

Numerical methods for accelerating transient simulation of dense parasitic RC networks

Citation for published version (APA):

Dang, N. T. K. (2019). *Numerical methods for accelerating transient simulation of dense parasitic RC networks*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mathematics and Computer Science]. Technische Universiteit Eindhoven.

Document status and date:

Published: 01/10/2019

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Numerical methods for accelerating transient simulation of dense parasitic RC networks

Citation for published version (APA):

Dang, N. T. K. (2019). Numerical methods for accelerating transient simulation of dense parasitic RC networks
Eindhoven: Technische Universiteit Eindhoven

Document status and date:

Published: 01/10/2019

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Numerical methods for accelerating transient simulation of dense parasitic RC networks

Nhung T.K. Dang

Cover design by Nhung Dang, based on the picture from *berya113* (istockphoto.com)

A catalogue record is available from the Eindhoven University of Technology Library

ISBN: 978-90-386-4867-5

Copyright © 2019 by T.K.N. Dang

All rights are reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of the author.

Printed by: Gildeprint - The Netherlands

**Numerical methods for accelerating transient simulation of
dense parasitic RC networks**

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een
commissie aangewezen door het College voor
Promoties, in het openbaar te verdedigen
op dinsdag 1 oktober 2019 om 13:30 uur

door

Nhung T. K. Dang

geboren te Tayninh, Vietnam

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr. M.A. Peletier
promotor:	prof.dr. W.H.A Schilders
copromotor:	dr. J.M.L Maubach
	dr. J. Rommes (Mentor Graphics, France)
leden:	dr.ir. N.P. van der Meijs (TU Delft, The Netherlands)
	prof.dr.ir. C. Vuik (TU Delft, The Netherlands)
	prof.dr. G. Ciuprina (Polytechnic University Bucharest)
	dr. M.E. Hochstenbach

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Contents

1	Introduction	5
1.1	Motivation	5
1.2	Approaches and results - outline of the thesis	7
2	Mathematical models for electronic circuits with parasitics	11
2.1	Introduction	11
2.2	Formulation of network equations	11
2.2.1	Basic elements and their characteristic equations	12
2.2.2	Topological constraints	12
2.2.3	Modified Nodal Analysis	13
2.3	The network with voltage or/and current sources	15
2.4	RC circuits with DAE index 1	23
2.5	Incomplete network issues	25
2.6	Circuit simulation	31
2.7	Round-off errors and input errors	32
2.8	Summary	33
3	Mathematical analysis	35
3.1	Introduction	35
3.2	The net current difference $I_p^{\text{orig}} - I_p^{\text{move}}$	36
3.2.1	The net current difference of a transmission line circuit	36
3.2.2	The net current difference of general RC circuits	41
3.3	Estimate for net current difference $ I_q^{\text{orig}} - I_q^{\text{move}} $	42
3.3.1	An estimate for $ I_q^{\text{orig}} - I_q^{\text{move}} $ using the 2-norm	43
3.3.2	An estimate for $ I_q^{\text{orig}} - I_q^{\text{move}} $ using ∞ -norm	45
3.3.3	An estimate of $\frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt}$	45
3.4	An upper bound for $\ \mathbf{v} - \hat{\mathbf{v}}\ $	46
3.5	Estimates for $\ \mathbf{C}\ _\infty$, $\ \Delta\mathbf{C}\ _\infty$, $\ \mathbf{1}_p^T \mathbf{C}\ _\infty$ and $\ \mathbf{1}_p^T \Delta\mathbf{C}\ _\infty$	54
3.5.1	The value of $\ \mathbf{C}\ _\infty$	54
3.5.2	The value of $\ \Delta\mathbf{C}_{\text{move}}\ _\infty$ and $\ \Delta\mathbf{C}_{\text{del}}\ _\infty$	55
3.5.3	The value of $\ \Delta\mathbf{C}_{\text{split}}\ _\infty$	56
3.5.4	Calculation of $\ \mathbf{1}_k^T \mathbf{C}\ _\infty$	57
3.5.5	Calculation of $\ \mathbf{1}_q^T \Delta\mathbf{C}_{\text{move/del}}\ _\infty$ and $\ \mathbf{1}_q^T \Delta\mathbf{C}_{\text{split}}\ _\infty$	58
3.6	An example system $\mathbf{C}\mathbf{x}' + \mathbf{G}\mathbf{x} = \mathbf{b}$ with exact solution	60

3.7	Conclusion	67
4	SplitC and MoveC: splitting and moving coupling capacitors	69
4.1	Introduction	69
4.2	The MoveC method	69
4.3	Error analysis for global and inter-net currents	73
4.3.1	Global error in term of net current of MoveC/SplitC	74
4.3.2	The delay error	77
4.4	Numerical results	77
4.5	Conclusion	80
5	SelectC: switching capacitors off and on at demand	81
5.1	Introduction	81
5.2	The SelectC method	81
5.3	SelectC error estimate	83
5.4	Numerical results	83
5.5	Conclusions and outlook	85
6	Conclusions and Recommendations	89
6.1	Conclusions	89
6.2	Recommendations	90

Bibliography

Nomenclature

$\mathbb{R}^{N \times M}$	the set of real matrices of dimension $N \times M$
\mathbb{R}^N	the set of real vectors of dimension N
G	a graph associated with the nonzero pattern of the conductance matrix
C	a graph associated with the nonzero pattern of the capacitance matrix
\mathcal{G}	original conductance matrix with <i>gnd</i> and <i>pows</i> node, unpartitioned
\mathcal{C}	original capacitance matrix with <i>gnd</i> and <i>pows</i> node, unpartitioned
\mathbf{G}	partitioned conductance matrix
\mathbf{C}	partitioned capacitance matrix
p, q	index of connected component of G
P	number of matrix partions or connected components
dt, h	time-step length of a numerical integration method
t	variable of time
\mathbf{J}	Jacobian matrix of dimension $N \times N$
$I(t) \in \mathbb{R}^{n_I}$	branch currents
n_I	number of branches
n_V	number of non-grounded nodes
$\mathbf{0}/\mathbf{1}$	matrix or vector zero/one
\mathbf{s}	function of input
s	scaled time
$nnz(\mathbf{A})$	number of non-zero elements of the matrix/vector \mathbf{A}
\mathbf{G} -net	diagonal block of \mathbf{G}
G -component	connected subgraph of G
<i>intra</i>	capacitors inside G -net
<i>inter</i>	coupling capacitors/capacitors between two G -nets
<i>extra</i>	capacitors connected to reference node

Acronyms

AC	Alternating Current
BE	Backward Euler method
DAE	Differential Algebraic Equation
KCL	Kirchhoff's Current Law
KVL	Kirchhoff's Voltage Law
MNA	Modified Nodal Analysis
ODE	Ordinary Differential Equation
DC	Direct Current
SPICE	Simulation Program with Integrated Circuit Emphasis
KKT	Karush–Kuhn–Tucker or saddle point system
PDI/PDV	Positive Definite system obtained when the power terminals are excited with current (I)/voltage (V) sources
SPD	Symmetric Positive Definite
BCE	Branch Constitutive Equations
PSD	Positive Semi-Definite

Chapter 1

Introduction

1.1 Motivation

Ever since its earliest days, evolutions in microelectronics have followed the trend of miniaturization, also well-known as Moore's law. Moore's law is the observation that the number of transistors that can be planted on an integrated circuit would double approximately every two years. The continuous increase in the number of transistors as well as electronic components integrated into a chip results in circuits that are smaller, faster, and cheaper than its predecessor. In short, the cost advantage and performance increase with each Integrated Circuit (IC) generation. However, in terms of circuit design, the high density of components (i.e. smaller and denser microelectronic components) cause cross-component interference. On the other hand, modern ICs are not only analogue or digital circuits but also mixed-signal integrated circuits. This diversification combined with the trend of miniaturization makes the design and manufacturing process even more complex. Due to the growth in the complexity of modern IC design, a Design and Verification (D&V) phase is essential to be carried out prior to manufacturing and deployment to discover undesirable behaviors and thus minimize the possibility of producing faulty devices, saving time and money. Thus a reliable, accurate and fast D&V phase is the main factor for enabling new technologies timely and competitive.

In order to meet the D&V challenges faced by the semiconductor industry, hundreds of Computer Aided Design (CAD) tools developed by the Electronic Design Automation (EDA) industry are used from early stage concept to manufacturing. As an example, SPICE (Simulation Program with Integrated Circuit Emphasis) is a CAD tool that helps to carry out the circuit simulations for such challenges of circuit complexity aforementioned. Nevertheless, most current EDA methods and their supported tools do not very easily integrate with mathematical research prototypes. Started in 2014, the ASIVA14 project (Analog SIMulation and Variability Analysis for 14 nm designs) aims to develop advanced numerical techniques which are imperative to address present-day challenges in the electronics industry, see project page [2]. The project is a joint work between Eindhoven University of Technology (TUE), Netherlands and the Mentor Graphics in France, funded by the European Marie Skłodowska-Curie Actions research grant. In particular, the ASIVA14 project

addresses three topics:

1. Speeding up simulations of a certain class of periodic circuits [10],
2. Speeding up simulations of large-scale circuits,
3. Speeding up circuit's variability analyzes [43].

This thesis focuses on the second topic, the acceleration of large-scale circuit simulations.

Due to the ever denser packed components on the ICs, interconnects and parasitic effects have become more dominant. It is crucial to take these density induced effects into account. This requires the simulation of very large scale electrical networks, with a very large number of electrical and parasitic elements. This simulation is usually time-consuming or infeasible for standard circuit simulation tools.

Very Large Scale Integration (VLSI) chips with interconnect layouts may contain up to millions of electrical elements, including extracted parasitic RLC elements and linear/nonlinear devices. Numerical simulation of such large networks can therefore become very time consuming or even infeasible. Consequently, the aim of this study is accelerating/enabling the transient simulation of large parasitic networks. Model Order Reduction (MOR) methods are used to accelerate simulation by reducing the size of the original model. Multiple MOR approaches have been introduced, for instance Krylov subspaces [16, 15, 26, 22, 5], balanced truncation methods [1, 35], and elimination methods [40, 12]. However, existing MOR methods can produce dense reduced models that become more expensive to simulate than the original systems. Moreover, for multi-terminal networks MOR is inefficient because of generating large reduced models and/or dense models such as when using PRIMA [33].

Many techniques have been proposed to deal with aforementioned outstanding issues [41]. ReduceR [36], SparseRC [27] and TurboMOR-RC [34] are presented and share the common goals which are creating sparse reduced models and working efficiently with multi-terminal networks. The proposed method, unlike the aforementioned MOR methods, does not reduce the dimension of the system matrices, therefore, it does not face the problems related to multi-terminal networks and creating dense resulting matrices as MOR methods. We refer to it as a simplification method. We do not provide a comparison between our methods and other MOR methods because methods such as SVD MOR [14] and ESVD MOR [29] consider and modify the transfer function, which our methods do not do (instead, our methods concentrate on the time-domain transient simulation). Also, ReduceR deals with R networks while ours modify C networks, sparseRC [27] and TurboMOR [34] reduce the sub-nets of RC networks. A comparison would require the three methods to be combined with sub-net reduction, which is not the scope of the current research. This thesis proposes three new methods for the speeding up of the transient RC network simulation with parasitic capacitors. The considered speed up/acceleration methods are named:

1. SplitC;
2. MoveC;
3. SelectC,

and they are briefly described in section 1.2.

The methods deal with linear and nonlinear circuits. The linear circuits include only linear resistors and capacitors, whereas the nonlinear circuits may include diodes. To speed up the simulation time our three methods mentioned above focus on the reduction of non-zeros in \mathbf{C} , i.e. on the (re-)removal of capacitances or even on the dynamic capacitor selection per time-step.

