoelektroniikan kehitys ja yleistyminen asettavat magneettipiirejä sisältäville sähkölaitteille uusia vaatimuksia. On tärkeää pystyä mallintamaan sähkömagneettisia ilmiöitä laitteiden sisällä sekä laitteen käyttäytymistä ulkoisen piirin osana. Tässä työssä tutkitaan mallin redusointimenetelmiä sekä redusoitujen mallien liittämistä piirisimulaattoreihin. Aluksi työssä esitellään sähkömagneettisen mallintamisen perusteoriaa. Tämän jälkeen esitellään lyhyesti elementtimenetelmä sekä työssä käytettyjä muita numeerisia ratkaisumenetelmiä.

Työn päätarkoitus on esitellä mallien redusointitekniikoita. Kaksi lupaavinta redusointitekniikkaa, proper orthogonal decomposition (POD) ja discrete empirical interpolation method (DEIM), käsitellään työssä tarkemmin. Näitä kahta menetelmää sovelletaan esimerkinomaisesti yksivaiheisen muuntajan mallintamiseen verkon osana. Näin tullaan esitellyksi menetelmä, jolla kyseisiä mallin redusointimenetelmiä voidaan käyttää yleisesti dynaamisten epälineaaristen sähkömagneettisten mallien redusoimiseen. Muodostettu redusoitu malli liitetään Simscape-piirisimulaattorilla mallinnettuun yksinkertaiseen piiriin tulosten laskemista varten.

Saatujen tulosten perusteella voidaan sanoa, että POD- ja DEIM-menetelmät sopivat tähän käyttötarkoitukseen ja niiden tuottamat tulokset ovat riittävän tarkkoja. Lisäksi ne vähentävät tuntuvasti mallien laskentatyötä nopeuttaen piirisimulaattoriin liitettyjen mallien laskenta-aikoja. Elementtimenetelmän avulla saadun yhtälöryhmän koko pienenee 99 %, yhden aika-askeleen kohdalla laskenta-aika vähenee 75 % ja piirisimulaattorin suoritusaika vähenee 44% alkuperäiseen malliin verrattuna. Piirisimulaattorikytkentä on tässä työssä suorituskyvyltään huono sekä redusoitava tehtävä alkujaan kevyt. Siksi onkin odotettavissa, että mikäli näitä tekniikoita käytetään työläämpiin 3D-tehtäviin, ja mikäli piirisimulaattorikytkentää tehostetaan päästään parempiin tuloksiin. Jatkossa voisi olla mahdollista kehittää liitännäinen elementtimenetelmäsovellukseen, joka voisi generoida piiriin liitettävän mallin automaattisesti yksityiskohtaisen mallin perusteella. Tekniikoita voidaan käyttää myös hienorakenteisten materiaalien mallien homogenisointiin.

PREFACE

This thesis was written in Tampere while working as a part of Electromechanics research group at Laboratory of Electrical Energy Engineering of Tampere University of Technology. This thesis is a part of a research project which focuses to develop new modelling techniques for magnetic materials. The resulting reduction process is later on utilized to make circuit analysis on devices which are modelled using the new techniques developed in the project.

I would like to thank assistant professor Paavo Rasilo who is the leader of our research group for examining this thesis and for providing the opportunity to work in his group. He has been an inspiring superior and it has been exiting and interesting time to work here. Thanks for the Emil Aaltonen foundation for providing the funding for this thesis. I would also like to thank all the people in our laboratory for coffee room discussions and lunch company. Also thanks for the people of the old Electromagnetics research group who have inspired me along the studies to make make my own choices and study the topics that have interested me the most.

Finally I would like to thank my loving wife Miia for supporting me during the changes of the first half of the year and my daughter Ronja for all the smiles and laughter we have experienced. I love you both with all my heart.

In Tampere 5.7.2017

Antero Marjamäki

CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS

AC	Alternating current
AC-DC	Alternating current to direct current
DC	Direct current
DC-DC	Direct current to direct current
DEIM	Discrete empirical interpolation method
FEM	Finite element method
MOR	Model order reduction
ODE	Ordinary differential equation
PDE	Partial differential equation
POD	Proper orthogonal decomposition
PWM	Pulse width modulation
ROM	Reduced order model
SST	Solid state transformer
SVD	Singular value decomposition
$\mathbf{A}$	Magnetic vector potential
$\mathbf{a}$	Nodal values of the vector potential
$\tilde{\mathbf{a}}$	Nodal values of the vector potential in the reduced system
$\mathbf{B}$	Magnetic flux density
$\mathbf{D}$	Electric flux density
Δt	Length of a timestep
$\mathbf{E}$	Electric field strength
ε	Electric permittivity
$\mathbf{H}$	Magnetic field strength
J	Jacobian matrix

$\mathbf{J}$	Current density
$\mathbf{J}_{\mathrm{ed}}$	Eddy current density
$\mathbf{J}_{\mathrm{s}}$	Source current density
μ	Magnetic permeability, parameter configuration
M	Damping matrix
$\hat{\mathbf{n}}$	Normal vector
N_i, $\mathbf{N}_i$	Shape function corresponding node i
Ω	The domain of a field problem
ϕ	Reduced scalar potential
ψ	A basis vector of POD basis
Ψ	POD projection mapping
$\mathbf{R}$	Residual term of equation system
R_{L}	Load resistance
S	An arbitrary surface
S	The stiffness matrix
σ	Electric conductivity
t	Time variable
V	An arbitrary volume
Ξ	POD parameter space

1. INTRODUCTION

With modern electromagnetic energy converters it is important to be able to accurately model the internal and external behaviour of the device. This will reduce the production costs of prototypes, make devices safer and more reliable and sometimes enable us to invent things that we easily overlook.

Many devices are parts of a larger system of sometimes very complex interactions. In these cases it is not only needed to ensure that the device works internally as intended but also ensure that the device behaves externally well enough to work as a part of a larger system. This requires modelling the device in multiple levels. Firstly the internal model makes it possible to develop smaller, safer and more efficient devices. Secondly an external model is needed to account for the behaviour of the devices. These two models naturally need to be coupled together at their interface. The external behaviour is often studied by using circuit simulators.

An example of such a device is a transformer. Some new design of a transformer must be modeled in high detail in order to verify the design. To verify the behaviour the transformer could be attached to a circuit simulator to simulate it with different loads and different feeding voltages. This is important for example in the design of solid state transformers (SST) in which the transformer itself operates in a medium frequency range contrary to a more traditional transformer which operates mostly in the low frequency of the power grid [1].

Evaluating results from models that accurately predict phenomena inside electronic devices is usually computationally expensive. Several model reduction methods have been developed to lower the computational costs without sacrificing the accuracy of the results [2]–[4]. Usually these methods aim to reduce the number of the unknowns in the equation system. Especially this complexity reduction is welcome when we are more interested in the external behaviour of the device than the internal phenomena. It is often enough to treat the system at hand as a black-box with inputs and outputs. The calculation of the outputs based on the inputs should be as light-weight as possible within reasonable accuracy.

In this thesis the most popular model reduction methods are investigated and the most promising methods are applied to model an example transformer. Methods which are well suited for this use are the proper orthogonal decomposition (POD) method together with the discrete empirical interpolation method (DEIM). These methods have been succesfully utilized to reduce electromagnetic models [5]–[8].

The same goal has been achieved by creating equivalent circuit models of systems. Some devices can be quite naturally decoupled so that their behaviour can be composed out of the basic electrical circuit components. It is naturally possible to use nonlinear components as well. This work however requires some manual work and heuristic justification by using both theory and trial and error to obtain approximation and some bounds for the operating area. Model reduction methods can be used to achieve automatic creation of lightweight reduced models that can be used instead of equivalent circuits [9].

The primary goal of this thesis is to narrow the gap between the highly detailed models and the circuit models. Some of the more popular model reduction methods are reviewed. The aim is to produce a systematic procedure for applying model reduction methods to models that are computationally heavy and thus enable them to be coupled with circuit simulators. The coupling of a reduced model and a circuit simulator is also investigated. In this thesis we focus mostly to electromagnetic problems which are nonlinear and have dynamical components such as eddy currents. The formulations are done in such a way that hysteresis is also possible to be incorporated into the models.

The model reduction methods introduced in this thesis can also be used in homogenization of finely structured regions in modelling domains [10]. For example magnetic cores compressed out of pulverized material. To model this kind of material we would require a very dense finite element mesh. This is why a homogenized material model has to be developed first.

Chapter 2 introduces the problem area. Some necessary background about electromagnetics is introduced along with necessary background on solving techniques. In Chapter 3 the general idea behind the model reduction methods analysed in this thesis is introduced. Mathematical background is briefly discussed. Chapter 4 deals with circuit analysis. Some background about circuit theory and applications is given. In Chapter 5 we apply the theory presented in Chapters 2-4 to a single-phase transformer. We also present the results and compare the accuracy and computation speed advantages that model reduction gives in this particular case. Finally in Chapter 6 we summarize and discuss the results and discuss the feasibility of model reduction in these kinds of problems.

2. ELECTROMAGNETIC FIELD PROBLEMS

This chapter introduces the problem application area handled in this thesis. Some mathematical background for modelling electromagnetic devices using field quantities and the finite element method are also introduced. The foundations of electromagnetic theory are covered well in literature for example in [11].

2.1 Theory for electromagnetic modelling

The modelling of phenomena in electronic devices and systems has its foundations in the theory of classical electromagnetism. At the heart of this theory are the Maxwell's equations. These equations together with the Lorentz force law govern the behaviour of electromagnetic phenomena inside electronic devices. These equations give accurate predictions in both small scale systems such as transistors and large scale systems such as motors and transformers. Many frameworks such as the circuit theory have their basis in these equations. First the Maxwell's equations in integral form will be presented. Each term and equation and its implications will also be discussed.

The field quantities discussed in this thesis are mostly dependent of spatial variables. In three dimensions the spatial variables are usually marked with x, y and z. For example an arbitrary field $\mathbf{F}(x, y, z)$ is denoted with $\mathbf{F}$ for simplicity if it is clear from the context that the field is not a constant field. The spatial dependency is emphasized wherever it is important. We also omit the time dependency notation for those fields which are clearly dependent of time and emphasize the dependency wherever it is important.

2.1.1 Maxwell's equations

The Maxwell's equations contain relationships between the following field quantities: the electric field $\mathbf{E}$, the electric flux density $\mathbf{D}$, the current density $\mathbf{J}$, the magnetic field $\mathbf{H}$, the magnetic flux density $\mathbf{B}$, and the charge distribution ρ. The three-dimensional domain space is denoted with Ω. We denote an arbitrary *volume* in the domain space with V and the boundary surface that encloses that volume with ∂V. Similarly we denote an arbitrary two-dimensional *surface* with S and the boundary curve of that surface with ∂S.

The first Maxwell equation which is presented is the Gauss's law

$$\int_{\partial V} \mathbf{D} \cdot \hat{\mathbf{n}} \, \mathrm{d}S = \int_V \rho \, \mathrm{d}V, \ \forall \, V \subset \Omega \, . \tag{2.1}$$

The Gauss's law denotes that if there is free charge inside a volume, we must have an electric flux flowing through the boundary of the volume.

The second equation is the Gauss's law for magnetism

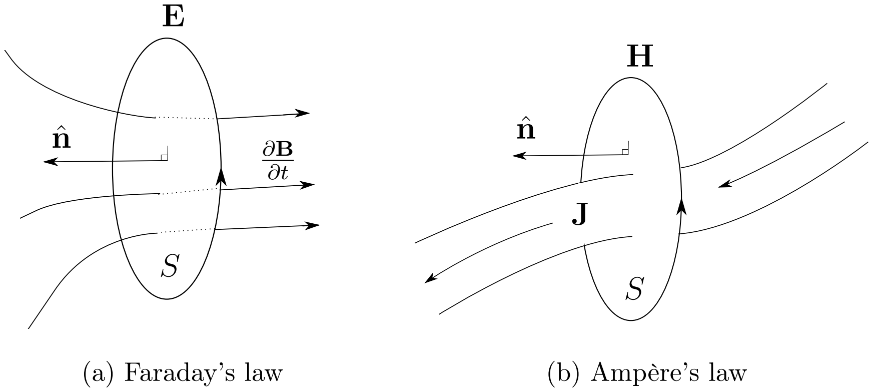
$$\int_{\partial V} \mathbf{B} \cdot \hat{\mathbf{n}} \, \mathrm{d}S = 0, \ \forall \, V \subset \Omega \, . \tag{2.2}$$

This equation states that magnetic monopoles do not exist. It implies the known fact that if one cuts a magnet in half one gets two magnets which both have north and south poles not two pieces which are the separated north and south poles.

The third equation is known as the Maxwell-Faraday equation and also as the Faraday's law of induction

$$\int_{\partial S} \mathbf{E} \cdot \mathrm{d}l = - \int_S \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{B} \cdot \hat{\mathbf{n}} \, \mathrm{d}S, \ \forall \, S \subset \Omega \, . \tag{2.3}$$

This equation states that a voltage which is induced to a closed loop around a surface is equal to the time derivative of the magnetic flux that penetrates that surface. This law is illustrated in Figure 2.1a. This law is a basic principle of how transformers work. The change in the magnetic flux of the transformer core induces a voltage to the secondary coil of the transformer.



(a) Faraday's law

(b) Ampère's law

Figure 2.1: Illustration of two laws which are important for transformers: Faraday's law of induction and Ampère's law.

The final remaining equation is known as the Ampère's law

$$\int_{\partial S} \mathbf{H} \cdot \mathrm{d}l = \int_{S} \left(\mathbf{J} + \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{D} \right) \cdot \hat{\mathbf{n}} \, \mathrm{d}S, \ \forall \, S \subset \Omega \, . \tag{2.4}$$

The Ampère's law states that an electric current penetrating a surface will induce a magnetomotive force to the closed loop around that surface. This formulation also separates *source currents* $\mathbf{J}$ from *displacement currents* $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{D}$. This law is illustrated in Figure 2.1b. This phenomenon is also an important part of a transformer as the magnetic field in the core is induced by the currents in the primary coil.

2.1.2 Connection to energy

It is visible in the integral form that although all vector field quantities introduced here are ordinary vector fields the nature of $\mathbf{E}$ and $\mathbf{H}$ is different to the nature of $\mathbf{D}$, $\mathbf{J}$ and $\mathbf{B}$. Fields $\mathbf{E}$ and $\mathbf{H}$ are integrated along a *path* and fields $\mathbf{D}$, $\mathbf{J}$ and $\mathbf{B}$ are integrated along a *surface*. This duality is related to energy. Each pair $(\mathbf{E}, \mathbf{D})$, $(\mathbf{E}, \mathbf{J})$, $(\mathbf{H}, \mathbf{B})$ has a property that deals with power dissipation or energy storage. The energy stored in the electric field inside a volume V can be calculated by

$$E = \int_V \int_0^{||\mathbf{D}||} \mathbf{E} \cdot \mathrm{d}\mathbf{D} \, \mathrm{d}V \, . \tag{2.5}$$

The power that is dissipated as heat inside volume V can be calculated with

$$P = \int_V \mathbf{E} \cdot \mathbf{J} \, \mathrm{d}V \, , \tag{2.6}$$

and the energy stored in the magnetic field can be calculated with

$$E = \int_V \int_0^{||\mathbf{B}||} \mathbf{H} \cdot \mathrm{d}\mathbf{B} \, \mathrm{d}V \, . \tag{2.7}$$

Electromagnetic energy converters such as a transformer transform the energy from one form to another. The energy flowing through the transformer flows through the magnetic field in the core of the transformer.

2.1.3 Constitutive equations

In addition to the Maxwell's equations the theory needs constitutive equations which are

$$\mathbf{D} = \varepsilon \mathbf{E} \tag{2.8}$$

$$\mathbf{B} = \mu \mathbf{H} \tag{2.9}$$

$$\mathbf{J} = \sigma \mathbf{E} \; , \tag{2.10}$$

where the material operators ε, μ, σ are called permittivity, permeability and conductivity.

In the simplest case when material is linear and isotropic these operators act as scalar constants. In the opposite end for example the relation between **B** and **H** of the core material in a transformer can be highly nonlinear, hysteretic and anisotropic. Anisotropy is present for example when the core of a transformer is formed from thin laminations to reduce eddy currents. In those cases which are not linear we use notation such as $\mathbf{B} = \mu(\mathbf{H})\mathbf{H}$ or $\mathbf{B}(\mathbf{H})$ to emphasize that the relationships between the two quantities are possibly nonlinear.

2.1.4 Differential formulation

Although the integral form of Maxwell's equations is informative and general it is not typically used in problem formulations. A more useful representation is the differential presentation of the equations. Using spatial derivatives the equations can be given as

$$\operatorname{div}(\mathbf{D}) = \rho \tag{2.11}$$

$$\operatorname{div}(\mathbf{B}) = 0 \tag{2.12}$$

$$\operatorname{curl}(\mathbf{E}) = -\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{B} \tag{2.13}$$

$$\operatorname{curl}(\mathbf{H}) = \mathbf{J} + \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{D} \; . \tag{2.14}$$

The constitutive equations (2.8) still hold. In this formulation special care needs to be taken in material boundaries. The field quantities are not continuous at material boundaries because constitutive relations have a sudden change. The pointwise formulation of the Maxwell's equations is obtained from the integral form by taking limits where the volume V and surface S shrink closer and closer to zero i.e. being a point.

2.1.5 Boundary conditions

Partial differential equation (PDE) systems are not well posed without appropriate boundary conditions. There are different kinds of boundary conditions that can be posed for field quantities. The Dirichlet boundary condition and the Neumann boundary condition are popularly used.

In the Dirichlet boundary condition the value of a field is fixed on the boundary. In the Neumann boundary condition we fix the normal component of the derivative of the field on the boundary. For example if we have a time varying field $\mathbf{f} : \Omega \times T \mapsto \mathbb{R}^n$ in domain Ω and divide the boundary into the Dirichlet boundary Ω_d and the Neumann boundary Ω_n so that $\Omega_\mathrm{d} \cap \Omega_\mathrm{n} = \partial\Omega$ we define the boundary conditions as

$$\mathbf{f}(\mathbf{x}, t) = \mathbf{f}_\mathrm{d}(\mathbf{x}, t), \text{ when } \mathbf{x} \in \Omega_\mathrm{d} \tag{2.15}$$

$$\frac{\partial}{\partial n}\mathbf{f}(\mathbf{x}, t) = \mathrm{f}_\mathrm{n}(\mathbf{x}, t), \text{ when } \mathbf{x} \in \Omega_\mathrm{n} \ . \tag{2.16}$$

Here the notation $\frac{\partial}{\partial n}$ means the directional derivative to the direction of the normal of the boundary curve or surface.

