Interpolatory Nonlinear Model Order Reduction and its Application in Circuit Simulation

by

Seyed-Ali Nouri

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

Master of Applied Science

in

Electrical and Computer Engineering

Carleton University
Ottawa, Ontario, Canada

Abstract

This thesis presents a new approach to construct parametrized reduced-order models for nonlinear circuits. The reduced model is obtained such that it matches the variations in the DC operating point of the original full circuit in response to variations in several of its key design parameters. The new approach leverages, through the use of circuit moments with respect to the design parameters, the discrete empirical interpolation approach developed for model reduction in other domains and enables its efficient application to the problem of DC operating point in nonlinear circuits. Utilizing the idea of rooted trees, the proposed approach constructs orthogonal bases that are used in projecting the full equations of the large original nonlinear circuit onto a reduced system of nonlinear equations in a space with a much smaller dimension. The variations in the DC operating point of the full circuit are then obtained by solving the reduced system of equations, yielding significant computational savings. Numerical examples are presented to demonstrate the efficiency and accuracy of the reduced model in predicting the change in the DC operating point of the original circuit.

Acknowledgments

I want to express my deepest gratitude to my supervisor Professor Michel Nakhla. Throughout my undergraduate studies, he provided me internship opportunities to introduce me to the world of CAD. He encouraged me to pursue my Master's degree, which provided me with the most fulfilling years of engineering studies. His passion for CAD and learning was an inspiration for me.

I would also like to express my gratitude to my co-supervisor, Professor Emad Gad. His attention to detail, passion for CAD, and drive were sources of inspiration throughout my masters. From our routine Skype calls to endless email chains, it was a once-in-a-lifetime adventure to adjust to the challenges of remote working during COVID-19.

I want to thank Professor Ram Achar for guiding my fourth-year undergraduate project. The implementation of Data-Driven Macromodeling (Vector Fitting) on GPUs was a source of inspiration to pursue CAD and software development further.

Additionally, I would like to thank Ye Tao. It was a pleasure being a Teaching Assistant with him for the CAD course. As a senior year Ph.D. student in CAD, he has always been readily available for some friendly deliberations that made my graduate life more enjoyable. I treasure our friendship.

I am thankful to the staff at the Department of Electronics at Carleton University for having been so helpful, supportive, and resourceful.

I want to thank my father, Dr. Behzad Nouri, for being a role model throughout

my life. His love for Electrical Engineering, Mathematics, and CAD has always been an inspiration for me. He has stood by me as an endless source of knowledge throughout my high school, undergraduate, and now post-graduate journey. I aspire to be as knowledgeable, passionate, and patient as him as I continue my engineering career. I would also like to thank my mother for her endless support. She has always been an invaluable source of encouragement and positivity. I am grateful for my fun and happy brother Ryan, who is always a steady source of happiness and smiles.

I want to thank Laura Machado for her love and support. She has always encouraged me to follow my passion and has shown incredible patience throughout my masters.

Table of Contents

List of Tables

List of Figures

List of Acronyms

Acronyms	Definition
CAD	Computer Aided Design
CPU	Central Processing Unit
DAE	Differential-Algebraic Equation
EIG	Eigenvalue (diagonal) Decomposition
EM	Electro-Magnetic
FD	Frequency Domain
IC	Integrated Circuit
I/O	Input-Output
KCL	Kirchoff's Current Law
KVL	Kirchoff's Voltage Law
DC-OP	DC Operating Point
NL	Nonlinear
N-R	Newton-Raphson
MOR	Model-Order Reduction
ROM	Reduced-Order Model
PMOR	Parameterized Model-Order Reduction
PDN	Power Distribution Network
LNA	Low Noise Amplifier
DEIM	Discrete Empirical Interpolation Method
SVD	Singular Value Decomposition
MPE	Missing Point Estimation
EIM	Empirical Interpolation Method
TPWL	Trajectory Piecewise Linear
RFIC	Radio Frequency Integrated Circuits
MIP	Moment-based Interpolation Projection

List of Symbols

Symbols	Definition
$\mathbb{N}$	The field of natural numbers
$\mathbb{R}$	The field of real numbers
$\mathbb{C}$	The field of complex numbers, *e.g.:* s-plane
$\mathbb{R}^{n\times m}$	The set of real matrices of size n $\times$ m
$\mathbb{C}^{n\times m}$	The set of complex matrices of size n $\times$ m
$\mathcal{C}^n$	n differentiable (n-smooth)
$\overline{a}$ or a^*	The complex conjugate of a complex number $a \in \mathbb{C}$
$\mathbf{A}^{\mathrm{H}}$	The complex conjugate of complex matrix $\mathbf{A} = [a_{ij}]$ defined as: $\overline{\mathbf{A}}^{\mathsf{T}} = [\overline{a}_{ji}]$
$\mathbf{A}^{\mathsf{T}}$	The transpose of matrix $\mathbf{A}$
span	The subspace spanned (or generated) by a set of vector
colsp	Column space of a matrix

Chapter 1

Introduction

1.1 Background and Motivation

The problem of determining the quiescent (DC) operating point constitutes an indispensable component in the Computer-Aided Design (CAD) tools of integrated circuits. Typically, this problem requires solving a system of nonlinear equations representing the circuit where the difficulty has been traditionally in guaranteeing the convergence to the DC operating point [4].

Another frequently demanded task in circuit design is to perform the so-called "parameter sweep" where the DC operating point is reevaluated at numerous values for several design parameters. Often, the designer considers these parameters as key to the circuit performance. This demand becomes challenging when the circuit is large as it requires repeatedly solving a large system of nonlinear equations at the various values of the design parameters. The cumulative computational cost becomes prohibitively large as the number of parameters increases.

The general concept of Model-Order Reduction (MOR) emerged as a response to the need to handle the increasing complexity of the circuits and the mathematical problems that they spawn in the course of their simulations [5–7]. MOR addresses the increasing complexity by *projecting* the full circuit model onto a reduced space.

The reduced system shares the same essential features as the original (large) system, allowing it to be simulated in its stead, thereby alleviating its high computational cost. The idea of Parameterized MOR (PMOR) was born out of MOR to mimic the original system's behaviour with respect to the key design parameters identified by the designer.

Projection-based PMOR methodologies may be grouped by their approach to constructing the orthogonal basis used in the projection. Moment matching MOR is one class of projection methods that has been widely adopted in linear circuits mainly due to its efficiency [8]. Another class of projection-based methods that is frequently used when the computation of the moments basis is not feasible, such as in the case of nonlinear systems, is the proper orthogonal decomposition (POD). POD-based projection is carried out by simulating the full system model to generate the so-called snapshots used to construct the projection basis.

Moment matching PMOR has been utilized successfully in linear circuits for purposes analogous to the DC operating point (e.g., computing frequency response) [8–12]. Part of the popularity of moment matching in linear circuits is the low computational cost of constructing the moments projection basis [13, 14]. For nonlinear systems, attempts to apply moment matching PMOR have been made in [15, 16] through approximation of the nonlinear system by a collection of parametric linear models obtained via linearizing the nonlinearity along training trajectories created from multiple simulations of the unreduced nonlinear system.

On the other hand, POD-based PMOR can be viewed as a continuation of this idea but without engaging moment matching [17, 18]; it uses the multiple simulations (snapshots) of the full nonlinear model to create the projections basis, thereby avoiding the need to collect a possibly large ensemble of linear systems.

Obviously, using moment matching PMOR directly on the nonlinear systems still represents a gap in the literature that, if properly filled, will help in circumventing

the need for those snapshots. The goal of this thesis is to fill this gap and use the proposed approach to address the problem of DC operating point.

1.2 Objective and Contribution

The proposed approach enables using moment matching directly on the nonlinear system that models the full circuit to construct a reduced-order model that matches the original circuit's behaviour with regards to a selected set of design parameters. The key idea in the proposed approach utilizes the notion of rooted trees to create the projection basis. Using the constructed projection basis, and to perform the projection efficiently, the proposed approach adopts the interpolatory projection methodology proposed in [19, 20] which is known as discrete empirical interpolation method or (DEIM). The new approach is dubbed as moment-based interpolation projection (MIP) for its reliance on the moments of the system, as opposed to empirical snapshots created through simulation, to construct the interpolator operator.

1.3 Organization of Thesis

This thesis is organized as follows:

- Chapter 2 introduces the concept of Modified Nodal Analysis (MNA) to mathematically model nonlinear electrical circuits and reviews the strategies to "stamp" common circuit components in the MNA matrices.
- Chapter 3 presents a common strategy to solve for the DC operating point of nonlinear circuits. It introduces the application of Newton-Raphson (N-R) as a method to iteratively find the DC operating point of a nonlinear circuit based on an initial estimation of the solution. It also discusses common strategies to help guarantee the solution of N-R, such as "Source Ramping".

- Chapter 4 reviews the application of MOR techniques in the context of circuit simulation. It highlights the common difficulties and inefficiencies experienced when apply existing MOR techniques to solve nonlinear circuits. It then presents the Discrete Empirical Interpolation Method (DEIM) to address the shortcomings of existing MOR techniques.

- Chapter 5 discusses the application of the theory introduced in Chapters 3 and 4 to evaluate the DC solution of nonlinear circuits.

- Chapter 6 presents examples to demonstrate the accuracy and efficiency of the proposed method.

- Chapter 7 provides closing remarks and discusses possible future work based on the findings of this thesis.

Chapter 2

Formulation of Nonlinear Systems

2.1 Nonlinear Dynamical Systems

The main paradigm for linear systems is superposition, which is defined in terms of "additivity" and "homogeneity" with respect to inputs and initial conditions. Any system that does not satisfy the superposition property is considered nonlinear (NL). General nonlinear dynamical systems, illustrated in Figure 2.1, can be characterized by a finite number of nonlinear first-order differential equations, often along with a set of algebraic equations, as given in (2.1) [21–23],

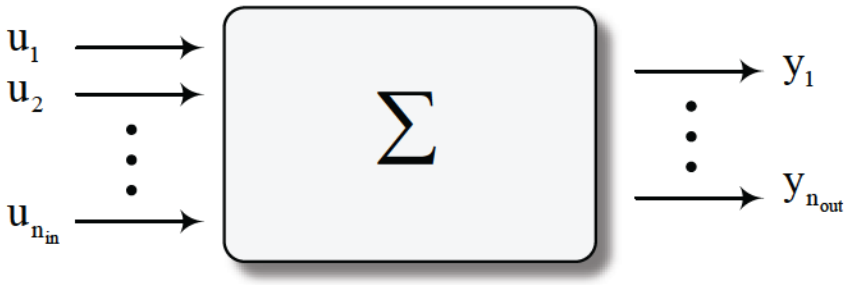


Figure 2.1: Finite-dimensional dynamical system

$$\boldsymbol{\Sigma}: \begin{cases} \mathbf{F}(\, t,\, \dfrac{d}{dt}\mathbf{x}(t),\, \mathbf{x}(t),\, \mathbf{u}(t)\,) = \mathbf{0} & \text{(state equations)} \quad (2.1a) \\ \mathbf{y}(t) = \mathbf{h}(\, t,\, \mathbf{x}(t),\, \mathbf{u}(t)\,) & \text{(output equations)} \quad (2.1b) \end{cases}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is a vector of system variables, $\mathbf{u}(t) \in \mathbb{R}^{n_{in}}$ is a vector of input sources, $\mathbf{F} \in \mathbb{R}^n$ is a vector-valued nonlinear function, $\mathbf{y}(t) \in \mathbb{R}^{n_{out}}$ is the vector of the responses at the outputs, and n is the order of system. The mathematical model in (2.2a) falls in the category of nonlinear Differential-Algebraic Equation (DAE) systems.

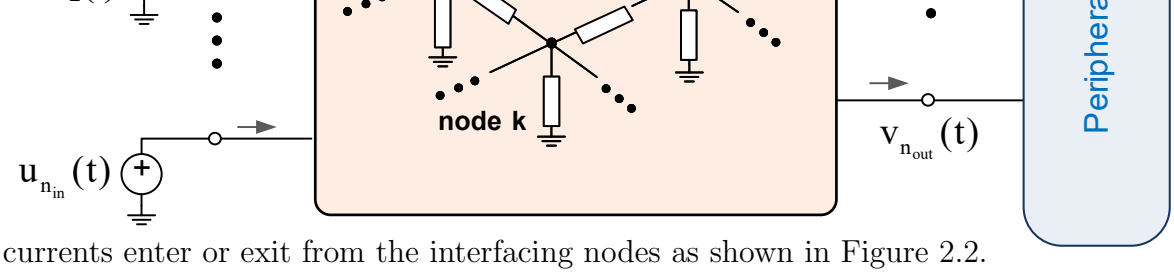
A system is defined as time-invariant if it only consists of time-invariant components for which the direct and explicit dependency of equations on the time variable can be dismissed as

$$\boldsymbol{\Sigma}: \begin{cases} \mathbf{F}(\, \dfrac{d}{dt}\mathbf{x}(t),\, \mathbf{x}(t),\, \mathbf{u}(t)\,) = \mathbf{0} & (2.2a) \\ \mathbf{y}(t) = \mathbf{h}(\, \mathbf{x}(t),\, \mathbf{u}(t)\,) & (2.2b) \end{cases}$$

2.1.1 Nonlinear Circuits as Nonlinear Dynamical Systems

A circuit is defined as nonlinear if it consists of at least one nonlinear component (not counting the independent- voltage and current sources). A circuit element is defined as nonlinear when the "constitutive relationship" between its voltage (established across) and its current (flowing through) is a nonlinear function $i_e = f(v_e)$. A diode ($I_d = I_s(e^{\frac{V_d}{\eta V_T}} - 1)$) is common example of a nonlinear component.

Electrical circuits are examples of dynamical systems. It was emphasized by Gear (1968) in [24] that circuits can be well characterized using a system of first-order differential-algebraic equations (2.2). A complex design is generally comprised of several circuit elements that are connected together at the nodes. A circuit interacts with the rest of the design (peripherals) through input/output terminals where the



currents enter or exit from the interfacing nodes as shown in Figure 2.2.

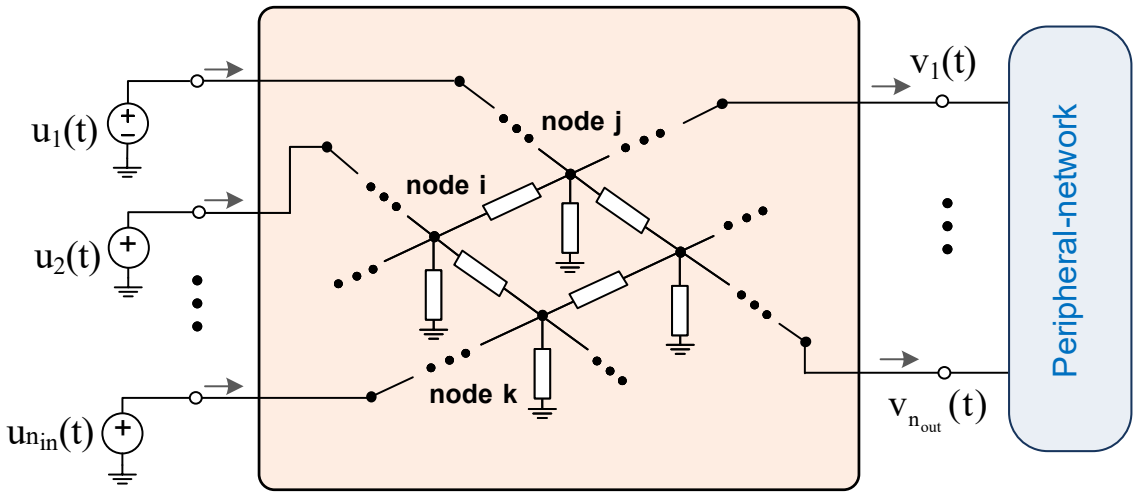


Figure 2.2: A network that accepts inputs and interacts with Peripherals.

The currents from the sources are added to (or subtracted from) the Kirchhoff's Current Law (KCL) equation for the corresponding nodes. The voltage at the terminal nodes (inputs) is directly decided (equated) by the voltage sources at corresponding nodes. The general idea is that the effect of the sources is "linearly injected" [25, 26] into the system at the associated nodes. Being linearly incorporated in the nodal formulation, the effect of input sources $\mathbf{u}(t)$ admits an affine realization for inputs (*c.f.* (2.3)).

Following the steps of the nodal analysis [27], the nonlinear state-models in (2.2a) (for a general class of nonlinear time-invariant circuits) can be recast as [28–30]

$$\frac{d}{dt}\mathbf{g}\left(\mathbf{x}(t)\right) = \boldsymbol{\mathcal{F}}\left(\mathbf{x}(t)\right) + \mathbf{B}\mathbf{u}(t). \tag{2.3}$$

The following remarks are noteworthy about (2.3)

- The input matrix $\mathbf{B}$ is directly applied to the source vector $\mathbf{u}(t)$ to map each source to the connected nodes;

- It will be shown in Section 2.2 that under a mild practical assumption, both the nonlinear differential term at the left-hand side ($\frac{d}{dt}\mathbf{g}(\mathbf{x}(t))$) and the nonlinear function $\boldsymbol{\mathcal{F}}(\mathbf{x}(t))$ can be recast in a simpler form, easing the application of numerical solvers;

- In the output equation (2.2b), $\mathbf{y}(t)$ can be a selection of the voltages and currents in $\mathbf{x}(t)$. The selection can be performed simply by multiplying a selection matrix $\mathbf{L}$ by $\mathbf{x}(t)$.

2.2 Formulation of Nonlinear Circuits

There are several established methods to obtain a representation for electrical networks, namely,

- Graph-based formulation [31–34],
- Sparse-Tableau Analysis (STA) formulation [33, 35],
- Modified Nodal Analysis (MNA) formulations [33, 36–38].

One may find a certain method more efficient for specific circuit typologies. The formulation of the MNA approach is straightforward to implement and is equally suitable for both frequency and time-domain analyses. Therefore, MNA can be reliably used to describe both linear and nonlinear circuits mathematically. Due to these properties, MNA is the standard formulation method for computer-based circuit analysis and is commonly used in general-purpose commercial circuit simulators such as SPICE [39].

MNA is an extension of the nodal analysis method in standard circuit theory [27]. Its strength lies in its ability to handle all types of circuit elements. Constructing the MNA formulation is usually done on an element-by-element basis and generates

a large system containing a mixed set of differential and algebraic equations through [33, 36–38],

- Writing the Kirchoff currents law (KCL) at each node;
- Considering node voltages as the unknowns in the formulation, The circuit equations will be solved to obtain these unknowns;
- Expressing the currents in the circuit elements in terms of the node voltages using the admittance form of constitutive relation of the circuit elements;
- Casting a special representation for the elements which either do not have an explicit representation in an admittance form, e.g., voltage sources or elements whose constitutive relation requires integration in the time-domain, e.g., inductors;
- Casting the charge or flux based formulation for nonlinear capacitors or nonlinear inductors, respectively, where the pertained charges and fluxs are included in the unknowns.

2.2.1 MNA Matrix Equations of Nonlinear Circuits

In general, a nonlinear circuit is described using the Modified Nodal Analysis (MNA) formulation as

$$\begin{aligned} \mathbf{G}\mathbf{x}(t) + \mathbf{C}\frac{d}{dt}\mathbf{x}(t) + \mathbf{f}\left(\mathbf{x}(t)\right) &= \mathbf{b}(t) \\ \mathbf{y}(t) &= \mathbf{L}\mathbf{x}(t) \end{aligned} \tag{2.4}$$

where

- $\mathbf{x}(t) \in \mathbb{R}^{n\times 1}$ is a vector of node voltages appended by currents in inductors, independent and dependent voltage sources, charges in nonlinear capacitors, and magnetic flux in nonlinear inductors.

- $\mathbf{G} \in \mathbb{R}^{n \times n}$ is a matrix representing the memory-less components of the circuit such as resistors
- $\mathbf{C} \in \mathbb{R}^{n \times n}$ is a matrix representing the components dependent on the changes in node voltages (components with memory) such as capacitors.
- $\mathbf{f}(\mathbf{x}(t)) \in \mathbb{R}^{n \times 1}$ is a vector of functions describing the nonlinear components. It contains the currents of nonlinear components such as diodes and the nonlinear entries corresponding to nonlinear capacitors and inductors.
- $\mathbf{b}(t) := \mathbf{B}\mathbf{u}(t) \in \mathbb{R}^{n \times 1}$ is a vector representing the independent voltage and current sources.
- $\mathbf{y}(t) \in \mathbb{R}^{n_{out}}$ contains the outputs that are the selections of the voltages and currents in $\mathbf{x}(t)$. The selection can be performed by multiplying $\mathbf{x}(t)$ by a selection matrix $\mathbf{L}$.
- n is the number of variables in the MNA formulation as seen in $\mathbf{x}(t)$.

The MNA formulation is constructed on an element-by-element basis. The process of placing the circuit components in the corresponding MNA matrices ($\mathbf{C}$, $\mathbf{G}$, $\mathbf{f}(\mathbf{x}(t))$, or $\mathbf{b}(t)$) is called "*component stamping*". Figure 2.3 shows samples of stamps for Resistors (R), Capacitors (C), Inductors (L), and Independent Sources.

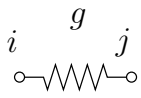
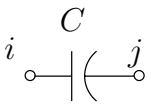
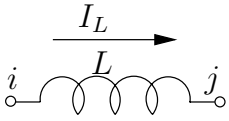
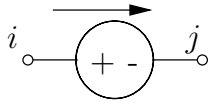
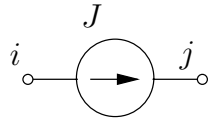
Element	Circuit Schematic	Stamp
Resistor		$\begin{array}{c} \\ i \\ \\ j \end{array} \begin{array}{c} \begin{array}{ccc} i & \cdots & j \end{array} \\ \begin{bmatrix} g & \cdots & -g \\ \vdots & \cdots & \vdots \\ -g & \cdots & g \end{bmatrix} \end{array}$
Capacitor		$\begin{array}{c} \\ i \\ \\ j \end{array} \begin{array}{c} \begin{array}{ccc} i & \cdots & j \end{array} \\ \begin{bmatrix} C & \cdots & -C \\ \vdots & \cdots & \vdots \\ -C & \cdots & C \end{bmatrix} \end{array}$
Inductor		$\begin{array}{c} \\ i \\ \\ j \\ \\ \end{array} \begin{array}{c} \begin{array}{cccc} i & \cdots & j & \end{array} \\ \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \cdots & \vdots & 0 \\ 0 & \cdots & 0 & -1 \\ 1 & 0 & -1 & -sL \end{bmatrix} \end{array}$
Voltage Source		$\begin{array}{c} \\ i \\ \\ j \\ \\ \end{array} \begin{array}{c} \begin{array}{cccc} i & \cdots & j & \end{array} \\ \begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \cdots & \vdots & 0 \\ 0 & \cdots & 0 & -1 \\ 1 & 0 & -1 & 0 \end{bmatrix} \end{array} \quad \begin{bmatrix} \\ \\ \\ E \end{bmatrix}$
Current Source		$\begin{array}{c} \\ i \\ \\ j \end{array} \begin{bmatrix} J \\ \vdots \\ -J \end{bmatrix}$

Figure 2.3: Component Stamps in MNA formulation.
(Courtesy of [40])

Chapter 3

DC Analysis of Nonlinear Circuits

A common challenge in simulating an electrical circuit's behavior using Computer-Aided Design (CAD) tools is to calculate the DC Operating Point (DC-OP) reliably and efficiently. As part of this document, I outline the challenges of constructing a model that tracks the DC-OP variation versus the variations in a selected set of parameters considered to be key to circuit performance. Before I dive into these challenges, I must briefly describe the process behind calculating/simulating the DC operating point.