The methods lead to a reduction of the transient simulation time. Their accuracy can be controlled by a chosen threshold. In the worst case scenario (i.e. maximize the speed-up over accuracy), MoveC and SelectC will turn out to be preferable over SplitC.

Last but not least, also the circuit representation and modification must be addressed. Firstly, our examples (concerning both industrial and academic examples) are based on matrices whose entries are extracted from physical circuits. For example, the resistor and capacitor sub-networks are saved in the forms of capacitance matrix and resistance matrix, respectively. This representation is round-off error sensitive because, for instance, the values of coupling capacitance can be extremely small (roughly between 10^{-14} and 10^{-25}). Therefore, matrix operations could change the non-zero pattern of matrices or create components which are not present in the physical circuit. Worse, our industrial based inputs \mathbf{C} and \mathbf{G} are not even symmetric for some examples. As a consequence, the transient simulation time increases (the Matlab "backslash" solver implicitly selects different algorithms for symmetric and asymmetric systems).

Very important is that our circuits do consist of R and C components which are left *after removing inductors, diodes, transistors, etc.* from certain industrial networks. Thus, our RC networks can consist of many resistor-based connected components and its related conductivity matrix is likely to be singular. To facilitate the existence of a valid DC solution we *must modify the RC network*, which is non-trivial and will be addressed in chapter 2.

1.2 Approaches and results - outline of the thesis

The new methods presented in this thesis are based on network reduction in order to reduce the computational time needed for solving the equations $\mathbf{C}\mathbf{x}'(t) = -\mathbf{G}\mathbf{x}(t) - \mathbf{B}\mathbf{s}(t)$ deduced from linear circuits and $\mathbf{C}\mathbf{x}'(t) = -\mathbf{G}\mathbf{x}(t) - \mathbf{d}(\mathbf{x}(t)) - \mathbf{B}\mathbf{s}(t)$ deduced from nonlinear circuits. SplitC, MoveC and SelectC change \mathbf{C} taking the parasitic capacitances into account.

The thesis is organized as follows:

- In *Chapter 2*, we recapitulate the properties of basic electrical elements. We show how the DAE can be derived from physical laws (i.e. Kirchhoff's laws and the branch equations). Furthermore, as mentioned in the previous section our network is incomplete and we only work with RC matrices, we explain how to fix the incomplete RC matrices such that we get a physically and mathematically ideal RC network.
- *Chapter 3* presents general mathematical analysis for the \mathbf{C} matrices resulting from SplitC, MoveC and DeleteC. Moreover, analytical solution and mathematical analysis for transmission line circuit are also shown as simple example.

- SplitC is the method of, electrically speaking, replacing a coupling capacitor between two nodes by two coupling capacitors with the same capacitance from each node to ground. Mathematically speaking, two non-zero entries from \mathbf{C} are removed for each split coupling capacitor. Note that only coupling capacitors which are smaller than a chosen threshold are split. Thus, if we use a large threshold, there is a high probability that all coupling capacitors between nets are removed, resulting in disconnection between nets and hence accuracy loss. Therefore, to maintain the connections between nets (at least one coupling capacitor between every two nets), there is an additional option for SplitC. The option is to keep the maximum capacitances per two nets regardless of the threshold value. *Chapter 4* zooms in on the SplitC method.
- MoveC is developed to improve on SplitC. There are two MoveC approaches. In the first approach, for every two distinct connected components of G , we select a maximum-capacitance coupling capacitor, after which we move all other coupling capacitances (below a threshold) to join (add up to) the selected maximum-capacitance coupling capacitor. The second approach is identical except that we select the maximum-RC value capacitor instead of the maximum-capacitance capacitors. Moving other coupling capacitors (below a threshold) to join (add up to) the selected inter-component capacitors causes fewer non-zero entries in \mathbf{C} (i.e. $s\mathbf{C} + \mathbf{G}$ is replaced by $s\mathbf{C}_{move} + \mathbf{G}$). *Chapter 4* discusses the MoveC method.
- Regarding SelectC, for each step t_n in the transient simulation, selects the active capacitors (above a threshold) to constitute the \mathbf{C}_n capacitance matrix at that time step t_n (i.e. $s\mathbf{C} + \mathbf{G}$ is replaced by $s\mathbf{C}_n + \mathbf{G}$). A coupling capacitor is active if the current passing through it is approximately zero. Because the SelectC matrix uses a potentially different capacitance matrix per time-step, it is not possible to use off-shelf ODE simulators such as Matlab's ODE toolbox (ode23t, ode15s, etc.). We therefore wrote a Matlab research prototype time integrator based on a simple trapezoid (or θ -method) time integration which can handle the dynamic selection of capacitors. Matlab was chosen because it facilitates rapid prototyping, but even in Matlab a fast enough prototype requires specialized vectorized code for about any operation. Important issue is the assessment of the improved speed of Cselect. We present Matlab timings which compare the solution time for the full system against the solution time for the Cselect approach. To this end our simulator aims to provide the fastest possible Matlab implementation of the time integrator. Of interest to the project's participant was the potential improved speed of the Cselect approach in the industrial's partner commercially available circuit simulation software, but providing these timings was deemed to be out of scope (would mean a full C++ implementation in ELDO, an industrial circuit simulator, see [11]). Working with matrix representation rather than a graph representation of the networks (based on the provided matrix input) caused extra numerical round-off issues to pop up in many parts of the simulator. All of these issues are addressed in *Chapter 5*.
- Finally, our conclusions are presented at the end of this thesis in *Chapter 6*. In general, we conclude that the reductions lead to fewer non-zeros in \mathbf{C} which

speeds up the transient simulation without "too much" loss of accuracy in voltages and delays, depending on the industrial/academical examples at hand.

Chapter 2

Mathematical models for electronic circuits with parasitics

2.1 Introduction

We recall the formulation of network equations which is a set of Differential Algebraic Equations (DAEs) in section 2.2. In sections 2.3 and 2.4 the transformation from a system of KKT (Karush–Kuhn–Tucker or saddle point system [3, 30]) to PDV (**P**ositive **D**efinite system obtained when the power terminals are excited with **V**oltage sources) and the DAEs index-1 of our interest are discussed. In section 2.5 we address the issues of incomplete RC networks used for experiments and the technique to deal with it. The time-domain analysis for general DAEs is discussed in section 2.6. Section 2.7 mentions round-off errors appearing due to computations or extracting processes and how we minimize these effects.

Our circuits contain parasitic capacitors which are unavoidable and usually unwanted modelled capacitances (do not created by) that exist between the parts of an electronic component or circuit. Because the parasitic capacitances are not part of the circuit and in general are much smaller than the circuit's capacitors themselves, it is thought possible to modify/relocate them without changing the circuit's state (too) much. Since we are dealing with circuits which contain (as part of the circuit) the parasitic capacitors we treat them alike all other capacitances and when needed determine which are the parasitics by using a threshold τ_{cap} , capacitance smaller than τ_{cap} are assumed to be parasitics if the capacitor is between two distinct nets of a resistor network.

2.2 Formulation of network equations

The mathematical model or a set of equations is generated by network constraints. There are two types of network constraints which are branch constraints, characterized

Element	Branch equation
Resistor	$\iota_R = G \cdot V_R$
Capacitor	$\iota_C = C \cdot \frac{dV_C}{dt}$
Inductor	$V_L = L \cdot \frac{d\iota_L(t)}{dt}$
Diode	$\iota_D(t) = d(V_D(t))$

Table 2.1: Basic electronic elements and their branch equations.

by branch equations, and topology constraints known as Kirchhoff's laws. Fulfilling these constraints gives us a set of differential algebraic equations which has to be completed with appropriate initial conditions for the state variables.

2.2.1 Basic elements and their characteristic equations

The physical behaviour of basic network elements is described by the characteristic equations or current - voltage relationship (Branch Constitutive Equations (BCE)). In general, the relationship may be linear/nonlinear and may be implicitly described by an $\iota - v$ equation $\iota(t) = f(v(t))$, where f can be any function $f : \mathbb{R} \rightarrow \mathbb{R}$ [32]. A linear resistor with resistance R is described by the Ohm's law equation $v(t) = R \cdot \iota(t)$ or $\iota(t) = G \cdot v(t)$ with $G = 1/R$. On the other hand, the linear capacitor with capacitance C is characterized by a relationship between the electrical charge $q(t)$ it stores and the voltage across it, $q(t) = C \cdot v$. Capacitor charge is related to the current through it by $\iota(t) = q' = C \cdot v'(t) = C \cdot \frac{dv(t)}{dt}$. For a linear inductor, the characteristic equation is a relationship between the magnetic flux and inductor's current $\phi(t) = L \cdot \iota(t)$. The magnetic flux is related to the voltage across the inductor by $v(t) = \phi' = L \cdot \frac{d\iota(t)}{dt}$. When resistor, capacitor and inductor are nonlinear, the constants R, C and L in the characteristic functions become $R(v), C(v)$ and $L(\iota)$ respectively. An example of a nonlinear element is a diode whose element equation is $\iota(t) = I_s \cdot (e^{\frac{v(t)}{\eta V_T}} - 1)$, where e.g. $\eta \approx 1$ and $V_T := (kT/q) \approx 26$ mV at $T = 298$ K (see [32]). For simplicity we abbreviate $d(z) = I_s \cdot (e^{\frac{z(t)}{\eta V_T}} - 1)$, then $\iota(t) = d(v(t))$. The basic elements and their characteristic functions are summarized in Table 2.1. The circuit symbols for two-terminal elements (diode, linear resistor, linear capacitor and linear inductor) are shown in Fig. 2.1. The positive direction for current is from the *plus* (+) node of higher potential v^+ to the *minus* (-) node of lower potential v^- (convention).

2.2.2 Topological constraints

Topological constraints arise from the structure of the network itself, as known as Kirchhoff's Current Law (KCL) and Kirchhoff's Voltage Law (KVL). Denote branch currents $I(t) \in \mathbb{R}^{n_I}$, n_I the number of branches, branch voltages $V(t) \in \mathbb{R}^{n_V}$, n_V the number of non-grounded nodes, and node voltages $v(t) \in \mathbb{R}^{n_v}$ (the voltage difference between a node and a reference node considered as ground $v_{ref} = 0$). *Kirchhoff's Current Law* states that the algebraic sum of all the currents entering and leaving a junction (node) must be equal to zero

$$\mathbf{A} \cdot I(t) = \mathbf{0}$$

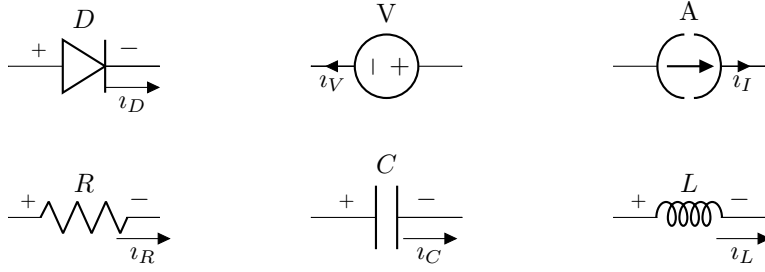


Figure 2.1: The symbols for some specific components. From left to right and bottom to top, a resistor (bottom-left), a capacitor (bottom-mid), an inductor (bottom-right), a diode (top-left), a voltage source (top-mid) and a current source (top-right).

where $\mathbf{A} \in \{-1, 0, 1\}^{n_v \times n_I}$ describes the branch-node connections of the network graph. *Kirchhoff's Voltage Law* states that for a closed loop path the algebraic sum of all the voltages around any closed loop in a circuit is equal to zero

$$\mathbf{A}^T \cdot \mathbf{v}(t) = \mathbf{V}(t).$$

A set of network equations is formed by the topological constraints and element equations. In the next section, we introduce Modified Nodal Analysis which is commonly used in industrial applications to generate the network equations.