The values at the boundary don't have to be constants, it is required that they are known. In (2.15) $\mathbf{f}_\mathrm{d}$ and f_n can be functions of both spatial and time coordinates.

2.2 Formulation of a magnetodynamic problem

Usually the equations (2.11)-(2.14) are generally not all taken into account. It depends on the system we are modelling what terms are negligible.

To model magnetodynamic systems such as a transformer we usually assume that displacement currents are negligible compared to source currents. We are furthermore not interested of charge densities over our problem domain. Therefore we can neglect equation (2.11) and the displacement current term from equation (2.14).

After simplifications we are left with three equations. To satisfy (2.12) we can exploit a vector identity which states that the divergence of a curl of a field is always zero. Hence we introduce a *vector potential* $\mathbf{A}$ such as $\mathbf{B} = \mathrm{curl}\,(\mathbf{A})$. Now $\mathrm{div}\,(\mathbf{B}) = \mathrm{div}\,(\mathrm{curl}\,(\mathbf{A})) = 0$ is always satisfied.

2.2.1 Accounting for eddy currents

Eddy currents are the result of equation (2.13). If flux changes are present in conductive regions the induced voltage will cause circulating currents to flow inside the material. To account for the eddy currents we substitute the vector potential to equation (2.13) and we get

$$\operatorname{curl}(\mathbf{E}) = -\operatorname{curl}\left(\frac{\partial}{\partial t}\mathbf{A}\right) \ . \tag{2.17}$$

The equality between curls of two fields implies that the fields itself are equal up to a difference by a gradient of a field. So (2.17) implies that

$$\mathbf{E} = -\frac{\partial}{\partial t}\mathbf{A} - \operatorname{grad}(\phi) \tag{2.18}$$

since $\operatorname{curl}(\operatorname{grad}(\phi)) = 0,\ \forall \phi$. The scalar field ϕ is called reduced scalar potential.

Now we can divide the current density into two parts. First part $\mathbf{J}_{\mathrm{s}}$ is the source current which is known and the second part $\mathbf{J}_{\mathrm{ed}}$ is the induced eddy currents. By using material relation $\mathbf{J} = \sigma\mathbf{E}$ we can write

$$\begin{aligned} \mathbf{J} &= \mathbf{J}_{\mathrm{s}} + \mathbf{J}_{\mathrm{ed}} && (2.19) \\ &= \mathbf{J}_{\mathrm{s}} - \sigma \frac{\partial}{\partial t}\mathbf{A} - \sigma \operatorname{grad}(\phi) && (2.20) \end{aligned}$$

So after we combine (2.14), (2.19) and the material relation $\mathbf{B} = \mu\mathbf{H}$ together we get

$$\operatorname{curl}\left(\mu^{-1}\operatorname{curl}(\mathbf{A})\right) + \sigma \frac{\partial}{\partial t}\mathbf{A} + \sigma \operatorname{grad}(\phi) = \mathbf{J}_{\mathrm{s}} \tag{2.21}$$

This form is a formulation that will be valid in all regions of problem domain. In those regions where $\mathbf{J}_{\mathrm{s}} \neq 0$ we set $\sigma = 0$ and those regions where $\mathbf{J}_{\mathrm{s}} = 0$ we can allow the eddy currents to flow. [12, p. 7:13]

2.2.2 Vector potential in two dimensions

When the system that is modelled has symmetries the dimension of the problem domain can usually be reduced from three dimensional space to two dimensional space. Dimension reduction will be made with an assumption that the fields will act specially in the direction of the dimension being reduced.

Assume a three dimensional system being modelled in a Euclidian space with an orthonormal (x, y, z) coordinate system. Assume also that the system is thick in z-direction. Thick means here that a plane $z = \alpha$ can be found where the fields that are inside the system do not change with respect to the z-direction. A slice of the system can then be taken and the system modelled only in the two dimensional subset which will usually result in a simpler and less computationally heavy model than in three dimensional case.

In this kind of system it can be assumed that $\mathbf{B}$ and $\mathbf{H}$ are vectors that lie inside the two dimensional plane. This implies that they are of the form $\mathbf{B} = (B_{\mathrm{x}}, B_{\mathrm{y}}, 0)$ and $\mathbf{H} = (H_{\mathrm{x}}, H_{\mathrm{y}}, 0)$. The connection $\operatorname{curl}(\mathbf{A}) = \mathbf{B}$ then states that $\mathbf{A} = (0, 0, A_{\mathrm{z}})$. Also the source current must be z-directional $\mathbf{J}_{\mathrm{s}} = (0, 0, J_{\mathrm{s}})$.

The restriction to two dimensions forces $\operatorname{grad}(\phi)$ to be z-directional. Scalar potential ϕ is then at most a linear function of z-coordinate only. The assumption we make to reduce the domain to two dimensions denies any existence of potential differences in z-direction. Hence $\phi = 0$. [12, 7:15]

Because of this in the two dimensional case where $\mathbf{B}$ and $\mathbf{H}$ are in xy-plane the main governing equation is

$$\operatorname{curl}\left(\mu^{-1} \operatorname{curl}(\mathbf{A})\right) + \sigma \frac{\partial}{\partial t}\mathbf{A} = \mathbf{J}_{\mathrm{s}} \tag{2.22}$$

and it can be stated as a scalar field Poisson problem with respect to A_{z}

$$-\operatorname{div}\left(\mu^{-1} \operatorname{grad}(A_{\mathrm{z}})\right) + \sigma \frac{\partial}{\partial t}A_{\mathrm{z}} = J_{\mathrm{s}} \; . \tag{2.23}$$

Note however that when $\mathbf{B}$ is calculated from $\mathbf{A}$ a curl operator must be used, so in two dimensional case

$$B_{\mathrm{x}} = \frac{\partial}{\partial y}A_{\mathrm{z}} \tag{2.24}$$

$$B_{\mathrm{y}} = -\frac{\partial}{\partial x}A_{\mathrm{z}} \tag{2.25}$$

this is sometimes called as two dimensional curl or surface curl although the whole curl operator is only defined in three dimensional space.

2.2.3 Modelling laminated core material in two dimensions

In equation (2.21) it is assumed that the material is continuous magnetic and conducting material. Most magnetic cores which use conducting material are built from thin plates of core material which are insulated from each other to prevent eddy currents from flowing from plate to plate.

The benefits of laminated core structure are firstly that it reduces eddy current losses a lot which reduces excess heat generation and improves the energy efficiency of the component. Using laminated cores it is also possible to alter the geometry of the device by adding or subtracting plates from the core. Plates can be manufactured with same specifications and used to create devices with similar cross sections but which differ in length. Some core plates of the transformer used as an example case in this thesis are shown in Figure 2.2.

In this thesis we use a low-frequency approximation for the eddy currents in thin plates which is introduced in [13]. The main idea is to approximate $\mathbf{B}$ and $\mathbf{H}$ as

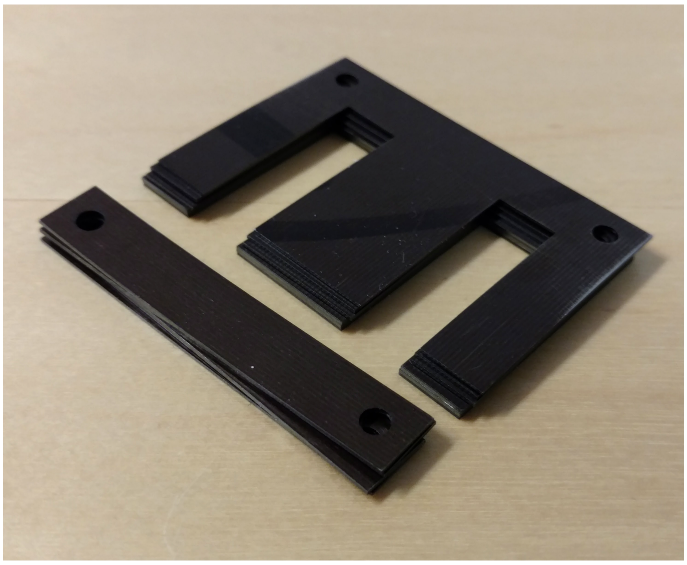


Figure 2.2: Piles of core laminations of the transformer used in the example case

a Fourier series. The low frequency approximation is obtained by truncating this series from the first term and assuming that $\mathbf{B}$ is constant throughout the plate in z direction. The result of this is that equation (2.22) will transform into

$$\operatorname{curl}\left(\mu^{-1}\operatorname{curl}(\mathbf{A}) + \sigma\frac{L^2}{12}\frac{\partial}{\partial t}\mathbf{B}\right) = \mathbf{J}_{\mathrm{s}}\ , \tag{2.26}$$

in which L is the thickness of a single plate in the laminated core. Here we can insert $\mathbf{B} = \operatorname{curl}(\mathbf{A})$ to obtain

$$\operatorname{curl}\left(\mu^{-1}\operatorname{curl}(\mathbf{A}) + \sigma\frac{L^2}{12}\frac{\partial}{\partial t}\operatorname{curl}(\mathbf{A})\right) = \mathbf{J}_{\mathrm{s}}\ . \tag{2.27}$$

The conductivity present here is the conductivity in x and y directions. The model for eddy currents in thin laminations presented here accounts only those currents which flow in the xy-plane. We can also see that the magnitude of the eddy currents is scaled with the factor of $\frac{L^2}{12}$. With typical thickness of the plate this scaling factor is usually very small so the eddy current effects are reduced.

2.3 Finite element discretization

The finite element method (FEM) is a popular way to calculate approximate solutions to PDE systems. In FEM the domain is divided in elements. A set of basis functions is used to approximate the fields in question. This discretization results in an algebraic system of equations which can be solved to obtain an approximate solution.

2.3.1 Weak formulation

In FEM the equations governing the situation will be required to hold in a weak sense. This kind of formulation is called the weak formulation. For simplicity we will derive the weak formulation for (2.22) and discuss the effects of the laminated core afterwards. Weak sense means that we require the equations to hold averagely when they are integrated over the domain and multiplied by an arbitrary test function $\mathbf{w}$ which is an element in the test function space W

$$\int_\Omega \left[\mathbf{w} \cdot \operatorname{curl}\left(\mu^{-1} \operatorname{curl}(\mathbf{A})\right) + \mathbf{w} \cdot \sigma \frac{\partial}{\partial t}\mathbf{A} - \mathbf{w} \cdot \mathbf{J}_\mathrm{s}\right] \mathrm{d}\Omega = 0 \ , \ \forall \, \mathbf{w} \in W. \tag{2.28}$$

When the domain is divided into elements we can construct a function basis which we can use to approximate the solution. It is possible to use different kinds of basis functions as a basis but a popular choice is orthogonal polynomials. Among these it is popular to use linear polynomials.

A reference element is introduced and a mapping from the reference element to each element in the element network is created. Using this mapping *shape functions* can be constructed which are defined for each global node. Shape functions are often denoted with $N_i(\mathbf{x})$ which corresponds to shape function of node i evaluated at point $\mathbf{x}$.

The fields $\mathbf{A}$ and $\mathbf{J}_\mathrm{s}$ are z directional, so the basis functions will be chosen to be of the form $\mathbf{N}(\mathbf{x}) = (0, 0, \mathrm{N}_z(\mathbf{x}))$. With these basis functions the vector potential in point $\mathbf{x}$ can be approximated with

$$\mathbf{A}(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{N}_i(\mathbf{x}) a_i \ , \tag{2.29}$$

where N is the amount of nodes in element network and $\mathbf{N}_i$ the shape function related to node i. Now following Petrov-Galerkin formulation and selecting the same basis for approximating the test function $\mathbf{w}$ the following form is archieved

$$\mathbf{w}(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{N}_i(\mathbf{x}) w_i \ . \tag{2.30}$$

By inserting (2.30) to (2.28) and exploiting the linearity of integral

$$\sum_{i=1}^{N} w_i \int_\Omega \left[\mathbf{N}_i \cdot \operatorname{curl}\left(\mu^{-1} \operatorname{curl}(\mathbf{A})\right) + \mathbf{N}_i \cdot \sigma \frac{\partial}{\partial t}\mathbf{A} - \mathbf{N}_i \cdot \mathbf{J}_\mathrm{s}\right] \mathrm{d}\Omega = 0 \ , \tag{2.31}$$

is obtained.

Since $\mathbf{w}$ was an arbitrary function, equation (2.31) will be true only if every member of the sum is zero. We obtain N equations

$$\int_\Omega \left[\mathbf{N}_i \cdot \mathrm{curl}\left(\mu^{-1}\,\mathrm{curl}\,(\mathbf{A})\right) + \mathbf{N}_i \cdot \sigma \frac{\partial}{\partial t}\mathbf{A} - \mathbf{N}_i \cdot \mathbf{J}_\mathrm{s} \right] \mathrm{d}\Omega = 0\,,\ \forall i \in [1, N]. \tag{2.32}$$

Next we can insert (2.29) to (2.32) and exploit the linearity of curl and integral to move the sum outside of the integral terms again. This time we also split the integral term in three parts.

$$\sum_{j=1}^{N} \underbrace{\int_\Omega \mathbf{N}_i \cdot \mathrm{curl}\left(\mu^{-1}\,\mathrm{curl}\,(\mathbf{N}_j)\right) \mathrm{d}\Omega\ \, a_j}_{\text{term 1}} + \sum_{j=1}^{N} \int_\Omega \mathbf{N}_i \cdot \sigma\, \mathbf{N}_j \,\mathrm{d}\Omega \frac{\partial}{\partial t} a_j = \int_\Omega \mathbf{N}_i \cdot \mathbf{J}_\mathrm{s}\,\mathrm{d}\Omega\,. \tag{2.33}$$

Note here that a_j is a scalar representing the nodal value of the vector potential which does not depend on spatial coordinates. However a_j will depend on time, but again $\frac{\partial}{\partial t}a_j$ does not depend on spatial coordinates. This is why we can move them outside the integrals.

We shall manipulate the term 1 of (2.33) with integration by parts and by using vector derivative identity $\mathrm{div}\,(\mathbf{A}\times\mathbf{B}) = \mathbf{B}\cdot\mathrm{curl}\,(\mathbf{A}) - \mathbf{A}\cdot\mathrm{curl}\,(\mathbf{B})$

$$\begin{aligned}\int_\Omega \mathbf{N}_i \cdot \mathrm{curl}\left(\mu^{-1}\,\mathrm{curl}\,(\mathbf{N}_j)\right) \mathrm{d}\Omega = &\int_\Omega \mathrm{curl}\,(\mathbf{N}_i)\cdot\mu^{-1}\,\mathrm{curl}\,(\mathbf{N}_j)\,\mathrm{d}\Omega \\ &\underbrace{-\int_{\partial\Omega} \mathbf{N}_i \times \mu^{-1}\,\mathrm{curl}\,(\mathbf{N}_j)\cdot\hat{\mathbf{n}}\,\mathrm{d}S}_{=0\,,\ \text{boundary term}}\ .\end{aligned} \tag{2.34}$$

The value of boundary term will be zero in those parts of the boundary where Dirichlet and Neumann boundary conditions are posed and set to zero. This is because the Dirichlet boundary condition forces the shape functions to be zero on the boundary and the Neumann condition forces

$$\mathbf{H}\times\hat{\mathbf{n}} = 0\,, \tag{2.35}$$

which in turn implies that

$$\Rightarrow \mu^{-1}\mathrm{curl}\,(\mathbf{N})\times\hat{\mathbf{n}} = 0\,. \tag{2.36}$$

Now we can use vector identity for dot and cross products $\mathbf{a}\times\mathbf{b}\cdot\mathbf{c} = \mathbf{a}\cdot\mathbf{b}\times\mathbf{c}$ to show that

$$\Rightarrow \mathbf{N}_i\times\mu^{-1}\,\mathrm{curl}\,(\mathbf{N}_j)\cdot\hat{\mathbf{n}} = \mathbf{N}_i\cdot\mu^{-1}\,\mathrm{curl}\,(\mathbf{N}_j)\times\hat{\mathbf{n}} = 0\,, \tag{2.37}$$

and this makes the Neumann part of the boundary term zero.

The highest order of derivatives in term 1 of (2.33) has now been reduced from two to one. If this wouldn't been possible, we would need to use higher order basis functions, since the second derivative of a linear function is always zero.

$$\sum_{j=1}^{N} \int_{\Omega} \mathrm{curl}\,(\mathbf{N}_i) \cdot \mu^{-1}\, \mathrm{curl}\,(\mathbf{N}_j)\, \mathrm{d}\Omega\, a_j + \sum_{j=1}^{N} \int_{\Omega} \mathbf{N}_i \cdot \sigma\, \mathbf{N}_j\, \mathrm{d}\Omega\, \frac{\partial}{\partial t} a_j = \int_{\Omega} \mathbf{N}_i \cdot \mathbf{J}_{\mathrm{s}}\, \mathrm{d}\Omega \tag{2.38}$$

We can now state this equation by using matrix notation. We collect matrices S and M which are called *stiffness* and *damping* matrices respectively as follows

$$S_{ij} = \int_{\Omega} \mathrm{curl}\,(\mathbf{N}_i) \cdot \mu^{-1}\, \mathrm{curl}\,(\mathbf{N}_j)\, \mathrm{d}\Omega \tag{2.39}$$

$$M_{ij} = \int_{\Omega} \mathbf{N}_i \cdot \sigma\, \mathbf{N}_j\, \mathrm{d}\Omega \tag{2.40}$$

and vector $\mathbf{F}$ as

$$F_i = \int_{\Omega} \mathbf{N}_i \cdot \mathbf{J}_{\mathrm{s}}\, \mathrm{d}\Omega\ . \tag{2.41}$$

If we model a laminated core the *damping matrix* M changes. The equation for the laminated core is presented in (2.27). In this case the damping matrix will be

$$M_{ij} = \sigma \frac{L^2}{12} \int_{\Omega} \mathrm{curl}\,(\mathbf{N}_i) \cdot \sigma\, \mathrm{curl}\,(\mathbf{N}_j)\ \mathrm{d}\Omega\ . \tag{2.42}$$

Here instead of the vector potential itself the curl of the vector potential is present.

We assemble the nodal values a_j to a vector $\mathbf{a}$. Now we can formulate the following linear algebraic equation system.

$$\mathrm{M}\frac{\partial}{\partial t}\mathbf{a} + \mathrm{S}\,\mathbf{a} = \mathbf{F} \tag{2.43}$$

which is a linear ordinary differential equation (ODE) and can be solved with standard tools.