3.1 DC Circuit Equation

In this chapter, I focus on the steps required to find the DC solution of a general circuit described by Equation (2.4). Unlike this general form, the DC solution is not time-dependent. Thus, $\frac{d\mathbf{x}(t)}{dt} = 0$ and $\mathbf{x}(t)$ can be denoted simply as $\mathbf{x}$. This simplifies Equation (2.4) into Equation (3.1). Please note the absence of the C matrix as multiplying it by $\frac{d\mathbf{x}(t)}{dt} = 0$ eliminates its effects.

$$\mathbf{G}\mathbf{x} + \mathbf{f}(\mathbf{x}) = \mathbf{b} \tag{3.1}$$

3.2 Newton-Raphson Iteration to Solve Nonlinear Equations

The DC solution of Equation (3.1) is defined as finding the values of the vector $\mathbf{x}$ that satisfies

$$\mathbf{G}\mathbf{x} + \mathbf{f}(\mathbf{x}) - \mathbf{b} = 0 \tag{3.2}$$

In the following, we will use $\mathbf{\Phi}(\mathbf{x})$ to stand for the residual error that arises from not satisfying the DC equation for an arbitrary $\mathbf{x}$. In other words $\mathbf{\Phi}(\mathbf{x})$ is defined by

$$\mathbf{\Phi}(\mathbf{x}) := \mathbf{G}\mathbf{x} + \mathbf{f}(\mathbf{x}) - \mathbf{b} \tag{3.3}$$

As explained previously, $\mathbf{x}$ is the node voltages and the currents of the voltage sources. Finding the DC solution of Equation (3.1) can be challenging as the equation's nonlinear elements ($\mathbf{f}(\mathbf{x})$) depend on the value of $\mathbf{x}$. To circumvent this obstacle, we can use a numerical method such as Newton-Raphson Iteration.

Newton-Raphson (N-R) is an iterative method that requires the user to make an initial guess for the value of $\mathbf{x}$. Using this guess, we can compute $\mathbf{f}(\mathbf{x})$. At this point, $\mathbf{x}$ and consequently $\mathbf{f}(\mathbf{x})$, are an estimation of their true value ("the solution"). Using this guess as a starting point, the algorithm can iteratively improve the guess and automatically correct the value of $\mathbf{x}$. Being a numerical method, the number of iterations that it takes to achieve convergence is dependent on the proximity of the initial guess to the true solution and the expected accuracy of the solution.

As explained above, for N-R method to satisfy Equation (3.2), it is required to make an initial guess denoted as $\mathbf{x}^{(0)}$. Naturally, the initial guess will likely not be the correct solution to Equation (3.2). This leaves an error represented by $\mathbf{\Phi}(\mathbf{x}^{(0)})$, as seen in Equation (3.4). The next step is to correct $\mathbf{x}^{(0)}$ so that the new estimation

($\mathbf{x}^{(1)}$) reduces the error such that $\mathbf{\Phi}(\mathbf{x}^{(1)}) < \mathbf{\Phi}(\mathbf{x}^{(0)})$.

$$\mathbf{\Phi}(\mathbf{x}^{(0)}) = \mathbf{G}\mathbf{x}^{(0)} + \mathbf{f}(\mathbf{x}^{(0)}) - \mathbf{b} \tag{3.4}$$

To calculate the correction required ($\mathbf{\Delta x}^{(1)}$) to obtain the new estimation ($\mathbf{x}^{(1)}$), we use Equation 3.5, where $\mathbf{\Psi}(\mathbf{x})$ is the matrix of partial derivatives defined by Equation (3.6).

$$\mathbf{\Delta x}^{(1)} = \mathbf{\Psi}(\mathbf{x}^{(0)})^{-1}\mathbf{\Phi}(\mathbf{x}^{(0)}) \tag{3.5}$$

$$\mathbf{\Psi}(\mathbf{x}) = \frac{\partial \mathbf{\Phi}(\mathbf{x})}{\partial x} = \mathbf{G} + \mathbf{J}(\mathbf{x}) \tag{3.6}$$

$$\mathbf{J} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \tag{3.7}$$

Having obtained $\mathbf{\Delta x}^{(1)}$ from Equation (3.5), we can now modify the initial trial vector ($\mathbf{x}^{(0)}$) to obtain the next trial vector ($\mathbf{x}^{(1)}$) using Equation (3.8)

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \mathbf{\Delta x}^{(0)} \tag{3.8}$$

The following process is repeated until convergence is achieved under the following two conditions, where i represents the ith iteration (assuming i=0 is the initial guess).

$$i) \quad \|\mathbf{x}^{(i+1)} - \mathbf{x}^{(i)}\| < \varepsilon_1$$
$$ii) \quad \|\mathbf{\Phi}(\mathbf{x}^{(i+1)})\| < \varepsilon_2$$

In the above two conditions, ε_1 and ε_2 are predefined error tolerances. Due to the iterative nature of the N-R method, it is important to choose the initial guess ($\mathbf{x}^{(0)}$) and error tolerances carefully. Each iteration requires the computationally involved process of taking the LU decomposition of the Jacobian matrix and Forward/Backward substitution. For this reason, the objective should be to minimize the number of iterations required to achieve an acceptably accurate solution.

Algorithm 1 presents a pseudo-code implementation of the N-R method to find the circuits' DC solution following the steps described above.

Algorithm 1: Finding DC solution of nonlinear circuits using N-R.

Input: $\mathbf{G}$,$\mathbf{b}$,$\mathbf{f}(\mathbf{x})$, $\mathbf{x^{(0)}}$ `// initial guess`
Output: $\mathbf{x_{sol}}$ `// solution`

$\mathbf{\Phi}(\mathbf{x^{(0)}}) = \mathbf{G}\mathbf{x^{(0)}} + \mathbf{f}(\mathbf{x^{(0)}}) - \mathbf{b}$
$\mathbf{\Psi}(\mathbf{x^{(0)}}) = \mathbf{G} + \mathbf{J}(\mathbf{x^{(0)}})$

while $(\|\mathbf{x}^{(i+1)} - \mathbf{x}^{(i)}\| > \varepsilon_1)$ ***or*** $(\mathbf{\Phi}(\mathbf{x^{(i)}}) > \varepsilon_2)$ **do**
    `// calculate delta x using LU decomposition`
    $[\mathbf{L}, \mathbf{U}, \mathbf{P}, \mathbf{Q}] = \mathrm{lu}(\mathbf{\Psi}(\mathbf{x^{(i-1)}}))$
    $\mathbf{Y} = \mathbf{L}(\mathbf{P} \times \mathbf{\Phi}(\mathbf{x^{(i-1)}}))$
    $\mathbf{Z} = \mathbf{U} \times \mathbf{Y}$
    $\mathbf{\Delta x^{(i)}} = \mathbf{U} \times \mathbf{Y}$

    `// calculate new estimation of x`
    $\mathbf{x^{(i)}} = \mathbf{x^{(i-1)}} - \mathbf{\Delta x^{(i)}}$ `// calculate new estimation of X`

    `// update using new x estimation`
    $\mathbf{\Phi}(\mathbf{x^{(i)}}) = \mathbf{G}\mathbf{x^{(i)}} + \mathbf{f}(\mathbf{x^{(i)}}) - \mathbf{b}$
    $\mathbf{\Psi}(\mathbf{x^{(i)}}) = \mathbf{G} + \mathbf{J}(\mathbf{x^{(i)}})$
end
return $\mathbf{X_{sol}} = \mathbf{X^{(i)}}$

3.3 Continuation Method for DC Analysis

As previously outlined, a common challenge when using numerical methods such as N-R is guaranteeing convergence to find the solution. Parameter embedding methods, also known as Continuation Methods and Homotopy, efficiently address the convergence problems.

This section will go over Homotopy methods and their application in source ramping to efficiently find the DC solution of nonlinear circuits.

3.3.1 Homotopy

Homotopy methods are effective at solving systems of nonlinear algebraic equations. In the context of circuit simulation, we attempt to tackle a zero finding problem such as Equation (3.9), where $\mathbf{x}$ is $\mathbb{R}^n$ and $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^n$.

$$\mathcal{F}(\mathbf{x}) = 0 \tag{3.9}$$

To find the solution of Equation (3.9), we embed a continuation parameter (λ) into $\mathcal{F}(x)$ to create the Homotopy function $\mathcal{H}(\mathbf{x}, \lambda)$ as seen in Equation (3.10).

$$\mathcal{H}(\mathbf{x}, \lambda) = 0 \tag{3.10}$$

The Homotopy parameter $\lambda \in \mathbb{R}$ and the mapping is $\mathcal{F} : \mathbb{R}^{n+1} \to \mathbb{R}^n$.

To better explain this, consider Equation (3.11) as an example of a simple Homotopy problem.

$$\mathcal{H}(\mathbf{x}, \lambda) = (1 - \lambda)\mathcal{G}(\mathbf{x}) + \lambda\mathcal{F}(\mathbf{x}) \tag{3.11}$$

By setting the continuation parameter (λ) to zero, we get $\mathcal{H}(\mathbf{x}, 0) = \mathcal{G}(\mathbf{x})$ which has a simple solution. If we set $\lambda = 1$, we get $\mathcal{H}(\mathbf{x}, 1) = \mathcal{F}(\mathbf{x}) = 0$ which is our original problem. Thus, by gradually increasing λ from 0 to 1 and finding the solution of $\mathcal{H}(\mathbf{x}, \lambda)$, we can more reliably calculate the solution of $\mathcal{F}(\mathbf{x}) = 0$.

3.3.2 Source Ramping

Source ramping is a special use-case of Homotopy to increase the probability of convergence. As the name implies, source ramping requires the gradual increase of the DC source voltages from zero using a multiplication factor (α). Equation (3.12) demonstrates the incorporation of the scaling factor in the general nonlinear MNA

equation.

$$\mathbf{\Phi}(\mathbf{x}, \alpha) = \mathbf{G}\mathbf{x} + \mathbf{f}(\mathbf{x}) - \alpha\mathbf{b} = 0 \tag{3.12}$$

As explained in Section 3.2, the first step of the process is to set α to an arbitrarily small number such 0.2 and set the initial guess as zero ($\mathbf{x}_1^{(0)} = 0$). N-R is then used to find the solution which takes ν iterations denoted as $\mathbf{x}^{(\nu)} = \mathbf{x}^{(\text{sol})}$. Now that the solution is found, the DC source voltages are gradually scaled up in stages, and the solution from the previous stage is used as the initial guess of the next stage. This process is illustrated in Table 3.1.

Table 3.1: Scaling the source to solve for the DC solution

Source Ramping Index	**Source Scale (α)**	**Initial Guess**	**Solution**
1	0.2	$\mathbf{x}_1^{(0)} = \mathbf{0}$	$\mathbf{x}_1^{(\text{sol})}$
2	0.4	$\mathbf{x}_2^{(0)} = \mathbf{x}_1^{(\text{sol})}$	$\mathbf{x}_2^{(\text{sol})}$
3	0.6	$\mathbf{x}_3^{(0)} = \mathbf{x}_2^{(\text{sol})}$	$\mathbf{x}_3^{(\text{sol})}$
4	0.8	$\mathbf{x}_4^{(0)} = \mathbf{x}_3^{(\text{sol})}$	$\mathbf{x}_4^{(\text{sol})}$
DC Solution	**1**	$\mathbf{x}_5^{(0)} = \mathbf{x}_4^{(\text{sol})}$	$\mathbf{x}_5^{(\text{sol})}$

As explained previously, N-R is an iterative method that can take several iterations to converge. However, convergence is not always guaranteed. By implementing source ramping, it increases the probability of convergence. However, with each gradual increase in α, the entire N-R process has to be rerun, making this method even more computationally demanding than N-R alone. That is why it is important to carefully choose the step size at which α increases to guarantee convergence but minimizes the computational complexity (total number of N-R iterations).

Evidently, optimizing the computational cost and reducing simulation times is important when tackling circuit simulation problems. With the combination N-R and source ramping, the total number of N-R iterations rapidly balloons. Chapter 4

will introduce methods to reduce the computational cost of each N-R iteration. This optimization will have a favourable effect on reducing the overall computational cost of DC circuit simulation.

Chapter 4

Advanced Simulations Using MOR Techniques

Model-order reduction (MOR) has proven to be an effective tool in reducing the computational complexity of simulating large systems. MOR has been successfully used in a broad spectrum of linear and nonlinear applications [41–48] such as microelectronics [39, 40, 49–51], high-speed and RF circuits [52–55], uncertainty quantification [9,56–58], electromagnetic [59,60] and thermal analysis [61]. The recent evolution of MOR techniques have also been fueled by their popularity and success in broader fields such as mechanical, biomedical, civil, and aerospace engineering [62–64].

This chapter is intended as the primary platform for reviewing the notations related to Model-Order Reduction (MOR) and its application in circuit simulation. Naturally, the concepts, techniques, and applications of MOR are too vast to be covered in one chapter. Rather, this chapter is more focused on the techniques that form the basis for the contribution of this thesis. The organization of the chapter proceeds along the following sections.

Section 4.1 introduces the idea of MOR as a general concept, while Section 4.2 shows its application in the domain of linear circuits through the concept of projection based MOR. Section 4.3 then presents a more detailed look in the computation of

the projection operator. Section 4.4 then moves to the more specialized idea of Parameterized MOR (PMOR) which is the principal technique used in this thesis. The presentation of this section reviews the application of PMOR in linear circuits. It subsequently explores the application of PMOR in general nonlinear circuits to highlight the main challenges which are addressed by the contribution in this thesis.

4.1 MOR in the Context of Circuit Simulation

Model-Order Reduction is an effective technique used to reduce the computation time required to simulate large circuits. The high-level strategy behind MOR techniques is to speed up the simulation time by reducing the order of the circuit's mathematical model. By reducing the order of the model, it is possible to reduce the computation time but at the risk of reduced accuracy. However, in most applications, the model's order can be reduced to a certain extent to minimize the CPU-cost without noticeable degradation of accuracy.

4.2 MOR for Linear Circuits

Model-Order Reduction (MOR) for linear systems is a well-established area, and a rich body of literature is available on the subject [43, 46, 51, 65–67]. This section will discuss a common approach to reducing linear systems based on using orthogonal projection operators.

As previously explained, the MNA equation for a general linear circuit is obtained by dismissing the nonlinearity from (2.4) to get (4.1).

$$\begin{aligned} \mathbf{G}\mathbf{x}(t) + \mathbf{C}\frac{d\mathbf{x}(t)}{dt} &= \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{L}\mathbf{x}(t) \end{aligned} \tag{4.1}$$

where $\mathbf{G}, \mathbf{C} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times n_{\text{in}}}$, $\mathbf{u}(t)$ is the circuit's n_{in} stimuli at the inputs, $\mathbf{L} \in \mathbb{R}^{n_{\text{out}}\times n}$, and n is the size of the original system.

The key idea behind MOR is to reduce the number of state variables by projecting the vector $\mathbf{x}(t)$ on to an m-dimensional subspace spanned by the column vectors of the orthogonal matrix $\mathbf{Q} \in \mathbf{R}^{n\times m}$ as

$$\mathbf{x}(t) \;=\; \mathbf{Q}\hat{\mathbf{x}}(t) \tag{4.2}$$

where $\hat{\mathbf{x}}(t) \in \mathbb{R}^{m\times 1}$ and $m \ll n$. Next, using (4.2), the Reduced Order Model (ROM) is obtained through the Galerkin projection scheme [48] as

$$\begin{aligned} \hat{\mathbf{G}}\hat{\mathbf{x}}(t) + \hat{\mathbf{C}}\frac{d\hat{\mathbf{x}}(t)}{dt} &= \hat{\mathbf{B}}\mathbf{u}(t) \\ \mathbf{y}(t) &= \hat{\mathbf{L}}\hat{\mathbf{x}}(t) \end{aligned} \tag{4.3a}$$

where

$$\begin{aligned} \hat{\mathbf{C}} &\triangleq \mathbf{Q}^\top\mathbf{C}\mathbf{Q} \in \mathbb{R}^{m\times m} \qquad & \hat{\mathbf{B}} &\triangleq \mathbf{Q}^\top\mathbf{B} \in \mathbb{R}^{m\times n_{in}} \\ \hat{\mathbf{G}} &\triangleq \mathbf{Q}^\top\mathbf{G}\mathbf{Q} \in \mathbb{R}^{m\times m} \qquad & \hat{\mathbf{L}} &\triangleq \mathbf{L}\mathbf{Q} \in \mathbb{R}^{n_{out}\times m} \end{aligned} \tag{4.3b}$$

By preserving specific properties in the original system, the reduced model in (4.3a) should ideally provide a sufficiently accurate approximation for the system response $\mathbf{x}$.

The framework introduced above for MOR leaves out the details of the construction of the projection matrix $\mathbf{Q}$ or the requirements that it needs to satisfy in order for the reduced system (4.3a) to preserve the essential features of the original system (4.1) and guarantee its accuracy. The computation of $\mathbf{Q}$ will be discussed in more depth in Section 4.3.

4.3 Construction of Projection Operator

Using the projection framework described above as an effective reduction strategy has been extensively explored in literature for linear [51, 68–74] and nonlinear [44, 50, 75–84] systems. The key steps in the projection operation is the construction of the projection operator matrix $\mathbf{Q}$. The method employed to obtain $\mathbf{Q}$ is the main differentiating factor between the reduction methods.

Finding suitable projection matrices is the main task in any MOR method. Therefore, the next section will elaborate on the construction of $\mathbf{Q}$ as related to the subject of this thesis.

4.3.1 Construction of Q Using Krylov-Subspace Methods

The Krylov-Subspace approach to constructing the projection matrix for use in the model-order reduction of linear circuits necessitates working in the Laplace-domain instead of time-domain.

By applying Laplace transformation to the MNA equations in (4.1), the corresponding complex-valued matrix transfer function (in complex frequency s-domain) is obtained as a relationship between the Laplace-domain output $\mathbf{Y}(s)$ and the Laplace-domain input $\mathbf{U}(s)$, in the form

$$\mathbf{Y}(s) = \mathbf{H}(s)\mathbf{U}(s) \tag{4.4}$$

where

$$\mathbf{H}(s) = \mathbf{L}\left(\mathbf{G} \; + \; s\mathbf{C}\right)^{-1}\mathbf{B}, \tag{4.5}$$

It is reasonable to assume that the matrix pencil $(\mathbf{G}, \mathbf{C})$ is "regular" which means that the matrix $(\mathbf{G} + s\mathbf{C})$ is non singular except at a finite number of points in the Laplace-domain. With this assumption, a properly selected value s_o is chosen in the

Laplace-domain such that $(\mathbf{G} + s_o\mathbf{C})$ is non-singular. Using a properly selected s_o as an expansion point, the transfer function (4.5) can be expanded in Taylor series in the vicinity of s_o as

$$\mathbf{H}(s) = \mathbf{L} \sum_{j=0} \mathbf{M}_j(s_o)(s - s_o)^j \tag{4.6}$$

where the coefficient of $(s - s_o)^j$,

$$\mathbf{M}_j(s_o) = \frac{1}{j!} \frac{\partial^j \mathbf{H}(s)}{\partial s^j}\Big|_{s=s_o}$$

is called the j^{th} moment of the system's variables at s_o and is given by [51]

$$\mathbf{M}_j(s_o) \triangleq \mathbf{A}^j \, \mathbf{R} \tag{4.7a}$$

where

$$\mathbf{R} \triangleq (\mathbf{G} + s_o\mathbf{C})^{-1} \mathbf{B} \qquad \in \mathbb{R}^{n \times n_{in}} \tag{4.7b}$$

$$\mathbf{A} \triangleq -(\mathbf{G} + s_o\mathbf{C})^{-1} \mathbf{C} \quad \in \mathbb{R}^{n \times n} . \tag{4.7c}$$

The moments for the transfer function can be obtained using the system's moments in (4.7) as

$$\mathcal{M}_i(s_o) = \mathbf{L}\mathbf{M}_j(s_o) \quad \in \mathbb{R}^{n_{\text{out}} \times n_{\text{in}}} \tag{4.8}$$

The so-called Krylov-Subspace MOR methods constructs the orthogonal projection matrix by ensuring that

1. The m columns of $\mathbf{Q}$ are orthonormal, i.e.,

$$\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_m \tag{4.9}$$

2. The m-dimensional space of the columns of $\mathbf{Q}$ span the space formed by the columns of $\mathbf{R}$, $\mathbf{AR}$, ..., $\mathbf{A}^{(m-1)}$. This feature of $\mathbf{Q}$ is typically expressed by

$$\operatorname{colsp} \mathbf{Q} = \mathcal{K}\boldsymbol{r}(\mathbf{A}, \mathbf{R}, m) = \operatorname{span} \left\{ \mathbf{R},\ \mathbf{AR},\ \ldots,\ \mathbf{A}^{(m-1)}\mathbf{R} \right\} \tag{4.10}$$

where $m = j \times n_{\text{in}}$, j is the number of the block-moments, and n_{in} is the number of the columns of $\mathbf{B}$ in the circuit formulation. The Arnoldi algorithm is an example of an efficient method for computing $\mathbf{Q}$ [85].

Once $\mathbf{Q}$ is constructed, it is then used to construct the reduced system as shown in (4.2) and (4.3).

Through Galerkin projection [65, 86] and using the projection matrix $\mathbf{Q}$ which satisfies the above two conditions, the reduced order model (4.3a) and the original (4.1) will share the first m moments about the expansion frequency point s_o. This is key to ensure that the reduced system will approximate the original system with good accuracy [49, 69].

4.4 MOR for Parametric Circuits

The term "parametric circuits" is used in this thesis to identify circuits with several design parameters that have been designated as key or critical to the performance of the circuit with respect to certain metrics.

During the circuit design phase, specific design parameters (e.g., layout and material features) influence the circuit's behaviour and must be further analyzed by designers to satisfy the design specifications. This leads to design tasks such as design space exploration, optimization, and variability analysis that are, in general, computationally cumbersome. The high computational complexity arises from having to repeatedly run the simulation of the circuit at numerous points in the space of

the design parameters.

The objective of this section is to review the role that MOR techniques play in reducing the computational cost associated with the idea of design space explorations. In this context, MOR is often referred to as Parameterized MOR (PMOR) to emphasise the fact that the obtained reduced system approximates the original system in regards to its behaviour with respect to a selected set of design parameters. The term "design space exploration" is also known under different names, such as the "parameter sweep" which is used in circuit simulation to refer to the notion of repeating the simulation under different values for the selected set of design parameters. Several PMOR methods have been developed for linear [8–10, 12, 59, 87–95] and nonlinear systems [15, 16, 96, 97] to capitalize on the savings achieved by using reduced order models.

The plan for this review proceeds along the following lines:

1. First, the mathematical formulation of the general nonlinear circuit (MNA) is modified to account for the presence of the design parameters whose effect on the circuit performance needs to be explored. This is done in section 4.4.1.

2. An overview of the PMOR approach that is proposed to handle the task of design space exploration in *linear circuits* is presented in Section 4.4.2.

3. Section 4.4.2 will push the idea of PMOR to the domain of *nonlinear circuits* using the same scheme employed in linear circuits. It will be shown that this attempt will engender new challenges not seen in the application of PMOR in linear circuits. More particularly, using the linear-type of PMOR directly on the nonlinear circuit will force the computation to alternate between the reduced-dimension space and the original large dimensional space of the circuit, creating a computational bottleneck whose resolution will be presented

in the following section (section 4.5) through the idea of Discrete Empirical Interpolation Method (DEIM).