2.2.3 Modified Nodal Analysis

The electrical network is fully described by both topological constraints, the characteristic equations and the initial conditions for the state variables. Based on these constraints, most computer programs employ one of three schemes to set up the network equations: *Sparse Tableau Approach* (STA) [23], *Nodal Analysis* (NA) [7], or *Modified Nodal Analysis* (MNA) [25, 13]. STA and NA have their own advantages and disadvantages which are specified in [32]. On the other hand, MNA gives a compromise between STA and NA [20], thus it is often used to generate network equations. In this section we concentrate on using MNA to form DAEs. To show how to formulate the MNA equations, we take the negative clamper circuit as an example. The schematic of the circuit is shown in Fig. 2.2. The circuit contains two nodes and a ground node (or reference node). The independent voltage source $V(t)$ has a positive terminal connected to node 1 and a negative terminal connected to the ground node. The source signal is known. The capacitor is connected to the positive node 1 and the negative node 2. The diode and resistor are placed in parallel connected to the positive node 2 and the ground node.

We will formulate the MNA equations for the circuit in Fig. 2.2. For each node other than the ground node, we write the KCL with the convention that the current outgoing from a node possesses positive sign.

$$\text{At node 1: } i_{V_s} + i_C = 0$$

$$\text{At node 2: } -i_C + i_D + i_R = 0$$

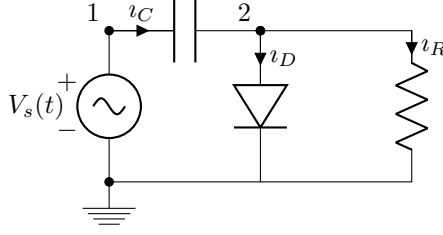


Figure 2.2: Schematic of negative diode clamper circuit.

with the provided BCEs in Table 2.1 and observing that:

$$\begin{aligned} V_C(t) &= v_1(t) - v_2(t) \\ V_D(t) &= v_2(t) \\ V_R(t) &= v_2(t) \end{aligned}$$

because the ground node has zero voltage. The equations for each node become

$$\begin{aligned} \text{At node 1: } i_{V_s} + C \cdot \frac{d(v_1 - v_2)}{dt} &= 0, \\ \text{At node 2: } -C \cdot \frac{d(v_1 - v_2)}{dt} + d(v_2) + \frac{v_2}{R} &= 0. \end{aligned}$$

At this point there are two equations but three unknowns. The last equation is the constitutive equation for the voltage source, thus $v_1(t) = V_s(t)$ leading to three equations and three unknowns:

$$\begin{cases} i_{V_s} + C \cdot \frac{d(v_1 - v_2)}{dt} = 0, \\ -C \cdot \frac{d(v_1 - v_2)}{dt} + d(v_2) + \frac{v_2}{R} = 0, \\ v_1(t) - V_s(t) = 0 \end{cases} \quad (2.2.1)$$

Let $\mathbf{x} = [\mathbf{v}; i_{V_s}]$ with $\mathbf{v} = [v_1; v_2]$. From now on, prime ' indicates the derivative with respect to the time variable t . Equations (2.2.1) can be rewritten from *conventional formulation of MNA* (for more detail about the formulation, see [13, 20]) as

$$\begin{aligned} &\begin{cases} \begin{bmatrix} 1 \\ -1 \end{bmatrix} C \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} v_1' \\ v_2' \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \frac{1}{R} \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} i_{V_s} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d(v_2) = 0, \\ \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} - V_s(t) = 0, \end{cases} \\ \Leftrightarrow &\begin{cases} \mathbf{A}_C \cdot C \cdot \mathbf{A}_C^T \cdot \mathbf{v}'(t) + \mathbf{A}_R \cdot \frac{1}{R} \cdot \mathbf{A}_R^T \cdot \mathbf{v}(t) + \mathbf{A}_V \cdot i_{V_s} + \mathbf{A}_I \cdot d(v_2) = 0, \\ \mathbf{A}_V^T \cdot \mathbf{v} - V_s(t) = 0, \end{cases} \end{aligned} \quad (2.2.2)$$

with element related incidence matrices \mathbf{A}_C , \mathbf{A}_R , \mathbf{A}_I , and \mathbf{A}_V describe the branch-current relations for capacitive branches, resistive branches, diode branches and branches of voltage sources, respectively.

Equations (2.2.1) can also be put in the form

$$\underbrace{\begin{bmatrix} C & -C & 0 \\ -C & C & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} v_1'(t) \\ v_2'(t) \\ v_{V_s}'(t) \end{bmatrix}}_{\mathbf{x}'(t)} + \underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 0 & \frac{1}{R} & 0 \\ 1 & 0 & 0 \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} v_1(t) \\ v_2(t) \\ v_{V_s}(t) \end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix} 0 \\ d(v_2) \\ 0 \end{bmatrix}}_{\mathbf{d}(\mathbf{x}(t))} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} V_s(t) \\ \mathbf{s}(t) \end{bmatrix}}_{\mathbf{s}(t)} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad (2.2.3)$$

or more generally

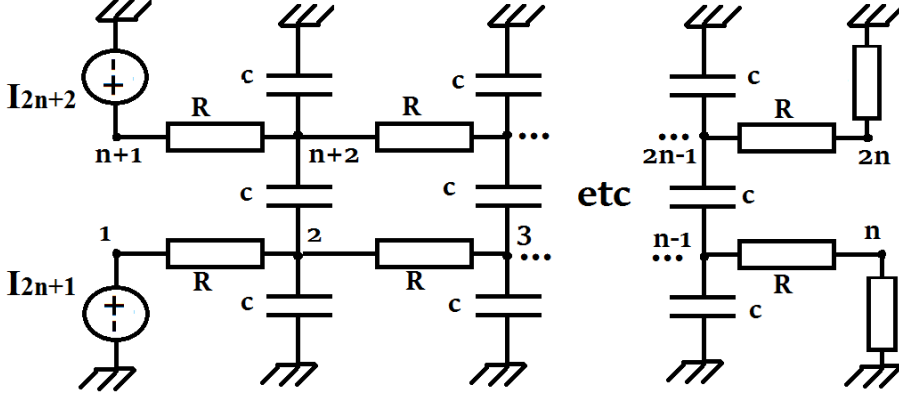
$$\mathbf{F}(\mathbf{x}'(t), \mathbf{x}(t), t) = \mathbf{C}\mathbf{x}'(t) + \mathbf{G}\mathbf{x}(t) + \mathbf{d}(\mathbf{x}(t)) + \mathbf{B}\mathbf{s}(t) = 0, \quad (2.2.4)$$

where $\mathbf{F} : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R} \mapsto \mathbb{R}^N$ and $\mathbf{x} \in \mathbb{R}^N$ is the vector solution, $\mathbf{C} \in \mathbb{R}^{N \times N}$ is the matrix of (linear) dynamic elements (in our case only capacitors), \mathbf{G} is the matrix of (linear) conductance, $\mathbf{d} : \mathbb{R}^N \mapsto \mathbb{R}^N$ is the vector function of the nonlinear elements (in our case only diodes), $\mathbf{s}(t) \in \mathbb{R}^M$ is the vector of all sources, $\mathbf{B} \in \mathbb{R}^{N \times M}$ is a full column-rank incidence matrix related to source, N and M are the dimension of the problem and the number of sources, respectively. Equation (2.2.4) is a differential algebraic equation (DAE) because it involves equations in both \mathbf{v} and its derivatives.

2.3 The network with voltage or/and current sources

For doing transient simulations with provided \mathbf{G} , \mathbf{C} matrices, we need to drive sources to externals. We could inject current or apply voltage sources or both of them. Observe that if we insert voltage sources, the linear DAE system will become a saddle point system (in this thesis called KKT system (2.3.6)). Because a KKT system also contains unknown branch currents \mathbf{y}_u , the number of equations increases by the number of voltage sources. For experiments with matrices \mathbf{G} , \mathbf{C} from industrial circuits with many terminals, this leads to a larger indefinite and potentially less well conditioned system. Thus in order to not increase the dimension of the system (in case there are voltage sources), we could either inject current sources (resulting in what we call the PDI system) or switch the system from voltage sources to current sources by elimination of the voltage sources from the KKT system. For this latter option, the resulting system is called the PDV system.

To illustrate this, consider a R-C segmented model of transmission line example.


 Figure 2.3: Transmission line example, $m = 2$ lines, each $n = 5$ nodes.

The system of equations (based on Kirchhoff's and constitutive law) with the current inputs is

$$\begin{array}{c}
 \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array}
 \begin{array}{c}
 \left[\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c}
 6 & & & & & & & & & & & & \\
 -1 & 2 & & & & & -1 & -1 & -1 & & & & \\
 -1 & & 2 & & & & & -1 & & & & & \\
 -1 & & & 2 & & & & & -1 & & & & \\
 -1 & & & & 2 & & & & & -1 & & & \\
 & & & & & & & & & & & & \\
 -1 & & -1 & & & & 2 & & & & & & \\
 -1 & & & -1 & & & & 2 & & & & & \\
 -1 & & & & -1 & & & & 2 & & & & \\
 & & & & & & & & & 2 & & &
 \end{array} \right]
 \begin{bmatrix} x' \\ y' \end{bmatrix}
 +
 \end{array}
 \end{array}$$

$$\begin{array}{c}
 \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array}
 \begin{array}{c}
 \left[\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c}
 2g & & & & & -g & & & & & -g & & \\
 g & -g & & & & & & & & & & 1 & \\
 -g & 2g & -g & & & & & & & & & & \\
 & -g & 2g & -g & & & & & & & & & \\
 -g & & -g & 2g & -g & & & & & & & & \\
 & & & -g & 2g & -g & & & & & & & \\
 & & & & -g & 2g & & & & & & 1 & \\
 & & & & & g & -g & & & & & & \\
 & & & & & -g & 2g & -g & & & & & \\
 & & & & & & -g & 2g & -g & & & & \\
 & & & & & & & -g & 2g & -g & & & \\
 & & & & & & & & -g & 2g & & &
 \end{array} \right]
 \begin{bmatrix} x \\ y \end{bmatrix}
 = 0
 \end{array}
 \end{array}$$

The 0 node is the reference node. We reorder the system such that the equation corresponding to the reference node is placed at the end. In addition, because the equation corresponding to the reference node equals zero and need to be removed to make the system consistent, we replace the row and column of the reference node in the system matrix by $[\mathbf{N}_0, \mathbf{0}]$ and $[\mathbf{N}_0^T; \mathbf{0}]$. The reordered system is as follows

$$\begin{bmatrix} \mathbf{C} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}
 \begin{bmatrix} \mathbf{x}' \\ \mathbf{x}'_{ref} \\ \mathbf{y}' \end{bmatrix}
 +
 \begin{bmatrix} \mathbf{G} & \mathbf{N}_0^T & \mathbf{N}_y^T \\ \mathbf{N}_0 & \mathbf{0} & \mathbf{0} \end{bmatrix}
 \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_{ref} \\ \mathbf{y} \end{bmatrix}
 = \mathbf{0}, \quad (2.3.1)$$

where $\mathbf{C}, \mathbf{G} \in \mathbb{R}^{10 \times 10}$, $\mathbf{N}_y^T \in \mathbb{R}^{10}$, $\mathbf{N}_0^T \in \mathbb{R}^{10}$ and $\mathbf{x} \in \mathbb{R}^{10}$ denotes the nodal voltages, $\mathbf{x}_{ref} \in \mathbb{R}$ is the reference node with zero voltage, and $\mathbf{y} \in \mathbb{R}^2$ is the current inputs.