2.3.2 Accounting for nonlinear materials

If the medium is not linear the material relation between $\mathbf{B}$ and $\mathbf{H}$ will depend on the magnetic field itself

$$\mathbf{H} = \mu^{-1}(\mathbf{B})\, \mathbf{B} \tag{2.44}$$

this causes that the system (2.43) is not linear and stiffness matrix S will depend on **a**

$$\mathrm{M}\frac{\partial}{\partial t}\mathbf{a} + \mathrm{S}(\mathbf{a})\,\mathbf{a} = \mathbf{F}\ , \tag{2.45}$$

where the stiffnes matrix is computed componentwise as

$$S_{ij}(\mathbf{a}) = \int_{\Omega} \mathrm{curl}\,(\mathbf{N}_i) \cdot \mu^{-1}(\mathbf{a})\,\mathrm{curl}\,(\mathbf{N}_j)\,\mathrm{d}\Omega\ . \tag{2.46}$$

The resulting system must be solved by using iterative solvers, such as Newton-Raphson (NR) method. This poses some challenges as in NR method the Jacobian matrix of the residual term must be derived.

2.3.3 A general formulation for nonlinear case

In this section we apply a different approach to obtain the solution. It is not necessary to derive a stiffness matrix based formulation to solve nonlinear field problems.

When equations (2.14) and (2.21) are combined we can leave the material relation out of the equation for now. **H** can be calculated from **B** which in turn can be calculated from **A** hence we use the notation **H**(**A**). This will result in

$$\mathrm{curl}\,(\mathbf{H}\,(\mathbf{A})) + \sigma\,\frac{\partial}{\partial t}\mathbf{A} = \mathbf{J}_{\mathrm{s}}\ . \tag{2.47}$$

A weak formulation can be created by using the weighted residual method as with equation (2.28)

$$\int_{\Omega} \left[\mathbf{w} \cdot \mathrm{curl}\,(\mathbf{H}(\mathbf{A})) + \mathbf{w} \cdot \sigma\,\frac{\partial}{\partial t}\mathbf{A} - \mathbf{w} \cdot \mathbf{J}_{\mathrm{s}}\right] \mathrm{d}\Omega = 0\ ,\ \forall \mathbf{w} \in W. \tag{2.48}$$

Now following Petrov-Galerkin formulation and approximating weight functions with shape functions again we obtain

$$\sum_{i=1}^{N} w_i \int_{\Omega} \left[\mathbf{N}_i \cdot \mathrm{curl}\,(\mathbf{H}(\mathbf{A})) + \mathbf{N}_i \cdot \sigma\,\frac{\partial}{\partial t}\mathbf{A} - \mathbf{N}_i \cdot \mathbf{J}_{\mathrm{s}}\right] \mathrm{d}\Omega = 0\ , \tag{2.49}$$

and because **w** is arbitrary it must be that all terms of the sum are zero

$$\int_{\Omega} \left[\mathbf{N}_i \cdot \mathrm{curl}\,(\mathbf{H}(\mathbf{B})) + \mathbf{N}_i \cdot \sigma\,\frac{\partial}{\partial t}\mathbf{A} - \mathbf{N}_i \cdot \mathbf{J}_{\mathrm{s}}\right] \mathrm{d}\Omega = 0\ ,\ \forall i \in [1, N]. \tag{2.50}$$

When we apply integration by parts similarly as in (2.33) for the first term of the integrand we can move the curl from $\mathbf{H}$ to $\mathbf{N}_i$.

$$\int_\Omega \left[\mathrm{curl}\,(\mathbf{N}_i) \cdot \mathbf{H}(\mathbf{B}) + \mathbf{N}_i \cdot \sigma \frac{\partial}{\partial t}\mathbf{A} - \mathbf{N}_i \cdot \mathbf{J}_\mathrm{s} \right] \mathrm{d}\Omega = 0 \ , \ \forall i \in [1, N]. \tag{2.51}$$

This results in N nonlinear equations from which we can solve $\mathbf{A}$. The equation system can be formulated as

$$\mathrm{M}\frac{\partial}{\partial t}\mathbf{a} + \mathbf{f}(\mathbf{a}) = \mathbf{F}(t) \ , \tag{2.52}$$

where $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ is a function which encapsulates the same characteristics of the system as the term $\mathrm{S}(\mathbf{a})\mathbf{a}$ but in a more general way and M is the damping matrix given in (2.40). If we are modelling a laminated core the damping matrix will be the one given in Equation (2.42). The nonlinear term is

$$\mathbf{f}_i(\mathbf{a}) = \int_\Omega \mathrm{curl}\,(\mathbf{N}_i) \cdot \mathbf{H}(\mathbf{B})\mathrm{d}\Omega \ . \tag{2.53}$$

In this formulation the relation between $\mathbf{B}$ and $\mathbf{H}$ can be a general function.

Vector potential $\mathbf{A}$ can be approximated with shape functions as before. Here the relation between $\mathbf{H}$ and $\mathbf{B}$ can be nonlinear and also hysteretic. We can evaluate $\mathbf{B}$ from $\mathbf{A}$ with

$$\mathbf{B} = \mathrm{curl}\,(\mathbf{A}) = \mathrm{curl}\left(\sum_{i=1}^{N} \mathbf{N}_i(\mathbf{x}) a_i \right) = \sum_{i=1}^{N} \mathrm{curl}\,(\mathbf{N}_i(\mathbf{x}))\, a_i \ . \tag{2.54}$$

Magnetic field strength $\mathbf{H}$ can be evaluated from $\mathbf{B}$ by using any kind of material model. The downside is that this evaluation must be done at each iteration and it is difficult to separate linear and nonlinear regions from each other. With stiffness matrix formulation it is possible to separate a linear and a nonlinear stiffness matrix and if the nonlinear region of the domain is small updating the nonlinear stiffness matrix requires only minimal amount of work.

2.3.4 Applying a time stepping scheme

There are numerous time stepping schemes available. In this thesis we will select a simple and robust scheme which is known as Backward-Euler (BW-Euler) scheme. In BW-Euler the time derivative term is approximated as a difference between consecuent timesteps

$$\frac{\partial}{\partial t}x = \frac{x_t - x_{t-1}}{\Delta t} \ , \tag{2.55}$$

where Δt is the length of the timestep. If we have a PDE system

$$\mathrm{M}\frac{\partial}{\partial t}\mathbf{x} = \mathbf{f}(\mathbf{x}) + \mathbf{b}(t) \ , \tag{2.56}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state of the system, $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a function of the state, $t \in \mathrm{T}$ a time variable and $\mathbf{b} : \mathrm{T} \rightarrow \mathbb{R}^n$ is a source term we can apply BW-Euler scheme as

$$\mathrm{M}\,\frac{\mathbf{x}_t - \mathbf{x}_{t-1}}{\Delta t} = \mathbf{f}(\mathbf{x}_t) + \mathbf{b}(t_t) \ . \tag{2.57}$$

Since the next state value $\mathbf{x}_t$ is also used as an argument to $\mathbf{f}$ we must solve $\mathbf{x}_t$ from (2.57). If $\mathbf{f}$ is linear i.e $\mathbf{f}(\mathbf{x}) = \mathrm{A}\,\mathbf{x}$, where A is some matrix the next step can be solved using the tools of linear algebra as

$$\mathbf{x}_t = \left(\frac{1}{\Delta t}\mathrm{M} - \mathrm{A}\right)^{-1}\left(\frac{1}{\Delta t}\mathrm{M}\,\mathbf{x}_{t-1} + \mathbf{b}(t_t)\right) \ . \tag{2.58}$$

If $\mathbf{f}$ is nonlinear some iterative solving method for example Newton-Rapson method must be used.

2.3.5 Newton-Raphson method

Newton-Raphson (NR) method is a popular numerical solving method which can be used to solve a wide variety of nonlinear equation systems. In NR method an initial guess is made. Then the Jacobian matrix of a residual term is evaluated and the next iterate is calculated from the previous one. As an example we use the time discretized form of (2.56) which is given in (2.57). Firstly we introduce a residual term which is achieved by moving all terms to the same side on the equation

$$\mathbf{R}(\mathbf{x}_t) = \mathrm{M}\,\frac{\mathbf{x}_t - \mathbf{x}_{t-1}}{\Delta t} - \mathbf{f}(\mathbf{x}_t) - \mathbf{b}(t_t) \ . \tag{2.59}$$

We are searching solution for the following equation system

$$\mathbf{R}(\mathbf{x}_t) = 0 \ . \tag{2.60}$$

The Jacobian matrix of the residual is calculated as

$$\mathrm{J}(\mathbf{x}_t) = \frac{\partial \mathbf{R}(\mathbf{x}_t)}{\partial \mathbf{x}_t} \ , \tag{2.61}$$

Here the notation implies that we take a partial derivative of $\mathbf{R}$ with respect to each state variable. Componentwise this is stated as

$$\mathrm{J}_{ij}(\mathbf{x}_t) = \frac{\partial R_i(\mathbf{x}_t)}{\partial(\mathbf{x}_t)_j} \ . \tag{2.62}$$

We mark the k:th iteration of the solution with superscripts i.e. the k:th iteration of $\mathbf{x}_t$ is $\mathbf{x}_t^k$. We mark the initial guess with $\mathbf{x}_t^0$. The next iterate can be calculated from the previous one with

$$\mathbf{x}_t^{k+1} = \mathbf{x}_t^k - \mathrm{J}^{-1}(\mathbf{x}_t^k)\,\mathbf{R}(\mathbf{x}_t^k) \ . \tag{2.63}$$

The iteration is continued until a stopping condition is reached. Popular choices for a stopping condition are for example if the norm of $\mathbf{R}$ is smaller than some tolerance ε_{R}

$$||\mathbf{R}(\mathbf{x}_t^{k+1})|| < \varepsilon_{\mathrm{R}} \tag{2.64}$$

or the norm of the step size in relation to current iterate is smaller than some tolerance ε_{s}

$$\frac{||\mathrm{J}^{-1}(\mathbf{x}_t^k)\,\mathbf{R}(\mathbf{x}_t^k)||}{||\mathbf{x}_t^k||} < \varepsilon_{\mathrm{s}} \ . \tag{2.65}$$

Here the norms are usually Euclidean norms of the vector spaces $\mathbb{R}^n$.

NR method works well in dynamic problems as we can usually have the last timestep state as an initial quess to the next timestep. The continuity and differentiability of the state variables in electromagnetics ensure that next timestep solution is found relatively close from the previous one.

2.3.6 General formulation for time-stepping analysis

When BW-Euler method is used to (2.52) we get

$$\mathrm{M}\,\frac{\mathbf{a}_t - \mathbf{a}_{t-1}}{\Delta t} + \mathbf{f}(\mathbf{a}_t) = \mathbf{F}(t_t) \tag{2.66}$$

from this equation $\mathbf{a}_t$ must be solved with NR method. We mark time instants of the quantities with subscripts and and NR iterations with superscripts. Residual term for (2.66) is

$$\mathbf{R}(\mathbf{a}_t^k) = \mathrm{M}\,\frac{\mathbf{a}_t^k - \mathbf{a}_{t-1}}{\Delta t} + \mathbf{f}(\mathbf{a}_t^k) - \mathbf{F}(t_t) \ . \tag{2.67}$$

Next the Jacobian matrix must be calculated. We start with equation (2.62) and insert the residual term

$$\mathrm{J}(\mathbf{a}) = \frac{\partial}{\partial \mathbf{a}} \left(\mathrm{M} \frac{\mathbf{a} - \mathbf{a}_{t-1}}{\Delta t} + \mathbf{f}(\mathbf{a}) - \mathbf{F}(t_t) \right) . \tag{2.68}$$

Since derivative is linear we can take the operator inside the term. Terms $\mathbf{a}_{t-1}$ and $\mathbf{F}(t_t)$ are constants with respect to $\mathbf{a}$ and M is a constant multiplier. Therefore the Jacobian reduces to

$$\mathrm{J}(\mathbf{a}) = \frac{\mathrm{M}}{\Delta t} + \frac{\partial}{\partial \mathbf{a}} \mathbf{f}(\mathbf{a}) . \tag{2.69}$$

Next the nonlinear term shall be handled separately using index notation. Recall from (2.53) how the nonlinear term was calculated

$$\frac{\partial}{\partial a_j} f_i(\mathbf{a}) = \frac{\partial}{\partial a_j} \int_\Omega \mathrm{curl}\,(\mathbf{N}_i) \cdot \mathbf{H}(\mathbf{B}) \mathrm{d}\Omega . \tag{2.70}$$

Again by using the linarity of the derivative and integral we can move the derivative term inside the integral. The basis functions don't depend on $\mathbf{a}$.

$$= \int_\Omega \mathrm{curl}\,(\mathbf{N}_i) \cdot \frac{\partial}{\partial a_j} \mathbf{H}(\mathbf{B}) \mathrm{d}\Omega \tag{2.71}$$

Next we can use the chain rule for derivative to change $\frac{\partial \mathbf{H}}{\partial a_j} = \frac{\partial \mathbf{H}}{\partial \mathbf{B}} \frac{\partial \mathbf{B}}{\partial a_j}$

$$= \int_\Omega \mathrm{curl}\,(\mathbf{N}_i) \cdot \frac{\partial \mathbf{H}(\mathbf{B})}{\partial \mathbf{B}} \frac{\partial \mathbf{B}}{\partial a_j} \mathrm{d}\Omega . \tag{2.72}$$

Magnetic flux density $\mathbf{B}$ can be calculated from $\mathbf{a}$ using (2.54) and we get the following form

$$= \int_\Omega \mathrm{curl}\,(\mathbf{N}_i) \cdot \frac{\partial \mathbf{H}(\mathbf{B})}{\partial \mathbf{B}} \frac{\partial}{\partial a_j} \sum_{k=1}^{N} \mathrm{curl}\,(\mathbf{N}_k)\, a_k \mathrm{d}\Omega . \tag{2.73}$$

The derivative can be moved inside the sum and over the basis functions

$$= \int_\Omega \mathrm{curl}\,(\mathbf{N}_i) \cdot \frac{\partial \mathbf{H}(\mathbf{B})}{\partial \mathbf{B}} \sum_{k=1}^{N} \mathrm{curl}\,(\mathbf{N}_k) \frac{\partial a_k}{\partial a_j} \mathrm{d}\Omega . \tag{2.74}$$

Since $\frac{\partial a_k}{\partial a_j} = \delta_{kj}$, where δ is the Kronecker delta we get

$$= \int_\Omega \mathrm{curl}\,(\mathbf{N}_i) \cdot \frac{\partial \mathbf{H}(\mathbf{B})}{\partial \mathbf{B}} \sum_{k=1}^{N} \mathrm{curl}\,(\mathbf{N}_k)\, \delta_{kj} \,\mathrm{d}\Omega , \tag{2.75}$$

which reduces to

$$= \int_\Omega \mathrm{curl}\,(\mathbf{N}_i) \cdot \frac{\partial \mathbf{H}(\mathbf{B})}{\partial \mathbf{B}} \mathrm{curl}\,(\mathbf{N}_j)\,\mathrm{d}\Omega \; . \tag{2.76}$$

The final result is then that the Jacobian matrix can be calculated elementwise as

$$\mathrm{J}_{ij}(\mathbf{a}) = \frac{\mathrm{M}_{ij}}{\Delta t} + \int_\Omega \mathrm{curl}\,(\mathbf{N}_i) \cdot \frac{\partial \mathbf{H}(\mathbf{B})}{\partial \mathbf{B}} \mathrm{curl}\,(\mathbf{N}_j)\,\mathrm{d}\Omega \; . \tag{2.77}$$

where the term $\frac{\partial \mathbf{H}(\mathbf{B})}{\partial \mathbf{B}}$ can be achieved from the material model. It is called *differential reluctivity*. The Jacobian matrix depends on **a** since it is required to evaluate **B** which must be calculated from **a**.

3. MODEL REDUCTION METHODS

Model reduction methods are used to reduce the computational cost of models without sacrificing the accuracy of the results. Most of the methods aim to reduce the amount of unknowns or degrees of freedom in equation systems. Usage of reduced models can for example replace a need to come up with an equivalent circuit.

The benefit of a reduced order model is that after it has been created, it can be evaluated faster than the ordinary model which makes it possible to experiment with different inputs. The drop on evaluation time also makes it possible to attach reduced models to circuit simulators which use iterative time stepping methods to solve equations. Attaching a full model to a circuit simulator results in long computation times.

3.1 General idea of model reduction

As presented in [2, p. 17] model order reduction methods can roughly be divided into two categories: white box and black box methods. Black box methods can not access the internal structure of the model. Black box methods usually tend to be heavy on fitting and interpolating the output variables based on the input. These kinds of methods can utilize for example regression and neural networks. In this thesis the emphasis is on systematic application of the model reduction method and hence the black box methods are left outside the scope.

White box methods can use the internal structure of the model as a part of the reduction process. These techiques often use projections and basis manipulations. White box methods generally have more deterministic error bounds and certain properties of the system can be guaranteed to exist.

If the model belongs to a subtype of PDE problems called affinely parametrizable PDE problems the system can be reduced properly and all computations of the errors and results of the reduced model are independent of the original dimension of the problem. If however the model is non-affine the situation is more complex. Affinely parametrizable systems have been studied excessively in the past. For non-affine systems the challenge to develop a way to estimate the error between reduced order model and full order model which does not depend on the full dimension of the system remains to be found. [14]

As an example and illustration of a projection based method we can study the

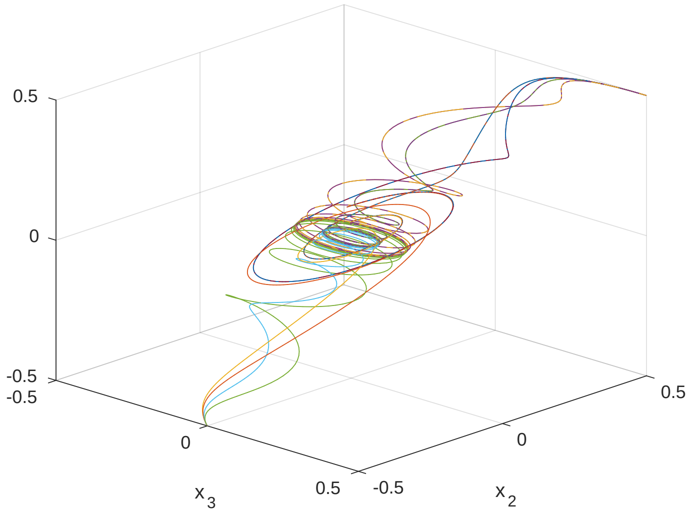


Figure 3.1: The trajectories in example system are attracted to the linear subspace $x_1 - x_2 = 0$. This follows from characteristics of operator A.

following system of equations presented in [2, pp. 36-37].