4.4.1 General Formulation of Parametric Circuits

The dependence of a circuit's response on the design parameters $\boldsymbol{\xi} = \{\xi_1, \ldots, \xi_d\} \in \boldsymbol{\Omega} \subset \mathbb{R}^d$ can be represented by adapting the modified nodal analysis (MNA) formulation to the following form

$$\begin{aligned} \mathbf{G}(\boldsymbol{\xi})\mathbf{x}(t,\boldsymbol{\xi}) + \mathbf{C}(\boldsymbol{\xi})\frac{d\mathbf{x}(t,\boldsymbol{\xi})}{dt} + \mathbf{f}(\mathbf{x}(t,\boldsymbol{\xi}),\boldsymbol{\xi}) &= \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t,\boldsymbol{\xi}) &= \mathbf{L}\mathbf{x}(t,\boldsymbol{\xi}) \end{aligned} \tag{4.11}$$

where $\mathbf{C}(\boldsymbol{\xi}), \mathbf{G}(\boldsymbol{\xi})$ are $n \times n$ parameter-dependent conductance and susceptance matrices, respectively. $\mathbf{x}(t,\boldsymbol{\xi}) \in \mathbb{R}^n$ is the vector of parameter-dependent MNA variables, $\mathbf{y}(t,\boldsymbol{\xi})$ contains the n_{out} parameter-dependent output response, $\mathbf{B} \in \{-1,0,1\}^{n \times n_{\text{in}}}$, and $\mathbf{L} \in \{-1, 0.1\}^{n_{\text{out}} \times n}$ are input and output selector matrices. The vector-valued function $\mathbf{f}(\mathbf{x}(t,\boldsymbol{\xi}),\boldsymbol{\xi})$ represents the nonlinearity in the circuits with a nonlinear dependence on both the parameters $\boldsymbol{\xi} \in \boldsymbol{\Omega}$ and parameter-dependent MNA variables $\mathbf{x}(t,\boldsymbol{\xi})$. The parametric linear circuits can be represented by dismissing the nonlinear term from (4.11).

Numerous design tasks, such as design optimization and sensitivity and variability analysis, require multiple simulations of the system for multiple variations in the design parameters (e.g., layout features of an electronic system). Simulating the original large-scale models of these systems (4.11) for each parameter variation can become computationally expensive.

The main ideas of PMOR will be introduced through its applications in computing the variation of the DC operating point with regards to variations in the design parameters $\boldsymbol{\xi}$. PMOR operates in the same framework of the MOR approach: projecting

the original system matrices onto a reduced space using a projection operator. However, the reduced system matrices of PMOR preserve the dependence of the system of equations on the parameters. As a result, an approximation of such dependencies is manifested in the reduced model. There are certain strategies in literature to achieve this goal [12].

4.4.2 PMOR For DC Solution of Linear Circuits

The problem formulation for computing the DC operating point of linear circuits is obtained by dismissing the nonlinear $\mathbf{f}(\mathbf{x}(t, \boldsymbol{\xi}), \boldsymbol{\xi})$ and the derivative terms from 4.11 as

$$\boldsymbol{\Phi}_{\text{dc}}(\mathbf{x}_{\text{dc}}(\boldsymbol{\xi}), \boldsymbol{\xi}) := \mathbf{G}(\boldsymbol{\xi})\mathbf{x}_{\text{dc}}(\boldsymbol{\xi}) - \mathbf{b}_{\text{dc}} = \mathbf{0} \tag{4.12}$$

where $\mathbf{x}_{\text{dc}}(\boldsymbol{\xi}) \in \mathbb{R}^N$ is the DC solution, $\boldsymbol{\xi} \in \boldsymbol{\Omega}$ is a vector collecting d circuit parameters $\{\xi_1, \ldots, \xi_d\}$, $\mathbf{b}_{\text{dc}} \in \mathbb{R}^N$ is a vector containing the DC sources in the circuit, and N is the number of circuit variables.

The purpose of solving parametric linear systems, as given in (4.12), is to track the variation of the DC operating point $\mathbf{x}_{\text{dc}}(\boldsymbol{\xi})$ of a circuit in response to variation in parameters $\boldsymbol{\xi}$. The large systems defining today's complex designs need to be repetitively solved for different values of the parameters.

The central idea in using PMOR is to approximate the large system in (4.12) with a reduced model, while preserving the same dependency on the parameters for the reduced solution. Tracking the variation of the DC operating point $\mathbf{x}_{\text{dc}}(\boldsymbol{\xi})$ can be performed by solving for the solution in the reduced space. This leads to a noticeable reduction in CPU-cost.

PMOR based on moment matching has been applied for linear circuits for purposes analogous to the DC operating point (e.g., computing the variation in the frequency response) [8–12]. In the context of DC operating points, moment-matching based

PMOR involves finding a reduced DC equation whose solution has a few leading parameter moments that match those of the original DC solution (4.12).

Single Parameter Moment-Matching PMOR for DC Operating Point of Linear Circuits:

In order to find the parametric reduced order model, the moments of the DC response of the circuit with respect to the random parameters are obtained.

To facilitate the presentation of the idea and without loss of generality, a single design parameter of interest is assumed in the following derivation (i.e., $d = 1$ and the vector $\boldsymbol{\xi}$ is replaced with the scalar ξ).

This approach proceeds by first expressing the set of variables $\mathbf{x}_{\rm dc}(\xi)$ as a Taylor series around the nominal value of the parameter, denoted ξ_0

$$\mathbf{x}_{\rm dc}(\xi) = \sum_{i=0}^{\infty} \mathbf{M}_i(\xi_0)\,(\xi - \xi_0)^i \tag{4.13}$$

where

$$\mathbf{M}_i(\xi_0) := \frac{1}{i!}\,\frac{\partial^i \mathbf{x}_{\rm dc}(\xi)}{\partial \xi^i}\bigg|_{\xi=\xi_0} \in \mathbb{R}^n \tag{4.14}$$

are the so-called moments. Forming the reduced system is carried out by first constructing an orthogonal basis set $\mathbf{Q} \in \mathbb{R}^{n\times m}$ whose column vectors form a basis for the subspace spanned by the moments.

$$\text{colsp}\,\mathbf{Q} = \text{span}\,\{\mathbf{M}_0(\xi_0), \mathbf{M}_1(\xi_0), \cdots \mathbf{M}_{m-1}(\xi_0)\} \tag{4.15}$$

For the sake of illustration, assume that $\mathbf{G}(\xi)$ is dependent on ξ in an affine manner, that is

$$\mathbf{G}(\xi) = \mathbf{G}(\xi_0) + \xi\tilde{\mathbf{G}}(\xi_0). \tag{4.16}$$

Next, the reduced system is obtained from the original system 4.12 using variable change as

$$\mathbf{x}_{\mathrm{dc}}(\xi) = \mathbf{Q}\,\hat{\mathbf{x}}_{\mathrm{dc}}(\xi) \tag{4.17a}$$

and through congruent transformation,

$$\hat{\mathbf{\Phi}}(\hat{\mathbf{x}}_{\mathrm{dc}}(\xi), \xi) = \hat{\mathbf{G}}(\xi)\,\hat{\mathbf{x}}_{\mathrm{dc}}(\xi) - \hat{\mathbf{b}}_{\mathrm{dc}} = \mathbf{0} \tag{4.17b}$$

where

$$\hat{\mathbf{G}}(\xi) \triangleq\ \mathbf{Q}^{\top}\mathbf{G}(\xi_0)\mathbf{Q} + \xi\mathbf{Q}^{\top}\tilde{\mathbf{G}}(\xi_0)\mathbf{Q}, \qquad \hat{\mathbf{b}}_{\mathrm{dc}} \triangleq\ \mathbf{Q}^{\top}\mathbf{b}_{\mathrm{dc}}. \tag{4.17c}$$

Multi-Parameter Moment-Matching PMOR for DC Operating Point of Linear Circuits:

Similar to the single parameter case, the development of multi-parameter moment-matching PMOR to find the DC operating point of nonlinear circuits is established based on the idea of expanding $\mathbf{x}_{\mathrm{dc}}(\boldsymbol{\xi})$ in the Taylor series around an expansion point $\boldsymbol{\xi}_0 = \{\xi_{1,0}, \ldots, \xi_{d,0}\}$, which is the vector of nominal values for the parameters. A multidimensional representation of the Taylor expansion of $\mathbf{x}(\boldsymbol{\xi})$ using the notion of multi-index $\boldsymbol{\alpha} \in \mathbb{N}^d$ is given as

$$\mathbf{x}_{\mathrm{dc}}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha}} \mathbf{M}_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_0) \prod_{i=1}^{d} (\xi_i - \xi_{i,0})^{\alpha_i} \tag{4.18}$$

where $\mathbf{M}_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_0)$ are the coefficients of the DC solution's Taylor series and $\boldsymbol{\alpha}$ in this case is referred to as a multi-index (a vector of indices). Let the Taylor series expansion (4.18) be truncated such that $\boldsymbol{\alpha}$ is confined to a set signified as $\boldsymbol{\Lambda}$. More details about (4.18) will be discussed in the next sections (c.f. Section 4.4.3).

Next, a set of orthogonal basis spanning the space defined by the columns of the

moments is computed [10, 12, 98]

$$\operatorname{colsp} \mathbf{Q} = \operatorname{span}\{\mathbf{M}_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_0)\}, \qquad \boldsymbol{\alpha} \in \boldsymbol{\Lambda} \tag{4.19}$$

where $\mathbf{Q} \in \mathbb{R}^{n \times m}$ and m is the number of elements in $\boldsymbol{\Lambda}$ ($m \ll n$).

Also, by an affine representation for $\mathbf{G}(\boldsymbol{\xi})$, as proposed in [10], we have

$$\mathbf{G}(\boldsymbol{\xi}) = \sum_{i=1}^{d} \tilde{\mathbf{G}}_i \xi_i. \tag{4.20}$$

Using the orthogonal matrix $\mathbf{Q}$ from (4.19), the ROM can be computed following the projection scheme as

$$\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi}) \;=\; \mathbf{Q}\mathbf{x}_{\mathrm{dc}}(\boldsymbol{\xi}). \tag{4.21}$$

In a similar fashion, as shown in (4.17), the parametrized ROM for the DC solution of multi-parameter systems can be written through congruent transformation and using (4.20) as

$$\boldsymbol{\Phi}_{\mathrm{dc}}(\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi}), \boldsymbol{\xi}) \;=\; \hat{\mathbf{G}}(\boldsymbol{\xi})\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi}) - \hat{\mathbf{b}}_{\mathrm{dc}} \;=\; \mathbf{0} \tag{4.22a}$$

where

$$\hat{\mathbf{G}}(\boldsymbol{\xi}) \triangleq \sum_{i=1}^{d} \mathbf{Q}^{\top} \tilde{\mathbf{G}}_i \mathbf{Q}, \qquad \hat{\mathbf{b}}_{\mathrm{dc}} \triangleq \mathbf{Q}^{\top} \hat{\mathbf{b}}_{\mathrm{dc}}. \tag{4.22b}$$

4.4.3 PMOR for DC Solution of Nonlinear Circuits

Similar to the linear circuits, the problem formulation for computing the DC operating point of nonlinear systems (4.11) is obtained by

$$\boldsymbol{\Phi}_{\mathrm{dc}}(\mathbf{x}_{\mathrm{dc}}(\boldsymbol{\xi}), \boldsymbol{\xi}) := \; \mathbf{G}(\boldsymbol{\xi})\mathbf{x}_{\mathrm{dc}}(\boldsymbol{\xi}) + \mathbf{f}(\mathbf{x}_{\mathrm{dc}}(\boldsymbol{\xi}), \boldsymbol{\xi}) - \mathbf{b}_{\mathrm{dc}} = \mathbf{0} \tag{4.23}$$

Applying PMOR on nonlinear circuits using moment matching has not been addressed before in the literature. The objective in this section is to attempt expanding the PMOR technique, as it is used in linear circuits through moment matching, to nonlinear circuits.

The steps for the reduction of (4.23) based on moment matching and projection are illustrated below to highlight the associated challenges and computational bottlenecks. These issues will be discussed in detail in the upcoming sections.

Applying the moment-matching approach used to reduce the above mentioned system (4.23) requires the following main steps,

1. Expanding $\mathbf{x}(\boldsymbol{\xi})$ in Taylor series around $\boldsymbol{\xi}_0$ with a multidimensional representation by using the notion of multi-index with $\boldsymbol{\alpha} \in \mathbb{N}^d$ as

$$\mathbf{x}_{\mathrm{dc}}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha}} \mathbf{M}_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_0) \prod_{i=1}^{d} (\xi_i - \xi_{i,0})^{\alpha_i} \tag{4.24}$$

   where $\mathbf{M}_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_0)$ are the moments of the truncated Taylor expansion.

2. Forming an orthonormal basis $\mathbf{Q} \in \mathbb{R}^{N \times m}$ for the subspace spanned by the set of moments collected from the truncation to the set Λ,

$$\operatorname{colsp} \mathbf{Q} = \operatorname{span} \left\{ \mathbf{M}_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_0) \right\}, \quad \boldsymbol{\alpha} \in \Lambda \tag{4.25}$$

3. Finally, using an affine representation for $\mathbf{G}(\boldsymbol{\xi})$, as proposed in [10],

$$\mathbf{G}(\boldsymbol{\xi}) = \sum_{i=1}^{d} \tilde{\mathbf{G}}_i \xi_i, \tag{4.26}$$

   premultiplying (5.1) by $\mathbf{Q}^\top$ and through the change of variables,

$$\mathbf{x}_{\mathrm{dc}}(\boldsymbol{\xi}) = \mathbf{Q} \hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi}) \tag{4.27}$$

a reduced system of the form

$$\hat{\boldsymbol{\Phi}}(\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi}),\boldsymbol{\xi}) := \hat{\mathbf{G}}(\boldsymbol{\xi})\,\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi}) + \hat{\mathbf{f}}(\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi}),\boldsymbol{\xi}) - \hat{\mathbf{b}}_{\mathrm{dc}}, \tag{4.28}$$

is obtained, where

$$\hat{\mathbf{G}}(\boldsymbol{\xi}) = \sum_{i=1}^{d} \mathbf{Q}^{\top}\tilde{\mathbf{G}}_i\mathbf{Q}\xi_i, \tag{4.29}$$

$$\hat{\mathbf{f}}(\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi}),\boldsymbol{\xi}) = \mathbf{Q}^{\top}\mathbf{f}(\mathbf{x}_{\mathrm{dc}}(\boldsymbol{\xi}),\boldsymbol{\xi}), \tag{4.30}$$

$$\hat{\mathbf{b}}_{\mathrm{dc}} = \mathbf{Q}^{\top}\mathbf{b}_{\mathrm{dc}} \tag{4.31}$$

The above steps show that in the course of using PMOR for tracking the variations in of the DC operating point for values of $\boldsymbol{\xi}$ different than the nominal values $\boldsymbol{\xi}_0$, one would solve the reduced system $\hat{\boldsymbol{\Phi}}\left(\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi}),\boldsymbol{\xi}\right) = \mathbf{0}$ for $\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi})$ at those values of $\boldsymbol{\xi}$ and use $\mathbf{Q}$ to map $\hat{\mathbf{x}}_{\mathrm{dc}}$ back to $\mathbf{x}_{\mathrm{dc}}$ and obtain the corresponding DC operating point of the original system, $\mathbf{x}_{\mathrm{dc}}(\boldsymbol{\xi}) = \mathbf{Q}\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi})$. Nonetheless, this simplistic take on the problem would incur computational difficulties that are discussed next.

4.4.4 The Computational Challenge

The above section highlighted two main issues that emerged as essential to applying the moment matching PMOR in the context of DC operating point. The first issue is the computation of the orthogonal basis $\mathbf{Q}$ that will be used in the projection. The second issue arises in the course projecting the nonlinear term in (4.30) in the reduced space. The rest of this section takes a closer look into the requirements needed in the implementation of the second issue, leaving the first issue for Section 5.3.

Solving for the reduced system (4.28) for $\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi})$ is typically done through applying the Newton-Raphson iterative scheme. The core of this process requires the

computation of the reduced nonlinear function and its Jacobian matrix, which are, respectively, obtained from,

$$\hat{\mathbf{f}}(\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi}), \boldsymbol{\xi}) = \mathbf{Q}^{\top}\mathbf{f}(\mathbf{Q}\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi}), \boldsymbol{\xi}) \tag{4.32}$$

and

$$\hat{\mathbf{J}}(\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi}), \boldsymbol{\xi}) = \frac{\partial \hat{\mathbf{f}}(\mathbf{x}_{\mathrm{dc}}(\boldsymbol{\xi}), \boldsymbol{\xi})}{\partial \hat{\mathbf{x}}_{\mathrm{dc}}} = \mathbf{Q}^{\top}\mathbf{J}(\mathbf{Q}\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi}), \boldsymbol{\xi})\mathbf{Q} \tag{4.33}$$

Using the values obtained for $\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi})$, it is possible to recover the values in $\mathbf{x}_{\mathrm{dc}}(\boldsymbol{\xi})$ through the mapping $\mathbf{Q}\hat{\mathbf{x}}_{\mathrm{dc}}(\boldsymbol{\xi})$.

It is obvious from the procedure, as outlined above, that the main advantage of using the proposed approach to track the variation of the DC operating point is having to factorize the reduced Jacobian matrix with size m, instead of the original one which of size N and given that typically $m << N$.

However, the basic computation of the reduced nonlinear $\hat{\mathbf{f}} \in \mathbb{R}^m$ and $\hat{\mathbf{J}} \in \mathbb{R}^{m \times m}$ (as presented in (4.32) and (4.33), respectively) would still require invoking the full circuit device model evaluation routines to compute its contribution to the N components of $\mathbf{f}(\cdot)$ as well as the nonlinear devices stamp on the $N \times N$ original Jacobian matrix $\mathbf{J}(\cdot)$. This, of course, comes in addition to the $\mathcal{O}(mN)$ computational complexity incurred in the linear algebra operations involving the projection $\mathbf{Q}$. This fact represents a significant disadvantage that could curb the computational savings hoped for by entirely moving the computation from $\mathbb{R}^N$ to the $\mathbb{R}^m$ domain.

4.5 Discrete Empirical Interpolation Method (DEIM)

As explained in Sections 4.1–4.2, model-order reduction (MOR), and by extension PMOR, have proven to be an effective tool in reducing the computational cost of

large linear circuits.

However, the application of MOR for nonlinear circuits remains a challenging task. As elaborated in Section 4.4.4 for the case of Nonlinear DC problems, the application of moment matching PMOR suffers two main issues that can easily outweigh any advantage of the reduction.

These issues are extremely important and are the main barriers in the application of projection based reduction even for the general case of nonlinear systems. Hence, when the MNA system of equations contains nonlinear components ($\mathbf{f}(\mathbf{x})$), the computational savings achieved by the projection-based MOR is not as evident. Although the dimension of the reduced nonlinear circuits (4.28) is smaller than the original (unreduced) system (5.1), the computational complexity for the evaluation of the reduced nonlinear function $\hat{\mathbf{f}}(\hat{\mathbf{x}})$ and its Jacobian $\hat{\mathbf{J}}(\hat{\mathbf{x}})$ does not improve. The computation of these reduced functions still requires evaluating $\mathbf{f}(\mathbf{x})$ and its Jacobian $\mathbf{J}(\mathbf{x})$ in the unreduced domain using the original system. It also requires taking $\mathbf{x}$ back and forth from the reduced space to the original space to evaluate $\mathbf{f}(\mathbf{x})$ and $\mathbf{J}(\mathbf{x})$. Once the computational investment has been made to evaluate each individual element of $\mathbf{f}(\mathbf{x})$ and $\mathbf{J}(\mathbf{x})$ only then can they be reduced. This makes the simulation of the reduced-order system inefficient.

There have been several attempts in the literature to circumvent this problem. Historically, the first was to replace (approximate) the nonlinearity by a weighted combination of linear systems which could then be efficiently treated by well-known linear reduction methods. The trajectory piecewise-based techniques in [75,76,99,100] are developed using this approach. For a more detailed insight one can refer to [101] and references therein.

Alternatively, there have been several recent papers attempting to address this issue such as Gappy-POD [102], Missing Point Estimation (MPE) [45,103], Empirical Interpolation Method (EIM) [56, 104], and its recent discrete variant called Discrete

Empirical Interpolation Method (DEIM) [19, 52, 105, 106].

This section will outline the general formulation of DEIM. DEIM aims to achieve an approximation for $\hat{\mathbf{f}}(\hat{\mathbf{x}})$ and $\hat{\mathbf{J}}(\hat{\mathbf{x}})$ based on combining projection and interpolation approaches. To this end, some selected k components of the nonlinear function, denoted $\mathbf{f}_k(\mathbf{x}_k(t))$ ($k \ll n$), are computed and used to approximate the full nonlinear vector.

4.5.1 Reduction of the computational complexity

An orthogonal basis $\mathbf{U} \in \mathbb{R}^{n\times k}$ that spans the subspace of nonlinear function $\mathcal{F}$ is constructed. In the classical DEIM, also known as POD-DEIM, $\boldsymbol{\mathcal{F}}$ is defined by the snapshots of the nonlinear function as

$$\boldsymbol{\mathcal{F}} = [\mathbf{f}(\mathbf{x}(t_1)), \mathbf{f}(\mathbf{x}(t_2)), \ldots, \mathbf{f}(\mathbf{x}(t_N))] \in \mathbb{R}^{n\times N}. \tag{4.34}$$

where $\mathbf{f}(\mathbf{x}(t_i)) \in \mathbb{R}^n$ is the evaluation of the nonlinear vector at time instance t_i and N is the number of time samples. These N samples can be collected, without any extra computational cost, from the initial simulation of the circuit which is required by the conventional POD reduction project operator $\mathbf{Q} \in \mathbb{R}^{n\times m}$ ($m \ll n$).

The orthogonal basis $\mathbf{U}_{\mathcal{F}} \in \mathbb{R}^{n\times k}$ is obtained through a Singular Value Decomposition (SVD) [107] of the nonlinear data matrix $\boldsymbol{\mathcal{F}}$ as,

$$\boldsymbol{\mathcal{F}} = \mathbf{U}_{\mathcal{F}}\boldsymbol{\Sigma}_{\mathcal{F}}\mathbf{W}_{\mathcal{F}}^{\mathsf{T}} \tag{4.35}$$

This results in N left-singular vectors in the orthonormal matrix $\mathbf{U}_{\mathcal{F}} \in \mathbb{R}^{n\times N}$ which are the basis spanning the space of the nonlinear trajectory data, i.e.,

$$\mathbf{U}_{\mathcal{F}} = \text{colspan}(\boldsymbol{\mathcal{F}}) \tag{4.36}$$

and $\mathbf{\Sigma}_F = \text{diag}\,[\sigma_1, \ldots, \sigma_N]$ $\mathbf{R}^{N\times N}$ is the singular-value matrix, where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_N \geq 0$, let $(N < n)$.