Now, after elimination of the last equation $\mathbf{N}_0 \mathbf{x} = \mathbf{0} = \mathbf{x}_{ref}$. We can rewrite (2.3.1) as

$$\begin{aligned}
 & \underbrace{c \left[\begin{array}{cccc|cccc} 0 & & & & 0 & & & \\ & 2 & & & & -1 & & \\ & & 2 & & & & -1 & \\ & & & 2 & & & & -1 \\ & & & & 0 & & & 0 \\ \hline 0 & & -1 & & 0 & & & \\ & & & -1 & & 2 & & \\ & & & & -1 & & 2 & \\ & & & & & -1 & & 2 \\ & & & & & & 0 & 0 \end{array} \right]}_{\mathbf{C}} \mathbf{x}' + \\
 & \underbrace{g \left[\begin{array}{cccc|cccc} 1 & -1 & & & & & & \\ -1 & 2 & -1 & & & & & \\ & -1 & 2 & -1 & & & & \\ & & -1 & 2 & -1 & & & \\ & & & -1 & 2 & -1 & & \\ & & & & -1 & 2 \\ \hline & & & & & & 1 & -1 \\ & & & & & & -1 & 2 & -1 \\ & & & & & & & -1 & 2 & -1 \\ & & & & & & & -1 & 2 \end{array} \right]}_{\mathbf{G}} \mathbf{x} + \underbrace{\begin{bmatrix} 1 \\ \\ \\ \\ 1 \end{bmatrix}}_{\mathbf{N}_y^T} \mathbf{y} = \mathbf{0}, \quad t > 0
 \end{aligned}$$

which can be written

$$\begin{cases} \mathbf{C} \mathbf{x}' + \mathbf{G} \mathbf{x} + \mathbf{N}_y^T \mathbf{y} = \mathbf{0}, \\ \mathbf{N}_0 \mathbf{x} = \mathbf{0}. \end{cases}$$

This system can be solved if the currents $\mathbf{y} = \begin{bmatrix} \mathbf{I}_{s_1} \\ \mathbf{I}_{s_2} \end{bmatrix} = \mathbf{s}_y$ are provided:

$$\begin{cases} \mathbf{C} \mathbf{x}' + \mathbf{G} \mathbf{x} = -\mathbf{N}_y^T \mathbf{y}, \\ \mathbf{N}_0 \mathbf{x} = \mathbf{0}. \end{cases}$$

In case the input is voltage source. Assuming that the currents flowing through sources are all unknown, one needs replacing voltage sources (x_1, x_6 or v_1, v_6) which leads to

$$\begin{aligned}
 & c \left[\begin{array}{cccccccc|c} 0 & & & & 0 & & & & 0 \\ & 2 & & & & -1 & & & \\ & & 2 & & & & -1 & & \\ & & & 2 & & & & -1 & \\ & & & & 0 & & & & 0 \\ 0 & & & & & 0 & & & \\ & -1 & & & & & 2 & & \\ & & -1 & & & & & 2 & \\ & & & -1 & & & & & 2 \\ & & & & 0 & & & & 0 \\ \hline & & & & & 0 & & & 0 \end{array} \right] \begin{bmatrix} \mathbf{x}' \\ \mathbf{y}' \end{bmatrix} + \\
 & g \left[\begin{array}{cccccccc|c} 1 & -1 & & & & & & & 1 \\ -1 & 2 & -1 & & & & & & \\ & -1 & 2 & -1 & & & & & \\ & & -1 & 2 & -1 & & & & \\ & & & -1 & 2 & -1 & & & \\ & & & & -1 & 1 & & & \\ & & & & & & 1 & -1 & \\ & & & & & & -1 & 2 & -1 \\ & & & & & & & -1 & 2 & -1 \\ & & & & & & & & -1 & 1 \\ \hline 1 & & & & & & & & & 1 \end{array} \right] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \hline V_{s_1} \\ V_{s_2} \end{bmatrix} \Leftrightarrow
 \end{aligned}$$

$$\begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}' \\ \mathbf{y}' \end{bmatrix} + \begin{bmatrix} \mathbf{G} & \mathbf{N}_s^T \\ \mathbf{N}_s & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{s}_x \end{bmatrix} \quad (2.3.4)$$

where $\mathbf{s}_x = \begin{bmatrix} \mathbf{V}_{s_1} \\ \mathbf{V}_{s_2} \end{bmatrix}$ are the voltage sources, and $\mathbf{y} = \begin{bmatrix} \iota_{s_1} \\ \iota_{s_2} \end{bmatrix}$ now denotes unknown currents. Thus, if no capacitors are connected to a source (as happens in our example here) elimination of $\mathbf{N}_s \mathbf{x} = \mathbf{s}_x$ from (2.3.4) leads to elimination of \mathbf{y} from

$$\begin{cases} \mathbf{C} \mathbf{x}' + \mathbf{G} \mathbf{x} + \mathbf{N}_s^T \mathbf{y} = \mathbf{0}, \\ \mathbf{N}_s \mathbf{x} = \mathbf{s}_x, \end{cases}$$

which leads to a new PDV right-hand side without derivative \mathbf{s}'_x . However, had there been capacitors connected to $v_1 = s_1$ or $v_6 = s_2$ then (2.3.4) would have been

$$\begin{bmatrix} \mathbf{C} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x}' \\ \mathbf{y}' \end{bmatrix} + \begin{bmatrix} \mathbf{G} & \mathbf{N}_s^T \\ \mathbf{N}_s & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{s}_x \end{bmatrix}$$

and elimination of \mathbf{y} would have caused a term \mathbf{s}'_x in the new PDV right-hand side.

We now show how to switch the KKT system to the PDV system. The KKT system of equations (for the sake of simplicity the non-linear element such as diode

$\mathbf{d}(\mathbf{x}(t))$ in (2.2.4) is omitted) is:

$$\begin{cases} \mathbf{C}\mathbf{x}' + \mathbf{G}\mathbf{x} + \mathbf{N}_s^T \mathbf{y} = 0, & t_0 \in (0, T], \\ \mathbf{N}_0 \mathbf{x} = \mathbf{0}, \end{cases} \quad (2.3.6)$$

with $\mathbf{y} = \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}^* \end{bmatrix}$ all branch currents, \mathbf{y}_0 related to 0V source and \mathbf{y}^* related to other branch currents, $\mathbf{N}_s^T = [\mathbf{N}_0^T, \mathbf{N}_y^T]$ including those related to 0V vertices \mathbf{N}_0^T and known branch currents \mathbf{N}_y^T . Note that the term $\mathbf{N}_s^T \mathbf{y}$ in (2.3.6) generate the term $\mathbf{B}\mathbf{s}(t)$ in (2.2.4). Input sources such as \mathbf{y} will be functions of time multiplying with a frequency f , and our time-end of interest is a few periods from the start, i.e., $T = k/f$ with typical $k \in [1, 5]$.

System (2.3.6) turns out to be a DAE (see later) and of course needs an initial solution

$$\mathbf{x}(0) = \mathbf{x}_0, \mathbf{y}(0) = \mathbf{y}_0,$$

to facilitate a unique solution (because it contains time derivatives). The initial solution is addressed in equation (2.3.11). Detailed attention is given to these aspects because moving/deactivating capacitors (or resistors) changes \mathbf{C} and or \mathbf{G} and can, therefore, have unintended effects on the solutions of (2.3.6) as commented on later.

In practice, many (most) of the currents flowing through sources are not known (unknown branch currents introduced by voltage sources). Assume a part of the branch currents $\mathbf{y}^* = \begin{bmatrix} \mathbf{y}_u \\ \mathbf{y}_s \end{bmatrix}$ are not known, currents \mathbf{y}_u related to other to be applied voltage sources and \mathbf{y}_s related to input current sources:

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_u \\ \mathbf{y}_s \end{bmatrix}.$$

Split $\mathbf{N}_s^T = [\mathbf{N}_x^T, \mathbf{N}_y^T]$ related to $[\mathbf{y}_0; \mathbf{y}_u]$ respectively \mathbf{y}_s , $\mathbf{N}_x^T = [\mathbf{N}_0^T, \mathbf{N}_u^T]$ and require $\mathbf{N}_x \mathbf{x} = \mathbf{s}_x$ (where \mathbf{s}_x also contain the 0V "source"). Note that by construction $\mathbf{N}_x \mathbf{N}_y^T = \mathbf{0}$ and $\mathbf{N}_y \mathbf{N}_x^T = \mathbf{0}$ as well as $\mathbf{N}_y^T \mathbf{y}_s = \mathbf{s}_y$ and $\mathbf{N}_x \mathbf{x}_s = \mathbf{s}_x$ (the latter two relation are not symmetric).

If needed, for the sake of simplicity we will assure an order of the \mathbf{x} degrees of freedom (dofs) $\mathbf{x} = [\mathbf{x}^*; \mathbf{x}_s]$ and write $\mathbf{N}_x \mathbf{x} := [\mathbf{0}, \mathbf{N}_x][\mathbf{x}^*; \mathbf{x}_s] = \mathbf{N}_x \mathbf{x}_s$. Thus with currents and voltage sources combined the system of equations is a saddle point system (KKT) in \mathbf{x}^* and \mathbf{x}_s dofs:

$$\begin{cases} \mathbf{C}\mathbf{x}' + \mathbf{G}\mathbf{x} + \mathbf{N}_x^T \mathbf{x}_s = -\mathbf{N}_y^T \mathbf{y}_s = -\mathbf{s}_y, & t_0 \in (0, T], \\ \mathbf{N}_x \mathbf{x}_s = \mathbf{s}_x. \end{cases}$$

In the case that $\mathbf{N}_x = \mathbf{N}_0$ the terms with \mathbf{N}_x are usually omitted (implicit elimination of the 0V nodes).

If desired, the voltage sources can be eliminated:

$$\begin{aligned}
 \mathbf{C}\mathbf{x}' + \mathbf{G}\mathbf{x} + \mathbf{N}_x^T \mathbf{x}_s &= -\mathbf{N}_y^T \mathbf{y}_s \\
 \Rightarrow \mathbf{N}_x \mathbf{C}\mathbf{x}' + \mathbf{N}_x \mathbf{G}\mathbf{x} + \underbrace{\mathbf{N}_x \mathbf{N}_x^T}_{\mathbf{I}} \mathbf{x}_s &= \underbrace{-\mathbf{N}_x \mathbf{N}_y^T \mathbf{y}_s}_0 \\
 \Rightarrow \mathbf{C}\mathbf{x}' + \mathbf{G}\mathbf{x} + \mathbf{N}_x^T (-\mathbf{N}_x \mathbf{C}\mathbf{x}' - \mathbf{N}_x \mathbf{G}\mathbf{x}) &= -\mathbf{N}_y^T \mathbf{y}_s \\
 \Rightarrow (\mathbf{I} - \underbrace{\mathbf{N}_x^T \mathbf{N}_x}_{\mathbf{M}_x}) \mathbf{C}\mathbf{x}' + (\mathbf{I} - \underbrace{\mathbf{N}_x^T \mathbf{N}_x}_{\mathbf{M}_x}) \mathbf{G}\mathbf{x} &= -\mathbf{s}_y \\
 \Rightarrow \underbrace{(\mathbf{I} - \mathbf{M}_x)}_{\mathbf{S}_x} \mathbf{C}\mathbf{x}' + \underbrace{(\mathbf{I} - \mathbf{M}_x)}_{\mathbf{S}_x} \mathbf{G}\mathbf{x} &= -\mathbf{s}_y \\
 \\
 \Leftrightarrow \mathbf{S}_x \mathbf{C}\mathbf{S}_x \mathbf{x}' + \mathbf{S}_x \mathbf{G}\mathbf{S}_x \mathbf{x} &= -\mathbf{s}_y - \mathbf{S}_x \mathbf{C}(\mathbf{I} - \mathbf{S}_x) \mathbf{x}' - \mathbf{S}_x \mathbf{G}(\mathbf{I} - \mathbf{S}_x) \mathbf{x} \\
 &= -\mathbf{s}_y - \mathbf{S}_x \mathbf{C}\mathbf{M}_x \mathbf{x}' - \mathbf{S}_x \mathbf{G}\mathbf{M}_x \mathbf{x} \\
 &= -\mathbf{s}_y - \mathbf{S}_x \mathbf{C}\mathbf{N}_x^T \mathbf{N}_x \mathbf{x}' - \mathbf{S}_x \mathbf{G}\mathbf{N}_x^T \mathbf{N}_x \mathbf{x} \\
 &= -\mathbf{s}_y - \mathbf{S}_x \mathbf{C}\mathbf{N}_x^T \mathbf{s}'_x - \mathbf{S}_x \mathbf{G}\mathbf{N}_x^T \mathbf{s}_x.
 \end{aligned}$$