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} = \underbrace{\begin{bmatrix} -10 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \end{bmatrix}}_{=\mathrm{A}} \mathbf{x} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u(t), \tag{3.1}$$

where $\mathbf{x} = [x_1, x_2, x_3]^{\mathrm{T}}$ are state variables and $u(t)$ is the input.

When we plot the trajectories with several random sinusoidal inputs in Figure 3.1, we can see that the trajectories lie approximately in a two dimensional linear subspace. The task at hand is to identify that subspace and generate a projection operator which then is used to project the system operator and state variables to the identified subspace and solve the system using the basis of that subspace.

As there are vast amounts of model reduction methods available [3], [4] the range must be narrowed to find a suitable method for further study and application in the problem area of this thesis. The problem domain poses some restrictions to the method. The method must be applicable to nonlinear and dynamic problems in time domain, it must be able to reduce multiple input and multiple output systems and be automatizable.

There are numerous methods that are suitable for linear problems, Krylov Subspace [15], Balanced Truncation [16] and Cross Gramian to name a few. These methods have been applied succesfully in the area of control theory. In most of the cases they use the transfer function of the system in order to capture the most important poles and sinks to capture the behaviour of the system. Many of these methods have also been succesfully used on nonlinear problems using different kinds of local linearization and piecewise-linear constructions as presented in [2, p. 52] and [17, p. 63].

3.2 Proper orthogonal decomposition

Proper orthogonal decomposition (POD) is essentially based on principal component analysis. It has been succesfully utilized in multiple cases also in nonlinear magnetodynamics [18]. The authors in [19] have applied POD to nonlinear magnetostatics and later in [5] to nonlinear magnetodynamics with circuit coupling. POD poses no restrictions to the system it is applied to so it's applicable on dynamic nonlinear systems. In [6] the authors have introduced some manifold theory to interpolate the projection mappings to improve the accuracy of reduced system outside the training data.

POD works as a white box method. Based on some training data it extracts the principal modes of the system using the singular value decomposition (SVD). SVD is applied to a *snapshot set* which is collected out of some outputs of the model. After that a reduced order model can be derived which works by using the principal modes as a basis. In POD the internal equations are projected into a subspace which is identified by the POD. After computing the solution it can be mapped back to the original state space to obtain the approximated full solution. Usually we are not interested about the full solution. We usually are interested on some results which are computed out of the full solution. In these cases it is required to derive the end result out of the reduced solution in order to avoid the costly projection back to the original state space.

To formulate the POD method, we must first decide how to parametrize the system we are modelling. Parameters are for example dimensions of the device, thickness of the plates in a laminated transformer or some material parameters. In dynamic systems parameters can be related to the input signals of the system.

Consider for example a single-phase transformer. If we want to model how the transformer's output voltage behaves in time given a certain input voltage we could parametrize the system so that the waveform of the input voltage is parametrized. In this case if we would restrict the study to sinusoidal input voltages we could have voltage amplitude and frequency as parameters. The input signal of the system would be the voltage. We could also choose other parameters such as dimensions of the transformer or winding turns.

3.2.1 Generation of snapshots

All parameters are collected together to form a *parameter space* and we denote it with Ξ. Out of the parameter space we pick a set of parameters $D \subset \Xi$ which we also call *training set*. This set of parameter configurations is used to generate a *snapshot set* which we denote with S. The snapshot set is formulated into a $n \times k$ matrix, where n is the dimension of the equation system and k is the amount of snapshots.

The snapshot set contains *trajectories* of the system which are calculated out of parameter configurations. In static models snapshots are captured by solving the full system with different inputs. In dynamic cases a time evolution of the system can be solved and state snapshots can be captured from each timestep. These steps can then be added to snapshot matrix S.

There are different strategies on how to pick the training set. A simple way is to sample the parameter space with a constant interval. This results in a large training set which can possibly be infeasible to process. One can also use some prior knowledge to heuristically achieve good results by guessing which areas of the parameter and input spaces have the most impact on the reduced model. Typically nonlinear region of the systems working domain requires more samples than a linear domain.

3.2.2 Forming the projection matrix

SVD will decompose the snapshot matrix S into a matrix product of three matrices

$$\mathrm{S} = \mathrm{V}^{\mathrm{T}}\,\Sigma\,\mathrm{W}\ , \tag{3.2}$$

where V contains the left singular vectors, W contains the right singular vectors and matrix Σ is a diagonal matrix which contains the singular values of matrix S in its diagonal ordered from the largest to the smallest

$$\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_n) \text{ where } \sigma_i > \sigma_j \text{ if } i < j\ . \tag{3.3}$$

The singular vectors in V and W are also ordered in a way that the i:th column corresponds to the i:th singular value.

The projection operator $\Psi : \mathbb{R}^n \to \mathbb{R}^m$ is a $m \times n$ matrix. It is formed from the rows of the product $\mathrm{V}^{\mathrm{T}}\Sigma$. To form a sufficiently accurate reduced model we only need to pick the singular vectors that correspond to the most dominant singular values. The tolerance limit affects the dimension m of the reduced space.

The question of how many singular vectors has to be picked to form a sufficiently accurate reduced model can be answered by using energy analysis of the POD modes. The energy associated with the system can be calculated from the singular values as

$$E = \sum_{i=1}^{n} \sigma_i^2 \ , \tag{3.4}$$

where n is the amount of singular values of the system. The POD approximation $\tilde{\mathbf{y}}$ of a state space vector $\mathbf{y}$ is defined as

$$\tilde{y}_j = \sum_{i=1}^{k} < y_j, \psi_i > \psi_i \ , \tag{3.5}$$

where k is the number of singular values taken into account, $\{\psi_i\}_{i=1}^{k}$ are the POD basis and $< \cdot, \cdot >$ is an inner product. It is known that POD basis minimizes the following 2-norm error

$$\sum_{j=1}^{k} ||y_j - \tilde{y}_j||^2 = \sum_{i=l+1}^{k} \sigma_i^2 \ , \tag{3.6}$$

where $\mathbf{y}$ is the exact solution of the problem being reduced. Furthermore, the error can be calculated from the singular values of the SVD as in 3.6 by summing squares of all singular values that are not taken into account [3]. So here we can actually set a tolerance parameter ε and include the POD basis vectors corresponding up to the l:th singular value where l is solved from

$$\sum_{i=1+l}^{k} \sigma_i^2 < \varepsilon \ . \tag{3.7}$$

This is the cut-off criterion which is proposed in [7].

3.2.3 Adaptive algorithms in snapshot generation

If training set Ξ is very large it quickly becomes infeasible to form the snapshot matrix S and to calculate the SVD of the matrix. To solve this problem there are several algorithms which help to pick only the most substantial parameter configurations that increase the accuracy of POD.

Some adaptive sampling algorithms are proposed in [20]. These rely on an error indicator function to calculate the error between the original model and the reduced model. The error indicator function is ideally computationally much lighter to evaluate than it is to solve the original model. From the parameter set a parameter configuration is searched which maximises the error between the solutions of the original system and the reduced system.

The original system is solved using the maximising parameters and the captured

```
Data: Parameter space $\mathcal{S} \subset \mathbb{R}^p$ , maximum iterations $N_{\max}$, error estimator $\mathcal{I}$,
      full order model $\mathcal{M}$. $\mathcal{M}$ takes an input $x$ and a parameter
      configuration $\mu$ and outputs a trajectory of the state $T_a$ and a
      trajectory of the nonlinear term $T_f$.
Result: Reduced order model $\mathcal{R}_{\text{out}}$
Initialize state snapshot set $S_a = \emptyset$ ;
Initialize nonlinear term snapshot set $S_f = \emptyset$;
Take the min and max values of each parameter;
Create a set $P = \{$ all combinations of min and max values $\}$;
for $\mu$ in $P$ do
    Evaluate model $(T_a, T_f) = \mathcal{M}(x(\mu),\, \mu)$;
    $S_x \leftarrow S_x \cup T_x,\; S_f \leftarrow S_f \cup T_f$;
end
Generate initial reduced order model $\mathcal{R}_0$ from $S_a$ and $S_f$;
for $i$ in $[1 \cdots N_{\max}]$ do
    Search for $\mu_{\max}$ which maximises $\mathcal{I}(\mathrm{S}_a)$, where $\mathrm{S}_a$ is a trajectory calculated
      using $\mathcal{R}_{i-1}$ and the parameters $\mu \in \mathcal{S}$;
    Evaluate model $(T_a, T_f) = \mathcal{M}(x(\mu),\, \mu_{\max})$;
    $S_a \leftarrow S_a \cup T_a,\; S_f \leftarrow S_f \cup T_f$;
    Generate new reduced order model $\mathcal{R}_i$ from $S_x$ and $S_f$;
end
$\mathcal{R}_{\text{out}} \leftarrow \mathcal{R}_{N_{\max}}$;
```

Algorithm 1: Adaptive reduced order model generation algorithm used in this thesis. Based on algorithms introduced in [20].

snapshots are added into the snapshot set and a new reduced model is derived at each iteration. This results in a more optimal reduced order model with a given parameter set. The process can also be interrupted when a certain tolerance is met or some time constraint is passed.

In this thesis Algorithm 1 is used to construct the reduced order model. The algorithm takes a bounded parameter space, an error estimator function and a full order model as its input and returns the reduced order model. In Algorithm 1 the full order model $\mathcal{M}$ is formulated to take input variables x and parameter configuration μ in and produce a time evolution of the state variables T_a and the values of the nonlinear term T_f. These trajectories are added to snapshot sets and in each iteration a new reduced order model is derived. In the example transformer case in Chapter 5 the input variables x which are the input currents of the coils actually depend on the parameter configuration μ.

The error estimator used here is introduced in [20]. It is based on a residual term which is defined as

$$\mathbf{r}(\tilde{\mathbf{a}}) = \mathrm{M}\frac{\partial}{\partial t}\Psi\tilde{\mathbf{a}} + \mathbf{f}(\Psi\tilde{\mathbf{a}}) - \mathbf{F}(t) \tag{3.8}$$

The actual error estimator function $\mathcal{I}$ is based on the norm of residual $\mathbf{r}$. For any time evolution of the state $\mathbf{a}$ which is solved from the full model the residual of the solution of each time step should be zero.

To estimate the error between the reduced order model and the full model we solve a time evolution of the reduced state. Then we feed every timestep of the reduced state to the residual by projecting it to the original state space with Ψ. The error at each timestep is given by the 2-norm of $\mathbf{r}$ and the complete error function is a root mean square error of the resulting error evolution. If we denote the reduced state trajectory with $\mathrm{S}_a = \{\tilde{\mathbf{a}}_i\}_{i=1}^{T}$ where T is the number of timesteps then

$$\mathcal{I}(\mathrm{S}_a) = \sqrt{\frac{1}{T}\sum_{i=1}^{T} ||\mathbf{r}(\Psi\tilde{\mathbf{a}}_i)||^2} \ , \tag{3.9}$$

where Ψ comes from the reduced model the error of which is being compared.

Yet better adaptive approach is to build an approximation of the error function over the parameter domain. In [20] a Kriging surrogate model is constructed at every iteration. With this model it is possible to calculate the areas of the parameter space where the likelihood of large errors is the greatest. These areas of parameter set are then refined. The benefits of this kind of algorithm are that one can start with a very sparse set of parameters and terminate when some tolerance is met. With simpler models this process can be fast compared to naive sampling strategies.

3.3 Discrete empirical interpolation method

Discrete empirical interpolation method (DEIM) can be used to reduce the amount of work required to evaluate some computations. It is an interpolation method which identifies the most meaningful components from a vector valued function. In DEIM interpolation coefficients are calculated which are used to evaluate the rest of the components based on the values for the most meaningful components.

DEIM requires a linear subspace and an orthogonal basis for it to work upon. In this thesis we use the POD basis as a basis for DEIM interpolation. This technique is known as POD-DEIM method. In this section the idea of DEIM and DEIM algorihm is presented as in [21].

3.3.1 Background and idea

Assume a nonlinear vector valued function $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n \ , n \in \mathbb{Z}_+$. Because $\mathbf{f}$ is actually a function which takes n real numbers as an input and produces n real numbers as an output the computation complexity must depend somehow on n. If the evaluation complexity of one component of $\mathbf{f}$ is a constant c for every component the total evaluation complexity is then nc. Consider for example a componentwise applied function $\mathbf{f}$

$$\mathbf{f}_i(\mathbf{x}) = f(x_i)\ , f : \mathbb{R} \to \mathbb{R} \tag{3.10}$$

where f takes each component in and produces an output corresponding to the input component. In this case nN_f computations must be made where N_f is the amount of computations required to evaluate f. DEIM interpolation reduces this so that only a subset of the components of $\mathbf{f}$ must be calculated for each input. This will result in a drop in required work especially if f is heavy to compute. If the set of selected DEIM indices is much smaller than n the drop in the amount of work will be significant.

3.3.2 Creating the interpolation

Firstly a linear subspace with a basis for the subspace is required. This subspace and the basis vectors will be obtained using the POD method. Assume that the subspace is of dimension $m < n$ and the n dimensional basis vectors of the subspace are $\{\mathbf{u}_i\}_{i=1}^m$. Now we can approximate $\mathbf{f}$ by projecting it to the subspace by using a projection matrix $\mathrm{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_m] \in \mathbb{R}^{n\times m}$

$$\mathbf{f}(\mathbf{x}) \approx \mathrm{U}\,\mathbf{c}(\mathbf{x})\ , \tag{3.11}$$

where $\mathbf{c}(\mathbf{x})$ are interpolation coefficients. Coefficients are solved from the overdetermined system $\mathbf{f}(\mathbf{x}) = \mathrm{U}\,\mathbf{c}(\mathbf{x})$ by selecting m rows to form a solvable system.

```
Data: POD basis $\mathrm{U} = \{\mathbf{u}_i\}_{i=1}^m$
Result: Column vector of DEIM indices $\mathbf{p} = \{p_i\}_{i=1}^m$
$p_1 = \texttt{indmax}(|\mathbf{u}_1|)$;
$\mathrm{U} = [\mathbf{u}_1],\ \mathrm{P} = [\hat{\mathbf{e}}_{p_1}],\ \mathbf{p} = [p_1]$;
for $l = 2$ to $m$ do
    Solve $(\mathrm{P}^\mathrm{T}\mathrm{U})\mathbf{c} = \mathrm{P}^\mathrm{T}\mathbf{u}_l$ for $\mathbf{c}$;
    $\mathbf{r} = \mathbf{u}_l - \mathrm{U}\mathbf{c}$;
    $p_l = \texttt{indmax}(|\mathbf{r}|)$;
    $\mathrm{U} \leftarrow [\mathrm{U}, \mathbf{u}_l],\ \mathrm{P} \leftarrow [\mathrm{P}, \hat{\mathbf{e}}_{p_l}],\ \mathbf{p} \leftarrow \begin{bmatrix} \mathbf{p} \\ p_l \end{bmatrix}$;
end
```

Algorithm 2: DEIM algorithm [21]. In this algorithm the function `indmax` takes a vector and returns the index of the maximum element of that vector.

To select the rows from the overdetermined system we use Algorithm 2 aka DEIM algorithm to form a set of indices $p = \{p_i\}_{i=1}^m$. These indices are used to form a selector matrix $\mathrm{P} = [\mathbf{e}_{p_1}, \cdots, \mathbf{e}_{p_m}] \in \mathbb{R}^{n \times m}$ where $\mathbf{e}_{p_i}$ is a natural n dimensional basis vector i.e vector where the p_i:th element is 1 and the rest are zero.

Multiplying a vector with the matrix P selects m elements from the vector. Coefficients $\mathbf{c}(\mathbf{x})$ can therefore be solved from

$$\mathrm{P}^{\mathrm{T}}\,\mathbf{f}(\mathbf{x}) = (\mathrm{P}^{\mathrm{T}}\mathrm{U})\,\mathbf{c}(\mathbf{x}) \tag{3.12}$$

$$\Leftrightarrow \mathbf{c}(\mathbf{x}) = (\mathrm{P}^{\mathrm{T}}\mathrm{U})^{-1}\mathrm{P}^{\mathrm{T}}\,\mathbf{f}(\mathbf{x})\ . \tag{3.13}$$

provided that $\mathrm{P}^{\mathrm{T}}\mathrm{U}$ is nonsingular. A proof which shows that $\mathrm{P}^{\mathrm{T}}\mathrm{U}$ actually is nonsingular at every step of DEIM algorithm if and only if the set of basis vectors $\{\mathbf{u}_i\}_{i=1}^{m}$ is linearly independent (which it is since it is a *basis*) is presented in [21].

When we insert the solved $\mathbf{c}(\mathbf{x})$ to (3.11) we get the final form of DEIM approximation

$$\mathbf{f}(\mathbf{x}) \approx \mathrm{U}\,(\mathrm{P}^{\mathrm{T}}\mathrm{U})^{-1}\mathrm{P}^{\mathrm{T}}\,\mathbf{f}(\mathbf{x})\ . \tag{3.14}$$

It is important to note that in (3.14) $\mathrm{U}\,(\mathrm{P}^{\mathrm{T}}\mathrm{U})^{-1}$ can be precomputed and this must be done only once. Other important aspect is that the term $\mathrm{P}^{\mathrm{T}}\,\mathbf{f}(\mathbf{x})$ can be evaluated so that only the elements of $\mathbf{f}(\mathbf{x})$ which are in the set of DEIM indices p must be evaluated. In the example (3.10) this results in mc computations, where c is the complexity of the computation of $\mathbf{f}$ to evaluate $\mathrm{P}^{\mathrm{T}}\,\mathbf{f}(\mathbf{x})$. Then a matrix product between $n \times m$ matrix and m dimensional vector is computed to obtain the values for the rest of the elements of $\mathbf{f}$.