The projection operator $\mathbf{U}$ (4.37) is constructed using the first k left-singular vectors in $\mathbf{U}_\mathcal{F}$ corresponding to the k largest singular values in $\mathbf{\Sigma}_F$.

$$\mathbf{U} = \{\mathbf{u}_i \in \mathbf{U}_\mathcal{F} \mid \text{for } i = 1, \ldots, k;\ k \ll n\} \in \mathbb{R}^{n\times k} \tag{4.37}$$

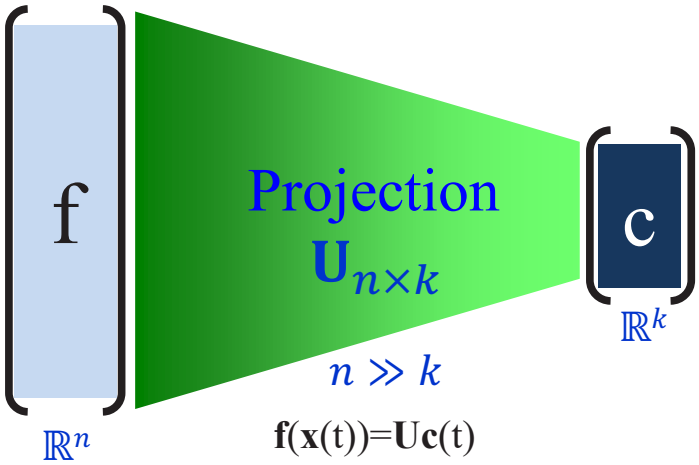


Figure 4.1: The subspace of nonlinear function is projecting onto the subspace spanned by the columns of $\mathbf{U}$.

$$\mathbf{f}(\mathbf{x}(t)) \approx \mathbf{U}\boldsymbol{c}(t) \tag{4.38}$$

A selector matrix $\mathbf{P} \in \mathbb{R}^{n\times k}$ is constructed in Algorithm 2. Using this algorithm, the k rows of the nonlinear function that have the most dominant contribution to the nonlinear space, spanned by the column of $\mathbf{U}$, are captured. Matrix $\mathbf{P}$ is formed as

$$\mathbf{P} \;=\; [\mathbf{e}_{\gamma_1}, \ldots, \mathbf{e}_{\gamma_k}] \in \{0,1\}^{n\times k}. \tag{4.39}$$

where $\mathbf{e}_{\gamma_i}$ is the γ_i-th column of an $n \times n$ identity matrix.

As a result, $\mathbf{f}_k(\mathbf{x}_k)$ from (4.40) includes the most significant k rows (components) in $\mathbf{f}(\mathbf{x})$

$$\mathbf{f}_k(\mathbf{x}_k) := \mathbf{P}^\top \mathbf{f}(\mathbf{x}) \tag{4.40}$$

$$\mathbf{f}_r(\mathbf{x}_r(t)) = \left[f_{\gamma_i}(\mathbf{x}_{f_{\gamma_i}}(t)) \,|\quad \text{for}\;\; \gamma_i \in \{\gamma_1,\, \ldots,\, \gamma_k\} \right] \;\in \mathbb{R}^{k\times 1} \tag{4.41}$$

where $\mathbf{x}_{f_{\gamma_i}}(t) \subseteq \mathbf{x}(t)$. The vector $\mathbf{x}_{f_{\gamma_i}}(t)$ collects the entries in unknown vector $\mathbf{x}(t)$ that are needed for the evaluation of the functions contributing to the γ_i-th row in $\mathbf{f}(t)$. Therefore, the truncated vector of unknowns $\mathbf{x}_k(t)$ can be defined as a collection of those entries in $\mathbf{x}(t)$ that are needed for the evaluation of all the rows in $\mathbf{f}_k$. This can be formalized as

$$\mathbf{x}_k(t) = \mathbf{x}_{f_{\gamma_1}}(t) \bigcup \mathbf{x}_{f_{\gamma_2}}(t) \bigcup \cdots \bigcup \mathbf{x}_{f_{\gamma_k}}(t) = \boldsymbol{\Upsilon}^\top \mathbf{x}(t) \in \mathbb{R}^{\eta} \tag{4.42}$$

where η is the number of the MNA variables in $\mathbf{x}_k(t)$ and $\boldsymbol{\Upsilon} \in \{0,1\}^{n\times\eta}$ is a selector matrix selecting the entries from $\mathbf{x}(t)$ required to compute $\mathbf{f}_k$.

Thus, the approximated nonlinear vector can be represented as below,

$$\mathbf{f}_k(\boldsymbol{\Upsilon}^\top \mathbf{x}(t)) = \mathbf{P}^\top \mathbf{U}\mathbf{c}(t) \tag{4.43}$$

Using (4.38), (4.43), and the projection $\mathbf{Q}$ for reduction (e.g., form POD method), an approximation for the reduced nonlinear function can be obtained as

$$\hat{\mathbf{f}}(\mathbf{x}(t)) \;\approx\; \boldsymbol{\Psi}\mathbf{f}_k(\boldsymbol{\Upsilon}^\top \mathbf{Q}\hat{\mathbf{x}}) \tag{4.44}$$

where $\mathbf{\Psi}$ is defined as,

$$\mathbf{\Psi} \;=\; \mathbf{Q}^{\top}\mathbf{U}(\mathbf{P}^{\top}\mathbf{U})^{-1} \in \mathbb{R}^{m\times k} \tag{4.45}$$

A similar technique can also be used to approximate the Jacobian as seen in Equation (4.46).

$$\hat{\mathbf{J}}(\hat{\mathbf{x}}) \approx \;\mathbf{\Psi}\mathbf{J}_k(\mathbf{x}_k)\mathbf{\Upsilon}^{\top}\mathbf{Q} \;\; \in \mathbb{R}^{m\times m} \tag{4.46}$$

where

$$\mathbf{J}_k(\mathbf{x}_k) \triangleq \frac{\partial \mathbf{f}_k(\mathbf{x}_k)}{\partial \mathbf{x}_k} = \mathbf{P}^{\top}\mathbf{J}(\mathbf{x})\mathbf{\Upsilon} \;\; \in \mathbb{R}^{k\times\gamma}. \tag{4.47}$$

Evidently, the evaluation of the reduced nonlinear function and its Jacobian are performed in the reduced domain. This reduces the computational complexity for the evaluation of $\mathbf{f}(\mathbf{x})$ from $\mathcal{O}(n)$ to $\mathcal{O}(k)$ where $(k \ll n)$. This translates to a significant reduction of CPU-cost.

Algorithm 2: Computation of the selector matrix $\mathbf{P}$.

Input: $\boldsymbol{\mathcal{F}} \in \mathbb{R}^{n\times N}$, (Nonlinear data matrix)
m, (The prescribed reduction order)
K, (The order for interpolation space)
Output: $\mathbf{U} \in \mathbb{R}^{n\times k}$, (Interpolation projector)
$\mathbf{P} \in \{0,1\}^{n\times k}$, (Row selector matrix)
$\boldsymbol{\Gamma} \in \mathbb{N}^{k}$ (Matrix of interpolation indices)

1: $(\mathbf{u}_i,\ \sigma_i) \leftarrow svd(\boldsymbol{\mathcal{F}})$; for $i = 1, \ldots, m$
2: $[\rho, \gamma_1] = max(|\mathbf{u}_1|)$; where $\rho \leftarrow |\mathbf{u}_1|_\infty$ and $\gamma_1 \leftarrow$ Row# of ρ
3: $\mathbf{I}_n \leftarrow eye(n, n)$; (Identity matrix of order n)
4: $\mathbf{U} \leftarrow \mathbf{u}_1$;
5: $\mathbf{e}_{\gamma_1} \leftarrow \mathbf{I}_n(:, \gamma_1)$; ($\gamma_1$-th column in identity matrix)
6: $\mathbf{P} \leftarrow \mathbf{e}_{\gamma_1}$;
7: $\boldsymbol{\Gamma} \leftarrow \gamma_1$;
8: $k \leftarrow 1$;
9: **while** $k < K$ **do**
10: $k \leftarrow k + 1$;
11: $Solve\ (\mathbf{P}^t\mathbf{U})\mathbf{c} = \mathbf{P}^t\mathbf{u}_k\ for\ \mathbf{c}$;
12: $\mathbf{r} = \mathbf{u}_k - \mathbf{U}\mathbf{c}$;
13: $[\rho, \gamma_k] = max\{|\mathbf{r}|\}$;
14: $\mathbf{U} \leftarrow [\mathbf{U}, \mathbf{u}_k]$;
15: $\mathbf{P} \leftarrow [\mathbf{P}, \mathbf{e}_{\gamma_k}]$;
16: $\boldsymbol{\Gamma} \leftarrow [\boldsymbol{\Gamma}, \gamma_k]$;
17: **end while**
18: **return** $\mathbf{U}$, $\mathbf{P}$, $\boldsymbol{\Gamma}$;

4.6 Summary and Discussions

The main objective in this chapter has been to outline the concepts of MOR and its application in the context of parameterized MOR (PMOR). The underlying goal of this presentation has been the creation of reduced system of nonlinear equations whose solution tracks the variations of the DC solution of the circuit with respect to variation in some design parameters. The approach used to achieve this goal relied on extending the concepts that are used in linear PMOR to the nonlinear domain. Although the procedural steps are well-defined and succeed in creating a reduced system with the desired characteristics, the development of those steps in the chapter exposed several issues and drawbacks that need to be addressed. More specifically, the contribution of this thesis is with regards to the following issues:

1. Computation of the orthogonal basis $\mathbf{Q}$ that is used in the projection of the original system onto the reduced space. For the case of linear circuits, this issue is handled through a well established approach that was described in Section 4.3. This approach, however, cannot be used for nonlinear circuits. The resolution of this issue is part of the contribution in this thesis and will be presented in the next chapter.

2. Projecting the nonlinear part of the DC nonlinear equations, $\mathbf{f}(\mathbf{x})$ on the reduced space. The development in this chapter demonstrated that the projection is a bottleneck in itself, since unlike the linear case, the projection needs to be repeated each time the reduced system needs to be solved. The DEIM framework which was presented in Section 4.5 was proposed to solve that projection problem but in different domains, e.g. aerodynamics and fluid mechanics [20,108,109]. DEIM, however, as it is implemented in those domains, requires using the POD via snapshots of $\mathbf{f}(\mathbf{x})$. This is not suitable in the context of DC problems since it forces solving the original DC problem in order to generate the

snapshots. The contributions of this thesis addresses this issue in using DEIM in the projection operation.

The following chapter presents the contributions of the thesis which handle the above-mentioned issues.

Chapter 5

DC-Centric Parameterized Reduced-Order Model via Moment-based Interpolation Projection (MIP) Algorithm

This chapter presents the main contributions of the thesis. The contribution introduced by this thesis targets the application of PMOR in nonlinear circuits as a computationally efficient tool to track the variations in the DC operating point of the circuit in response to variations in a selected set of design parameters.

The proposed approach uses the idea of direct moment matching to construct a reduced system that is used in tracking the DC solution of the original system.

The presentation style used in this chapter is top-to-bottom in the sense that the main ideas are first presented. Through this presentation, key building blocks are identified and detailed computational steps are presented next.

5.1 Proposed PMOR Approach Using Moment Matching

This section describes the main outline of the proposed approach. Through the developments presented in this section, the essential procedural steps involved will be highlighted. Further details on the implementation of those steps will be handled in the following sections.

The mathematical formulation for a general nonlinear circuit with d design parameters $\xi_i, i = 1, \cdots, d$ is typically obtained through the modified nodal (MNA) analysis approach [36]. Using MNA yields a system of nonlinear algebraic equations of the form

$$\mathbf{\Phi}(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}) := \mathbf{G}(\boldsymbol{\xi})\mathbf{x}(\boldsymbol{\xi}) + \mathbf{f}(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}) - \mathbf{b} \tag{5.1}$$

where $\mathbf{G}(\boldsymbol{\xi}) \in \mathbb{R}^{N\times N}$ is a matrix that carries the stamps of memoryless linear elements, $\mathbf{f}(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}) \in \mathbb{R}^N$ is a vector of N nonlinear functions that describe the nonlinear elements. $\mathbf{b} \in \mathbb{R}^N$ is a vector containing the quiescent (DC) sources in the circuit, N is the number of circuit variables, and $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_d]^\top$.

The task of finding the DC operating point for a specific (nominal) set of parameters, say $\boldsymbol{\xi} = \boldsymbol{\xi}_0$, is done by using an iterative technique such as the Newton-Raphson to solve $\mathbf{\Phi}(\mathbf{x}(\boldsymbol{\xi}_0), \boldsymbol{\xi}_0) = \mathbf{0}$.

5.1.1 Moment Matching PMOR for DC Operating Point

PMOR based on the concept of moment matching has been applied for linear circuits for purposes analogous to the DC operating point (e.g., frequency response) [8–12]. For nonlinear circuits, however, using PMOR through moment matching on the original nonlinear system of equations that models the circuit has not been addressed

before. The proposed approach handles the issues involved in this objective and uses the resulting algorithm to efficiently track the variations in the DC operating point in response to variations in key design parameters.

The proposed approach proceeds conceptually in the following four main steps,

1. Expand $\mathbf{x}(\boldsymbol{\xi})$ in Taylor series around $\boldsymbol{\xi}_0$. Typically, representing a multidimensional Taylor expansion of $\mathbf{x}(\boldsymbol{\xi})$ is better served by using the notion of multi-index, which defines a multi-index, say $\boldsymbol{\alpha}$, of size d as a vector with non-negative elements, i.e., $\boldsymbol{\alpha} \in \mathbb{N}^d$. Thus, a Taylor expansion for $\mathbf{x}(\boldsymbol{\xi})$, is represented by

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\alpha}} \mathbf{M}_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_0) \prod_{i=1}^{d} (\xi_i - \xi_{i,0})^{\alpha_i} \tag{5.2}$$

   where $\mathbf{M}_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_0)$ are the Taylor series coefficients, which are also known as the moments of the expansion,

2. Truncate the Taylor expansion to include only $\boldsymbol{\alpha}$ in a set, denoted Λ, defined according a certain criterion. For example, using the notion of total order, Λ is given by $\Lambda := \{\boldsymbol{\alpha} : \Sigma_i \alpha_i < p\}$ for some integer p. In the following we shall assume that Λ includes m elements ($|\Lambda| = m$) with $m << N$.

3. Forming an orthonormal basis $\mathbf{Q} \in \mathbb{R}^{N \times m}$ for the subspace spanned by the set of moments collected from the truncation,

$$\mathbf{Q} = \text{colspan}\left[\mathbf{M}_{\boldsymbol{\alpha}}(\boldsymbol{\xi}_0)\right], \quad \boldsymbol{\alpha} \in \Lambda \tag{5.3}$$

4. Finally, using an affine representation for $\mathbf{G}(\boldsymbol{\xi})$, as proposed in [10],

$$\mathbf{G}(\boldsymbol{\xi}) = \sum_{i=1}^{d} \tilde{\mathbf{G}}_i \xi_i, \tag{5.4}$$

premultiplying (5.1) by $\mathbf{Q}^\top$ and through the change of variables,

$$\mathbf{x}(\boldsymbol{\xi}) = \mathbf{Q}\hat{\mathbf{x}}(\boldsymbol{\xi}) \tag{5.5}$$

a reduced system of the form

$$\hat{\boldsymbol{\Phi}}(\hat{\mathbf{x}}(\boldsymbol{\xi}), \boldsymbol{\xi}) := \hat{\mathbf{G}}(\boldsymbol{\xi})\,\hat{\mathbf{x}}(\boldsymbol{\xi}) + \hat{\mathbf{f}}(\hat{\mathbf{x}}(\boldsymbol{\xi}), \boldsymbol{\xi}) - \hat{\mathbf{b}}, \tag{5.6}$$

is obtained, where

$$\hat{\mathbf{G}}(\boldsymbol{\xi}) = \sum_{i=1}^{d} \mathbf{Q}^\top \tilde{\mathbf{G}}_i \mathbf{Q} \xi_i, \tag{5.7}$$

$$\hat{\mathbf{f}}(\hat{\mathbf{x}}(\boldsymbol{\xi}), \boldsymbol{\xi}) = \mathbf{Q}^\top \mathbf{f}(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}), \tag{5.8}$$

$$\hat{\mathbf{b}} = \mathbf{Q}^\top \mathbf{b} \tag{5.9}$$

The above steps show that in the course of using PMOR for tracking the variations in of the DC operating point for values of $\boldsymbol{\xi}$ different than the nominal values $\boldsymbol{\xi}_0$, one would solve the reduced system $\hat{\boldsymbol{\Phi}}\left(\hat{\mathbf{x}}(\boldsymbol{\xi}), \boldsymbol{\xi}\right) = \mathbf{0}$ for $\hat{\mathbf{x}}(\boldsymbol{\xi})$ at those values of $\boldsymbol{\xi}$ and use $\mathbf{Q}$ to map $\hat{\mathbf{x}}$ back to $\mathbf{x}$ and obtain the corresponding DC operating point of the original system, $\mathbf{x}(\boldsymbol{\xi}) = \mathbf{Q}\hat{\mathbf{x}}(\boldsymbol{\xi})$. Nonetheless, this simplistic take on the problem would incur computational difficulties that are discussed next.

5.1.2 The Main Computations

The presentation of the previous subsection highlighted two main issues that emerged as essential to applying the moment matching PMOR in the context of DC operating point. The first issue is the computation of the orthogonal basis $\mathbf{Q}$ that will be used in the projection. The second issue arises in the course projecting the nonlinear term in (5.8) in the reduced space. The rest of this section takes a closer look into the

requirements needed in the implementation of the second issue, leaving the first issue for Section 5.3.

Solving for the reduced system (5.6) for $\hat{\mathbf{x}}(\boldsymbol{\xi})$ is typically done through applying the Newton-Raphson iterative scheme. The core of this process requires the computation of the reduced nonlinear function and its Jacobian matrix, which are, respectively, obtained from,

$$\hat{\mathbf{f}}(\hat{\mathbf{x}}(\boldsymbol{\xi}), \boldsymbol{\xi}) = \mathbf{Q}^{\top}\mathbf{f}(\mathbf{Q}\hat{\mathbf{x}}(\boldsymbol{\xi}), \boldsymbol{\xi}) \tag{5.10}$$

and

$$\hat{\mathbf{J}}(\hat{\mathbf{x}}(\boldsymbol{\xi}), \boldsymbol{\xi}) = \frac{\partial \hat{\mathbf{f}}(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi})}{\partial \hat{\mathbf{x}}} = \mathbf{Q}^{\top}\mathbf{J}(\mathbf{Q}\hat{\mathbf{x}}(\boldsymbol{\xi}), \boldsymbol{\xi})\mathbf{Q} \tag{5.11}$$

Using the values obtained for $\hat{\mathbf{x}}(\boldsymbol{\xi})$, it is possible to recover the values in $\mathbf{x}(\boldsymbol{\xi})$ through the mapping $\mathbf{Q}\hat{\mathbf{x}}(\boldsymbol{\xi})$.

It is obvious from the procedure, as outlined above, that the main advantage of using the proposed approach to track the variation of the DC operating point, is having to factorize the reduced Jacobian matrix with size m, instead of the original one which of size N and given that typically $m << N$.

However, the basic computation of the reduced nonlinear $\hat{\mathbf{f}} \in \mathbb{R}^m$ and $\hat{\mathbf{J}} \in \mathbb{R}^{m \times m}$ (as presented in (5.10) and (5.11), respectively) would still require invoking the full circuit device model evaluation routines to compute the contribution to the N components of $\mathbf{f}(\cdot)$ as well as the nonlinear devices stamp on the $N \times N$ original Jacobian matrix $\mathbf{J}(\cdot)$. This, of course, comes in addition to the $\mathcal{O}(mN)$ computational complexity incurred in the linear algebra operations involving the projection $\mathbf{Q}$. This fact represents a significant disadvantage that could curb the computational savings hoped for by entirely moving the computation from $\mathbb{R}^N$ to the $\mathbb{R}^m$ domain.

The difficulty described above has already been elaborated upon in Section 4.4.3 and is a result of trying to apply the projection on $\mathbf{f}(\mathbf{x})$, the nonlinear term. The

idea of DEIM, briefly outlined in section 4.5, is meant to solve this difficulty. The next section revisits this idea in the current context illustrating how it can be used to alleviate the computational difficulty of projecting $\mathbf{f}(\mathbf{x})$.

5.2 Proposed MIP-Based Reduction

This section addresses the issue of alternating in computation between the original domain $\mathbb{R}^N$ and the reduced domain $\mathbb{R}^m$ with each point of $\boldsymbol{\xi}$. The approach presented in this regard is termed as the moment-based interpolation, or MIP, because it relies on using an interpolatory matrix operator that is used by the DEIM framework. However, it departs from the DEIM approach in the way in which the operator U is constructed.

Similar to DEIM, MIP is premised on the idea that $\mathbf{f}\left(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}\right)$ can be approximated with another vector $\tilde{\mathbf{f}}\left(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}\right) \in \mathbb{R}^N$ (same size), which is obtained as an interpolation of k entries selected from the original $\mathbf{f}\left(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}\right)$, with $k << N$. This notion is formulated [19, 105] more succinctly by defining

$$\tilde{\mathbf{f}}\left(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}\right) := \mathbf{Y}\mathbf{f}_k\left(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}\right) \tag{5.12}$$

where,

- $\mathbf{f}_k\left(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}\right)$ is a set of k entries selected from, $\mathbf{f}\left(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}\right)$ and best described using a selector matrix $\mathbf{P} \in \{0, 1\}^{N \times k}$,

$$\mathbf{f}_k\left(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}\right) = \mathbf{P}^\top \mathbf{f}\left(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}\right) \tag{5.13}$$

  with $\mathbf{P}$ being a matrix whose k columns are defined as a set of k columns selected

from an $N \times N$ identity matrix, i.e.,

$$\mathbf{P} = [\mathbf{e}_{\gamma_1}, \cdots, \mathbf{e}_{\gamma_k}] \tag{5.14}$$

with $\gamma_i \in \mathbb{N}, 1 \leq \gamma_i \leq N$ and $\mathbf{e}_{\gamma_i}$ being the γ_i^{th} column of an $N \times N$ identity matrix.

- $\mathbf{Y} \in \mathbb{R}^{N \times k}$ is defined in terms of $\mathbf{P}$ and another *orthonormal* matrix $\mathbf{U}$

$$\mathbf{Y} = \mathbf{U}\left(\mathbf{P}^\top \mathbf{U}\right)^{-1} \tag{5.15}$$

where it is assumed that $\mathbf{P}^\top \mathbf{U}$ is invertible.

It should be obvious from (5.12) that the N entries of $\tilde{\mathbf{f}}(\cdot, \cdot)$ are interpolations of the k selected entries in $\mathbf{f}(\cdot, \cdot)$. Furthermore, this interpolation becomes exact for those k entries, in the sense that those entries in the former coincide with the same entries in the latter, a fact that can be easily seen from,

$$\mathbf{P}^\top \tilde{\mathbf{f}}(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}) = \mathbf{P}^\top \mathbf{f}(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}) \tag{5.16}$$

The interpolatory matrix operator in (5.12) then invites two issues to the forefront of the present development. The first issue is relevant to resolving problem of alternating in computation between the reduced $\mathbb{R}^m$ and the full original model space in $\mathbb{R}^N$. This issue is addressed by Section 5.2.1.

The second issue is concerned with constructing the selector matrix $\mathbf{P}$ and the orthonormal basis $\mathbf{U}$. The framework used for this purpose is predicated on the idea that for any orthonormal matrix $\mathbf{U} \in \mathbb{R}^{N \times k}$, there is a selector matrix $\mathbf{P} \in \{0, 1\}^{N \times k}$ that can be constructed such that the error in $\left|\left|\mathbf{f}\left(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}\right) - \tilde{\mathbf{f}}\left(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}\right)\right|\right|$ is minimized [19]. The algorithm used to construct $\mathbf{P}$ given an arbitrary matrix $\mathbf{U}$ is considered

later in Section 5.3.2 based on [19].

On the other hand, the choice of $\mathbf{U}$ is typically made based on empirical grounds, where it was observed that, for best results, the matrix $\mathbf{U}$ needs to span the subspace spanned by $\mathbf{f}$. In other areas of applications where the interpolatory operator has been used, e.g. aerodynamics and fluid mechanics [20, 108, 109], $\mathbf{U}$ was obtained through simulating the full nonlinear system that models the underlying problem domain and using the results of the simulation to create a snapshot of the nonlinear part of the model, which is analogous to $\mathbf{f}(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi})$ in the present work. This process is repeated many times to collect more snapshots that cover as much as possible the subspace spanned by the nonlinear part. Upon termination, a singular value decomposition (SVD) is run on the snapshots to extract the subspace that corresponds to the k largest singular values and use that to construct the orthonormal basis $\mathbf{U}$.