Because $\mathbf{N}_x^T \mathbf{N}_x \mathbf{x} = \mathbf{N}_x^T \mathbf{s}_x$ addition of this term leads to

$$\mathbf{S}_x \mathbf{C}\mathbf{S}_x \mathbf{x}' + (\mathbf{S}_x \mathbf{G}\mathbf{S}_x + \mathbf{N}_x^T \mathbf{N}_x) \mathbf{x} = \mathbf{N}_x^T \mathbf{s}_x - \mathbf{s}_y - \mathbf{S}_x \mathbf{C}\mathbf{N}_x^T \mathbf{s}'_x - \mathbf{S}_x \mathbf{G}\mathbf{N}_x^T \mathbf{s}_x, \quad t \in (0, T] \quad (2.3.7)$$

when the RHS (right-hand side) can be written as

$$\mathbf{S}_x \mathbf{C}\mathbf{S}_x \mathbf{x}' + (\mathbf{S}_x \mathbf{G}\mathbf{S}_x + \mathbf{N}_x^T \mathbf{N}_x) \mathbf{x} = \mathbf{B}\mathbf{s} + \mathbf{d}\mathbf{B}\mathbf{s}', \quad t > 0 \quad (2.3.8)$$

$$\text{with } \mathbf{s} = \begin{bmatrix} \mathbf{s}_x \\ \mathbf{s}_y \end{bmatrix}, \mathbf{B}\mathbf{s} = [\mathbf{N}_x^T - \mathbf{S}_x \mathbf{G}\mathbf{N}_x^T, -\mathbf{I}] \begin{bmatrix} \mathbf{s}_x \\ \mathbf{s}_y \end{bmatrix}, \text{ and } \mathbf{d}\mathbf{B}\mathbf{s}' = [-\mathbf{S}_x \mathbf{C}\mathbf{N}_x^T, \mathbf{0}] \begin{bmatrix} \mathbf{s}'_x \\ \mathbf{s}'_y \end{bmatrix}.$$

To simplify simulation choices we chose all sources to be either currents $\mathbf{s}_x = \mathbf{0}$ (\mathbf{s}_x related to a 0V) or all voltages ($\mathbf{s}_y = \mathbf{0}$) labelled (PDI respectively PDV). The elimination yields a system with positive (semi-)definite matrices $\mathbf{S}_x \mathbf{C}\mathbf{S}_x$ and $\mathbf{S}_x \mathbf{G}\mathbf{S}_x + \mathbf{N}_x^T \mathbf{N}_x$, drawback of the elimination is the appearance of the term $-\mathbf{S}_x \mathbf{C}\mathbf{N}_x^T \mathbf{s}'_x$ which contains derivatives of the inputs signals \mathbf{s}' which typically scales with the input frequency $f \in [10^{10}, 10^{11}]$ Hertz (Hz) in our numerical test examples with f .

This cannot be avoided but roundoff errors are mitigated by time scaling the system. Assume that $\mathbf{s}_x, \mathbf{s}_y$ are of the type $t \mapsto \mathbf{s}(2\pi f t)$ \mathbf{s} is the function of input, we can solve a time scaled system ($s := ft$)

$$\mathbf{S}_x(f \cdot \mathbf{C}) \mathbf{S}_x \mathbf{x}' + (\mathbf{S}_x \mathbf{G}\mathbf{S}_x + \mathbf{N}_x^T \mathbf{N}_x) \mathbf{x} = \mathbf{N}_x^T \widehat{\mathbf{s}}_x - \widehat{\mathbf{s}}_y - \mathbf{S}_x(f \cdot \mathbf{C}) \mathbf{N}_x^T \widehat{\mathbf{s}}'_x - \mathbf{S}_x \mathbf{G}\mathbf{N}_x^T \widehat{\mathbf{s}}_x \quad (2.3.9)$$

where $\widehat{\mathbf{s}}(t) := \mathbf{s}(2\pi s)$, $s \in (0, f \cdot T]$. Numerical tests show little difference in results, but Matlab ode/dae solvers (ode15s, ode23t, etc) require (in general) this scaling to function (a time-scale is not amongst their input parameters). We choose frequency $f \in [10^{10}, 10^{11}]$ Hz for two reasons: first, the application point of view, normally circuits are operated at a 1-100 GHz frequency and second, margin of frequency depends on time integration. As we choose simulation intervals of range $[10^{-11}, 10^{-10}]$ second (s) with $dt \in [10^{-12}, 10^{-13}]$ s, frequency should belong to $[10^{10}, 10^{11}]$ Hz.

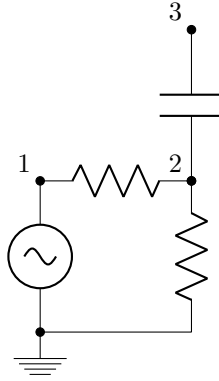


Figure 2.4: Schematic of a small circuit giving \mathbf{G} singular.

Capacitors connected to voltage sources: We notice that the resulting PDV system (2.3.7) has \mathbf{C} contributing to the RHS. The to be presented circuit reductions MoveC and SelectC alter \mathbf{C} (move or switch off capacitors) which could cause the RHS of (2.3.7) to change because $\mathbf{S}_x \mathbf{C} \mathbf{N}_x^T \neq \mathbf{S}_x \hat{\mathbf{C}} \mathbf{N}_x^T$ (where $\hat{\mathbf{C}}$ stands for \mathbf{C}_{move} , \mathbf{C}_{sel} , etc.). This is undesirable since we suppose to solve the modified problems (MoveC, SelectC, etc.) with the same sources (RHS with the $\mathbf{S}_x \mathbf{C} \mathbf{N}_x^T$ contribution). However, note that $\mathbf{S}_x \mathbf{C} \mathbf{N}_x^T$ in the RHS of (2.3.7) is a matrix which only contains the capacitors to 0V and those connected to the voltage sources. Therefore, $\hat{\mathbf{C}}$ gives identical $\mathbf{S}_x \hat{\mathbf{C}} \mathbf{N}_x^T = \mathbf{S}_x \mathbf{C} \mathbf{N}_x^T$ as long as all capacitors to 0V and connected to the voltage sources are kept in the $\hat{\mathbf{C}}$ matrix.

Equations (2.3.8) are our equations of interest, which can deal with the application of voltages as well as with the injection of currents. For our industrial examples, typical input frequencies are $f \in [10^{10}, 10^{11}]$, a typical input voltage has amplitude 1V (or 2V), and a typical injected current has amplitude 10mA (standard for current VLSI designs).

From now on, for the sake of simplicity, we consider the system of interest (2.3.8) to be

$$\mathbf{C} \mathbf{x}' + \mathbf{G} \mathbf{x} = \mathbf{b}, \quad t > 0 \quad (2.3.10)$$

where we omit \mathbf{S}_x in (2.3.8) and use $\mathbf{b} = \mathbf{B} \mathbf{s} + \mathbf{d} \mathbf{B} \mathbf{s}'$ from (2.3.8) (i.e. (2.3.10) represents system (2.3.6) with \mathbf{N}_x empty and 0V nodes eliminated).

For the initial condition, assume that at $t = 0$ the system has no active capacitors, i.e., $\mathbf{x}'(0) = 0$. However, \mathbf{x}_0 is bound to be non-zero due to voltage/current sources. Thus the initial condition is $\mathbf{G} \mathbf{x} = \mathbf{b}$. However, \mathbf{G} is likely to be singular, for instance, the small circuit shown in 2.4. \mathbf{G} is singular because the KCL equation at node 3 gives zero cross in \mathbf{G} .

Our networks are partial networks and can have so called floating nodes (singletons) which are nodes of degree of freedom one (degree of freedom is the number of nodes that a node connected to).

One can partition the nodes x_i to nodes also connected to a resistor, and nodes only connected to a capacitor. In the following, we keep using $\mathbf{C}, \mathbf{G}, \mathbf{x}, \mathbf{b}$ for the permuted system. Thus $\mathbf{x} = [\mathbf{x}_r, \mathbf{x}_c]$ and permute the system into

$$\begin{aligned} \mathbf{C}\mathbf{x}' + \mathbf{G}\mathbf{x} &= \mathbf{b} \Leftrightarrow \\ \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} \mathbf{x}' + \begin{bmatrix} \mathbf{G}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{x} &= \mathbf{b} \end{aligned}$$

where \mathbf{G}_{11} can still have multiple singular diagonal blocks.

Under the assumption that each connected components of \mathbf{C} is connected to at least one resistor, it is easy to show that \mathbf{C}_{22} is non-singular and

$$\begin{cases} \mathbf{C}_{11}\mathbf{x}'_r + \mathbf{C}_{12}\mathbf{x}'_c + \mathbf{G}_{11}\mathbf{x}_r = \mathbf{b}_1 \\ \mathbf{C}_{21}\mathbf{x}'_r + \mathbf{C}_{22}\mathbf{x}'_c = \mathbf{b}_2 \end{cases} \quad (2.3.11)$$

$$\Leftrightarrow \begin{cases} \mathbf{x}'_c = \mathbf{C}_{22}^{-1}(\mathbf{b}_2 - \mathbf{C}_{21}\mathbf{x}'_r) \\ \mathbf{C}_{11}\mathbf{x}'_r + \mathbf{G}_{11}\mathbf{x}_r = \mathbf{b}_1 - \mathbf{C}_{12}\mathbf{C}_{22}^{-1}(\mathbf{b}_2 - \mathbf{C}_{21}\mathbf{x}'_r). \end{cases} \quad (2.3.12)$$

Now finally one can solve for \mathbf{x}_r if \mathbf{G}_{11} is non-singular which is the case when all of its components are non-singular, i.e., when all its components (1) have a resistor to 0V or (2) a node connected to a (voltage or current) source. The formation of the Schur complement $\mathbf{C}_{12}\mathbf{C}_{22}^{-1}\mathbf{C}_{21}$ is (very) round-off sensitive in the sense that round-off can create connections between capacitors which should not exist. To prevent this, it is best to calculate this Schur-complement one C-node at a time. To avoid this extra complication, this thesis assumes \mathbf{C}_{22} is empty, i.e., for all test/input networks, capacitors not connected to a resistor are deleted from the network.