3.4 Reduced formulation of linear magnetodynamic problems

To create a reduced order model from (2.43) we need to project the state vector $\mathbf{a}$ and operators S and M by using the projection matrix Ψ. The starting point is that we approximate the full state $\mathbf{a}$ by using a reduced state vector $\tilde{\mathbf{a}}$ and Ψ

$$\mathbf{a} \approx \Psi\tilde{\mathbf{a}}\ . \tag{3.15}$$

We begin the derivation with rearranged equation (2.43)

$$\mathrm{M}\frac{\partial}{\partial t}\mathbf{a} + \mathrm{S}\mathbf{a} - \mathbf{F} = \mathbf{0} \tag{3.16}$$

into which the equation (3.15) is inserted. Since equation (3.15) is an approximation the new equation will not be exactly zero. We will define a residual $\mathbf{R}$ as

$$\mathbf{R}(\tilde{\mathbf{a}}) = \mathrm{M}\,\frac{\partial}{\partial t}\Psi\tilde{\mathbf{a}} + \mathrm{S}\,\Psi\tilde{\mathbf{a}} - \mathbf{F} \tag{3.17}$$

which must then be minimized. Now the residual is required to be ortoghonal and hence minimal to the full system by projecting the residual with Ψ^{T} and requiring the projection to be zero

$$\Psi^{\mathrm{T}}\mathbf{R}(\tilde{\mathbf{a}}) = \mathbf{0} \tag{3.18}$$

$$\Rightarrow \Psi^{\mathrm{T}}\mathrm{M}\,\frac{\partial}{\partial t}\Psi\tilde{\mathbf{a}} + \Psi^{\mathrm{T}}\mathrm{S}\,\Psi\tilde{\mathbf{a}} - \Psi^{\mathrm{T}}\mathbf{F} = \mathbf{0} \tag{3.19}$$

$$\Rightarrow \Psi^{\mathrm{T}}\mathrm{M}\,\Psi\frac{\partial}{\partial t}\tilde{\mathbf{a}} + \Psi^{\mathrm{T}}\mathrm{S}\,\Psi\tilde{\mathbf{a}} - \Psi^{\mathrm{T}}\mathbf{F} = \mathbf{0}\,. \tag{3.20}$$

From this form it can be seen that the reduced operators are

$$\tilde{\mathrm{M}} = \Psi^{\mathrm{T}}\mathrm{M}\Psi \;\; \in \mathbb{R}^{m\times m} \tag{3.21}$$

$$\tilde{\mathrm{S}} = \Psi^{\mathrm{T}}\mathrm{S}\Psi \;\; \in \mathbb{R}^{m\times m} \tag{3.22}$$

$$\tilde{\mathbf{F}} = \Psi^{\mathrm{T}}\mathbf{F} \;\; \in \mathbb{R}^{m}. \tag{3.23}$$

The reduced order system is the same as (2.43) but its dimension has reduced from n to $m << n$.

$$\tilde{\mathrm{M}}\frac{\partial}{\partial t}\tilde{\mathbf{a}} + \tilde{\mathrm{S}}\tilde{\mathbf{a}} = \tilde{\mathbf{F}} \tag{3.24}$$

The term $\tilde{\mathbf{F}}$ actually needs some special treatment as it is a source term which may vary in time. This causes that $\mathbf{F}$ must be projected again after each time step and the complexity of evaluating it still depends on n. The projection of linear problems is straightforward as the system operators need to be projected only once and they can then be used to solve the system in each timestep.

3.5 Reducing nonlinear hysteretic magnetodynamic problems

In this thesis we aim to produce a method which can be generally used to reduce problems which are both nonlinear and hysteretic. In [5], [6], [18], [19] POD-DEIM method has been applied to a nonlinear stiffness matrix formulation of the system which is presented in Equation 2.45. This formulation however does not allow hysteresis effect to be taken into account. A finite element formulation for the system where the behaviour of the material can be nonlinear and hysteretic was derived in Section 2.3.3. In this section we derive a reduced order model for the general formulation presented in Equation (2.52) by using both POD and DEIM.

3.5.1 State reduction using POD

In Section 2.3.3 we presented a general formulation for nonlinear and hysteretic problems. If we apply POD as we did previously starting with (3.15) to system (2.59) where time discretization is already applied we get a residual term

$$\mathbf{R}(\tilde{\mathbf{x}}) = \mathrm{M}\Psi \, \frac{\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-1}}{\Delta t} - \mathbf{f}(\Psi \tilde{\mathbf{x}}_t) - \mathbf{b}(t_t) \; . \tag{3.25}$$

Then by left multiplying with Ψ^{T} again and requiring orthogonality we get

$$\Psi^{\mathrm{T}}\mathrm{M}\Psi \, \frac{\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-1}}{\Delta t} - \Psi^{\mathrm{T}}\mathbf{f}(\Psi \tilde{\mathbf{x}}_t) - \Psi^{\mathrm{T}}\mathbf{b}(t_t) = \mathbf{0} \; . \tag{3.26}$$

The reduced damping matrix is as before $\tilde{\mathrm{M}} = \Psi^{\mathrm{T}}\mathrm{M}\Psi$. However the nonlinear term $\tilde{\mathbf{f}}(\Psi\tilde{\mathbf{x}}_t) = \Psi^{\mathrm{T}}\mathbf{f}(\Psi\tilde{\mathbf{x}}_t)$ requires some special attention.

First to evaluate the nonlinear term a projection to full state space must be made with $\Psi\tilde{\mathbf{x}}_t$. This operation depends on the full dimension n. After the projection $\mathbf{f}$ must be evaluated. The evaluation of $\mathbf{f}$ here is computationally as heavy as it is in the full model. Lastly we must project the result of this evaluation back to the reduced subspace which again depends on n. All this must be done for each NR iteration inside each timestep which results in a poor reduction with the term $\tilde{\mathbf{f}}$.

The benefit of POD reduction is that it reduces the equation system to be of size $m << n$. This results in less work when in NR iteration the next iterate is solved using the Jacobian and the residual. The complexity of solving a group of dense linear equations is approximately $\mathcal{O}(k^3)$ where k is the dimension of the system. In sparse systems the complexity depends on the amount of nonzero elements in the system matrix [22]. The workload per iteration will be reduced but at the same time we introduce two excess matrix products which both are also of similar complexity $\mathcal{O}(k^2)$. Therefore the overall speed boost with POD alone is not remarkable.

In [23, pp.1918-1924] the savings which POD can achieve in linear and nonlinear time invariant cases are investigated. It is stated that in a linear time-invariant system a full model with a sparse system matrix solved using the BW-Euler method is of complexity $\mathcal{O}(b^2 n)$, where b is the band width of the system matrix. In dense systems this complexity is $\mathcal{O}(n^3)$. The complexity of the reduced model in both cases is $\mathcal{O}(m^3)$. Since m should be a lot smaller than n the theoretical savings which can be achieved using POD can be significant. In nonlinear cases the complexity depends heavily on the evaluation complexity of the nonlinear term and therefore it is important to reduce the complexity of the nonlinear term evaluation as well which can be done using the DEIM.

3.5.2 Applying DEIM to nonlinear term

We can collect a second set of snapshots which consists of the values of the term $\mathbf{f}(\mathbf{x}_t)$. This can be done at the same time as we generate snapshots to form the POD reduction for the state.

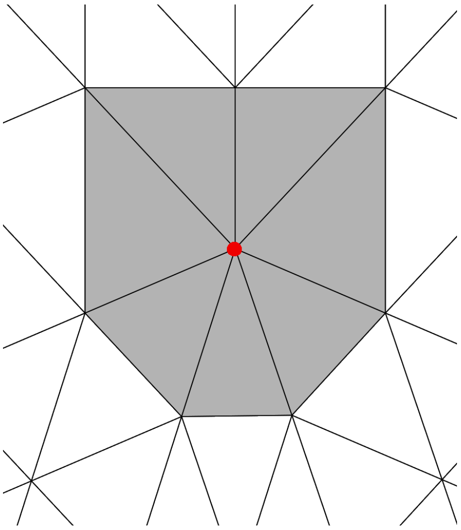


Figure 3.2: The support of the basis functions related to the red node are nonzero only inside the neighbouring elements which are grey.

Let's assume that we build a POD basis for $\mathbf{f}$ which is of dimension d which is the same as the amount of DEIM nodes. We can then interpolate $\mathbf{f}$ with

$$\mathbf{f}(\mathbf{x}_t) \approx \mathrm{U}\,(\mathrm{P}^{\mathrm{T}}\mathrm{U})^{-1}\mathrm{P}^{\mathrm{T}}\,\mathbf{f}(\mathbf{x}_t)\ , \tag{3.27}$$

where P is the selector matrix from DEIM algorithm and U the POD projector of the nonlinear term. When this is inserted into (3.26) we get

$$\tilde{\mathrm{M}}\,\frac{\tilde{\mathbf{x}}_t - \tilde{\mathbf{x}}_{t-1}}{\Delta t} - \Psi^{\mathrm{T}}\,\mathrm{U}\,(\mathrm{P}^{\mathrm{T}}\mathrm{U})^{-1}\mathrm{P}^{\mathrm{T}}\,\mathbf{f}(\Psi\tilde{\mathbf{x}}_t) - \Psi^{\mathrm{T}}\mathbf{b}(t_t) = \mathbf{0}\ . \tag{3.28}$$

Now as was presented in Section 3.3.2 we can evaluate the right hand side of the Equation (3.27) in parts. Term $\Psi^{\mathrm{T}}\,\mathrm{U}\,(\mathrm{P}^{\mathrm{T}}\mathrm{U})^{-1}$ can be evaluated only once and it will now result in a $m \times d$ matrix where m is the dimension of the reduced state space. The evaluation of the term $\mathrm{P}^{\mathrm{T}}\,\mathbf{f}(\Psi\tilde{\mathbf{x}}_t)$ is equivalent to a situation where we only calculate d rows from the output of $\mathbf{f}$. After this we need to perform a multiplication of an $m \times d$ matrix and a d-dimensional vector. At this stage there is no dependency on the original dimension n anymore.

Here $\mathbf{f}$ behaves similarly to a componentwise applied function (see Section 3.3.1). This follows from FEM formulation. Every component of $\mathbf{f}$ corresponds to a node in the FE mesh. If we evaluate a single component i of $\mathbf{f}$ we get only terms that emerge from all neighbouring elements of node i. This follows from the fact that the support of the shape functions which correspond to node i is nonzero only in the neighbouring elements. This is illustrated in Figure 3.2.

The amount of neighbouring elements denoted here with e_{n} is different depending on the mesh but typically in two-dimensional meshes $e_{\mathrm{n}} < 10$. Also e_{n} doesn't depend drastically on the amount of nodes or elements of the mesh. Let us denote the mean of the amount of nodes present in all neighbouring elements of a single DEIM node with l. It is therefore only required to calculate ld rows of $\Psi\tilde{\mathbf{x}}_t$ to do evaluations in the neighbouring elements of the DEIM nodes. This can be done efficiently by selecting only the rows corresponding the ld nodes from Ψ and multiplying the resulting $ld \times m$ matrix with $\tilde{\mathbf{x}}_t$.

After projection we can evaluate the nodal values of $\mathbf{f}$. We only need to evaluate those d components of $\mathbf{f}$ which are given by the DEIM algorithm. Each component requires that we evaluate terms in the nodes of the neighbouring elements of the node which corresponds to the component we are evaluating. So the complexity of this phase of calculation is $\mathcal{O}(ld)$ instead of $\mathcal{O}(ln)$. In every timestep it is then required to evaluate d elements of $\mathbf{f}$ and calculate the matrix-vector product between an $m \times d$ matrix and a d-dimensional vector which reduces the amount of work greatly because $m, d << n$.

4. COUPLING WITH CIRCUIT SIMULATOR

This chapter introduces some key concepts of the circuit theory. There exists a lot of literature concerning circuit theory. The basic concepts are introduced here by referring to [24, pp. 11-29].

4.1 Background of circuit theory

Circuit theory is an important tool for engineers. The mathematical model for a circuit is formed by a graph which basically is a set of *nodes* and a set of *arcs* which are connecting the nodes. Electrical components are modelled as arcs which have terminals that in turn are connected to nodes. In this way we can create a network of components. The wires between the terminals of components are assumed to be ideal conductors.

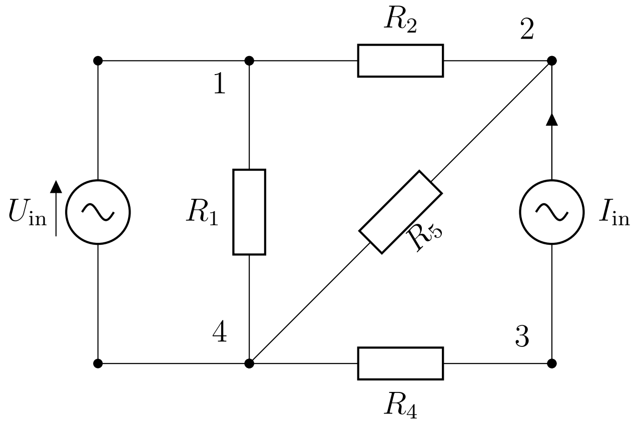


Figure 4.1: An example circuit which has a alternating voltage source, alternating current source and a network of resistors.

For example based on the circuit presented in Figure 4.1 a graph which is presented in Figure 4.2 can be created. The graphs adjacency matrix contains the information about the connections in the graph. The properties of the components in the network are associated with the edges as *weights* of the edges.

In general components have two quantities: through quantity and across quantity. In the case of electric circuits these are known as current and voltage. Arguably the simplest of the components, the resistor, has a through quantity which is current and an accross quantity which is the voltage over the resistor. This is illustrated

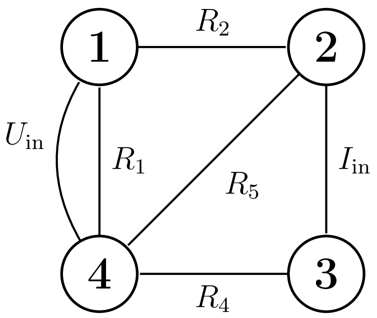


Figure 4.2: The graph representation of the circuit in Figure 4.1. The graph is used to generate required equation systems.

in Figure 4.3. The resistor also has a simple equation known as the Ohm's law describing it's behaviour. Two other basic components the inductor and the capacitor have also current and voltage as through and accross quantities but their behaviour is characterized by a first order differential equation. They bind together their through and accross quantities through the derivative of one of the quantities. Circuit theory can be used to model many other cases which have similar quatities such as pipelines or heat transfer.

The circuit theory is based on two physical laws which are derived from Maxwell's equations. These laws combined with topological information about the connections of the circuit are used to construct equation systems which are then solved. If all components of the circuit are linear the solution is then obtained by the resulting linear equation system. With nonlinear components iterative methods must be used.

The first law is known as Kirchoff's current law. This law is derived from Ampere's law (2.4) by choosing S to be a closed surface. Then it follows that ∂S is an empty set and the equation reduces to

$$0 = \int_S \left(\mathbf{J} + \frac{\partial}{\partial t}\mathbf{D} \right) \cdot \hat{\mathbf{n}} \, \mathrm{d}S \tag{4.1}$$

In circuit theory displacement currents are assumed to be zero in other parts of the circuit than capacitors. If volume enclosed by closed surface S contains just the node we are inspecting we can neglect the displacement current and the term reduces to the sum of the currents going in and out of the volume

$$0 = \sum_{i=1}^{N} I_i \; . \tag{4.2}$$

This equation states that the sum of the current coming into a node must equal to the sum of the currents going out from a node.

The second law is known as Kirchoff's voltage law. It is derived from Faraday's

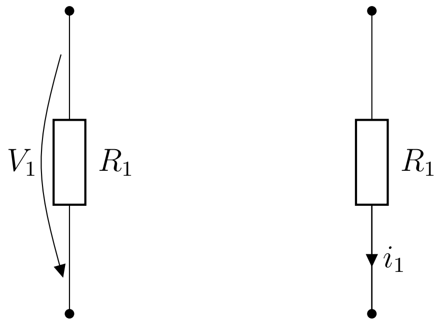


Figure 4.3: In graph theory voltage V_1 is called the across quantity and current i_1 is called the through quantity of the resistor R_1.

law (2.3). First we can define the voltage over an open path as $V_k = \int_{C_k} \mathbf{E} \cdot \mathrm{d}l$. Then we can define the induced voltage as $V_\mathrm{i} = \int_S \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{B} \cdot \hat{\mathbf{n}} \, \mathrm{d}S$. If we split the closed path around the surface S to separate paths we can state the integral over the boundary as a sum

$$0 = \sum_{k=1}^{N} V_k + V_\mathrm{i} \, . \tag{4.3}$$

This law states that the sum of all voltages in a closed loop inside a circuit is zero.

Using these basic building blocks we can construct solvable equation systems by using the adjacency data of the graph. Popular general methods are for example mesh current method or nodal analysis. The equations which describe the behaviour of the basic components are derived separately. In this thesis the most significant components are the resistor and the inductor as these are the components which are needed to create equivalent circuit models of transformers.

4.2 Simulink and Simscape

Simulink is a system simulator developed by MathWorks used especially to simulate control systems [25]. It is tightly integrated to Matlab scientific computing software. In Simulink the user can create blocks which have inputs, outputs and a certain behaviour. These blocks can then be connected together to model complex systems which can then be numerically solved. In Simulink the user has to create a mathematical model of the system he is modelling and then build a Simulink model based on that mathematical model.

Simscape is a physical system simulator also developed by MathWorks [26]. Simscape is aimed to be more physically oriented as Simulink. Simscape provides libraries to create for example actual physical electrical circuits and constructs the algebraic equation systems out of these circuit models. Simscape is implemented on top of Simulink so it is possible to combine them.

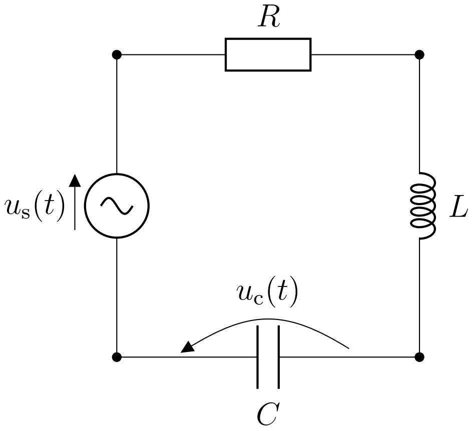


Figure 4.4: A simple RLC circuit. The voltage of the capacitor needs to be solved.