In the present application of tracking the DC operating point such a scenario is not plausible for several reasons, foremost among them is the fact that number of snapshots required to cover the d-dimensional space spanned by the design parameters $\boldsymbol{\xi}$ grows exponentially with the number of parameters d. This fact makes the snapshot-based approach, even if it is possible to achieve, stand to erode the computational gains anticipated in using the PMOR approach since it necessitates computing the DC operating point at large number of points within the parameters space. In addition, the concept of snapshots defeats the very purpose of using a moment matching approach to PMOR. This issue is addressed in Section 5.2.2 through proposing a new method to construct $\mathbf{U}$ based on utilizing the concept of rooted trees.

5.2.1 Efficient Projection using MIP

This section illustrates the role that using $\tilde{\mathbf{f}}(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi})$ instead of $\mathbf{f}(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi})$ in (5.1) plays in eliminating the problem of alternating in computation between the original domain $\mathbb{R}^N$ and the reduced domain $\mathbb{R}^m$. To this end, and to make this illustration

lucid, the following assumptions will be temporarily made,

- $\mathbf{f}\left(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}\right)$ has no explicit dependence on $\boldsymbol{\xi}$. Therefore, henceforth, $\mathbf{f}\left(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi}\right)$ will be replaced by $\mathbf{f}\left(\mathbf{x}(\boldsymbol{\xi})\right)$.
- The N nonlinear functions in $\mathbf{f}\left(\mathbf{x}(\boldsymbol{\xi})\right)$ are only component-wise functions of $\mathbf{x}$. Thus, $\mathbf{f}\left(\mathbf{x}(\boldsymbol{\xi})\right)$ can be captured by,

$$\mathbf{f}(\mathbf{x}(\boldsymbol{\xi})) = \left[f_1(x_1(\boldsymbol{\xi})), \cdots, f_N(x_N(\boldsymbol{\xi}))\right]^\top. \tag{5.17}$$

Next, replace $\mathbf{f}(\mathbf{x}(\boldsymbol{\xi}))$ with $\tilde{\mathbf{f}}(\mathbf{x}(\boldsymbol{\xi}))$ in (5.1), and $\mathbf{x}$ with $\mathbf{Q}\hat{\mathbf{x}}$, pre-multiplying both sides by $\mathbf{Q}^\top$ to obtain the new expression for $\hat{\mathbf{f}}\left(\hat{\mathbf{x}}(\boldsymbol{\xi})\right)$ (different than the one in (5.10)) that is given by

$$\begin{aligned} \hat{\mathbf{f}}(\hat{\mathbf{x}}(\boldsymbol{\xi})) &= \mathbf{Q}^\top\, \tilde{\mathbf{f}}(\mathbf{x}(\boldsymbol{\xi})) \\ &= \boldsymbol{\Psi}\, \boldsymbol{\Xi}(\hat{\mathbf{x}}(\boldsymbol{\xi})) \end{aligned} \tag{5.18}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{m\times k}$ is a constant matrix given by

$$\boldsymbol{\Psi} = \mathbf{Q}^\top \mathbf{Y} \tag{5.19}$$

and $\boldsymbol{\Xi}(\hat{\mathbf{x}}(\boldsymbol{\xi})) \in \mathbb{R}^k$ is given by

$$\boldsymbol{\Xi}(\hat{\mathbf{x}}(\boldsymbol{\xi})) = \mathbf{f}_k(\mathbf{Q}_k \hat{\mathbf{x}}(\boldsymbol{\xi})) \tag{5.20}$$

with $\mathbf{f}_k \in \mathbb{R}^k$ and $\mathbf{Q}_k \in \mathbb{R}^{k\times m}$ being, respectively, the k elements of $\mathbf{f}(\cdot)$ and k rows of $\mathbf{Q}$ selected by $\mathbf{P}^\top$.

It is obvious from (5.18) that computing the reduced nonlinear function $\hat{\mathbf{f}}(\hat{\mathbf{x}}(\boldsymbol{\xi}))$ of (5.18) can be carried out entirely in a reduced-domain $\mathbb{R}^k$, since $\boldsymbol{\Psi}$ is a constant $m \times k$

matrix, and evaluation of $\boldsymbol{\Xi}(\hat{\mathbf{x}}(\boldsymbol{\xi}))$ only requires using a constant matrix $\mathbf{Q}_k \in \mathbb{R}^{k\times m}$ obtained by *selecting* k rows from the matrix $\mathbf{Q}$ and evaluation of k components of the nonlinear function. Thus, utilizing the selector matrix $\mathbf{P}$ of (5.14) in (5.18) is the key idea that enables one to avoid computations proportional to $\mathcal{O}(N)$.

Similar arguments can be made to show that computing the reduced Jacobian matrix (5.11) can also be entirely performed in a reduced domain using (5.18) as evidenced by,

$$\hat{\mathbf{J}}\left(\hat{\mathbf{x}}(\boldsymbol{\xi})\right) = \frac{\partial \hat{\mathbf{f}}(\hat{\mathbf{x}}((\xi)))}{\partial \hat{\mathbf{x}}} = \underbrace{\boldsymbol{\Psi}}_{m\times k} \quad \underbrace{\frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_k}}_{k\times k} \quad \underbrace{\mathbf{Q}_k}_{k\times m}, \tag{5.21}$$

where $\mathbf{x}_k = \mathbf{P}^\top \mathbf{x}$ is the k components of $\mathbf{x}$ selected by $\mathbf{P}^\top$. Therefore, computing the reduced Jacobian can be executed while avoiding the type of $\mathcal{O}(N)$ computational complexity that arises from direct projection.

5.2.2 Orthonormal Basis U Based on Moment Matching

Computing $\mathbf{U}$ in the present work is based on utilizing the concept of the rooted trees to compute the moments of $\mathbf{f}(\mathbf{x}(\boldsymbol{\xi}))$. To simplify the mathematical illustration of the basic ideas involved in this process, we will consider, without any loss of generality, the case of a single design parameter ξ ($d = 1$). Proceeding forward, we posit that $\mathbf{f}(\mathbf{x}(\xi))$ varies smoothly enough in a neighbourhood around the nominal design point ξ_0 to allow a Taylor series expansion of $\mathbf{f}(\mathbf{x}(\xi))$ at $\xi = \xi_0$,

$$\mathbf{f}(\mathbf{x}(\xi)) = \sum_{i=0}^{\infty} \mathbf{F}_i(\xi_0)\ (\xi - \xi_0)^i \tag{5.22}$$

where, $\mathbf{F}_i(\xi_0)$ are referred to as the moments of $\mathbf{f}(\mathbf{x}(\xi))$ and are given by

$$\mathbf{F}_i(\xi_0) = \frac{1}{i!}\ \frac{\mathrm{d}^i \mathbf{f}(\mathbf{x}(\xi))}{\mathrm{d}\xi^i}\bigg|_{\xi=\xi_0} \tag{5.23}$$

The matrix $\mathbf{U}$ is then taken to span the column span of the first k moments of $\mathbf{f}(\mathbf{x}(\xi))$, i.e.,

$$\mathbf{U} = \text{colspan}\left[\begin{array}{cccc} \mathbf{F}_0(\xi_0) & \mathbf{F}_1(\xi_0) & \cdots & \mathbf{F}_{k-1}(\xi_0) \end{array}\right] \tag{5.24}$$

In computing $\mathbf{F}_i(\xi_0)$, one should note that since $\mathbf{f}(\mathbf{x}(\xi))$ depend on $\mathbf{x}(\xi)$, then its moments ($\mathbf{F}_1(\xi_0)$) will implicitly be determined by the moments of $\mathbf{x}(\xi)$. The moments of $\mathbf{x}(\xi)$ were introduced earlier in Section 5.1 in the general Taylor expansion of $\mathbf{x}(\xi)$ of (5.2), which for the case of a single design parameter used here for illustration simplifies to,

$$\mathbf{x}(\xi) = \sum_{i=0}^{\infty} \mathbf{M}_i(\xi_0)\,(\xi - \xi_0)^i \tag{5.25}$$

The implicit dependence of $\mathbf{F}_i(\xi_0)$ on $\mathbf{M}_i(\xi_0)$ suggests that in order to compute $\mathbf{F}_i(\xi_0)$, $\mathbf{M}_i(\xi_0)$ need to be computed first. On the other hand, (5.3) shows that $\mathbf{M}_i(\xi_0)$ are also needed to compute $\mathbf{Q}$. Therefore, both matrices $\mathbf{U}$ and $\mathbf{Q}$ are obtained by first computing the moments $\mathbf{M}_i(\xi_0)$. The process of computing $\mathbf{M}_i(\xi_0)$ is based on the notion of rooted tree and will be detailed in the next section.

5.3 Computing the Moments Using Rooted Trees

The objective in this section is to compute the moments of $\mathbf{x}(\xi)$, or $\mathbf{M}_i(\xi_0)$ and use that to construct the matrices $\mathbf{U}$ used for the iterpolatory operator and $\mathbf{Q}$ used in the projection operations described in Section 5.2.1. Computing the matrix $\mathbf{U}$ then enables computing the matrix $\mathbf{P}$, thereby completing the projection process.

5.3.1 Computing the Moments $\mathbf{M}_i(\xi_0)$ and $\mathbf{F}_i(\xi_0)$

Computing the moments $\mathbf{M}_i(\xi_0)$ is carried out first through differentiation of (5.1) with respect to ξ, and the substitution of $\mathbf{x}(\xi)$ from (5.25). This process yields,

$$\left(\mathbf{G}(\xi) + \mathbf{J}(\mathbf{x}(\xi))\right) \sum_{i=1} i\mathbf{M}_i(\xi_0)(\xi - \xi_0)^{i-1} + \frac{\mathrm{d}\mathbf{G}(\xi)}{\mathrm{d}\xi} \sum_{i=0} \mathbf{M}_i(\xi_0)(\xi - \xi_0)^i = 0 \quad (5.26)$$

where $\mathbf{J}(\mathbf{x}(\xi)) = \partial\mathbf{f}(\mathbf{x}(\xi))/\partial\mathbf{x}$ is the Jacobian matrix of the full circuit model.

Expanding $\mathbf{J}(\mathbf{x}(\xi))$ and $\mathbf{G}(\xi)$ as a Taylor series around $\xi = \xi_0$,

$$\mathbf{J}(\mathbf{x}(\xi)) = \sum_{i=0} \mathbf{J}_i(\xi_0)\,(\xi - \xi_0)^i \quad (5.27)$$

$$\mathbf{G}(\xi) = \sum_{i=0} \mathbf{G}_i(\xi_0)\,(\xi - \xi_0)^i \quad (5.28)$$

with $\mathbf{G}_k(\xi_0)$ and $\mathbf{J}_k(\xi_0)$ being the k^{th} (matrix-valued) Taylor series coefficient of $\mathbf{G}(\xi)$ and $\mathbf{J}(\mathbf{x}(\xi))$, respectively, when expanded around $\xi = \xi_0$. Define

$$\tilde{\mathbf{J}}_k(\xi_0) = \mathbf{G}_k(\xi_0) + \mathbf{J}_k(\xi_0) \quad (5.29)$$

In the following, we shall use $\tilde{\mathbf{J}}_k$, $\mathbf{G}_k$ and $\mathbf{M}_k$ in place of $\tilde{\mathbf{J}}_k(\xi_0)$, $\mathbf{G}_k(\xi_0)$ and $\mathbf{M}_k(\xi_0)$, respectively, to make the mathematical analysis terser.

Substitute from (5.27), (5.28) and (5.29) in (5.26)

$$\sum_{i=1}\sum_{j=0} i\tilde{\mathbf{J}}_j\mathbf{M}_i\,(\xi - \xi_0)^{i+j-1} = -\sum_{i=0}\sum_{j=1} j\mathbf{G}_j\mathbf{M}_i\,(\xi - \xi_0)^{i+j-1} \quad (5.30)$$

If we differentiate both sides with respect to ξ for p times, we get

$$\sum_{i=1}\sum_{j=0} i(i+j-1)(i+j-2)\cdots(i+j-p)\tilde{\mathbf{J}}_j\mathbf{M}_i\,(\xi - \xi_0)^{i+j-1-p}$$

$$= -\sum_{i=0}\sum_{j=1} j(i+j-1)(i+j-2)\cdots(i+j-p)\mathbf{G}_j\mathbf{M}_i\left(\xi-\xi_0\right)^{i+j-1-p} \quad (5.31)$$

The above equation is valid for all ξ. Thus setting $\xi = \xi_0$, all terms for which $i+j-1-p>0$ will vanish leaving only the terms for which

$$i+j-1-p=0 \quad (5.32)$$

On the left side of (5.31), this will lead to indices given by

$$\begin{aligned} i &= 1,2,\cdots,p+1 \\ j &= p,p-1,\cdots,0 \end{aligned} \quad (5.33)$$

On the right-side of (5.31), the indices i and j are given by

$$\begin{aligned} i &= 0,1,\cdots,p \\ j &= p+1,p,\cdots,1 \end{aligned} \quad (5.34)$$

The above two relations show that on both the left- and right-side, i and j are related by $j=p-i+1$, which upon substitution into (5.31) yields

$$\sum_{i=1}^{p+1} i\underbrace{(p)(p-1)\cdots(1)}_{p!}\tilde{\mathbf{J}}_{p-i+1}\mathbf{M}_i = \\ -\sum_{i=0}^{p}(p-i+1)\underbrace{(p)(p-1)\cdots(1)}_{p!}\mathbf{G}_{p-i+1}\mathbf{M}_i \quad (5.35)$$

Rearranging, we get

$$(p+1)\tilde{\mathbf{J}}_0\mathbf{M}_{p+1}+\sum_{i=1}^{p} i\tilde{\mathbf{J}}_{p-i+1}\mathbf{M}_i =$$

$$- (p+1)\mathbf{G}_{p+1}\mathbf{M}_0 - \sum_{i=1}^{p}(p-i+1)\mathbf{G}_{p-i+1}\mathbf{M}_i \quad (5.36)$$

Therefore,

$$\tilde{\mathbf{J}}_0\mathbf{M}_{p+1} = -\sum_{i=1}^{p}\left(\frac{i}{p+1}\tilde{\mathbf{J}}_{p-i+1} + \frac{(p-i+1)}{p+1}\mathbf{G}_{p-i+1}\right)\mathbf{M}_i - \mathbf{G}_{p+1}\mathbf{M}_0 \quad (5.37)$$

It is obvious that (5.37) provides a recursive formula to compute $\mathbf{M}_{p+1}(\xi_0)$ from $\mathbf{M}_k(\xi_0)$, $k \leq p$. It should be also clear that the main computational efforts in computing $\mathbf{M}_i(\xi_0)$ is a result of having to run the LU factorization of the Jacobian matrix $\tilde{\mathbf{J}}_0$ and use several forward/backward substitutions with different right-side vectors. However, given that the matrix $\tilde{\mathbf{J}}_0$ is the same matrix used in solving for the DC operating point at the nominal values for the design parameters, $\boldsymbol{\xi}$, its LU factors will be readily available upon convergence to the nominal DC operating point. Hence, the only extra computational cost required to compute $\mathbf{M}_i(\xi_0)$ is mainly the forward/backward substitutions, in addition to computing the Taylor coefficient matrices $\tilde{\mathbf{J}}_m$, which is discussed next.

Clearly $\mathbf{J}(\mathbf{x}(\xi))$ is a function of $\mathbf{x}$, and therefore $\mathbf{J}_i(\xi_0)$ is also a function of $\mathbf{M}_i(\xi_0)$. In fact, it can be shown that $\mathbf{J}_k(\xi_0)$ is a function of all $\mathbf{M}_p(\xi_0)$, for $p \leq k$ [110]. Handling the process of computing $\mathbf{J}_i(\xi_0)$ based on $\mathbf{M}_i(\xi_0)$ has been introduced in the literature and carried out through the notion of rooted trees [55,110]. This idea is used to represent each nonlinear function entry in the matrix $\mathbf{J}(\mathbf{x}(\xi))$ as a tree, with the root node of the tree representing the expression of the nonlinearity itself and the leaf nodes representing the values in $\mathbf{M}_i(\xi_0)$ or the constants in the expression of the nonlinear function. Other intermediate nodes in the tree represent atomic nonlinear functions such as the exponential function or arithmetic operators such multiplication. The process of computing the value of an entry in $\mathbf{J}_i(\xi_0)$ is done by traversing the rooted tree in the upward direction starting with the root node, with

each node simply triggering the node the next level. When a leaf node is reached, it sends back to the nodes at the lower level its assigned value, which could be a value in the vector $\mathbf{M}_i(\xi_0)$, computed earlier, or the value of the constant assigned to this leaf node. The tree is next traversed in the downward direction with each node computing a pre-specified recursive expression with inputs taken from the nodes at the upper level in the tree. The process is continued downwards, and upon arriving back at the root node, yields the required value in $\mathbf{J}_i(\xi_0)$.

The above process for computing $\mathbf{M}_i(\xi_0)$ can be used to compute $\mathbf{F}_i(\xi_0)$. Similar to $\mathbf{J}_k(\xi_0)$, $\mathbf{F}_k(\xi_0)$ is a function of all $\mathbf{M}_p(\xi_0)$ for all $p \leq k$. This makes $\mathbf{F}_i(\xi_0)$ function of $\mathbf{M}_i(\xi_0)$. In a similar manner, rooted trees can be used to compute $\mathbf{F}_i(\xi_0)$ by representing the nonlinear expressions entries in $\mathbf{f}(\mathbf{x})$ as the root node in a tree with the leaf nodes representing the $\mathbf{M}_i(\xi_0)$ or other constants that might be used by the nonlinear function components in $\mathbf{f}(\cdot)$.

The process is first started with the computation of $\mathbf{M}_0(\xi_0)$, which, as can be seen from (5.2), is the DC operating point at the nominal values of the design parameters. $\mathbf{M}_0(\xi_0)$ is then assigned to the leaf nodes on the rooted trees representing the expressions of both $\mathbf{J}(\mathbf{x})$ and $\mathbf{f}(\mathbf{x})$. The trees are traversed in both directions as mentioned above to obtain $\mathbf{F}_1(\xi_0)$ and $\mathbf{J}_1(\xi_0)$, which are then used to compute $\mathbf{M}_1(\xi_0)$ using the recursive formula in (5.37). Again, both of $\mathbf{M}_0(\xi_0)$ and $\mathbf{M}_1(\xi_0)$ are assigned to the leaf nodes, and traversing the tree is repeated again resulting in $\mathbf{F}_2(\xi_0)$ and $\mathbf{J}_2(\xi_0)$, and so forth. Further details on the construction and operation of the rooted tree can be found in [55, 110, 111].

Rooted Trees

For the illustration of the idea of rooted trees, consider the circuit at the left bottom corner of Figure 5.1. In the MNA formulation, this circuit has $N = 1$, with the diode current assumed to be related to the node voltage through, $i(v(t)) = I_0 \left(\mathrm{e}^{v(t)/V_T} - 1\right)$.

Let ξ represent the thermal voltage V_T. Thus, it should be straightforward to write for this circuit

$$\mathbf{J}_I\left(\mathbf{z}_p(\xi),\xi\right) = \frac{I_0}{\xi}\mathrm{e}^{\frac{\mathbf{z}_p(\xi)}{\xi}} \tag{5.38}$$

To explain the operation of rooted trees, we assume that $\mathbf{M}_0$ and $\mathbf{M}_1$ have been computed and the goal now is to compute $\mathbf{M}_2$ using (5.37) where $\bar{\mathbf{J}}_1$ will be needed on the right side to accomplish this goal. The rooted tree representation of (5.38) will be used for that purpose. The process is started by first computing $\mathbf{J}_1$. The rooted tree representation of $\mathbf{J}_I\left(\cdot,\cdot\right)$ is the technique that is used to compute $\mathbf{J}_1$. The structure to the right of the circuit in Figure 5.1 is the rooted tree corresponding to the expression in (5.38).

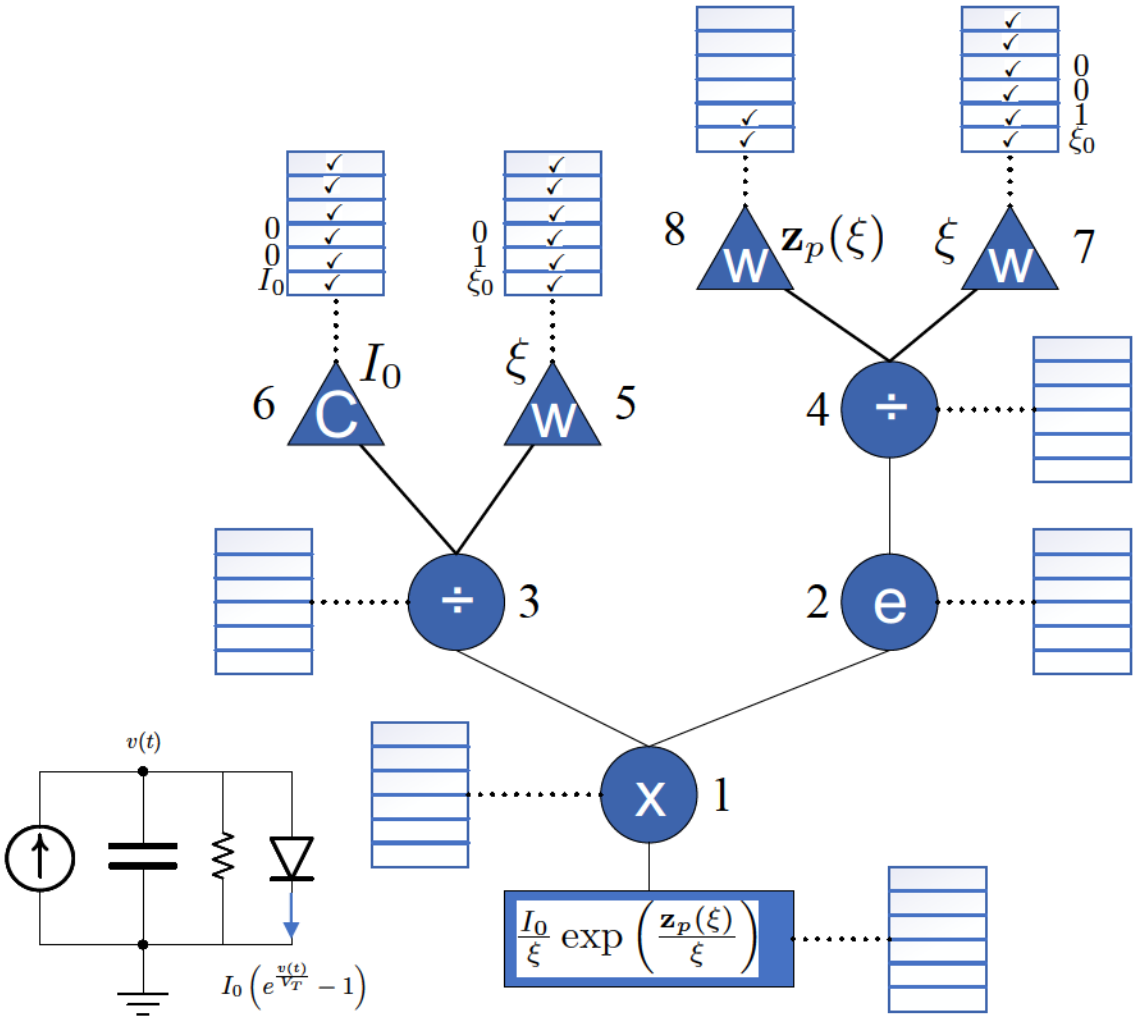


Figure 5.1: An example circuit and its rooted tree representation for $\mathbf{J}_I(\mathbf{x}(\xi), \xi)$.