The round-off analysis for Schur-complement and Schur-complement one-by-one is shown as follows: Suppose \mathbf{C}_{11} is the matrix block which corresponds to all nodes disconnected to a resistor (i.e. nodes that cannot be used for DC analysis, and must be eliminated from the system of interest), which can be done using a Schur-complement update $\mathbf{C}_{22} - \mathbf{C}_{21}\mathbf{C}_{11}^{-1}\mathbf{C}_{12}$. The typical way to proceed is to calculate numerically $\mathbf{C}_{11} = \tilde{\mathbf{L}}\tilde{\mathbf{L}}^T$ by Matlab Cholesky factorization followed by updated $\mathbf{C}_{21}\tilde{\mathbf{L}}^{-T}\tilde{\mathbf{L}}^{-1}\mathbf{C}_{12}$. Because of numerical round-off $\mathbf{C}_{11} \neq \tilde{\mathbf{L}}\tilde{\mathbf{L}}^T$ (in exact arithmetic $\mathbf{C}_{11} = \mathbf{L}\mathbf{L}^T$), in general, i.e., $[\tilde{\mathbf{L}}\tilde{\mathbf{L}}^T - \mathbf{C}_{11}]_{ij} = \epsilon \cdot \|\mathbf{C}_{11}\|_{\infty}$ ($\epsilon \approx 10^{-15}$). Many (i, j) pairs are created $i \neq j$ (also $i = j$), which implies that $\mathbf{L}\mathbf{L}^T$ is related to a circuit with many extra capacitors. To avoid this modification of the circuit note that a factorization of \mathbf{C}_{11} can be computed step by step:

Let

$$\mathbf{S}_0 = \mathbf{C}_{11} = \left[\begin{array}{c|c} d_1 & \mathbf{u}_1 \\ \hline \mathbf{l}_1 & \mathbf{S}_1 \end{array} \right] \in \mathbb{R}^{m \times m}, \mathbf{l}_1 \text{ and } \mathbf{u}_1^T \in \mathbb{R}^{m-1}$$

$$\mathbf{S}_1 = \mathbf{S}_1 - \mathbf{l}_1 d_1^{-1} \mathbf{u}_1 = \left[\begin{array}{c|c} d_2 & \mathbf{u}_2 \\ \hline \mathbf{l}_2 & \mathbf{S}_2 \end{array} \right] \in \mathbb{R}^{(m-1) \times (m-1)}, \mathbf{l}_2 \text{ and } \mathbf{u}_2^T \in \mathbb{R}^{m-2}$$

$$\mathbf{S}_2 = \mathbf{S}_2 - \mathbf{l}_2 d_2^{-1} \mathbf{u}_2$$

until

$$\mathbf{S}_{m-1} = \mathbf{S}_{m-1} - \mathbf{l}_{m-1} d_{m-1}^{-1} \mathbf{u}_{m-1} \in \mathbb{R}, \mathbf{l}_{m-1} \text{ and } \mathbf{u}_{m-1} \in \mathbb{R}$$

through node m which also produce a factorization

$$\mathbf{C}_{11} = (\bar{\mathbf{L}} + \bar{\mathbf{D}})\bar{\mathbf{D}}^{-1}(\bar{\mathbf{D}} + \bar{\mathbf{U}})$$

effectively $\mathbf{C}_{11} = \widehat{\mathbf{L}}\widehat{\mathbf{L}}^T$ with $\mathbf{L} = (\bar{\mathbf{L}} + \bar{\mathbf{D}})\sqrt{\bar{\mathbf{D}}^{-1}}$ (see [30]) where

$$\bar{\mathbf{L}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \mathbf{l}_1 & 0 & \cdots & \\ & \mathbf{l}_2 & \ddots & \vdots \\ & & \ddots & \\ & & & \mathbf{l}_{m-1} & 0 \end{bmatrix}, \quad \bar{\mathbf{D}} = \begin{bmatrix} d_1 & 0 & 0 & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & \\ 0 & 0 & & & \vdots \\ \vdots & \vdots & & \ddots & 0 \\ 0 & & \cdots & 0 & d_m \end{bmatrix},$$

$$\text{and } \bar{\mathbf{U}} = \begin{bmatrix} 0 & \mathbf{u}_1 & & & \\ 0 & 0 & \mathbf{u}_2 & & \\ \vdots & \vdots & \ddots & & \\ & & & 0 & \mathbf{u}_{m-1} \\ 0 & \cdots & & 0 & \end{bmatrix}.$$

Now we create no capacitor each step but still $[\widehat{\mathbf{L}}\widehat{\mathbf{L}}^T - \mathbf{C}_{11}]_{ij} \neq 0$ at certain $i = j$. Even so, the Schur-complement one-by-one has the advantage because it is easier to compute by non-zero elements instead of computing the inverse \mathbf{C}_{11}^{-1} .

2.4 RC circuits with DAE index 1

We consider networks with inputs are currents (known inputs) or currents derived from voltage sources. RC circuits only involve capacitors and resistors and satisfy the equation

$$\mathbf{C} \cdot \mathbf{x}'(t) + \mathbf{G} \cdot \mathbf{x}(t) + \mathbf{B} \cdot \mathbf{s}(t) = 0 \quad t > 0, \quad \mathbf{x}(t_0) = \mathbf{x}_0. \quad (2.4.1)$$

where the *state* \mathbf{x} is the solution of the system driven by the input(s) $t \mapsto \mathbf{s}(t)$. Depending on \mathbf{C} the system is

- an ODE system: \mathbf{C} non-singular. In this case, the system (2.4.1) can be transformed into the explicit ODE system

$$\mathbf{x}' = \mathbf{C}^{-1}(-\mathbf{G} \cdot \mathbf{x} - \mathbf{B} \cdot \mathbf{s})$$

- a DAE system: \mathbf{C} singular. We assume that \mathbf{G} is non-singular and that $\mathbf{x}'(0) = 0$, which allows for the computation of the steady-state (DC) solution.

In the case of a DAE, knowing the index of a DAE is an important prerequisite for its consistent initialization and its numerical solution [6]. We focus first on the nilpotency index, continue with the differentiation index.

Definition 2.4.1. Nilpotency index. A nilpotent matrix is a square matrix \mathbf{N} such that $\mathbf{N}^k = \mathbf{0}$, for some positive integer k . The smallest such k is called the index of the nilpotency of \mathbf{N} (i.e. $\mathbf{N}^k = \mathbf{0}$ but $\mathbf{N}^{k-1} \neq \mathbf{0}$) [24].

Left-multiplying (2.4.1) with \mathbf{G}^{-1} leads to

$$\mathbf{G}^{-1}\mathbf{C} \cdot \mathbf{x}' + \mathbf{x} = -\mathbf{G}^{-1} \cdot \mathbf{B} \cdot \mathbf{s} \quad (2.4.2)$$

after which a Jordan decomposition [21] to $\mathbf{G}^{-1}\mathbf{C}$ leads to:

$$\mathbf{G}^{-1}\mathbf{C} = \mathbf{T}^{-1} \begin{bmatrix} \tilde{\mathbf{G}} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix} \mathbf{T}$$

where \mathbf{T} is a non-singular, time independent matrix, $\tilde{\mathbf{G}}$ is a non-singular matrix and \mathbf{N} is a nilpotent matrix. Multiplying \mathbf{T} from the left-hand side, the equation (2.4.2) becomes

$$\begin{bmatrix} \tilde{\mathbf{G}} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix} (\mathbf{T}\mathbf{x})' + \mathbf{T}\mathbf{x} = -\mathbf{T}\mathbf{G}^{-1} \cdot \mathbf{B} \cdot \mathbf{s}.$$

and after splitting the transformed state into two components

$$\mathbf{T}\mathbf{x} := \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} \text{ and } -\mathbf{T}\mathbf{G}^{-1}\mathbf{B}\mathbf{s}(t) := \begin{bmatrix} \mathbf{s}_1(t) \\ \mathbf{s}_2(t) \end{bmatrix}$$

it reduces to

$$\begin{cases} \mathbf{y}' = \tilde{\mathbf{G}}^{-1}(\mathbf{s}_1(t) - \mathbf{y}), \\ \mathbf{N}\mathbf{z}' = \mathbf{s}_2(t) - \mathbf{z}. \end{cases} \quad (2.4.3)$$

Now the index of DAE (2.4.3) is one if $\mathbf{N} = \mathbf{0}$ and is greater than one otherwise (in the case of an index greater than one we would obtain

$$\mathbf{z} = \mathbf{s}_2(t) - \mathbf{N}\mathbf{s}_2'(t) + \mathbf{N}^2\mathbf{s}_2^{(2)}(t) - \dots + (-1)^{k-1}\mathbf{N}^{k-1}\mathbf{s}_2^{(k-1)}(t)$$

after k times taking a derivative).

Example 2.4.2. Equations (2.2.3) have DAE index-1 because:

$$\begin{aligned} \mathbf{G}^{-1}\mathbf{C} &= \begin{bmatrix} 0 & 0 & 0 \\ -RC & RC & 0 \\ C & -C & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ -\frac{1}{R} & 0 & 1 \end{bmatrix} \begin{bmatrix} RC & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 \\ 1 & 0 & 0 \\ -\frac{1}{R} & \frac{1}{R} & 1 \end{bmatrix} \\ &= \mathbf{T}^{-1} \begin{bmatrix} \tilde{\mathbf{G}} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix} \mathbf{T}, \quad \text{with } \mathbf{N} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

We introduce a new definition of index for general DAEs (2.2.4) which is known as differentiation index [4, 44, 18, 17]:

Definition 2.4.3. Differentiation index. The differentiation index of (2.2.4) is the minimum number of times that all or part of (2.2.4) must be differentiated with respect to t in order to extract an ordinary differential equation (ODE) from (2.2.4).

Assumption 2.4.4. The conductivity matrix \mathbf{G} is non-singular.

We make the assumption 2.4.4 because as shown below it ensures that our system (2.4.1) is a DAE of index-1, independent on \mathbf{C} , which is needed because of our limited control of the (used) capacitors in/of \mathbf{C} .

In the case of linear constant coefficient systems (2.4.1), the differentiation index is equivalent to the nilpotency index [42] of \mathbf{N} .

Throughout this thesis we solve the DAEs of index-1. By construction we have \mathbf{C} and \mathbf{G} are positive semi-definite and by assumption 2.4.4 \mathbf{G} is non-singular. Since

\mathbf{C} is positive semi-definite, there exists an unitary matrix \mathbf{Q} and eigenvalues Λ such that

$$\mathbf{Q}^{-1}\mathbf{C}\mathbf{Q} = \begin{bmatrix} \Lambda & \\ & 0 \end{bmatrix}.$$

Consider

$$\begin{aligned} \mathbf{Q}^{-1}(\mathbf{C}\mathbf{x}' + \mathbf{G}\mathbf{x}) &= -\mathbf{Q}^{-1}\mathbf{B}\mathbf{s} \Leftrightarrow \\ \mathbf{Q}^{-1}(\mathbf{C}\mathbf{Q}\mathbf{y}' + \mathbf{G}\mathbf{Q}\mathbf{y}) &= -\mathbf{Q}^{-1}\mathbf{B}\mathbf{s} \Leftrightarrow \\ \begin{bmatrix} \Lambda & \\ & 0 \end{bmatrix} \mathbf{y}' + \hat{\mathbf{G}}\mathbf{y} &= -\mathbf{Q}^{-1}\mathbf{B}\mathbf{s} \Leftrightarrow \\ \begin{bmatrix} \Lambda & \\ & 0 \end{bmatrix} \mathbf{y}' + \begin{bmatrix} \hat{\mathbf{G}}_{11} & \hat{\mathbf{G}}_{12} \\ \hat{\mathbf{G}}_{21} & \hat{\mathbf{G}}_{22} \end{bmatrix} \mathbf{y} &= -\mathbf{Q}^{-1}\mathbf{B}\mathbf{s}. \end{aligned}$$

Let $\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}$ and $\mathbf{Q}^{-1}\mathbf{B}\mathbf{s} = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix}$ we have

$$\begin{aligned} \Leftrightarrow \begin{cases} \Lambda\mathbf{y}'_1 + \hat{\mathbf{G}}_{11}\mathbf{y}_1 + \hat{\mathbf{G}}_{12}\mathbf{y}_2 = -\mathbf{s}_1, \\ \hat{\mathbf{G}}_{21}\mathbf{y}_1 + \hat{\mathbf{G}}_{22}\mathbf{y}_2 = -\mathbf{s}_2 \end{cases} \Leftrightarrow \\ \begin{cases} \Lambda\mathbf{y}'_1 + \hat{\mathbf{G}}_{11}\mathbf{y}_1 + \hat{\mathbf{G}}_{12}\mathbf{y}_2 = -\mathbf{s}_1, \\ \mathbf{y}_2 = \hat{\mathbf{G}}_{22}^{-1}(-\mathbf{s}_2 - \hat{\mathbf{G}}_{21}\mathbf{y}_1) \end{cases} \end{aligned}$$

Differentiate the second equation one time gives ODEs. Thus, system (2.4.1) with \mathbf{G}, \mathbf{C} positive semi-definite (PSD) and \mathbf{G} non-singular is of index-1. Observe that only \mathbf{G}_{22} needs to be non-singular. However, the \mathbf{G}_{22} part of \mathbf{G} is implicitly defined by the nullspace of \mathbf{C} , which we have not analyzed (yet).