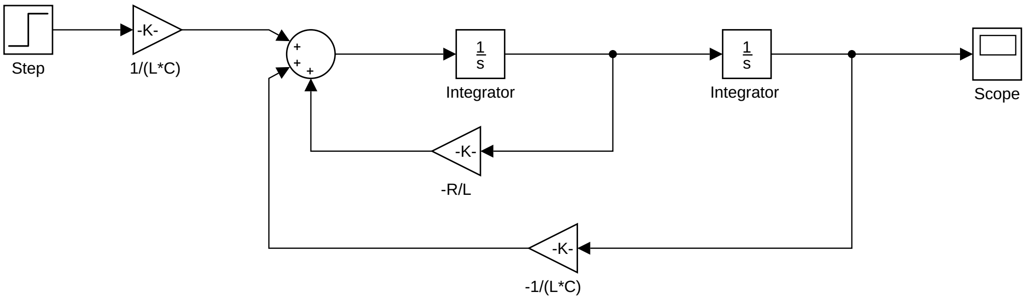


Figure 4.5: The RLC circuit modelled as a Simulink system. The system doesn't visually correspond to the original system which is being modelled.

The difference between Simscape and Simulink is easy to demonstrate using an example. If we have a RLC circuit which is presented in Figure 4.4 and we want to study the voltage of the capacitor during a transient we would first derive the following second order differential equation

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}u_c(t) + \frac{R}{L}\frac{\mathrm{d}}{\mathrm{d}t}u_c(t) + \frac{1}{LC}u_c(t) = u_s(t) \ , \tag{4.4}$$

where R, L, C are the resistance, inductance and the capacitance of the components respectively, u_c is the voltage of the capacitor and u_s the source voltage. The initial conditions are $u_\mathrm{c}(0) = 0\,\mathrm{V}$ and $\frac{\mathrm{d}}{\mathrm{d}t}u_\mathrm{c} = 0\,\frac{\mathrm{V}}{\mathrm{s}}$. Source voltage will step from $0\,\mathrm{V}$ to $1\,\mathrm{V}$ at $t = 0.05\,\mathrm{s}$.

In Simulink we implement the system following the differential equation at hand. To achieve the derivation of the state functions we must use integrator blocks. Then we add necessary gain multipliers and create feedback loops to model equation (4.4). The Simulink model of RLC circuit is presented in Figure 4.5.

In Simscape we just need to add the components and create the correct connections. Simscape will take care of the equation systems based on the topology

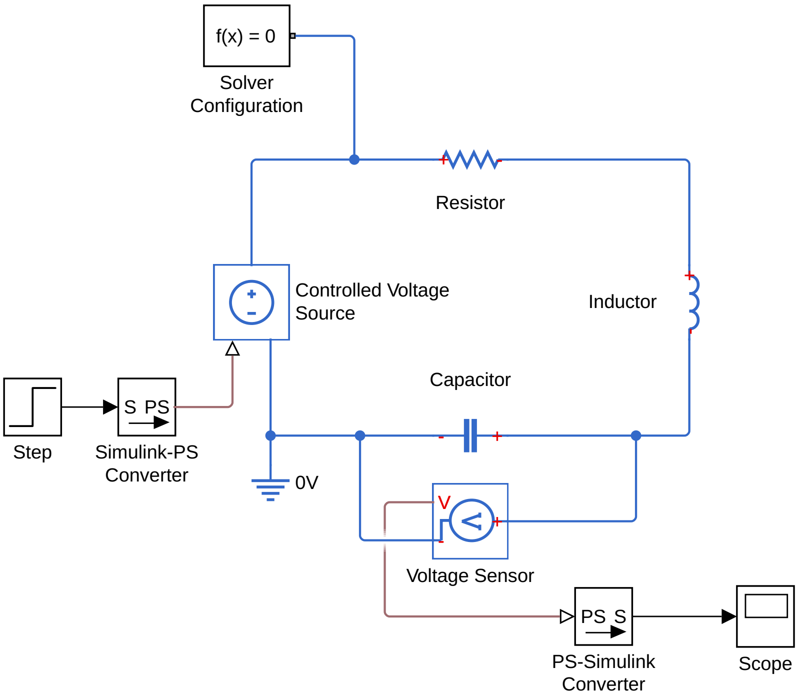


Figure 4.6: The RLC circuit modelled in Simscape. One can see the circuit, and the quantity we are interested in is measured using a voltage sensor block.

and the behaviour of the components. A Simscape model of the RLC circuit is presented in Figure 4.6. As can be seen the physical circuit modelling tools can greatly improve productivity as they abstract away the need to manually create the governing equation systems. Therefore modelling circuits in Simscape is somewhat more straightforward than modelling them in Simulink.

4.3 Integration of the reduced numerical model

The stategy of integrating the reduced model to Simscape is to do it in two steps. In step one the reduced model is integrated to Simulink environment. A Simulink subsystem is created which will be integrated to a physical circuit in step two. This high level approach will likely not produce the best possible performance but it will allow more flexibility.

Simulink offers a ”Matlab System” block which enables the user to derive a block that can utilize much of the tools Matlab provides inside its scripting environment. It works therefore as a good starting point to start the integration of the FEM model. The derivation of a new block which has inputs and outputs and an internal state is done by creating a new class and inheriting a base class of a Simulink block. If the component does not need an inner state a ”Matlab function” block would be sufficent. It allows the user to implement a standard Matlab script which is then called during the simulation.

Matlab system blocks can be used either in interpreted mode or in compiled mode. Compiled mode is faster but it has some rather constraining limitations. One can't for example use sparse matrices or file input or output in compiled mode. For these reasons it is decided to take the performance hit for now and use the system block in interpreted mode to have more flexibility. This limitation can later be worked around by implementing a sophisticated reduced model generator which capsulates all data it needs.

Simscape has voltage and current meters which can be used to couple quantities from physical circuit model to Simulink models. Simscape also has ideal voltage and current sources which can be used to couple quantities from Simulink model to Simscape model. These are the blocks which are used in the example system in Chapter 5.

In this thesis we are aiming to narrow the gap between detailed finite element models and external behavioral circuit models. Hence we aim to create a customizable physical model of a general electromagnetic device which can be connected to a physical circuit by just defining the physical terminals of the component.

5. CASE STUDY: EI TRANSFORMER

In this chapter the theory presented in previous chapters is applied to a simple single-phase transformer. A case study is needed to argue the feasibility of the circuit simulator connection and at the same time see the effects of the model reduction.

5.1 Description of the device

EI transformers have been used for a long time and they have become standardized components. The core of the transformer consists of two pieces shaped of the letters E and I. These pieces consist of some magnetic material. If the material is conducting the core is usually divided into thin plates. Each plate is insulated from one another to stop electrical currents from flowing between the plates. The result is that eddy current losses in the core drop dramatically in comparison to a massive core. Coils are inserted around the middle limb of the E letter shaped plate pack. Figure 5.1 shows how the coils are located inside the transformer core. It also shows the symmetries which are used to derive the modelling domain for the transformer. The modular design of this type of transformer allows quick prototyping and experimentation by changing the amount of plates in the core or by adjusting the air gap between the E and I parts. Coils can be changed easily as well.

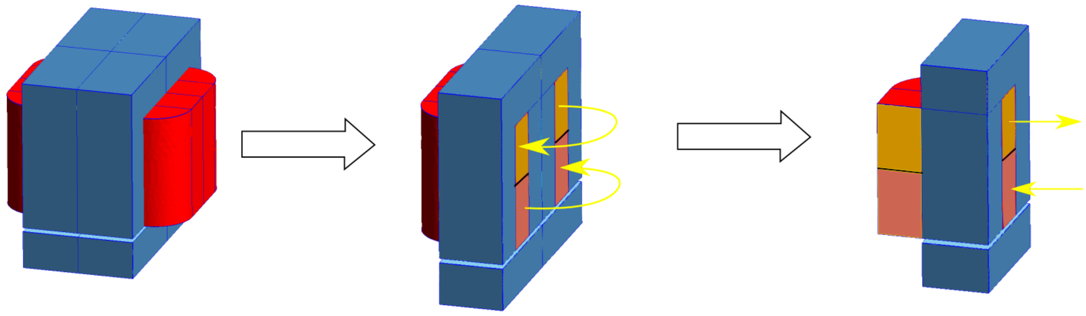


Figure 5.1: The construction of single-phase EI transformer. Final modelling domain is the right facing face of the quarter at the right. Coils are colored similarly as in Figure 5.2. Currents are marked with yellow arrows.

One specific use case for a single-phase transformer which needs to be modelled as a part of a larger system is in a solid state transformer (SST). SST is a DC-DC

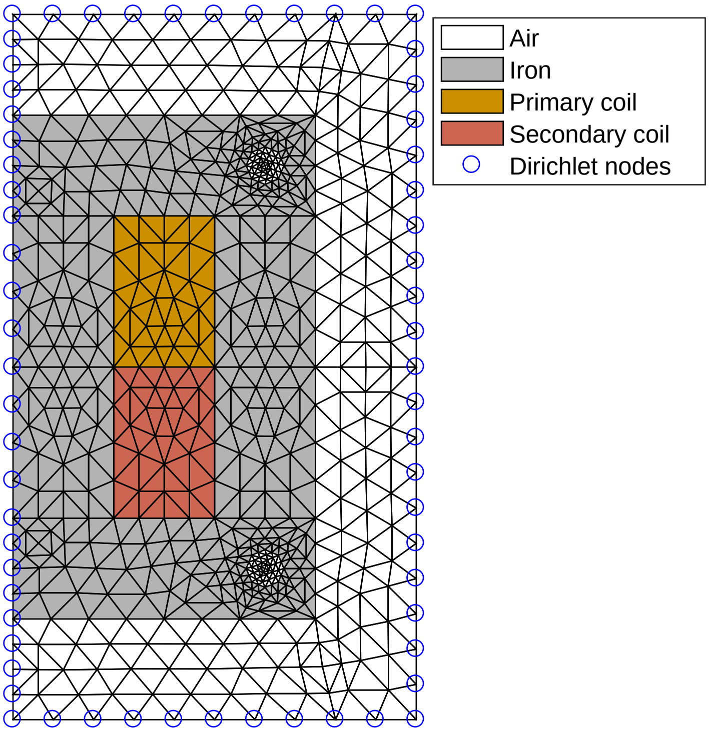


Figure 5.2: Finite element mesh used to solve the field equations. It is a cross section with anti-symmetry along the $x = 0$ axis.

voltage converter which has a DC-AC converter before the primary coil of the transformer and an AC-DC converter after the secondary coil of the transformer. These transformers operate on medium frequency and the input voltages have a square waveform. The benefit of the higher operation frequencies is that the transformer does not need as much core material than an ordinary transformer so it will be lighter. [1]

5.2 Derivation of full order model

The behaviour of this type of transformer can be modelled with reasonable accuracy by using a two-dimensional cross section of the device. Furthermore it is possible to exploit the symmetry of the device to reduce the domain into two-dimensional half-plane of the transformer. The domain area and FE mesh we use to solve the field problems is presented in Figure 5.2.

We will use two-dimensional vector potential formulation with eddy currents and nonlinear core material. A formulation that will allow hysteresis effect to be taken to account is used. The required mathematics is introduced in Chapter 2.

The full model is formulated so that the currents of the coils are given as an input and the solved vector potential is returned. After time discretization we also need to include the previous timestep solution and the timestep size as an input to be able to calculate the derivative using implicit Euler scheme. This is needed to calculate the effect of eddy currents. Timestep size is included as an input for flexibility since Matlab has solvers that can boost the simulation speed using variable step size.

5.3 Solver implementation

The finite element solver and model reduction algorithms were implemented with Matlab. A Matlab class was written that will hold the mesh data and be capable to solve full models and reduced models. Model data and reduced model data was stored in structures which contain similar fields so they can be used to achieve polymorphic behaviour. These structures can be saved to and loaded from files.

The solver class can solve a single timestep. The idea is that this same solving function can be used to generate the snapshots and later to solve the full and reduced order model steps. The time evolution is just a series of these single step solutions.

The solver class is also able to solve the magnetic vector potential from two input currents. This is used in the training phase of the reduced order model. The same component which solves the vector potential is used to solve the induced voltages in the circuit simulation phase. The primary goal is to create a hierarchical structure which could be abstracted so that the user just needs to reimplement the parts of the class which define the behaviour of the model if the class is later used on the reduction of some other model.

5.4 Applying POD and DEIM

There are at least two options to generate the snapshot matrices required in POD and DEIM: either we solve the system with time continuous input signals or we solve only individual timesteps with input current pairs. The second approach will result in a very large parameter set as we must provide the initial value of the timestep in order to calculate the solution of the timestep. Every possible combination of all possible nodal values of vector potential $\mathbf{a}$ is a possible initial value of the timestep. Most of these combinations are however not natural in a sense that they would not occur in a time simulation of the system. There could be situations where it may be beneficial to do reduction on timestep basis and the matter requires some further investigation.

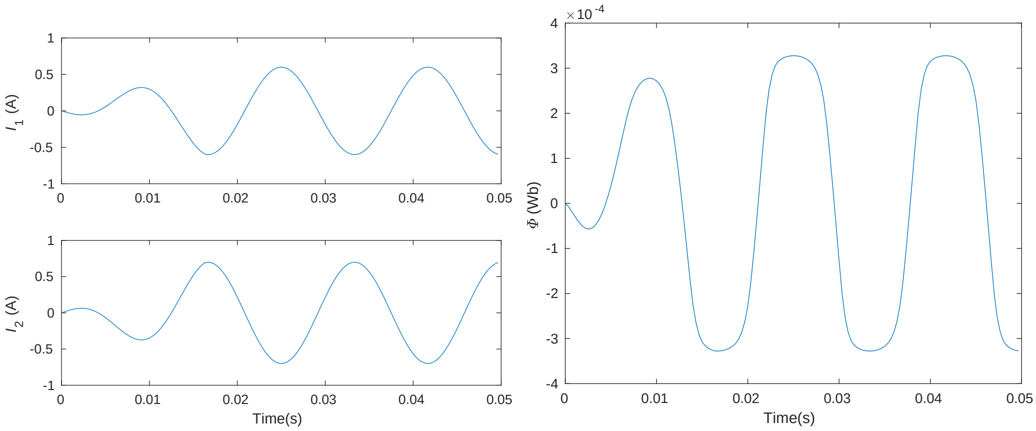
In this thesis we use time continuous input signals to generate the snapshots. This will result in a smaller parameter set and all generated snapshots are the result of the state evolving according to the governing equations. Furthermore we use sinusoidal input signals. The downside is that the reduced model may then lack some accuracy for input signals that are not similar to the training signals. In a voltage driven transformer this is less crucial as it is the integral of the voltage which matters the most. The integral function of a Riemann integrable function is always continuous so if the feeding voltage is at least Riemann integrable the effect it has on the flux is continuous and sinusoidal training signals will capture a lot of the relevant behaviour.

5.4.1 Parameter space and training set

Input parameters of the system are the amplitudes, frequencies and phases of the currents. It is assumed that the input and output current work with same base frequency. Therefore the parameter space in this case is $\Xi \subset \mathbb{R}^5$. With some prior knowledge of the system at hand we can restrict the training set.

Current amplitudes can be restricted between 0 A and 2 A. The maximum current depends on the system at hand. For this transformer 2 A corresponds to a high amplitude low frequency feeding voltage with practically short circuited load.

We can restrict the frequency range to the domain of interest. For example in SSTs the operating frequencies are in the kilohertz region. In SST engineered in [1] the operating frequency is 20 kHz. In our case the frequency is restricted between 40 Hz and 160 Hz. This allows us to investigate the behaviour of the transformer using the standard 50 Hz voltage and the multiples of the standard voltage frequency 100 Hz and 150 Hz.



(a) The input currents

(b) The core flux through primary coil

Figure 5.3: An example of the snapshot trajectory corresponding to a parameter configuration. The core flux is calculated through the primary coil. The solution is obtained over three periods. The first one is a ramp-up period to avoid large transients. Actual trajectory is the nodal values of the vector potential **a** from which the core flux is calculated.

The phases of the currents can be restricted so that the secondary current is behind in time of the primary current. In this case primary current phase is restricted to be between $-\pi$ rad and 0 rad and the secondary current phase was restricted to be between 0 rad and π rad. This is the case with resistive load. If reactive load is present one has to investigate the viable restrictions of the phase.

An example of a snapshot trajectory and the input currents is presented in Figure 5.3. The parameter configuration which was plotted is as follows: the amplitudes of the currents were $\hat{I}_1 = 0.6\,\mathrm{A}$, and $\hat{I}_2 = 0.7\,\mathrm{A}$, the phases of the currents were $\alpha_1 = -\frac{\pi}{2}$ rad, $\alpha_2 = \frac{\pi}{2}$ rad and the frequency $f = 60\,\mathrm{Hz}$. The actual data inserted into the snapshot matrix was the time evolution of the nodal values of the vector potential $\mathbf{a}$. The core flux of the primary coil shown in Figure 5.3b is calculated from the vector potential. There were three periods, 50 timesteps per period and both of the currents are ramped up during the first period to avoid large transients.