In Figure 5.1, the nodes in the graph define an arithmetic operations. Thus, node 1 corresponds to a multiplication, 2 an exponential function, 3 and 4 a division, etc. Each node has a local storage that contains the values of the Taylor coefficients corresponding to the term represented by the node. For example, node 2 stores the Taylor coefficients of $e^{\mathbf{z}_p(\xi)/\xi}$ in ξ. Leaf nodes with ✓ in their table entries show which Taylor coefficients are known, assuming $\mathbf{M}_0$ and $\mathbf{M}_1$ are computed already. As can be seen in the tree structure in Figure 5.1, nodes in the tree can be classified as, root (parent) node that represents the expression in (5.38), leaf nodes (triangles) which are

the terminal (childless) nodes, and intermediate nodes (circles). Leaf nodes represent either the constant terms in the expression, e.g., I_0, or the terms that have explicit[1] or implicit (indirect)[2] dependence on the parameter ξ, which are marked with "w". Intermediate nodes represent atomic functions, such as the exponential function, or arithmetic operators, e.g. division or multiplication.

The fundamental ideas of operation in the rooted trees can be summarized as follows. With exception of the leaf nodes, all nodes in the tree compute the Taylor coefficients of the part of the expression they represent, based on the Taylor coefficients of their children nodes and their own lower-order Taylor coefficients. Nodes with constant terms or explicit dependence on ξ have known Taylor coefficients in ξ, e.g., nodes 6, 5, and 7 in Figure 5.1. Each node maintains a local storage (shown as a table in the figure) which it uses to store its *own* Taylor coefficients. The process starts by first assigning the Taylor coefficients to the leaf nodes, as mentioned. Computing $\mathbf{J}_1$ is triggered at the root node, which instigates its immediate descendant, so to speak, to compute its 1-st order Taylor coefficient. The tree is traversed in the upward direction, with each node simply triggering the node at the next level prompting it to compute the 1-st coefficient. When a leaf node is reached, it responds by sending back to the nodes at lower level the values already stored in the local storage table. The tree is then traversed in the downward direction, with each node computing, using a pre-specified expression (specific to each node type), the value of the 1st Taylor coefficient at that node. The computed value is stored locally to the node (for future use), and sent down to the lower level. The process is continued downwards, and upon arriving back at the root node, will have accumulated in its path the required value of $\mathbf{J}_1$. Therefore, $\mathbf{M}_2$ can now be computed and the process is repeated to compute $\mathbf{J}_2$, and so forth. Further details on the automated construction, operation

[1] such as nodes 5 and 7 representing ξ.

[2] e.g., node 8, representing $x(\xi)$.

of the rooted tree can be found in [55, 110, 111]. Once the moments up to order m have been computed a "skinny" QR factorization [107] can be performed on them to obtain $\bar{\mathbf{V}}$.

5.3.2 Construction of the Matrix P

Once $\mathbf{U}$ is obtained using the rooted trees as explained in the previous subsection, then $\mathbf{P}$ can be constructed through $\mathbf{U}$ using the algorithm proposed in [19] and shown in the pseudo-code representation of Algorithm 3 for completeness. The operator max employed in this algorithm is the same as the Matlab function `max(u)`, which returns the maximum element in a vector and its location (index). The output of the algorithm is a set of k indices $\gamma_i, \in \mathbb{N}$, with $i = 1, \cdots, k$ that represents the set of columns selected from the identity matrix use to build $\mathbf{P}$. One needs to stress here that the resulting $\mathbf{P}$ from the above algorithm is $\mathbf{U}$-dependent.

Algorithm 3: Indices selection algorithm

input : $\{\mathbf{u}_l\}_{l=1}^{k}$ linearly independent
output: $\mathbf{\Gamma} = \{\gamma_i,\ i = 1, \cdots, k\}$
$[\rho_1, \gamma_1] = \mathsf{max}\,(\mathbf{u}_1)$;
$\mathbf{\Gamma} \leftarrow \gamma_1$;
for $l \leftarrow 2$ **to** k **do**
  Solve $\left(\mathbf{P}^\top \mathbf{U}\right) \mathbf{c} = \mathbf{P}^\top \mathbf{u}_l$ for $\mathbf{c}$;
  $\mathbf{r} = \mathbf{u}_l - \mathbf{U}\mathbf{c}$;
  $[\rho_l, \gamma_l] = \mathsf{max}\,(\mathbf{r})$;
  $\mathbf{U} \leftarrow [\mathbf{U} \quad \mathbf{u}_l]\,, \quad \mathbf{P} \leftarrow [\mathbf{P} \quad \mathbf{e}_{\gamma_l}], \quad \mathbf{\Gamma} \leftarrow \{\mathbf{\Gamma} \cup \gamma_l\}$;
end

5.4 Extension to General Nonlinearity

The previous development used a special form of nonlinearity in which each component in $\mathbf{f}(\cdot)$ was assumed to be a function of only one independent variable in $\mathbf{x}$. The

objective was to facilitate presenting the core ideas without having to utilize cumbersome notations. Handling the more general nonlinear $\mathbf{f}(\mathbf{x}(\boldsymbol{\xi}), \boldsymbol{\xi})$ is easily implemented by noting that in a typical nonlinear circuit with many variables the elements in $\mathbf{f}$ indexed by the set $\boldsymbol{\Gamma}$, i.e. $f_{\gamma_i}(\mathbf{x}(\boldsymbol{\xi}))$ are typically dependent on a few number of variables in $\mathbf{x}$. This fact can be captured by the introducing the following structures.

- A set $\mathbf{j}_{\gamma_i}$ of integer values with p_{γ_i} members ($p_{\gamma_i} = |\mathbf{j}_{\gamma_i}|$) that is comprised from the indices of the variables that control the expression of $f_{\gamma_i}(\mathbf{x}(\boldsymbol{\xi}))$,
- A local selector matrix $\mathbf{L}_{\gamma_i} \in \{0,1\}^{N \times p_{\gamma_i}}$. Columns in $\mathbf{L}_{\gamma_i}$ has "1" in the row corresponding to the elements in $\mathbf{j}_{\gamma_i}$, and "0" otherwise, and
- A global selector matrix $\mathbf{L} \in \{0,1\}^{N \times p}$, where

$$p = \Sigma_{i=1}^{k} \left|\mathbf{j}_{\gamma_i}\right|,$$

given by

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{\gamma_1} & \mathbf{L}_{\gamma_2} & \cdots & \mathbf{L}_{\gamma_k} \end{bmatrix} \tag{5.39}$$

With the above notation in place, (5.12) can be extended to general nonlinearity using

$$\tilde{\mathbf{f}}(\mathbf{x}(\boldsymbol{\xi})) = \mathbf{U}\left(\mathbf{P}^{\top}\mathbf{U}\right)^{-1}\mathbf{P}^{\top}\mathbf{f}(\mathbf{L}^{\top}\mathbf{Q}\hat{\mathbf{x}}(\boldsymbol{\xi})) \tag{5.40}$$

Proceeding with manipulation similar to what was followed in Section 5.2.1, the reduced nonlinear vector follows from,

$$\hat{\mathbf{f}}(\hat{\mathbf{x}}(\boldsymbol{\xi})) = \boldsymbol{\Psi}\,\boldsymbol{\Xi}(\hat{\mathbf{x}}(\boldsymbol{\xi})) \tag{5.41}$$

where $\boldsymbol{\Xi}(\hat{\mathbf{x}}(\boldsymbol{\xi}))$ in this case is given by

$$\boldsymbol{\Xi}(\hat{\mathbf{x}}(\boldsymbol{\xi})) = \mathbf{f}_k(\mathbf{Q}_p\hat{\mathbf{x}}(\boldsymbol{\xi})) \tag{5.42}$$

with $\mathbf{Q}_p$ being the p rows of $\mathbf{Q}$ selected by $\mathbf{L}^\top$. Similarly, the reduced Jacobian matrix is given by the (more general) formula

$$\hat{\mathbf{J}}\left(\hat{\mathbf{x}}(\boldsymbol{\xi})\right) = \underbrace{\boldsymbol{\Psi}}_{m\times k} \quad \underbrace{\frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_p}}_{k\times p} \quad \underbrace{\mathbf{Q}_p}_{p\times m}, \tag{5.43}$$

Chapter 6

Numerical Examples

Three examples are presented in this section. The first example presents a proof of concept and demonstrates the accuracy of the proposed approach compared with the direct sensitivity approaches.

The other two examples validate the accuracy and efficiency of the proposed approach. An error measurement based on Equation (6.1) was adopted to illustrate the accuracy of the proposed algorithm.

$$\epsilon = \frac{||\mathbf{x}_{\mathrm{DC}}^{\mathrm{full}} - \mathbf{x}_{\mathrm{DC}}^{\mathrm{proposed}}||_2}{||\mathbf{x}_{\mathrm{DC}}^{\mathrm{full}}||_2} \tag{6.1}$$

$\mathbf{x}_{\mathrm{DC}}^{\mathrm{full}}$ denotes the original circuit's DC operating point, and $\mathbf{x}_{\mathrm{DC}}^{\mathrm{proposed}}$ is the DC operating point evaluated using the proposed procedure described in this chapter.

A selected set of d parameters in each of the two circuits are marked as the design parameters represented by the parameters $\boldsymbol{\xi}$ in the formulation. Each of the parameters are assumed to vary between -20% to +20% around their nominal design values, generating 2^d "corners" in the $\mathbb{R}^d$ design parameter space. The DC operating point, evaluated using the original and reduced systems, is obtained, and the error between the two solutions at each parameter space corner is evaluated.

To implement the proposed method, k and m are defined for each example. k is

the number of columns of the orthonormal matrix $\mathbf{U}$ and selector matrix $\mathbf{P}$. In other words, it defines the number of entries chosen in $\mathbf{f}(\mathbf{x}(\xi), \xi)$ for MIP. m defines the order of the reduces system after applying PMOR.

6.1 Example 1: CMOS Operational Amplifier

The first example is the MOSFET-based OpAmp circuit shown in Figure 6.1. The variation in the DC operating point in this circuit is the result of variations in the length and width of the channel in several transistors. The nominal values of the length (L) and width (W) are provided in the circuit schematic (indicated by the ratio W/L in μm). For this example V_{DD}=2.5V.

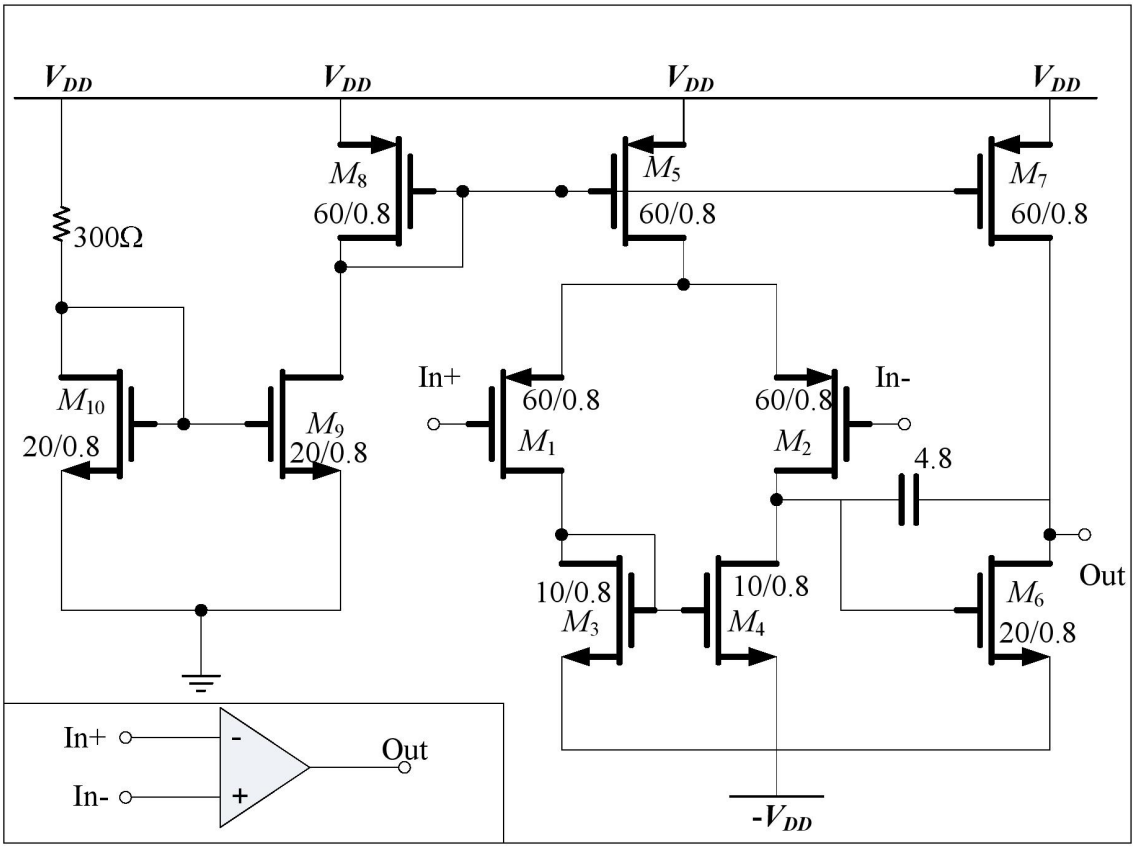


Figure 6.1: Schematic of operational amplifier.

Figure 6.2 shows the change in the DC operating point of the output node versus variations in the channel lengths of all the transistors. Likewise, Figure 6.3 shows the change in the DC voltage of the same node versus variations in the width of transistors M1, M2, M5, M7, M8. The variations illustrated in both figures also show a comparison for results obtained through different methods: the first approach

(dubbed "original") refers to the exact variations which are computed by solving the original system of nonlinear equations for the DC operating point; the second method is the proposed approach obtained with a reduced system with m = 6 moments; the third approach (labeled as "direct Sensitivity") is based on using the computed moments $\mathbf{M}_i(\xi_0)$, $i = 1, 2, \cdots, 6$ to evaluate the variations in DC solution.

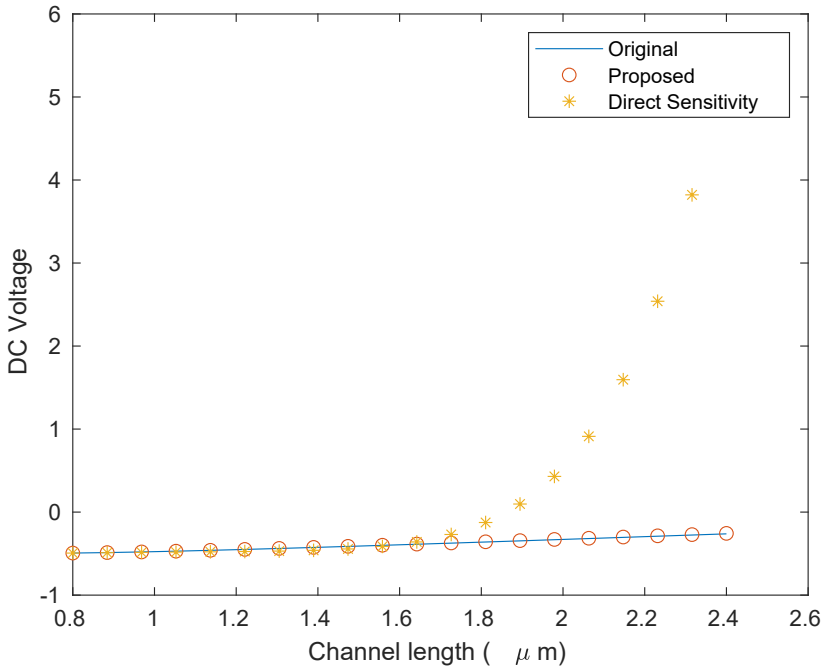


Figure 6.2: Variations of the DC voltage at the output node of OpAmp vs. variations in the length of the MOSFETs channels.

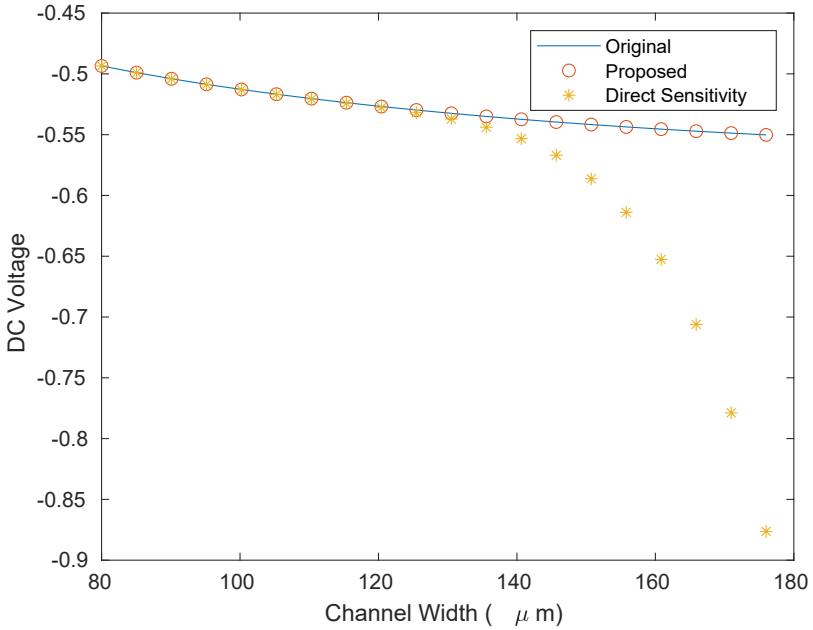


Figure 6.3: Variations of the DC voltage at the output node of OpAmp vs. variations in the width of the MOSFETs channels.

The above results indicate that the reduced system, created through the proposed approach, can accurately capture the variations in the DC operating point with up to 200% variations in the design parameters around their nominal values.

6.2 Example 2: 741 Op-Amp

The circuit used in this example is an inverting amplifier based on the μA741 OpAmp, whose schematic is shown in Figure 6.4. The internal schematic of the μA741 is presented in Figure 6.5. As seen from Figure 6.5, the circuit's DC bias is provided by two power supplies V_{CC}=15V and V_{EE}=−15V.

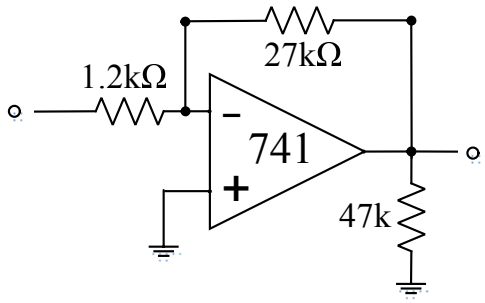


Figure 6.4: Schematic of inverting amplifier

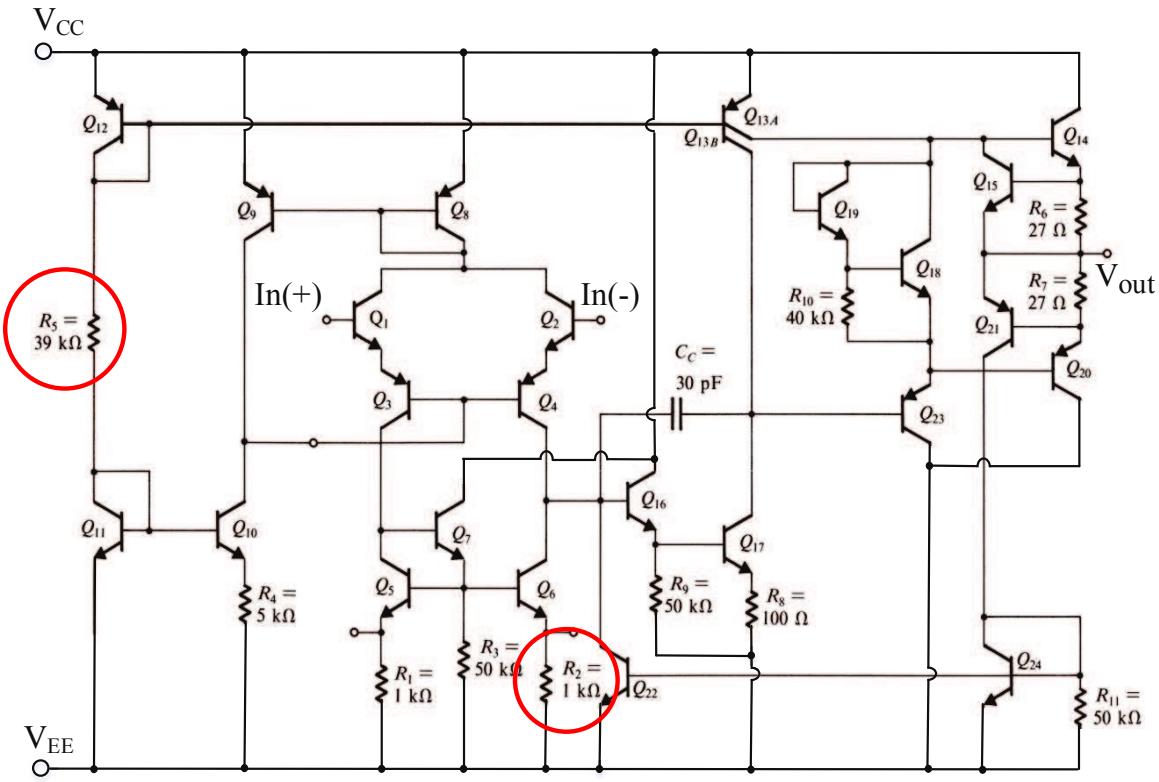


Figure 6.5: Internal schematic of μA741 OpAmp [1].

The resistors circled in Figure 6.5 are considered as the design parameters ξ_1 and ξ_2. Each parameter is varied by $\pm 20\%$. Table 6.1 provides the error computed using (6.1) at the four possible corners in the parameter space. In this example, k was defined as 5, whereas the size of the reduced system using PMOR is $m = 6$.

Table 6.1: Error at each corner case for the DC simulation of the inverting amplifier.

Param Combination ($\boldsymbol{\xi}$)	ϵ **in** $\%$
$(0\%, 0\%)$	1.125×10^{-11}
$(20\%, 20\%,)$	2.58×10^{-2}
$(-20\%, 20\%,)$	1.36×10^{-2}
$(-20\%, -20\%)$	2.16×10^{-2}
$(-20\%, -20\%)$	1.21×10^{-2}

Table 6.2 demonstrated the reduction in the size of the Jacobian matrix ($\mathbf{J}$) due to PMOR. By setting m as 6, the size of the system was reduced from 202 to 6.

The advantages of MIP do not only stem from the reduction in the size of the Jacobian matrix used during factorization. It is also a result of drastically reducing the number of nonlinear model evaluations, made possible through the projection operator $\mathbf{P}$. Table 6.3 demonstrates the optimization achieved by only evaluating selected nonlinear functions using MIP.

Table 6.2: A comparison between the size of the original system and the reduced model using PMOR.

Original system	**Size of $\mathbf{X}(\boldsymbol{\xi})$**	202x1
	Size of $\mathbf{J}(\boldsymbol{\xi})$	202x202
Reduced System ($m = 6$)	**Size of $\hat{\mathbf{X}}(\boldsymbol{\xi})$**	6x1
	Size of $\hat{\mathbf{J}}(\boldsymbol{\xi})$	6x6

Table 6.3: A comparison of number nonlinear function evaluation between the original system and DEIM.