2.5 Incomplete network issues

As mentioned in the previous section, our linear DAE system containing RC matrices extracted from an incomplete network including nonlinear, dynamic elements, etc., therefore, RC matrices are likely to be ill-conditioned. Thus the DAE system could not be solved, also they lack physical behaviour/network topology. After doing some statistics on the collected RC matrices, we observe that most \mathbf{G} matrices for experiments contain many zero rows/columns, which does not satisfy assumption 2.4.4 for the linear DAE case. Therefore, to ensure assumption 2.4.4 is satisfied a process called *\mathbf{G} -regularization* is needed before applying the methods. It includes two steps: removing zero rows/columns in \mathbf{G} and component singularity fix for every component of \mathbf{G} . \mathbf{G} -regularization aims to create a new regular \mathbf{G} and corresponding \mathbf{C} without making too many changes to the properties of original RC matrices.

Inputs: \mathbf{G} and \mathbf{C} supposed to be structurally symmetric i.e, the non-zero pattern is symmetric, and vectors of *gnd* (reference node), *exts* (externals) and *inputs*. Output: non-singular $\hat{\mathbf{G}}$, $\hat{\mathbf{C}}$ with smaller size compared to \mathbf{G} and \mathbf{C} , and updated vectors of *gnd*, *exts* and *inputs* with smaller size compared to their original. The changes in the matrices and vectors after \mathbf{G} -regularization are shown in Table 2.2.

Algorithm 1 \mathbf{G} -regularization

```

1: Remove zero rows, columns in  $\mathbf{G}$  and the same ones in  $\mathbf{C}$ 
2: Update the indices of  $gnd$ ,  $exts$  and  $inputs$ 
3: Reorder  $\mathbf{G}$  and  $\mathbf{C}$  by connected components of  $G$ 
4: Update the indices of  $gnd$ ,  $exts$  and  $inputs$ 
5: for  $i = 1 : P$  do
6:   if  $comp_G(i)$  singular then
7:      $GconnectG \leftarrow$  nodes only connected to resistor(s)
8:      $GconnectC \leftarrow$  nodes connected to resistor(s) and capacitor(s)
9:      $del \leftarrow \emptyset$ 
10:    if  $GconnectG \ \& \ (\text{no } GconnectC \parallel nr(GconnectC) == 1)$  then
11:       $del \leftarrow$  the index of min degree in  $G$ 
12:    else
13:       $del \leftarrow$  the index of min degree in  $G + C$ 
14:    end if
15:  end if
16: end for
17:  $\tilde{\mathbf{G}} \leftarrow$  remove rows and columns with index  $del$  in  $\mathbf{G}$ 
18:  $\tilde{\mathbf{C}} \leftarrow$  remove rows and columns with index  $del$  in  $\mathbf{C}$ 
19: Update the indices of  $gnd$ ,  $exts$  and  $inputs$ 
20: Reorder  $\tilde{\mathbf{G}}$  and  $\tilde{\mathbf{C}}$  by connected components of  $\tilde{\mathbf{G}}$ 
21: Update the indices of  $gnd$ ,  $exts$  and  $inputs$ .
    
```

Algorithm 1 shows the singularity-fix process: inside the loop, for each singular i th component of \mathbf{G} , $comp_G(i)$, a node is deleted. The deleted-node is chosen by order of priority as follows: nodes of $comp_G(i)$ have no connection to \mathbf{C} or only 1 connection to \mathbf{C} , then we delete the node has minimum degree of freedoms (dofs) to \mathbf{G} . Otherwise, the node has minimum dofs of $\mathbf{G} + \mathbf{C}$ is chosen. The why is that we avoid to create many $\mathbf{G} + \mathbf{C}$ components (or decouple the network), thus we have to minimize the act of destroying coupling connections.

Algorithm 1 selects resistors to be deleted, independent on their resistance. In the case of multiple resistors with minimal degree the first one is chosen. In the more general case we would only need \mathbf{G}_{22} to be non-singular. Because \mathbf{G} is positive semi-definite, its major block \mathbf{G}_{22} also is. To determine its rank consider the following.

Lemma 2.5.1. *\mathbf{C} is positive semi-definite. The dimension of its nullspace is equal to the amount of its singular connected components.*

Proof. Let \mathbf{e}_k be the k th column vector of \mathbb{R}^N , $\mathbf{C} \in \mathbb{R}^{N \times N}$. Let $\mathbf{d}_{ij} := \mathbf{e}_j - \mathbf{e}_i$. By construction

$$\mathbf{C} = \sum_{cap(i,j)} \mathbf{d}_{ij} c_{ij} \mathbf{d}_{ij}^T + \sum_{cap(k,k)} \mathbf{e}_k c_{kk} \mathbf{e}_k^T \quad (2.5.1)$$

\mathbf{C} can be permuted using the order of its connected components into $\mathbf{PCP}^T := \hat{\mathbf{C}}$ where $\hat{\mathbf{C}}$ is a block-diagonal matrix. The dimension of the nullspace of $\hat{\mathbf{C}}$ is equal to

the amount of its singular blocks because each connected component is also of form

$$\hat{\mathbf{C}}_p = \sum_{\text{cap}(i,j) \text{ in } E_p} \mathbf{d}_{ij} c_{ij} \mathbf{d}_{ij}^T + \sum_{\text{cap}(k,k) \text{ in } E_p} \mathbf{e}_k c_{kk} \mathbf{e}_k^T,$$

where E_p is the set of edges belongs to component p . $\hat{\mathbf{C}}_p$ is of maximal rank if and only if there exists a capacitor to ground, i.e., $\text{cap}(k,k) \neq \emptyset$, the empty set. If $\text{cap}(k,k) = \emptyset$ the single vector in the nullspace of $\hat{\mathbf{C}}_p$ is $\sum_{k \text{ in } E_p} \mathbf{e}_k$, and the related vector in the nullspace of \mathbf{C}_p is $\sum_{k \text{ in } E_p} \mathbf{P}^T \mathbf{e}_k$. \square

Corollary 2.5.2. *Because $\mathbf{G} \in \mathbb{R}^{n \times n}$ has the same structure (2.5.1) as $\mathbf{C} \in \mathbb{R}^{n \times n}$, also $s\mathbf{C} + \mathbf{G}$ has structure (2.5.1) which implies that the $s\mathbf{C} + \mathbf{G}$ is non-singular for $s \in \mathbb{R}$ if and only if its each connected component has a capacitor or resistor to ground.*

The elimination of voltage sources in the PDV version of the system of equations creates extra capacitors and or resistors to the reference node since elimination of e.g.

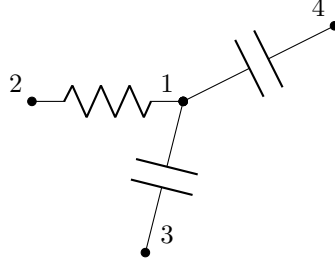


Figure 2.5: Circuit before elimination of node 1.

node 1 above (Fig. 2.5) alters the circuit to the form (Fig. 2.6)

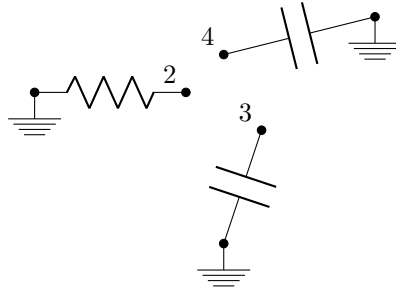


Figure 2.6: Circuit after elimination of node 1.

when the contribution resistor r_{12} between nodes 1 and 2 is altered in resistor to ground contributing $\mathbf{e}_1 \left(\frac{1}{r_{12}} \right) \mathbf{e}_1^T$, i.e., making the connected component non-singular.

Also the application of a voltage source (see Fig 2.7) in the KKT version of the system of equations leads to a singular component with a matrix representation as in (2.5.2)

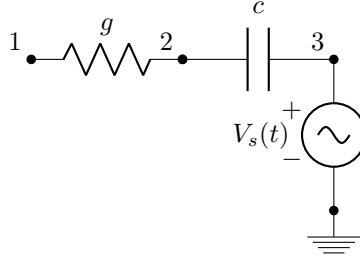


Figure 2.7: Apply a voltage source.

$$\mathbf{G} + \mathbf{C} = \left[\begin{array}{ccc|c} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 1 & 1 \\ \hline 0 & 0 & 1 & 0 \end{array} \right], \quad (2.5.2)$$

Suppose resistance and capacitance are equal to 1. The component with a connected source in (2.5.2) is non-singular because addition of $row2 := row1 + row2$, followed by $row3 := row3 + row2$ leads to

$$\left[\begin{array}{ccc|c} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & 0 \end{array} \right],$$

which is non-singular because it can be symmetrically. Permuted to unit upper triangular by swapping the last two columns/rows. Observe that the addition of a current source (version PDI) would modify (2.5.2) to

$$\left[\begin{array}{ccc} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{array} \right],$$

i.e., it keeps a singular component singular. To conclude the examination of the dimension of the nullspace of $s\mathbf{C} + \mathbf{G}$ we still have to consider (2.5.2) for the case that $g \neq c$.

$$\begin{bmatrix} g & -g \\ -g & g+c & -c \\ & -c & c \end{bmatrix}. \quad (2.5.3)$$

First consider the case of a component which is a tree, of which a simplified (chain without side-branches) version can be seen in (2.5.3) In this very simple case, replaced addition of equations ($row2 := row2 + row1$, $row3 := row3 + row2$) leads to a singular

matrix $\begin{bmatrix} g & -g \\ c & -c \\ 0 & 0 \end{bmatrix}$. In the case of a tree with a single branch, see Fig. 2.8

$$\begin{bmatrix} g & -g \\ -g & g+c_1+c_2 & -c_1 & -c_2 \\ & -c_1 & c_1 \\ & -c_2 & & c_2 \end{bmatrix}, \quad (2.5.4)$$

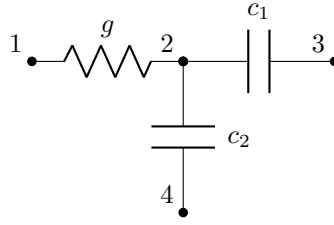


Figure 2.8: Circuit with a single branch.

after $row2 := row2 + row1$, one needs to scale the 2^{nd} equation to eliminate the $-v_2$ coefficient in the 3^{rd} row, ditto for elimination of $-c_2$ in row 4. Since the scaling constitutes a non-singular operation which does not alter the rank of the matrix, one ends up with a zero row at the end which shows that also (2.5.4) can be reduced to an upper triangular matrix, i.e.,

Corollary 2.5.3. *The PDV and KKT version of the system of equations still lead to non-singular $s\mathbf{C} + \mathbf{G}$ matrices.*