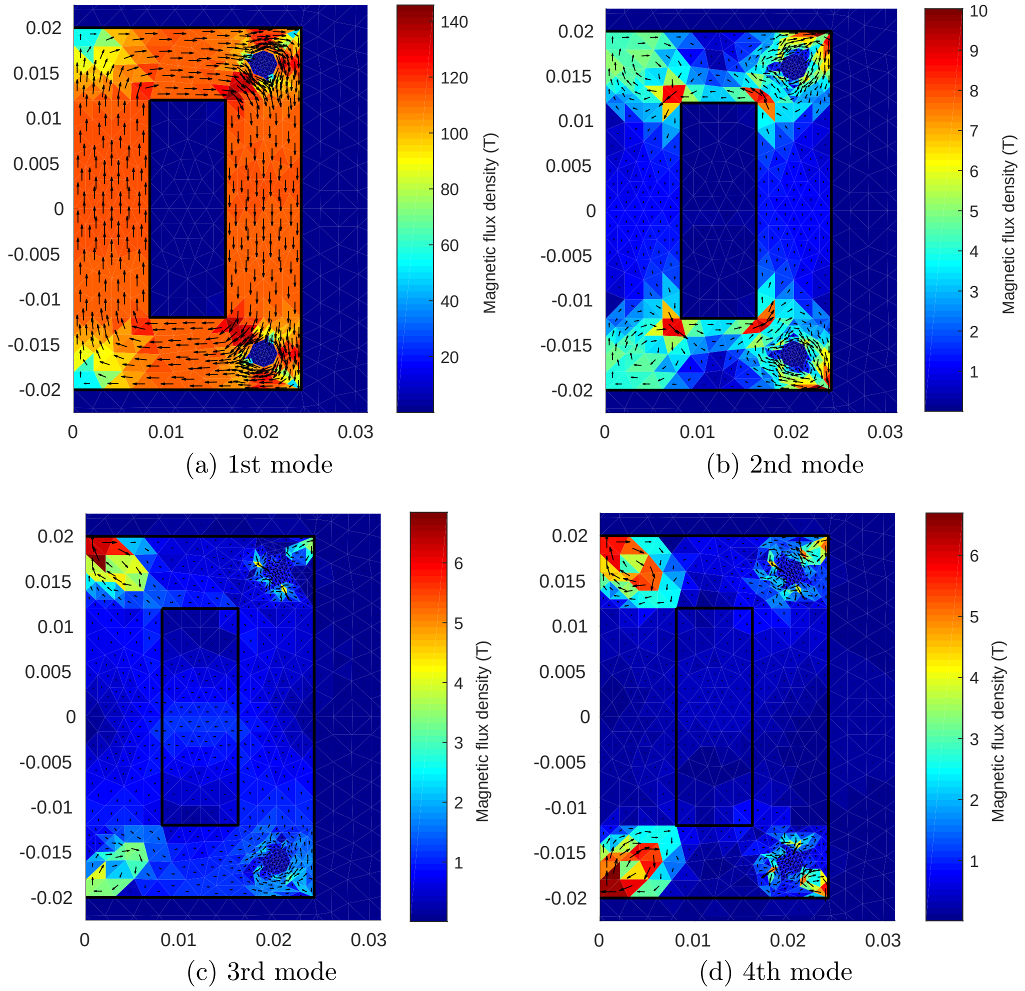
5.4.2 Training algorithm

The training of the system was done using Algorithm 1. Firstly a set of all combinations of the upper and lower boundary values of the parameter set was formed. Because the dimension of the parameter space is 5 the set will contain $2^5 = 32$ different combinations. These combinations were used to form a reduced order model which was then improved with adaptive iterations.

Some challenges rose from convergence issues with the NR method of the reduced order model. It seems that instead of smoothly growing error the error of the dynamic reduced order model explodes if it passes a certain treshold. After experimenting with different amounts of adaptive iterations it was concluded that in this case 20 adaptive iterations was enough to produce a stable reduced order model which had no convergence issues. This is highly dependent of the tolerance given to the POD which forms the basis for DEIM algorithm. It is very likely that the locations of the parameters which are used to calculate the snapshots also affect the stability [23, pp.1903-1918].

The convergence issues could be caused by a minor deviation from the state path which then amplifies at every timestep and after certain timestep the error grows so large that it goes out of the stability zone of the reduced model. Other explanation could be a bug in the implementation. This problem is present mostly when the dimension of the POD basis given for DEIM algorithm is too low. This follows for example if the tolerance given to the POD algorithm used to create the basis for the DEIM algorithm is too low. The problem was not detected if we used a smaller tolerance to calculate the DEIM approximation which resulted in a higher dimensional POD basis as an input to the DEIM algorithm. This resulted also in an increase in the amount of the DEIM nodes.

The convergence problem makes finding an optimum reduced order model difficult as one must make extra iterations to be certain that this issue is not present in real



(a) 1st mode

(b) 2nd mode

(c) 3rd mode

(d) 4th mode

Figure 5.4: The flux densities **B** calculated from the four POD modes for vector potential **A** of the system. All solutions of the reduced model are linear combinations of these modes.

calculations. The extra iterations take more time in the reducing phase and result in an increase of the reduced state space. This issue requires some more investigation to pinpoint the root cause for this behaviour.

5.5 Results of POD and DEIM

In the original model we have a mesh which has 593 nodes and 1120 elements. This translates to 593 degrees of freedom. This amount is very moderate as the mesh used here is kept rather coarse and the mesh is two-dimensional. If the device would be modelled in three dimensions with similarly coarse mesh the amount of nodes could be over 10^5.

The reduced order model was generated from the total of 32 snapshots from the boundaries of the parameter set and 20 iterations of the adaptive algorithm.

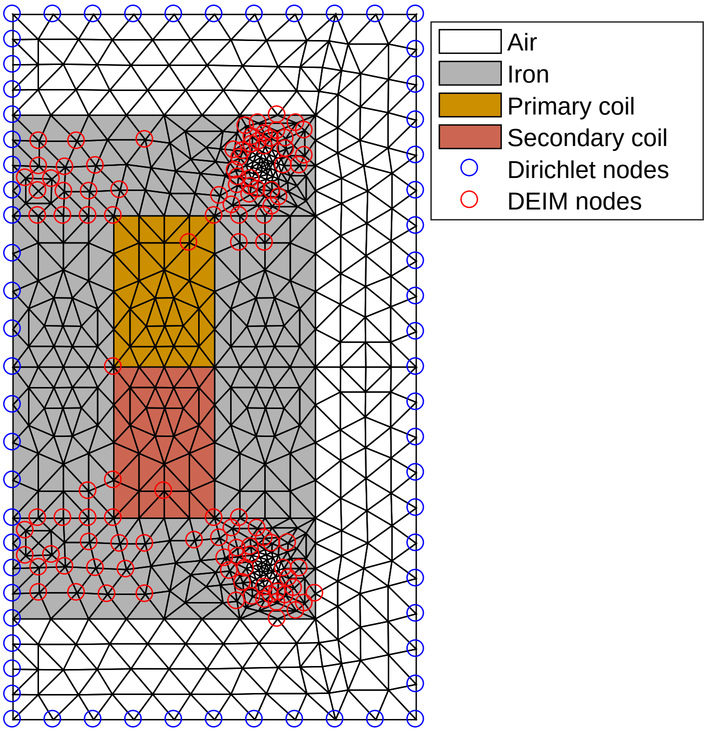


Figure 5.5: Nodes selected by DEIM algorithm are marked with red circles. They concentrate heavily on the regions where nonlinear material saturates first.

Therefore the total amount of snapshot trajectories was 52. The generation of the model took approximately 5 minutes.

The POD algorithm was used to generate the reduced basis of the state space from state snapshots. A tolerance of $\varepsilon_a = 10^{-3}$ was used. State space reduced from 593 degrees of freedom to 4 degrees of freedom. This is over 99 % reduction on the dimension of the state space. If the tolerance of the POD is set to $\varepsilon_a = 10^{-4}$ the dimension of the reduced space will be 16.

The POD basis can also be thought of as the principal modes of the system at hand. The four principal modes of the transformer are plotted in Figure 5.4. The flux density values in this figure seem unnaturally high. It must be kept in mind that the flux density is calculated from the vector potential which actually forms the POD basis. Therefore these flux densities are not present in the reduced system. The state of the reduced system is a linear combination of the POD basis of the vector potential $\mathbf{A}$. The figure merely illustrates how the basis behaves. The first mode is a dominant one and the other components which have a minor effect to the system focus on the corners where the material saturates the most.

For the nonlinear term the POD algorithm was used to generate the reduced basis from nonlinear term snapshots with a tolerance $\varepsilon_f = 10^{-3}$. Here the POD algorithm gave a base the dimension of which was 109. This POD basis was used to calculate a DEIM approximation for $\mathbf{f}_{\mathrm{nl}}$. If a tolerance $\varepsilon_f = 10^{-4}$ was used the dimension increased to 141.

The DEIM algorithm picked the 109 most significant nodes which are presented in Figure 5.5. As can be seen the nodes are mostly concentrated inside the core around corners and bolt holes. Few nodes are in the coils and none in the outside air region. The evaluation of the nonlinear term then requires that 389 elements and 270 nodes are processed. We can see a reduction of 65 % in the amount of elements and a reduction of 54 % in the amount of nodes to process.

If the tolerance is lowered the POD algorithm will result in a higher dimensional basis. It is yet to be investigated how dramatically changing the tolerance of either the state or the nonlinear term reduction affects the accuracy of the model. It was noted however that the amount of DEIM nodes plays a critical role in the convergence of NR method when the system was solved.

5.6 Benchmark system

A simple circuit presented in Figure 5.6 was used to calculate numerical results. Firstly the full order model was used to calculate full order results. A nonlinear system of equations which contained the full order finite element model was derived. The field equations which produce the induced voltages from the coil currents were solved using the developed finite element solver class.

The time-discretized equation system for the voltage controlled full order model is

$$\begin{cases} R_1 i_1 + \frac{1}{\Delta t}\mathrm{C}_1(\mathbf{x}-\mathbf{x}_{\mathrm{p}}) - U_{\mathrm{in}}(t) = 0 & (5.1) \\ \frac{1}{\Delta t}\mathrm{C}_2(\mathbf{x}-\mathbf{x}_{\mathrm{p}}) + (R_2 + R_{\mathrm{L}})i_2 = 0 & (5.2) \\ \frac{1}{\Delta t}\mathrm{M}(\mathbf{x}-\mathbf{x}_{\mathrm{p}}) + \mathbf{f}_{\mathrm{nl}}(\mathbf{x}) - \mathrm{D}_1 i_1 - \mathrm{D}_2 i_2 = 0\ , & (5.3) \end{cases}$$

where U_{in} is the input voltage, R_1, R_2 are the coil resistances, R_{L} is the load resistance, $\mathbf{x}_{\mathrm{p}}$ is the solution of the previous timestep, C_1, C_2 are matrices which are used to calculate the integral of the flux through the coils in the core and D_1, D_2 are matrices which divide the currents of the coils evenly to the coil area of the domain.

During the benchmark calculations the finite element solver class is used to calculate the induced voltages from the input currents. Therefore the system of equations needed to be solved is

$$\begin{cases} R_1 i_1 + u_1(i_1, i_2) - U_{\mathrm{in}}(t) = 0 & (5.4) \\ u_2(i_1, i_2) + (R_2 + R_{\mathrm{L}})i_2 = 0\ , & (5.5) \end{cases}$$

where u_1 and u_2 are the induced voltages over the primary coil and secondary coil

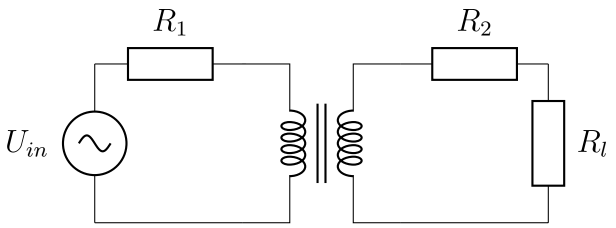


Figure 5.6: The circuit model used to benchmark the reduced order model of the transformer.

respectively given by the solver class. This equation system was solved at each timestep using Matlab's `fsolve` function. This made the model of the transformer work like a black box and made it possible to switch between the reduced order model and the full order model to compare results.

5.7 Coupling the solver to Simulink and Simscape

The solver class is used to solve the reduced order model during the time stepping of the simulation. A customized Matlab system block named `FEM_transformer` was derived to take the coil currents as inputs and return the induced voltages as outputs. The block holds the previous timestep solution as an internal state. This is used to compute the numerical derivative.

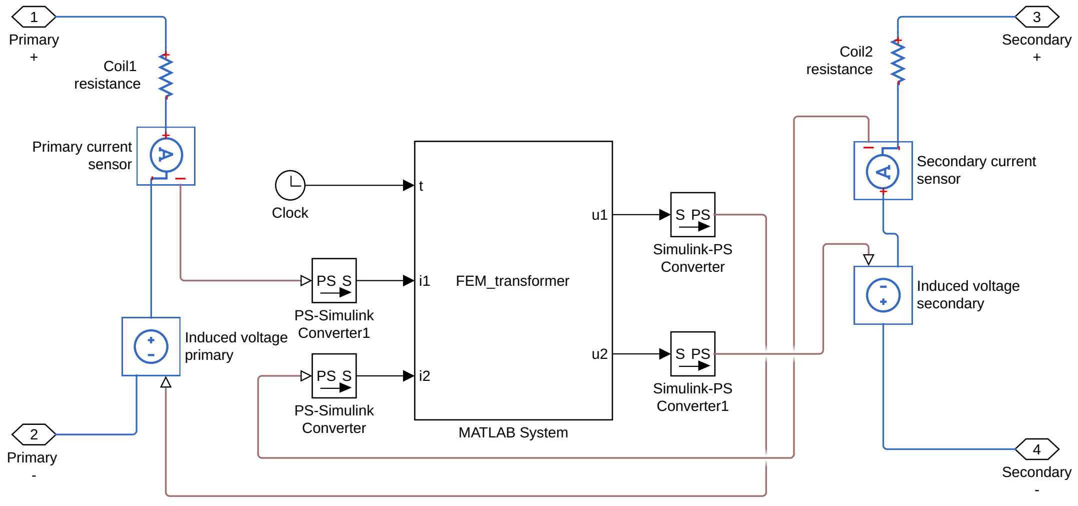


Figure 5.7: Simscape subsystem `FEM_connector_subsystem` which contains the transformer model and connections to physical circuit.

`FEM_transformer` block was integrated to a Simulink model as presented in Figure 5.7. The Simulink block was connected to the physical circuit model of Simscape by using ideal voltage sensors and ideal voltage meters. Coil resistances were also added to the subsystem containing the Simulink model. This creates a subsystem

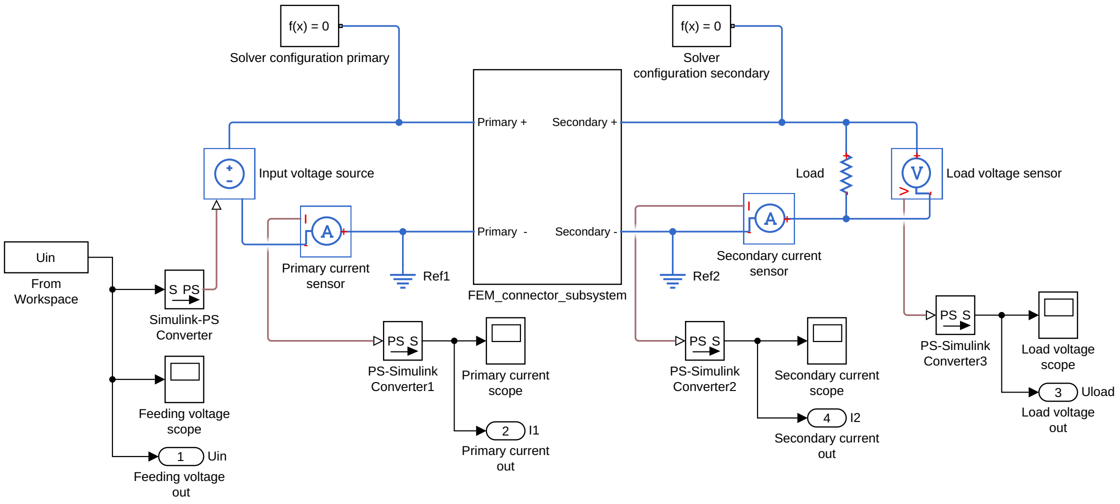


Figure 5.8: The Simscape circuit model of the benchmark circuit.

`FEM_connector_subsystem` which contains all details of the coil making it look like a black box from the other parts of the circuit. This subsystem has no inputs or outputs. It has physical terminals for primary and secondary coil.

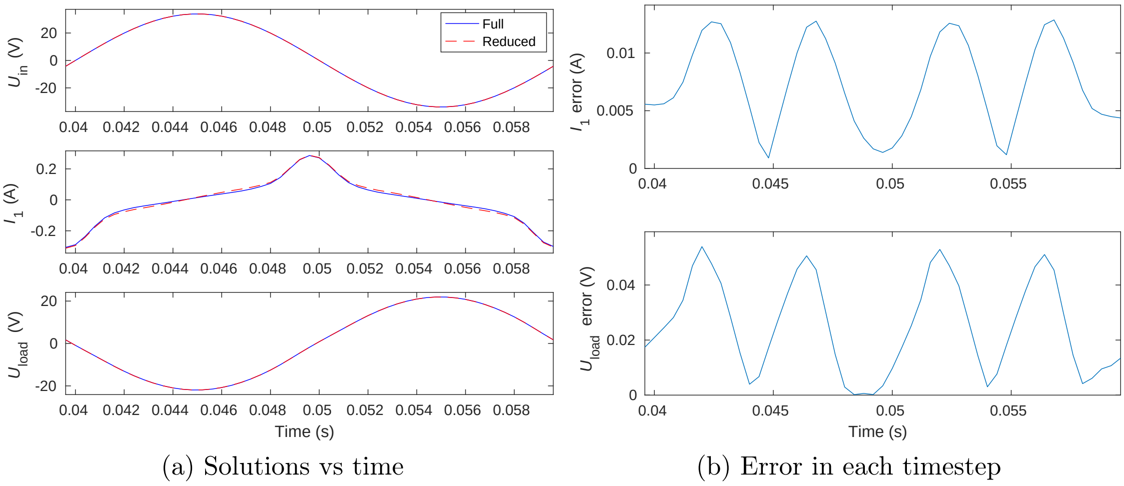
The benchmark circuit itself is modelled in Simscape and the model is presented in Figure 5.8. The primary side circuit and the secondary side circuit are customizable which makes investigating and experimenting with different components easier.

The solver in Simscape was configured so that it will use fixed timestep size. Fixed step size made it possible to know beforehand for which timesteps the result will be available and the hand made circuit model could then be solved in parallel with the Simscape model and the results could be compared timestepwise. The solver was set to use BW Euler method. Since we use BW Euler scheme in the formulation of the reduced model we must restrict the circuit simulator to use the same scheme. Otherwise instabilities may occur and the energy balance of the electromagnetic device is suboptimal [27].

5.8 Numerical results

The solutions of the reduced order model coupled with the circuit simulator were compared to the results of the full model solved with a standard nonlinear equation solver. Different kinds of feeding voltages and load resistances were used. The amplitude, frequency and waveform of the voltage were varied. The reduced order model produces rather accurately the same results as the full order model even with non-sinusoidal waveforms.

The largest errors were detected when the frequency is low and the transformer was running idle. In this case the magnetic flux density in the core is at its highest and nonlinear properties of the core are dominating. If frequency is increased the flux density inside the core will decrease. The higher the frequency of the input voltage is the more the eddy current losses affect the system. The flux density will also decrease if the load resistance is increased.