Original system	**Size of $\mathbf{f}(\mathbf{x}(t,\boldsymbol{\xi}),\boldsymbol{\xi})$**	202x1
	NL function evals in $\mathbf{f}(\mathbf{x}(t,\boldsymbol{\xi}),\boldsymbol{\xi})$	75
Reduced System ($k=5$)	**Size of $\mathbf{f}_k(\mathbf{x}(t,\boldsymbol{\xi}),\boldsymbol{\xi})$**	5x1
	NL function evals in $\mathbf{f}(\mathbf{x}(t,\boldsymbol{\xi}),\boldsymbol{\xi})$	5

The impact in the reduction of nonlinear function evaluations are further highlighted when considering the total number of N-R iteration used to solve the DC solution and all the parameter combinations that must be evaluated. Table 6.4 demonstrates this by showing the total number of nonlinear device model evaluations in both the original and reduced models, indicating a significant reduction in the number of times the device model had to be evaluated. The "Total Evaluations" was calculated by adding the number of nonlinear function evaluations in $\mathbf{f}(\mathbf{x})$ and $\mathbf{J}(\mathbf{x})$ and multiplying it by the total number of N-R iterations required to evaluate all parameter combinations.

Table 6.4: A comparison between the savings achieved during nonlinear function evaluation using traditional method vs. the proposed method.

	# of NL function eval.		**Total Iter.**	**Total Evals.**
	$\mathbf{f}(\mathbf{x}(t,\boldsymbol{\xi}),\boldsymbol{\xi})$	$\frac{\partial \mathbf{f}(\mathbf{x}(t,\xi),\xi)}{\partial \mathbf{x}(t,\xi),\xi)}$		
Orig.	75	175	36	**9,000**
Prop.	5	12	25	**425**

6.3 Example 3: Power Distribution Network (PDN)

The circuit in this example is based on a **Power Distribution Network (PDN)** conceptually represented by Figure 6.6. A PDN is a grid used on Silicon to distribute power to the various devices in an integrated circuit. The PDN is typically modelled by a mesh of resistors with capacitors connected to Ground. The circuit in this example uses a PDN, modelled by a mesh of resistors with 10,494 nodes, where the nodes are connected to the DC supply rail of inverter circuits via resistors. There is a total of 10 SN7404 inverters (Figure 6.7) used in this circuit. The inverters are connected to the ground plane via resistors.

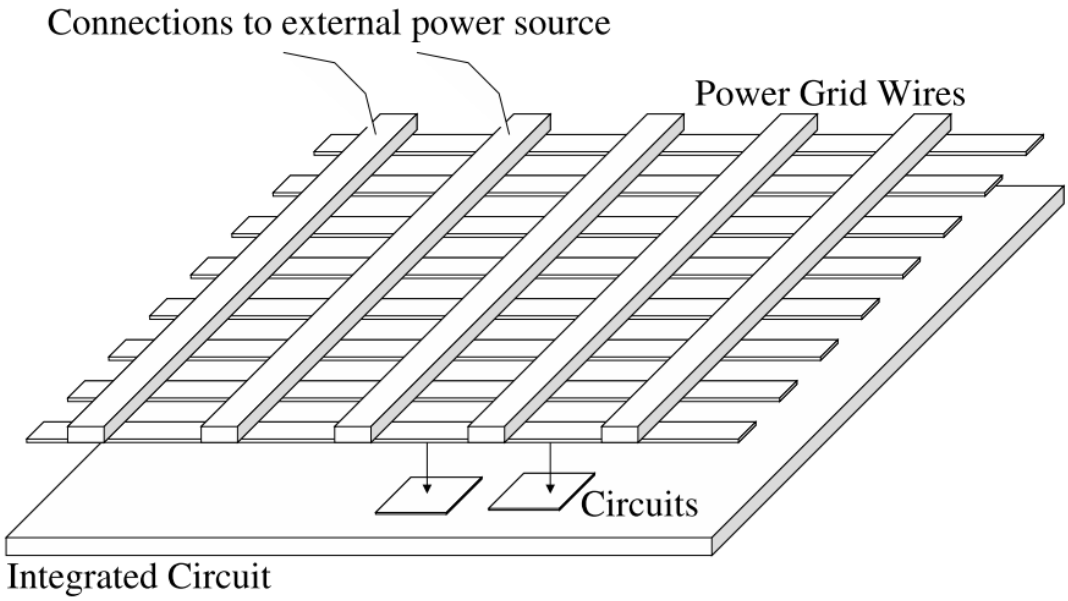


Figure 6.6: Power distribution network [2].

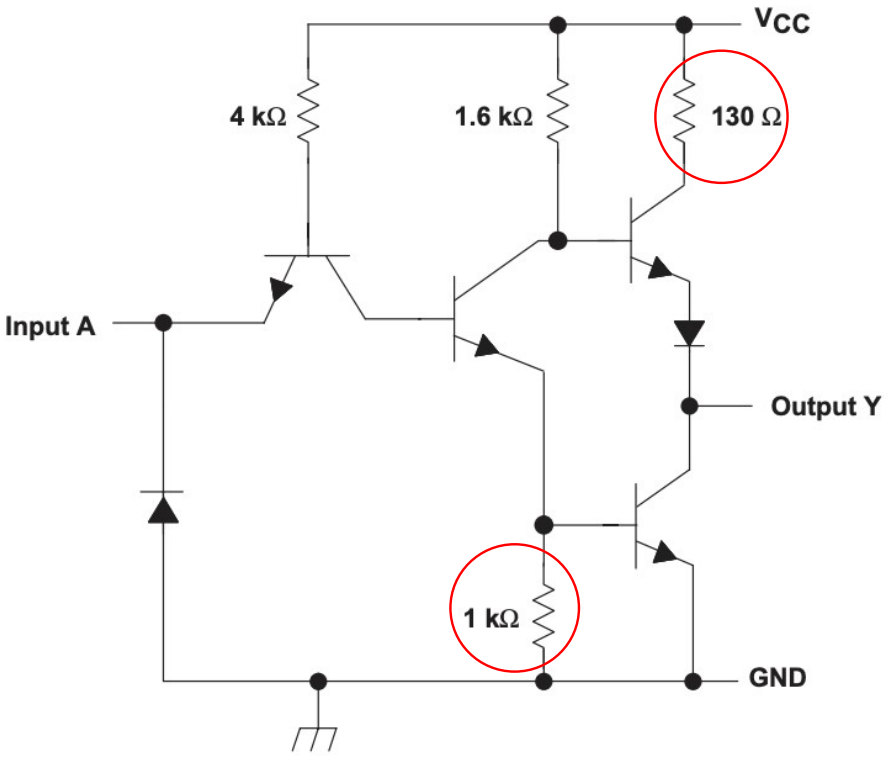


Figure 6.7: Circuit schematic of the SN7404 inverter [3].

Four design parameters were used in this example: two parameters were used to represent the group of resistors connecting the PDN nodes to the supply rail of the inverters and the group of resistors connecting the inverter ground node to the ground plane. The other two parameters are taken from the two resistors circled in Figure 6.7 in all the inverters. This configuration resulted in 16 corners in the design space.

In this example, k was chosen to be 5, and m was set to 7. The accuracy of the reduced system in approximating the variation in the DC operating point of the original system is illustrated by Table 6.5. It tabulates the error in the DC operating point obtained at the 16 corners in the design space.

Table 6.5: Error at each corner case for the DC simulation of the PDN.

Param Combination ($\boldsymbol{\xi}$)	ϵ **in** %
$(0\%, 0\%, 0\%, 0\%)$	5.13×10^{-11}
$(+20\%, +20\%, +20\%, +20\%)$	7.97×10^{-6}
$(-20\%, +20\%, +20\%, +20\%)$	6.62×10^{-6}
$(-20\%, -20\%, +20\%, +20\%)$	1.55×10^{-6}
$(-20\%, -20\%, -20\%, +20\%)$	1.55×10^{-6}
$(-20\%, -20\%, -20\%, -20\%)$	1.55×10^{-6}
$(+20\%, -20\%, -20\%, -20\%)$	3.49×10^{-7}
$(+20\%, +20\%, -20\%, -20\%)$	3.78×10^{-6}
$(+20\%, +20\%, +20\%, -20\%)$	7.97×10^{-6}
$(+20\%, -20\%, -20\%, +20\%)$	3.49×10^{-7}
$(-20\%, +20\%, +20\%, -20\%)$	6.62×10^{-6}
$(+20\%, -20\%, +20\%, +20\%)$	2.45×10^{-6}
$(+20\%, +20\%, -20\%, +20\%)$	3.77×10^{-6}
$(-20\%, +20\%, -20\%, +20\%)$	3.07×10^{-6}
$(+20\%, -20\%, +20\%, -20\%)$	2.45×10^{-6}
$(-20\%, -20\%, +20\%, -20\%)$	1.55×10^{-6}
$(-20\%, +20\%, -20\%, -20\%)$	3.07×10^{-6}

Table 6.6 demonstrates the reduction in the size of the Jacobian matrix ($\mathbf{J}$) due to PMOR. By setting m as 7, the size of the system was reduced from 10706 to 7.

As explained previously, MIP also drastically reduced the number of nonlinear model evaluations, which is made possible through the projection operator P. Table 6.7 presents the optimization observed by only evaluating selected nonlinear functions using MIP.

Table 6.6: A comparison between the size of the original system and the reduced model using PMOR.

Original system	**Size of $\mathbf{X}(\boldsymbol{\xi})$**	10706x1
	Size of $\mathbf{J}(\boldsymbol{\xi})$	10706x10706
Reduced System $(m = 7)$	**Size of $\hat{\mathbf{X}}(\boldsymbol{\xi})$**	7x1
	Size of $\hat{\mathbf{J}}(\boldsymbol{\xi})$	7x7

Table 6.7: A comparison of number NL function evaluation between the original system and DEIM.

Original system	**Size of $\mathbf{f}(\mathbf{x}(t, \boldsymbol{\xi}), \boldsymbol{\xi})$**	10706x1
	NL function evals in $\mathbf{f}(\mathbf{x}(t, \boldsymbol{\xi}), \boldsymbol{\xi})$	150
Reduced System $(k = 5)$	**Size of $\mathbf{f}_k(\mathbf{x}(t, \boldsymbol{\xi}), \boldsymbol{\xi})$**	5x1
	NL function evals in $\mathbf{f}(\mathbf{x}(t, \boldsymbol{\xi}), \boldsymbol{\xi})$	5

The impact in the reduction of nonlinear function evaluations are further highlighted when considering the total number of N-R iteration used to solve the DC solution and all the parameter combinations that must be evaluated. Table 6.8 demonstrates this by recording the total number of nonlinear device model evaluations in both the original and reduced models. It demonstrates a significant reduction in the number of times the device model had to be evaluated.

Table 6.8: A comparison between the savings achieved during NL function evaluation using traditional method vs. the proposed method.

	# of NL function eval.		**Total Iter.**	**Total Evals.**
	$\mathbf{f}(\mathbf{x}(t, \boldsymbol{\xi}), \boldsymbol{\xi})$	$\frac{\partial \mathbf{f}(\mathbf{x}(t,\xi),\xi)}{\partial \mathbf{x}(t,\xi),\xi)}$		
Orig.	150	330	89	**42,720**
Prop.	5	13	95	**1,710**

6.3.1 Summary

As demonstrated using the numerical examples in this Chapter (6), the proposed method is able to accurately and efficiently simulate nonlinear circuits. For both examples it was demonstrated that the advantages in the MIP approach do not stem merely from reducing the size of the Jacobian matrix that needs to be factorized. It also results from drastically reducing the number of nonlinear model evaluations which is made possible through the projection operator $\mathbf{P}$.

Chapter 7

Conclusion

7.1 Concluding Remarks

The main objective in this thesis has been to wield the idea of Model Order Reduction (MOR) to tackle the problem of repeated DC analysis of nonlinear circuits.

The problem of DC analysis considered in the thesis refers to the task of repeating the computation of the DC quiescent point at values for a selected set of design parameters, different than their nominal values.

The approach developed in the thesis is based on the ideas of Discrete Empirical Interpolation Method (DEIM) and concepts of moment-matching model-order reduction techniques.

The unique contribution of the thesis lies in adapting DEIM and moment matching to the context of the DC problems. More particularly, the notion of rooted trees was developed to enable the construction of the projection matrices used in the reduction process.

Through the concept of rooted trees, it was demonstrated that it is possible to construct the projection basis at the nominal design values and without having to create the so-called "snapshots" that necessitated simulating the full system at numerous points in the parameter space.

Several examples have been presented to demonstrate the validity, accuracy and efficiency of the proposed methodology.

7.2 Future Work

The ideas and approaches developed in this thesis are sufficiently broad to be used in tackling different problems and in different domains of applications. Indeed, the DC nonlinear equations, which represented the core and target of this thesis had no significant restriction in their formulation that would limit the applicability of the proposed approach to the DC problem. This observation represents the main motivation in pursuing problems that posit similar questions to the DC problem and the parameter variations.

One of the main problems that will be the immediate focus in future research is the problem of Harmonic Balance (HB). In HB, as in the DC problems, the goal is to solve a set of nonlinear equations but with a more complex structure. It also happens that in HB problems there are parameters that need to varied from their nominal design values and the HB analysis needs to be repeated at those new values.

The ideas presented in this thesis are ideal tools to tackle the HB problem. In fact, the author has developed initial prototypes that showed significant efficiency for the HB problem and plans to pursue those ideas in his Ph.D. program. A sample of the results obtained is described in the following example to demonstrate the validity of using the methodology in this thesis for HB problems.

7.2.1 Example: Low Noise Amplifier

The Low Noise Amplifier (LNA) with the schematic shown in Figure 7.1 is considered for this example, where the Eber-Moll model has been used to model the bipolar junction transistors. The steady-state output voltage in response to a sinusoidal

input of 0.2V amplitude is shown Figure 7.2.

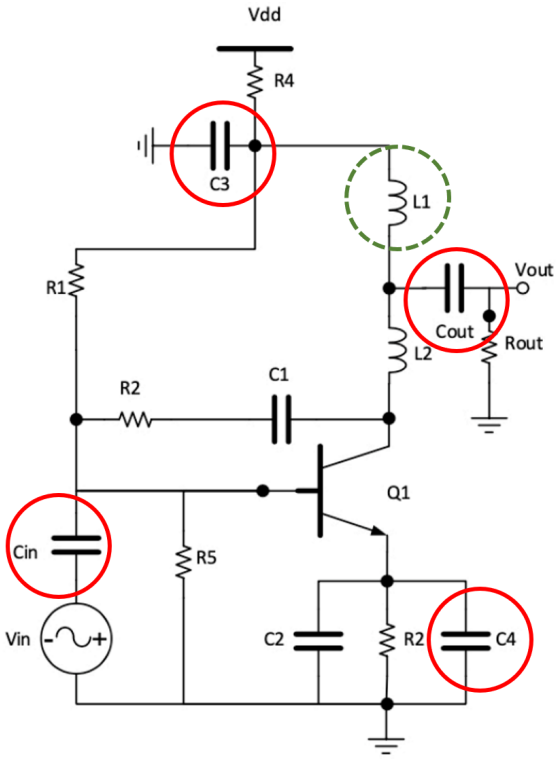


Figure 7.1: LNA schematic.

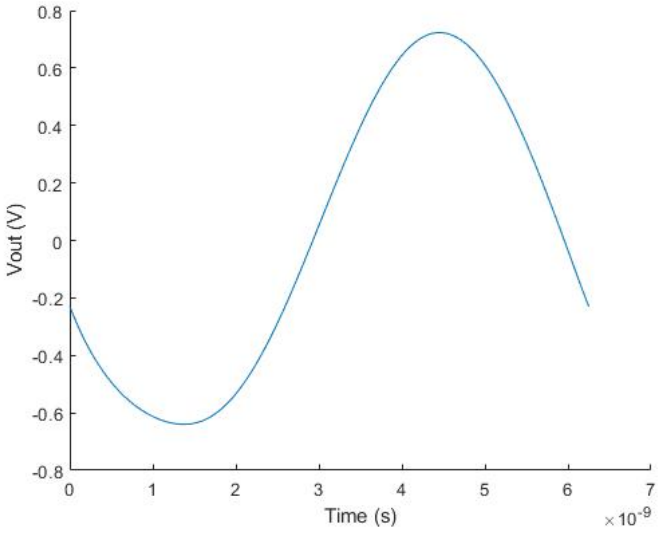


Figure 7.2: Steady-state of the output response.

Two parameters ξ_1 and ξ_2 are used in this example: ξ_1 controls the values of circled capacitors in Figure 7.1 using the relationship in Equation (7.1) and ξ_2 controls the value of the circled inductor L_1 using the relationship in Equation (7.2).

$$C(\xi_1) = \frac{\epsilon(L(1+\xi_1) \times W(1+\xi_1))}{d} \tag{7.1}$$

$$L_1(\xi_2) = L(1+\xi_2) \tag{7.2}$$

The Proposed HB prototype based on PMOR was used in this example to perform the parameter sweep on the first, second, and third harmonics of the steady-state output voltage. This produced the results depicted in the bottom of the plots of Figures 7.3-7.5 with a CPU time of 1.63 mili-seconds. Figures 7.3-7.5 also demonstrate the accuracy of the results by comparing them with the results obtained from the full HB parameter sweep which required 0.185 seconds. This indicates that the proposed approach is around 113 times faster than the conventional parameter sweep of the full HB.

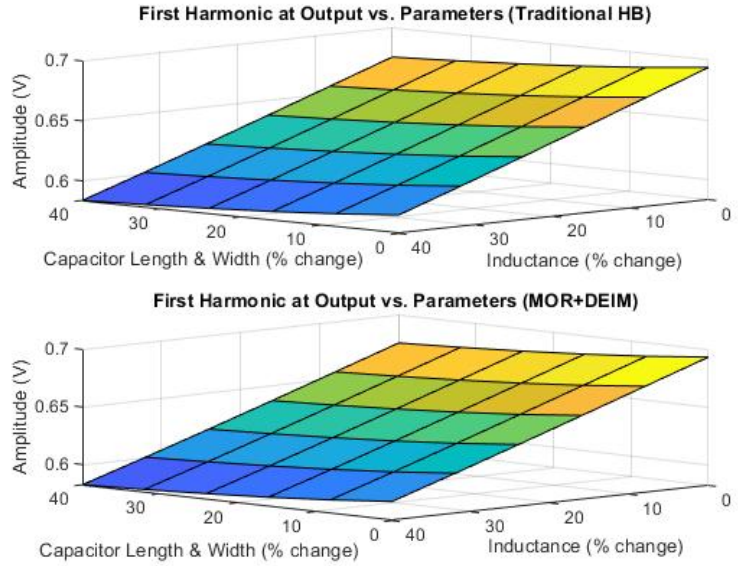


Figure 7.3: Comparison between the proposed method and traditional HB for the variation in the first harmonic of the output node vs. the variation in the design parameters

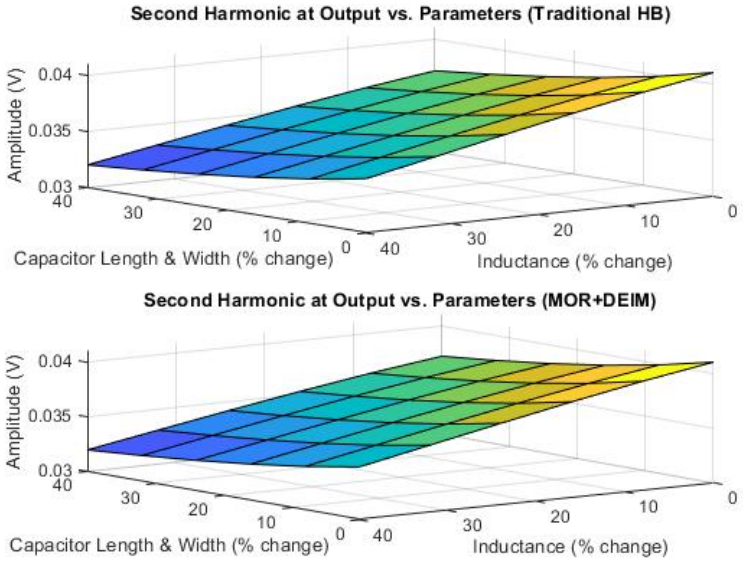


Figure 7.4: Comparison between the proposed method and traditional HB for the variation in the second harmonic of the output node vs. the variation in the design parameters.

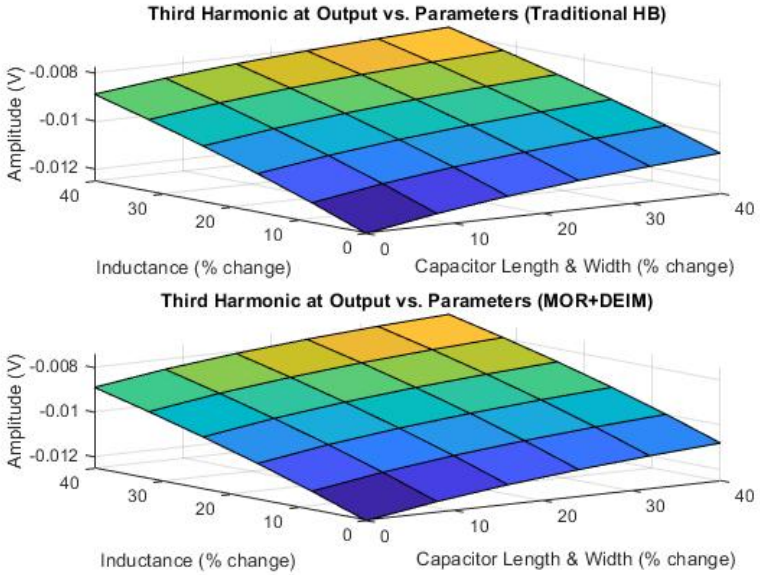


Figure 7.5: Comparison between the proposed method and traditional HB for the variation in the third harmonic of the output node vs. the variation in the design parameters.

List of References

[1] A. Sedra and K. Smith, *Microelectronic Circuits*. New York, NY, USA: Oxford Univ. Press, seventh ed., 2014.

[2] S. R. Nassif, "Power grid analysis benchmarks," in *2008 Asia and South Pacific Design Automation Conference*, pp. 376–381, 2008.

[3] Texas Instruments Inc., Dallas, TX, USA, *Hex Inverters Datasheet*, 1983.

[4] R. C. Melville, L. Trajković, S.-C. Fang, and L. T. Watson, "Artificial parameter Homotopy methods for the DC operating point problem," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 12, pp. 861–877, June 1993.

[5] M. Striebel and J. Rommes, "Model order reduction of nonlinear systems in circuit simulation: Status and applications," in *Model Reduction for Circuit Simulation* (P. Benner, M. Hinze, and E. J. W. ter Maten, eds.), vol. 74 of *Lecture Notes in Electrical Engineering*, ch. 17, pp. 289–301, Berlin, Heidelberg, Germany: Springer-Verlag, 2011.

[6] D. De Jonghe and G. Gielen, "Characterization of analog circuits using transfer function trajectories," *IEEE Trans. Circuits Syst. I*, vol. 59, pp. 1796–1804, Aug. 2012.

[7] H. Aridhi, M. H. Zaki, and S. Tahar, "Enhancing model order reduction for nonlinear analog circuit simulation," *IEEE Trans. VLSI Syst.*, vol. 24, pp. 1036–1049, Mar. 2016.

[8] P. Benner and L. Feng, "A robust algorithm for parametric model order reduction based on implicit moment matching," in *Reduced order methods for modeling and computational reduction* (A. Quarteroni and G. Rozza, eds.), vol. 9 of *MS&A Modeling, Simulation and Applications*, ch. 6, pp. 159–185, Berlin, Heidelberg, Germany: Springer-Verlag, 2014.

[9] Y. Tao, B. Nouri, M. S. Nakhla, M. Farhan, and R. Achar, "Variability analysis via parameterized model order reduction and numerical inversion of Laplace transform," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 7, pp. 678–686, Jan. 2017.

[10] L. Daniel, O. C. Siong, L. S. Chay, K. H. Lee, and J. White, "A multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, pp. 678–693, May 2004.