(a) Solutions vs time

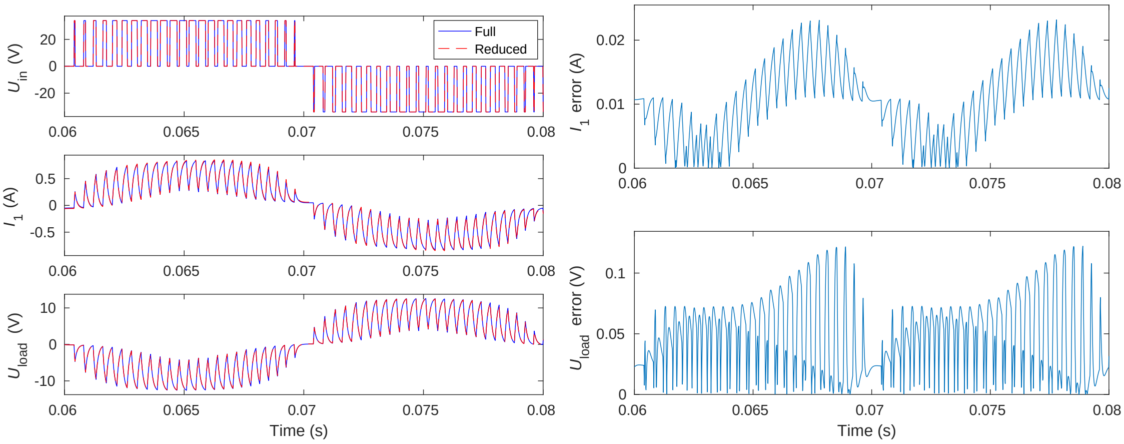
(b) Error in each timestep

Figure 5.9: The results of the last period of the reduced order model and the full model solved over three periods of sinusoidal waveform input voltage. U_{load} is the secondary voltage and I_1 the primary current. The transformer is idling.

In Figure 5.9 the transformer is fed with a sinusoidal voltage when the transformer is running idle. The figure contains only the last period of the simulation. In total three periods were simulated with the first period being a ramp up period to reduce transients. The voltage amplitude is $\sqrt{2} \cdot 24\,\text{V}$ and the frequency is 50Hz. One period was split into 50 timesteps. The dashed line which shows the reduced order solution is almost exactly overlapping the full order solution in the case of the load voltage. In the primary current I_1 one can see slight deviations.

The average relative error of I_1 over the final period is 15 %. At the peak value of the current the error is 4.8 % which is an acceptable value. The error of the voltage U_{load} is smaller than the error of the current. The average error of U_{load} is 0.4 % and the highest relative error is 2.5 %. It must be noted that the errors can be made smaller if we increase the dimension of the reduced order model.

An example of a harsh signal is a pulse width modulated (PWM) signal. In Figure 5.10 we have fed the transformer using a PWM signal which has a carrier wave with the frequency of 50 times the frequency of the base signal. Input waveform is presented in Figure 5.10a. The base signal which was modulated was a sinusoidal voltage with an amplitude of $\sqrt{2} \cdot 24\,\text{V}$ and the frequency of 50 Hz. The frequency of the carrier wave was then 2.5 kHz. This time the value of the load resistance R_{L} was 10 Ω. This calculation requires a small timestep size to avoid convergence issues. One period was split into 2000 steps. The Figure 5.10 contains only the



(a) Solutions vs time

(b) Error in each timestep

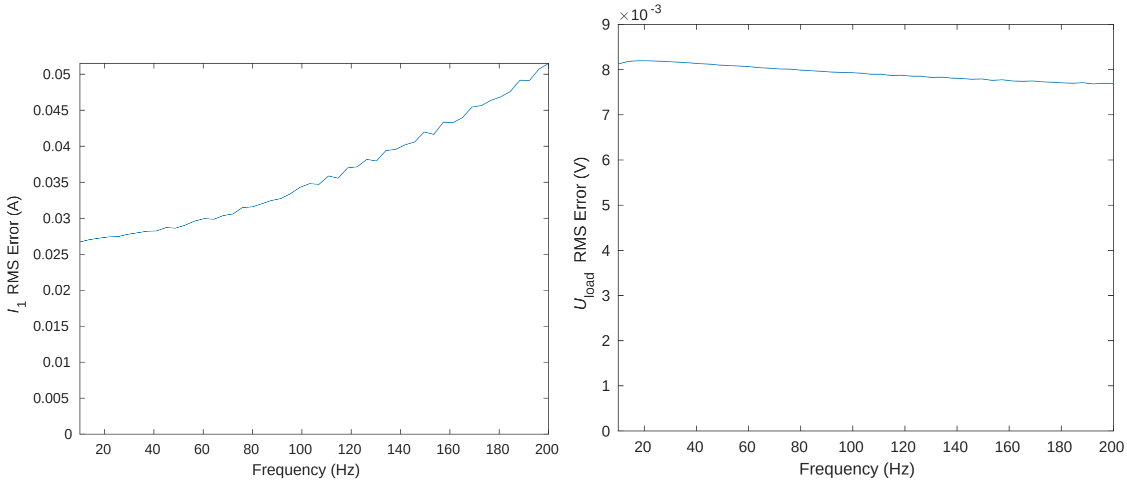
Figure 5.10: The results of the last period of the reduced order model and the full model solved over five periods of PWM input voltage. U_{load} is the secondary voltage, U_{in} the primary voltage and I_1 the primary current. Here the load resistance $R_{\mathrm{L}} = 10\,\Omega$.

last period of the signal. A total of five periods were calculated. During the first period the voltage was ramped up to sinusoidal and during the second period it was ramped from sinusoidal waveform to PWM waveform. This was done to avoid large transients.

From Figure 5.10b we see the error during each timestep. The average of the relative error of I_1 here is 7 %. There are time instances where the relative error is substantially high. With this waveform however even a slightest phase difference can cause a large error to a number of timesteps. The average error of the load voltage U_{load} is 3.5 %. Although the error for individual time instances is high it can be said that the reduced order model gives in general rather accurate results also in this case.

In Figure 5.11 the frequency is swept from 30 Hz to 170 Hz to show the behaviour of the error with respect to frequency. The waveform of the input voltage is sinusoidal and the amplitude of the voltage was chosen so that the magnitude of the flux density stays constant. To compensate the increasing frequency a voltage of $\frac{f}{50\mathrm{Hz}} \cdot \sqrt{2} \cdot 24\,\mathrm{V}$ was chosen where f is the frequency of the voltage. This keeps the flux density at the same level as it is at 50 Hz when the results of Figure 5.9 were calculated.

As the model was generated using a frequency range from 40 Hz to 160 Hz here we can also see how the reduced model behaves in the edge and outside the training domain. At least up to 500 Hz the error stayed moderate and no convergence issues were detected. It was not thoroughly investigated how far the reduced model is valid



(a) Error of primary current I_1

(b) Error of output voltage U_{load}

Figure 5.11: RMS error of output voltage U_{load} and primary current I_1 calculated for different values of frequency using sinusoidal voltage. The error stays feasible for the whole region from 50Hz to 150Hz but shows an increasing trend.

or how reliable the convergence of the reduced model is in the regions outside the training set.

Based on the results it can be said that this reduced order model instance gives rather accurate results compared to the full model. The results in this thesis are not compared to measurements from the actual transformer. Here only the difference between full order model and reduced order model is emphasized and the full order model is assumed to be accurate.

It is always dangerous to compare execution times without factoring out all interfering details such as memory allocations. In this case if a full model is connected to Simscape it takes 186 seconds to compute a time simulation of three periods with 150 timesteps altogether. With reduced order model connected the same computation takes 102 seconds. We are seeing a reduction of 44 % in execution time which is not as much as we would expect from the reduction statistics of the model. This is mainly because of the overhead and the lack of performance in the connection between the reduced order model and the circuit simulator.

Another test was done so that only the time which it took to solve the induced voltages from the input currents was captured. The mean value of the time it takes to solve a single step like this using the full order model was 12.6 ms. The same time by using the reduced order model was 3.1 ms. Here the reduction of the computational time is 75 %. Naturally the second benchmark test had less overhead involved. Based on this it can argued that a significant speed boost can be obtained using model reduction if it is utilized properly.

6. CONCLUSIONS

In this thesis model reduction methods were investigated and two specific techniques, POD and DEIM, were inspected with more depth. The reduced model implemented with Matlab was connected to a circuit simulator Simscape and results were computed using the coupled system. The computational complexity, execution times and numerical results were compared between the full order model and the reduced order model. The results are promising in a sense that the amount of work decreases greatly, the execution time decreases and the numerical results remain feasibly accurate.

6.1 Discussion of the results

From the results presented in Sections 5.5 and 5.8 we can conclude that the accuracy of the computed reduced model instance is sufficient. It is always possible to improve the accuracy by tightening the tolerance of the POD method. Furthermore we can conclude that the workload of the solving phase has reduced because the dimension of the state space reduced over 99 %. The workload of the evaluation of the nonlinear term has also reduced since the amount of nodes to loop through reduced by 54 % and the amount of elements to loop through reduced by 65 %. This indicates that the total workload is greatly reduced.

In this case the circuit simulator coupling has very poor performance which is caused mainly by Matlab's Simulink block being executed as interpreted code. This adds much overhead to the timestepping phase and makes the reduction seem worse. The reduction in computation time of the simulation was only 44 %. When the overhead of the circuit simulator coupling was removed to some extend we saw a reduction of 75 % in the computation time. Also it must be noted that solving the original system in this case is not a heavy task to begin with.

No direct connection between the accuracy and, for example, the dimension of the reduced basis were investigated. It is clear that if the dimension is higher the reduced order model will be more accurate and if dimension is lower the accuracy is worse. It would be beneficial to study the errors and to be able to derive some kind of error bounds for the method. This would allow us to answer questions like what is the optimal reduced model and how one can achieve the optimal reduced model with minimal work.

The reduced order model instance presented in Section 5.5 was trained using sinusoidal input currents for the coils. It could be possible to use other waveforms as well but this was left outside the scope of the thesis. It could also be possible for example in this case to create a specific reduced order model for a specific input voltage by using a Fourier transform of the input voltage to investigate which frequencies would be important to include into the snapshot set of the reduced order model to obtain accurate results.

6.2 Feasibility of model reduction

FEM systems are all the time coupled with circuit models but these systems usually are tailored for specific cases. They are modelled to the level of differential equation systems and if the circuit is changed the equation systems must be changed accordingly. If FEM systems are coupled directly to a circuit simulator software the performance takes a hit and the solving times of such systems can be high. With the techniques presented in this thesis it is possible to lower the time of calculation greatly while still maintaining the flexibility benefits of using a circuit simulator software.

The results show that theoretically the amount of work to solve the system is reduced heavily. The results also demonstrate that the accuracy of the reduced order model remains feasible. Based on this we can conclude that this technique is promising and likely will result in a lot better reduction rates if the implementation of the solver and the connection between the reduced order model and the circuit simulator are optimized. Also three-dimensional problems usually result in equation systems which are orders of magnitude larger than the one analysed in this thesis. Therefore these reduction techniques are expected to provide better results if they are applied to three-dimensional problems.

6.3 Issues with convergence

A phenomenon was seen when the required tolerance of POD was investigated that the Newton-Raphson algorithm seized to converge if the tolerance was too high and the dimension of the reduced basis for DEIM too low. This results in lesser amount of DEIM nodes. It seems that the snapshot set also affects the convergence. If parameters are picked from bad spots of the training set the POD basis will not be as optimal as it would be if more optimal spots of the parameter set are picked.

The problems can of course be caused by a bug in the code or then it is a phenomenon which would need some further study. This could be caused by the nonlinear behaviour so that if the error grows slightly the state will then wander off the trajectory more and more during each timestep and finally explode. It could also be that the convergence problems are caused by some numerical instabilities.

Another convergence issue was seen when Simscape simulations were ran. The iterative method used by Simscape's nonlinear solver failed to find solutions on some timesteps. This issue was mostly fixed by decreasing the step size. However the same step size which caused trouble during Simscape simulations was working fine with the hand made time-stepping solver where `fsolve` was used. This indicates that the circuit simulator coupling introduces some numerical inaccuracy and care must be taken to obtain reliable results.

6.4 Further study

It would be crucial to know what causes the NR method to stop converging during the solving of the reduced model and how to prevent it from happening during real calculations. The phenomenon was not detected in any of the models that were generated by adaptive algorithms with a relatively large amount of adaptive iterations. The phenomenon was also not detected if reduction was applied to a linear version of the transformer system. Hence it is likely that this issue is caused by DEIM approximation. Lowering the tolerance i.e. increasing the dimension of the POD basis used in DEIM algorithm seems to mitigate this problem.

The dependence of the accuracy of reduced model and the techniques used in the reduction phase could be investigated further. As mentioned before the error indicators and error bounds for non-affine PDEs which do not depend on the original dimension of the model seems still to be a topic which needs further research.

The formulations of the systems were done in a way that hysteresis can be taken into account which is a main point how this thesis differs from the existing research. This improves the generality of the technique. Based on this thesis it would be possible to investigate the behaviour of reduction methods with hysteretic materials.

Based on the work on this thesis it would be possible to develop a model reduction module to some FEM software such as Elmer [28]. Another module would then be added to some circuit simulator software. This could be a new tool for product development of electromagnetic energy converters. It would make investigating the external behaviour of the designed device more straightforward and less time consuming. Another positive effect is that the circuit model is automatically generated from the full order model. The same reduction technique described in this thesis could also be used to homogenize for example models of materials which have detailed microstructures.

REFERENCES

[1] M. Leibl, G. Ortiz, and J. W. Kolar, “Design and Experimental Analysis of a Medium-Frequency Transformer for Solid-State Transformer Applications”, *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 5, no. 1, pp. 110–123, 2017.

[2] C. Gu, *Model Order Reduction of Nonlinear Dynamical Systems*, 2012, Available (accessed on 29.6.2017) `http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-217.html`.

[3] P. Benner, S. Gugercin, and K. Willcox, “A survey of model reduction methods for parametric systems”, *MPI Magdeburg Preprints*, vol. MPIMD, no. 13-14, pp. 1–36, 2013.

[4] V. Sharma and G. Daga, “Comparison between Model Order Reductions Techniques”, *International Journal of Electronics and Computer S cience Engineering*, vol. 1, no. 3, pp. 906–915, 2012.

[5] T. Henneron and S. Clénet, “Model-order reduction of multiple-input nonlinear systems based on POD and DEI methods”, *IEEE Transactions on Magnetics*, vol. 51, no. 3, 2015.

[6] Y. Paquay, O. Bruls, and C. Geuzaine, “Nonlinear Interpolation on Manifold of Reduced-Order Models in Magnetodynamic Problems”, *IEEE Transactions on Magnetics*, vol. 52, no. 3, pp. 18–21, 2016.

[7] M. F. Far, A. Belahcen, P. Rasilo, S. Clenet, and A. Pierquin, “Model order reduction of electrical machines with multiple inputs”, *IEEE Transactions on Industry Applications*, vol. PP, no. 99, pp. 1–1, 2017.

[8] M. Mordhorst, T. Strecker, D. Wirtz, T. Heidlauf, and O. Röhrle, “POD-DEIM reduction of computational EMG models”, *Journal of Computational Science*, vol. 19, pp. 1–11, 2017.

[9] W. Aquino, “An object-oriented framework for reduced-order models using proper orthogonal decomposition (POD)”, *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 41-44, pp. 4375–4390, 2007.

[10] E. Monteiro, J. Yvonnet, and Q. C. He, “Computational homogenization for nonlinear conduction in heterogeneous materials using model reduction”, *Computational Materials Science*, vol. 42, no. 4, pp. 704–712, 2008.

[11] J. R. Reitz, F. J. Milford, and R. W. Christy, *Foundations of electromagnetic theory. 4th ed.* Addison-Wesley, 1992.

[12] J. Luomi, *Finite element methods for electrical machines, Lecture notes*, 1993.

[13] P. Rasilo, E. Dlala, K. Fonteyn, J. Pippuri, A. Belahcen, and A. Arkkio, “Model of laminated ferromagnetic cores for loss prediction in electrical machines”, *IET Electric Power Applications*, vol. 5, no. 7, p. 580, 2011.

[14] G. Rozza, D. B. P. Huynh, and A. T. Patera, “Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: Application to transport and continuum mechanics”, *Archives of Computational Methods in Engineering*, vol. 15, no. 3, pp. 229–275, 2008.

[15] B. Salimbahrami and B. Lohmann, “Krylov Subspace Methods in Linear Model Order Reduction: Introduction and Invariance Properties”, *Researchgate*, 2002.

[16] M. Heinkenschloss, T. Reis, and A. C. Antoulas, “Balanced truncation model reduction for systems with inhomogeneous initial conditions”, *Automatica*, vol. 47, no. 3, pp. 559–564, 2011.

[17] D. M. Vasilyev, *Theoretical and practical aspects of linear and nonlinear model order reduction techniques*, 2008, Available (accessed on 29.6.2017) `http://www.rle.mit.edu/cpg/documents/vasilyev.pdf`.

[18] A. Chatterjee, “An introduction to the proper orthogonal decomposition”, *Current Science*, 2000.

[19] T. Henneron and S. Clénet, “Model Order Reduction of Non-Linear Magnetostatic Problems Based on POD and DEI Methods”, *The European Physical Journal Applied Physics*, vol. 50, no. 2, pp. 1–7, 2013.

[20] A. Paul-Dubois-Taine and D. Amsallem, “An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models”, *International Journal for Numerical Methods in Engineering*, vol. 102, no. March, pp. 1261–1292, 2015.

[21] S. Chaturantabut and D. C. Sorensen, “Nonlinear Model Reduction via Discrete Empirical Interpolation”, *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2737–2764, 2010.

[22] *Solving Systems of Linear Equations*, Available (accessed on 5.6.2017) `http://www.mcs.anl.gov/~anitescu/CLASSES/2012/LECTURES/S310-2012-lect3.pdf`, 2012.

[23] M. Rathinam and L. R. Petzold, “A New Look at Proper Orthogonal Decomposition”, *SIAM Journal on Numerical Analysis*, vol. 41, no. 5, pp. 1893–1925, 2003.

[24] O. Wing, *Classical circuit theory*. Springer US, 2008, p. 296.

[25] *Mathworks Simulink product page*, Available (accessed on 28.6.2017) `https://se.mathworks.com/products/simulink.html`.

[26] *Mathworks Simscape product page*, Available (accessed on 28.6.2017) `https://se.mathworks.com/products/simscape.html`.

[27] P. Rasilo, L. Perkkiö, A. Hannukainen, B. Silwal, T. Eirola, and A. Arkkio, "Instantaneous Power Balance in Finite-Element Simulation of Electrical Machines", *IEEE Transactions on Magnetics*, vol. 50, no. 5, 2014.

[28] *Elmer web page*, Available (accessed on 5.6.2017) `https://www.csc.fi/web/elmer`.