[11] F. Ferranti, M. S. Nakhla, G. Antonini, T. Dhaene, L. Knockaert, and A. E. Ruehli, "multipoint full-wave model order reduction for delayed PEEC models with large delays," *IEEE Trans. Electromagn. Compat.*, vol. 53, pp. 959–967, Nov. 2011.

[12] P. Benner, S. Gugercin, and K. Willcox, "A survey of projection-based model reduction methods for parametric dynamical systems," *SIAM Review*, vol. 57, pp. 483–531, Nov. 2015.

[13] P. Benner and T. Breiten, "Two-sided projection methods for nonlinear model order reduction," *SIAM Journal on Scientific Computing*, vol. 37, pp. B239–B260, Jan. 2015.

[14] A. Astolfi, "Model reduction by moment matching for linear and nonlinear systems," *IEEE Trans. Autom. Control*, vol. 55, pp. 2321–2336, Oct. 2010.

[15] B. N. Bond and L. Daniel, "A piecewise-linear moment-matching approach to parameterized model-order reduction for highly nonlinear systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, pp. 2116–2129, Dec. 2007.

[16] B. Bond and L. Daniel, "Parameterized model order reduction of nonlinear dynamical systems," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, (San Jose, CA, USA), pp. 487–494, Nov. 2005.

[17] L. Sirovich, "Turbulence and the dynamics of coherent structures, parts: I-III," *Q. Appl. Math.*, vol. 45, pp. 561–590, Oct. 1987.

[18] T. Bui-Thanh, M. Damodaran, and K. E. Willcox, "Proper orthogonal decomposition extensions for parametric applications in transonic aerodynamics," in *Proc. 15th AIAA Computational Fluid Dynamics Conference*, (Orlando, FL, USA), pp. 4213–4233, June 2003.

[19] S. Chaturantabut and D. C. Sorensen, "Nonlinear model reduction via discrete empirical interpolation," *SIAM J. Sci. Comput.*, vol. 32, no. 5, pp. 2737–2764, 2010.

[20] D. Amsallem and J. Nordström, "Energy stable model reduction of neurons by nonnegative discrete empirical interpolation," *SIAM J. Sci. Comput.*, vol. 38, no. 2, pp. B297–B326, 2016.

[21] D. G. Luenberger, "Nonlinear descriptor systems," *Journal of Economic dynamics control 1*, pp. 219–242, Apr. 1979.

[22] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ, USA: Prentice Hall, third ed., 2002.

[23] K. Brenan, S. Campbell, and L. Petzold, *Numerical solution of initial-value problems in differential-algebraic equations*. Philadelphia, PA, USA: SIAM, 1996.

[24] C. W. Gear, "The automatic integration of stiff ordinary differential equations," in *Proc. the IFIPS Congress*, pp. A81–A85, 1968.

[25] A. Verhoeven, M. Striebel, J. Rommes, E. Maten, and T. Bechtold, "Model order reduction for nonlinear IC models with POD," in *Progress in Industrial Mathematics at (ECMI'08)* (A. D. Fitt, J. Norbury, H. Ockendon, and E. Wilson, eds.), Mathematics in Industry, pp. 441–446, Berlin, Heidelberg, Germany: Springer-Verlag, 2010.

[26] S.-B. Nouri, *Advanced Model-Order Reduction Techniques for Large-Scale Dynamical Systems*. PhD thesis, Dept. Elect., Carleton Univ., Ottawa, Canada, Sept. 2014.

[27] C. A. Desoer and E. S. Kuh, *Basic Circuit Theory*. New York, NY, USA: Prentice-Hall, 1969.

[28] E. S. Kuh and R. A. Rohrer, "The state-variable approach to network analysis," *Proc. IEEE*, vol. 53, pp. 672–686, July 1965.

[29] R. A. Rohrer, *Circuit Theory: an introduction to the State Variable Approach*. New York, NY, USA: McGraw-Hill, 1970.

[30] L. O. Chua, C. A. Desoer, and E. S. Kuh, *Linear and Nonlinear Circuits*. New York, NY, USA: McGraw-Hill, 1987.

[31] V. István, *Graph Theory: Application to the Calculation of Electrical Networks*, vol. 15 of *Studies in electrical and electronic engineering*. New York, NY, USA: Elsevier Science Pub., 1985.

[32] W.-K. Chen, *Graph Theory and Its Engineering Applications*, vol. 5 of *Advanced series in electrical and computer engineering*. River Edge, NJ, USA: World Scientific, 1997.

[33] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. Boston, MA, USA: Kluwer Academic, second ed., 2003.

[34] F. Dörfler, J. W. Simpson-Porco, and F. Bullo, "Electrical networks and algebraic graph theory: Models, properties, and applications," *Proc. IEEE*, vol. 106, pp. 977–1005, May 2018.

[35] G. Hachtel, R. Brayton, and F. Gustavson, "The sparse tableau approach to network analysis and design," *IEEE Trans. Circuit Theory*, vol. 18, pp. 101–113, Jan. 1971.

[36] C. W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits Syst.*, vol. 22, pp. 504–509, June 1975.

[37] L. T. Pillage, R. A. Rohrer, and C. Visweswariah, *Electronic Circuit and System Simulation Methods*. New York, NY, USA: McGraw-Hill, 1999.

[38] F. N. Najm, *Circuit Simulation*. Hoboken, NJ, USA: Wiley, 2010.

[39] B. Nouri, E. Gad, M. Nakhla, and R. Achar, "Model-order reduction in microelectronics," in *Handbook on Model-Order Reduction: Theory, Research Aspects and Applications* (P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. M. Silveira, eds.), vol. 3 of *MORNET series*, ch. 4, pp. 111–143, Berlin, Germany: Walter de Gruyter GmbH, 2020.

[40] E. Gad, M. Nakhla, and R. Achar, "Model-order reduction of high-speed interconnects using integrated congruence transform," in *Model Order Reduction: Theory, Research Aspects and Applications* (W. H. A. Schilders, H. A. van der Vorst, and J. Rommes, eds.), vol. 13 of *Mathematics in Industry*, ch. 17, pp. 361–401, Berlin, Heidelberg, Germany: Springer-Verlag, 2008.

[41] U. Baur, P. Benner, and L. Feng, "Model order reduction for linear and nonlinear systems: A system-theoretic perspective," *Arch. Computat. Methods Eng.*, vol. 21, pp. 331–358, Dec. 2014.

[42] P. Benner, M. Hinz, and E. J. W. ter Maten, eds., *Model Reduction for Circuit Simulation*. Berlin, Heidelberg, Germany: Springer-Verlag, 2011.

[43] W. H. A. Schilders, H. A. van der Vorstand, and J. Rommes, eds., *Model Order Reduction: Theory, Research Aspects and Applications*. Berlin, Heidelberg, Germany: Springer-Verlag, 2008.

[44] R. Pinnau, "Model reduction via proper orthogonal decomposition," in *Model Order Reduction: Theory, Research Aspects and Applications* (W. H. A. Schilders, H. A. van der Vorstand, and J. Rommes, eds.), ch. 2, pp. 95–110, Berlin, Heidelberg, Germany: Springer-Verlag, 2008.

[45] P. Astrid, S. Weiland, K. Willcox, and T. Backx, "Missing point estimation in models described by proper orthogonal decomposition," *IEEE Trans. Autom. Control*, vol. 53, no. 10, pp. 2237–2251, 2008.

[46] S. X. D. Tan and L. He, *Advanced model order reduction techniques in VLSI design*. Cambridge, MA, USA: Cambridge Univ. Press, 2007.

[47] P. Benner, V. Mehrmann, and D. C. Sorensen, eds., *Dimension Reduction of Large-Scale Systems*, vol. 45 of *Lecture Notes in Computational Science and Engineering*. Berlin, Heidelberg, Germany: Springer-Verlag, 2005.

[48] A. C. Antoulas, *Approximation of large-scale dynamical systems*. Philadelphia, PA, USA: SIAM, 2005.

[49] B. Nouri, M. S. Nakhla, and X. Deng, "Stable model-order reduction of active circuits," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 7, pp. 710–719, May 2017.

[50] M. J. Rewieński and J. White, "A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, pp. 155–170, Feb. 2003.

[51] M. Celik, L. Pileggi, and A. Odabasioglu, *IC interconnect analysis*. Boston, MA, USA: Kluwer Academic, 2002.

[52] B. Nouri and M. Nakhla, "Model-order reduction of nonlinear transmission lines using interpolatory proper orthogonal decomposition," *IEEE Trans. Microw. Theory Tech.*, vol. 66, pp. 5429–5438, Dec. 2018.

[53] B. Nouri, M. S. Nakhla, and R. Achar, "Efficient simulation of nonlinear transmission lines via model-order reduction," *IEEE Trans. Microw. Theory Tech.*, vol. 65, pp. 673–683, Mar. 2017.

[54] P. Li and L. T. Pileggi, "Compact reduced-order modeling of weakly nonlinear analog and RF circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, pp. 184–203, Feb. 2005.

[55] E. Gad, R. Khazaka, R. Nakhla, and R. Griffith, "A circuit reduction technique for finding the steady-state solution of nonlinear circuits," *IEEE Trans. Microw. Theory Tech.*, vol. 48, no. 12, pp. 2389–2396, 2000.

[56] B. Nouri, E. Gad, A. Nouri, and M. Nakhla, "DC-centric parameterized reduced-order model via moment-based interpolation projection (MIP) algorithm," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 10, pp. 1348–1357, Aug. 2020.

[57] P. Chen and C. Schwab, "Model order reduction methods in computational uncertainty quantification," in *Handbook of Uncertainty Quantification* (R. Ghanem, D. Higdon, and H. Owhadi, eds.), ch. 27, pp. 937–990, Basel, Switzerland: Springer Int. Publishing AG, 2017.

[58] D. Spina, F. Ferranti, G. Antonini, T. Dhaene, and L. Knockaert, "Efficient variability analysis of electromagnetic systems via polynomial chaos and model order reduction," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 4, pp. 1038–1051, June 2014.

[59] L. Lombardi, Y. Tao, B. Nouri, M. Nakhla, F. Ferranti, and G. Antonini, "Parameterized model order reduction of delayed PEEC circuits," *IEEE Trans. Electromagn. Compat.*, vol. 62, pp. 859–869, June 2020.

[60] D. Szypulski, G. Fotyga, V. de la Rubia, and M. Mrozowski, "A subspace-splitting moment-matching model-order reduction technique for fast wideband fem simulations of microwave structures," *IEEE Trans. Microw. Theory Tech.*, vol. 68, no. 8, pp. 3229–3241, 2020.

[61] L. Feng and P. Benner, "Parametric model order reduction for electro-thermal coupled problems," in *Nanoelectronic Coupled Problems Solutions*, pp. 293–309, Springer, 2019.

[62] G. Thrivikraman, S. L. Johnson, Z. H. Syedain, R. C. Hill, K. C. Hansen, H. S. Lee, and R. T. Tranquillo, "Biologically-engineered mechanical model of a calcified artery," *Acta biomaterialia*, vol. 110, pp. 164–174, 2020.

[63] M. Xiao, D. Lu, P. Breitkopf, B. Raghavan, S. Dutta, and W. Zhang, "On-the-fly model reduction for large-scale structural topology optimization using principal components analysis," *Structural and Multidisciplinary Optimization*, pp. 1–22, 2020.

[64] S. Ponsioen, S. Jain, and G. Haller, "Model reduction to spectral submanifolds and forced-response calculation in high-dimensional mechanical systems," *Journal of Sound and Vibration*, vol. 488, p. 115640, 2020.

[65] A. C. Antoulas and D. C. Sorensen, "Approximation of large-scale dynamical system: An overview," *International Journal of Applied Mathematics and Computer Science*, vol. 11, no. 5, pp. 1093–1121, 2001.

[66] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. Silveira, eds., *Model Order Reduction. Volume 1: System- and Data-Driven Methods and Algorithms*. Berlin: De Gruyter, 2020.

[67] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, and L. Silveira, eds., *Model Order Reduction. Volume 3: Applications*. Berlin: De Gruyter, 2020.

[68] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, pp. 639–649, May 1995.

[69] R. W. Freund, "Krylov-subspace methods for reduced-order modeling in circuit simulation," *Comput. Appl. Math.*, vol. 123, pp. 395–421, Nov. 2000.

[70] R. W. Freund, "SPRIM: structure-preserving reduced-order interconnect macromodeling," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Des.*, (San Jose, CA, USA), pp. 80–87, Nov. 2004.

[71] L. M. Silveira, M. Kamon, and J. White, "Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3-D interconnect structures," *IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part B*, vol. 19, pp. 283–288, May 1996.

[72] A. Odabasioglu, M. Celik, and L. Pileggi, "PRIMA: passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, pp. 645–654, Aug. 1998.

[73] E. J. Grimme, *Krylov projection methods for model reduction*. PhD thesis, Dept. Elect. Comput. Eng., Univ. Illinois Urbana-Champaign, Champaign, IL, USA, 1997.

[74] C. de Villemagne and R. E. Skelton, "Model reductions using a projection formulation," in *Proc. 26th IEEE Conference on Decision and Control*, (Los Angeles, CA, USA), pp. 461–466, 1987.

[75] T. Bechtold, M. Striebel, K. Mohaghegh, and E. J. W. ter Maten, "Nonlinear model order reduction in nanoelectronics: Combination of POD and TPWL," *Proc. Applied Mathematics and Mechanics*, vol. 8, pp. 10057—-10060, May 2008.

[76] N. Dong and J. Roychowdhury, "General-purpose nonlinear model-order reduction using piecewise-polynomial representations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, pp. 249–264, Feb. 2008.

[77] Y. Zhang, N. Fong, and N. Wong, "Piecewise-polynomial associated transform macromodeling algorithm for fast nonlinear circuit simulation," in *Proc. Proc. 18th Asia and South Pacific Design Automation Conference*, (Yokohama, Japan), pp. 515–520, Jan. 2013.

[78] J. M. A. Scherpen, *Balancing for nonlinear systems*. PhD thesis, University of Twente, Enschede, Netherlands, 1994.

[79] J. Hahn and T. Edgar, "Reduction of nonlinear models using balancing of empirical gramians and Galerkin projections," in *Proc. American Control Conference*, vol. 4, pp. 2864–2868, 2000.

[80] S. Sahyoun, J. Dong, and S. M. Djouadi, "Reduced order modeling for fluid flows based on nonlinear balanced truncation," in *Proc. American Control Conference*, pp. 1284–1289, June 2013.

[81] K. Willcox and J. Peraire, "Balanced model reduction via the proper orthogonal decomposition," *AIAA J.*, vol. 40, pp. 2323–2330, Nov. 2002.

[82] P. Astrid, *Reduction of Process Simulation Models: a Proper Orthogonal Decomposition Approach*. PhD thesis, Eindhoven Univ. of Technology, Eindhoven, Netherlands, Nov. 2004.

[83] a. Verhoeven, t. voss, p. astrid, e. ter maten, and t. bechtold, "Model order reduction for nonlinear problems in circuit simulation," *Proc. Applied Mathematics and Mechanics*, vol. 7, pp. 1021603–1021604, Dec. 2007.

[84] A. Verhoeven, M. Striebel, J. Rommes, E. Maten, and T. Bechtold, "Proper orthogonal decomposition model order reduction of nonlinear IC models," in *Progress in Industrial Mathematics at ECMI 2008* (A. D. Fitt, J. Norbury, H. Ockendon, and E. Wilson, eds.), Mathematics in Industry, pp. 441–446, Berlin, Heidelberg, Germany: Springer-Verlag, 2010.

[85] W. E. Arnoldi, "The principle of minimized iteration in the solution of the matrix eigenvalue problem," *Quat. Appl. Math.*, vol. 9, pp. 17–29, Apr. 1951.

[86] N. L. Schryer, "A tutorial on galerkin's method, using b-splines, for solving differential equations," Tech. Rep. 52, Bell Laboratories, Murray Hill, NJ, USA, Sept. 1976.

[87] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced basis methods for partial differential equations: an introduction*, vol. 92 of *MS&A – Modeling, Simulation and Applications*. New York, NY, USA: Springer, 2016.

[88] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. SpringerBriefs in Mathematics, Basel, Switzerland: Springer Int. Publishing AG, 2016.

[89] P. Benner, M. Ohlberger, A. Patera, G. Rozza, and K. Urban, eds., *Model reduction of parametrized systems*, vol. 17 of *Modeling, Simulation and Applications (MS&A)*. Basel, Switzerland: Springer Int. Publishing AG, 2017.

[90] X. Li, P. Li, and L. T. Pileggi, "Parameterized interconnect order reduction with explicit-and-implicit multi-parameter moment matching for inter/intra-die variations," in *Proc. ICCAD-2005 Computer-Aided Design IEEE/ACM International Conference on*, pp. 806–812, Nov. 6–10, 2005.

[91] Y.-T. Li, Z. Bai, Y. Su, and X. Zeng, "Model order reduction of parameterized interconnect networks via a two-directional arnoldi process," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, pp. 1571–1582, Sept. 2008.

[92] Y. Liu, L. Pileggi, and A. Strojwas, "Model order-reduction of RC(L) interconnect including variational analysis," in *Proc. 36th IEEE/AMC Des. Autom. Conf.*, pp. 201–206, 1999.

[93] F. Ferranti, G. Antonini, T. Dhaene, and L. Knockaert, "Guaranteed passive parameterized model order reduction of the partial element equivalent circuit (PEEC) method," *IEEE Trans. Electromagn. Compat.*, vol. 52, pp. 974–984, Nov. 2010.

[94] F. Ferranti, G. Antonini, T. Dhaene, L. Knockaert, and A. E. Ruehli, "Physics-based passivity-preserving parameterized model order reduction for PEEC circuit analysis," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 1, pp. 399–409, Mar. 2011.

[95] F. Ferranti, M. Nakhla, G. Antonini, T. Dhaene, L. Knockaert, and A. E. Ruehli, "Interpolation-based parameterized model order reduction of delayed systems," *IEEE Trans. Microw. Theory Tech.*, vol. 60, pp. 431–440, March 2012.

[96] K. C. Sou, A. Megretski, and L. Daniel, "A quasi-convex optimization approach to parameterized model order reduction," in *Proc. 42nd Design Automation Conference*, pp. 933–938, June 2005.

[97] K. C. Sou, A. Megretski, and L. Daniel, "A quasi-convex optimization approach to parameterized model order reduction," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, pp. 456–469, Mar. 2008.

[98] P. K. Gunupudi, R. Khazaka, M. S. Nakhla, T. Smy, and D. Celo, "Passive parameterized time-domain macromodels for high-speed transmission-line networks," *IEEE Trans. Microw. Theory Tech.*, vol. 51, pp. 2347–2354, Dec. 2003.

[99] K. Mohaghegh, M. Striebel, E. ter Maten, and R. Pulch, "Nonlinear model order reduction based on trajectory piecewise linear approach: Comparing different linear cores," in *Scientific Computing in Electrical Engineering SCEE 2008* (J. Roos and L. R. Costa, eds.), Mathematics in Industry, pp. 563–570, Berlin, Heidelberg, Germany: Springer-Verlag, 2010.

[100] M. J. Rewieński and J. White, "Model order reduction for nonlinear dynamical systems based on trajectory piecewise-linear approximations," *Linear Algebra and its Applications*, vol. 415, pp. 426–454, June 2006.

[101] M. J. Rewieński, *A Trajectory Piecewise-Linear Approach to Model Order Reduction of Nonlinear Dynamical Systems*. PhD thesis, Massachusetts Institute of Technology MIT, Cambridge, MA, USA, June 2003.

[102] K. Willcox, "Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition," *Comput. Fluids*, vol. 35, pp. 208–226, Feb. 2006.

[103] R. Zimmermann and K. Willcox, "An accelerated greedy missing point estimation procedure," *SIAM J. Sci. Comput.*, vol. 38, pp. A2827–A2850, Nov. 2016.

[104] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, "An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations," *Comptes Rendus Mathématique*, vol. 339, pp. 667–672, Nov. 2004.

[105] Z. Drmač and S. Gugercin, "A new selection operator for the discrete empirical interpolation method–improved a priori error bound and extensions," *SIAM J. Sci. Comput.*, vol. 38, no. 2, pp. A631–A648, 2016.

[106] D. Wirtz, D. C. Sorensen, and B. Haasdonk, "A posteriori error estimation for DEIM reduced nonlinear dynamical systems," *SIAM J. Sci. Comput.*, vol. 36, no. 2, pp. A311–A338, 2014.

[107] G. H. Golub and C. F. van Van Loan, *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins Univ. Press, fourth ed., 2013.

[108] A. Radermacher and S. Reese, "POD-based model reduction with empirical interpolation applied to nonlinear elasticity," *Int J Numer Meth Eng*, vol. 107, pp. 477–495, Dec. 2016.

[109] T. Henneron and S. Clénet, "Application of the PGD and DEIM to solve a 3-D non-linear magnetostatic problem coupled with the circuit equations," *IEEE Trans. Magn.*, vol. 52, pp. 1–4, Dec. 2016.

[110] Y. Zhou, E. Gad, M. S. Nakhla, and R. Achar, "Structural characterization and efficient implementation techniques for a-stable high-order integration methods," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 1, pp. 101–108, 2012.

[111] M. R. Rufuie, E. Gad, M. Nakhla, and R. Achar, "Generalized hermite polynomial chaos for variability analysis of macromodels embeddedin nonlinear circuits," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 4, no. 4, pp. 673–684, 2014.

[112] N. Loehr, *Advanced Linear Algebra*. Boca Raton, FL, USA: CRC Press, 2014.

[113] S. Roman, *Advanced linear algebra*. Graduate texts in mathematics, New York, NY, USA: Springer, third ed., 2007.

[114] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge ; New York: Cambridge University Press, 1999.

[115] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM / Cambridge University Press, 2000.

[116] D. C. Lay, S. R. Lay, and J. McDonald, *Linear Algebra and its Applications - (5ed)*. Boston: Pearson, 5th ed ed., 2016.

Appendix A

Fundamental Notions

A few fundamental notions from linear algebra related to the subject of this thesis that are repeatedly used are briefly formalized. For more details any fundamental linear algebra references, e.g. [112–116] to name a few, can be consulted with.

Definition A.1. ***subspace:*** *A subspace of a vector space $\mathbb{R}^n$ is a subset $\mathbf{V}$ of $\mathbb{R}^n$ that has three following properties and hence, is is a vector space in its own right.*

- *It includes the zero vector; it is $\mathbf{0} \in \mathbf{V}$,*
- *$\forall\ \mathbf{v}_i, \mathbf{v}_j \in \mathbf{V}$, we have $\mathbf{v}_i + \mathbf{v}_j \in \mathbf{V}$,*
- *$\forall\ \mathbf{v}_i \in \mathbf{V},\ \alpha \in \mathbb{R}$, we have $\alpha \mathbf{v}_i \in \mathbf{V}$,*

Definition A.2. ***span:*** *Given a set of vectors $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$, the set of all linear combinations of these vectors is a subspace referred to as the "span" of $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$. It is*

$$\text{span}\,\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\} \;=\; \left\{ \mathbf{v} \;\middle|\; \mathbf{v} = \sum_{i=1}^{m} \beta_i \mathbf{v}_i \quad \forall \beta_i \in \mathbb{R} \right\} \subset \mathbb{R}^n \tag{A.1}$$

Definition A.3. ***Column Spaces of a Matrix:*** *Let $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$ be an $n \times m$ matrix. The* colsp *of matrix $\mathbf{V}$ is the set of all linear combinations of the*

columns of matrix $\mathbf{V}$*. Hence, it is*

$$\text{colsp}(\mathbf{V}) = \text{span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\} \tag{A.2}$$

Definition A.4. ***Basis:*** *Given a subspace defined as* $\text{span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\} \subset \mathbb{R}^n$*, the set of vectors* $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$ *are a basis for the subspace if they are a linearly independent set.*