Table 2.2: \mathbf{G} , \mathbf{C} after \mathbf{G} -regularization

Ex	Matrix	Size \mathbf{G}	#0-crosses in \mathbf{G}	cond(\mathbf{G})	cond($\mathbf{G} + s\mathbf{C}$)	#comps \mathbf{G}	#comps $\mathbf{G} + \mathbf{C}$	#comps \mathbf{G} have 1 dof	#comps $\mathbf{G} + \mathbf{C}$ have 1 dof	#singular comps	Largest \mathbf{G} comps	Size largest \mathbf{G} comps	Size gnd comp
2	in	34356	7771	inf	9.1E+17	8107	1	7771	0	-	-	-	-
	bf fix	26584	0	inf	5.3E+5	336	3	0	0	336	334	1886	1
	af fix	26248	0	3.6E+7	2.9E+5	336	3	0	0	0	334	1885	1
3	in	18928	0	inf	inf	1474	197	1	0	-	-	-	-
	bf fix	18927	0	inf	inf	1473	197	0	0	1473	1226	1813	1
	af fix	17454	0	1.7E+6	2.8E+6	1473	197	199	196	0	1226	1812	1
3.5	in	6889	2	inf	inf	1475	197	1	0	-	-	-	-
	bf fix	6885	0	inf	inf	1471	197	0	0	1471	1226	1447	2
	af fix	5414	0	2.9E+6	3.6E+5	1471	198	1163	197	0	1226	1446	2
4	in	15582	40	inf	3.7E+19	1117	1	43	0	-	-	-	-
	bf fix	15539	0	inf	1.3E+8	1074	1	0	0	1074	747	265	3
	af fix	14465	0	1E+8	4.2E+6	1078	1	2	0	0	749	264	3
4.5	in	6020	227	inf	6.6E+19	1117	1	230	0	-	-	-	-
	bf fix	5790	0	inf	8.2E+6	887	1	0	0	887	854	115	3
	af fix	4903	0	1E+7	3.7E+5	887	1	296	0	0	854	114	3
6.5	in	8080	524	inf	2.6E+19	1417	1	527	0	-	-	-	-
	bf fix	7553	0	inf	1.1E+4	890	1	0	0	890	34	1039	3
	af fix	6663	0	3.8E+6	5.4E+3	890	1	45	0	0	34	1038	3
26	in	16862	44	inf	2.2E+19	152	1	44	0	-	-	-	-
	bf fix	16817	0	inf	2.2E+9	161	1	7	0	107	1	6856	1
	af fix	16710	0	1.6E+10	7.9E+8	239	1	75	0	0	1	6856	1

- **Matrix:** **in** implies as loaded from the .mat file (include gnd/powers); **bf fix** implies after removal of 0-crosses and **af fix** implies after \mathbf{G} -regularization (exclude gnd/powers)
- **size \mathbf{G} :** order of \mathbf{G}
- **#0-crosses in \mathbf{G} :** number of zero columns/rows in \mathbf{G}
- **cond(\mathbf{G})** and **cond($\mathbf{G} + s\mathbf{C}$)**: condition number of \mathbf{G} and $\mathbf{G} + s\mathbf{C}$, respectively, where $s = 1/dt = 10^{13}$
- **#comps \mathbf{G}** and **#comps($\mathbf{G} + \mathbf{C}$)**: number of components (diagonal blocks) of \mathbf{G} and $\mathbf{G} + \mathbf{C}$, respectively
- **singular comps**: number of singular components which is identical to the number of removed nodes (no gnd comp)
- **largest comp**: the i^{th} component which is the largest

- **#compsG have only 1 dof** and **#comps(G + C) have only 1 dof**: nb of 1-dof comps in **G** and **G + C**, respectively
- **size gnd comp**: size of the gnd/power component
- cells with " - " stand for uninteresting information

In table 2.2, we observe that the number of connected components in G (or diagonal blocks in \mathbf{G}) increases (especially in example 26). This is because in the regularization process the row and column in between the first and the last rows/columns of a diagonal block of \mathbf{G} could be deleted, resulting in two diagonal blocks in \mathbf{G} . Electrically speaking, when a node which is neither the first nor the last node in a connected component is deleted, two smaller connected components of G will be generated. In the process, we exclude the gnd/power node in **af fix** because we only concentrate on the properties of the matrix which will be used in the simulation. Practically, we should remove the equation(s) corresponding to reference (gnd/power) node(s) to make the systems consistent and **af fix** is the system matrix of the system of equations.

Now for simplicity, we still use the notations \mathbf{C}, \mathbf{G} , etc. for the matrices after G -regularization.

2.6 Circuit simulation

The Modified Nodal Analysis (MNA) method leads to a system of differential - algebraic equations, reformulated (2.2.4):

$$\mathbf{C}\mathbf{x}'(t) + \mathbf{G}\mathbf{x}(t) + \mathbf{d}(\mathbf{x}(t)) + \mathbf{B}\mathbf{s}(t) = 0, \quad t > 0 \quad (2.6.1)$$

The vector $\mathbf{x}(t) \in \mathbb{R}^N$ represents the unknown node voltages and branch currents of the system, and $\mathbf{B}\mathbf{s} \in \mathbb{R}^N$ is the input vector. The system (2.6.1) is usually solved from a given initial condition or DC solution $\mathbf{x}(t_0) = \mathbf{x}_0$. Usually, \mathbf{x}_0 is obtained by solving for the DC solution:

$$\mathbf{G}\mathbf{x}(t_0) + \mathbf{d}(\mathbf{x}(t_0)) + \mathbf{B}\mathbf{s}(t_0) = 0,$$

when all the dynamic elements are omitted from the circuit and a DC input (constant input) is applied. Next, starting from the initial condition \mathbf{x}_0 , the system (2.6.1) is discretized for time points $t_k \in (0, T], k = 1, \dots, T_N$, where T represents the stop time of the simulation. To this end the backward Euler formula implicit integration scheme is used at time point t_k :

$$\mathbf{C} \cdot \frac{\mathbf{x}(t_k) - \mathbf{x}(t_{k-1})}{t_k - t_{k-1}} + \mathbf{G}\mathbf{x}(t_k) + \mathbf{d}(\mathbf{x}(t_k)) + \mathbf{B}\mathbf{s}(t_k) = 0.$$

More in general (2.6.1) can be written in form

$$\mathbf{C}\mathbf{x}'(t) = -\mathbf{G}\mathbf{x}(t) - \mathbf{d}(\mathbf{x}(t)) - \mathbf{B}\mathbf{s}(t) = \mathbf{f}(\mathbf{x}, t),$$

and we implemented the integration method

$$\mathbf{C} \cdot \frac{\mathbf{x}(t_k) - \mathbf{x}(t_{k-1})}{t_k - t_{k-1}} = \theta \mathbf{f}(\mathbf{x}(t_k), t_k) + (1 - \theta) \mathbf{f}(\mathbf{x}(t_{k-1}), t_{k-1}),$$

which reduces to Euler implicit (backward) for $\theta = 1$. This leads to

$$\mathbf{F}_k(\mathbf{x}(t_k)) = 0,$$

where

$$\mathbf{F}_k(\mathbf{x}(t_k)) = \mathbf{C} \cdot \frac{\mathbf{x}(t_k) - \mathbf{x}(t_{k-1})}{t_k - t_{k-1}} - \theta \mathbf{f}(\mathbf{x}(t_k), t_k) - (1 - \theta) \mathbf{f}(\mathbf{x}(t_{k-1}), t_{k-1}),$$

which system we solve with an undamped Newton method. To approximate $\mathbf{x}(t_k)$ we set $\mathbf{x}_k^{(0)} := \mathbf{x}(t_{k-1})$ and iterate $\mathbf{x}_k^{(l+1)} := \mathbf{x}_k^{(l)} - \mathbf{J}_{\mathbf{F}}^{-1}(\mathbf{x}_k^{(l)}) \mathbf{F}(\mathbf{x}_k^{(l)})$. The multiplication with $\mathbf{J}_{\mathbf{F}}^{-1}(\mathbf{x}_k^{(l)})$ is equivalent to solving a linear system of equations, which we do with the Matlab backslash operator, a (k -independent in case of RC circuit) LU factorization of $\mathbf{J}_{\mathbf{F}}(\mathbf{x}_k^{(l)}) := \mathbf{J}_{\mathbf{F}}(\mathbf{x}(t_0))$. Most of the computation time is spent in assembling the derivative matrix $\mathbf{J}_{\mathbf{F}}(\mathbf{x}^{(l)})$ and in solving systems with it. Therefore, increasing the sparsity of \mathbf{C} and/or \mathbf{G} would potentially help to accelerate the transient simulation.

Because \mathbf{G} can be ill-conditioned our simulator can employ scaling, i.e., instead of (2.6.1) we solve $\mathbf{SCS}\mathbf{y}' + \mathbf{SGS}\mathbf{y} + \mathbf{Sd}(\mathbf{S}\mathbf{y}) + \mathbf{SBs} = \mathbf{0}$ where \mathbf{S} is a diagonal matrix such that $s_{ii} = \frac{1}{g_{ii}}$ if $g_{ii} \neq 0$ and $s_{ii} = 1$ if $g_{ii} = 0$, where g_{ii} is a diagonal entry of \mathbf{G} .

2.7 Round-off errors and input errors

The circuits under consideration are either contributed by the industry or self-made small academical ones. The industrial circuits are defined by means of conductance and capacitance matrices, which in a few cases are not symmetric. To continue with these examples, we symmetrize \mathbf{C} and \mathbf{G} :

$$\mathbf{G} = \frac{\mathbf{G} + \mathbf{G}^T}{2}, \quad \mathbf{C} = \frac{\mathbf{C} + \mathbf{C}^T}{2}.$$

Some of the properties of the conductance and capacitance matrices we do not/cannot verify for the (larger) industrial examples. For instance, that they are positive semi-definite.

Later on, to ensure that the components of \mathbf{G} are non-singular (needed to ensure \mathbf{G} itself is non-singular, which is needed to have a DAE of index 1), we will either remove resistors from \mathbf{G} or add resistors-to-0V to \mathbf{G} . This procedure is also round-off error sensitive. In addition, to confirm that whether a G-component is non-singular or not, we have to determine all edges related to this component. However, to determine all edges, we have to convert the matrix related to the G-component into edge information and that process is round-off sensitive. Typically, assume a vertex is connected to three resistors, with resistances r_1, r_2 , and r_3 . The G-component's matrix will then have a diagonal entry $r_1 + r_2 + r_3$, and computing the edges will lead to edges with related conductivities, r_1, r_2, r_3 and r_0 , the latter being a round-off resistance. These round-off resistances can only be removed with proper care. One way to avoid the issue with the round-offs' presence (yes or no) is to estimate the condition number of the G-component with Matlab `condet()`, but this only provides an indication of potential singularity.

In addition, in order to determine *intra* (capacitors inside G-net), *inter* (coupling capacitors/capacitors between two G-nets) and *extra* (capacitors connected to reference node) net capacities, round-off errors also can increase the amount of *extra* net capacities. Also here, round-off error treatment was put into place. Issue is that a round-off induced capacitance loop typically is of order 10^{-15} times the capacitance value. There is no problem, as long as the capacitance range (value from minimum capacitance to maximum capacitance) is small: If the smallest capacitor is in the order of the largest one times 10^{-15} then we no longer can distinguish round-off loops from real capacitors to 0V.

MoveC, see Chapter 4, is also round-off error sensitive. In matrix form, this happens when we move/remove capacitors connected to the same node. When some of these capacitors are removed (by subtracting i.e., $\mathbf{C}_{move} = \mathbf{C} - \Delta\mathbf{C}$), there remains a small ϵ which we cannot say whether it represents a capacitor to ground or a round-off error. To avoid unwanted round-off error, we construct an element related incidence matrix \mathbf{A}_C and a diagonal matrix (whose elements are capacitances) as constructing \mathbf{C} in (2.2.2). In this way, \mathbf{C}_{move} is also created by $\hat{\mathbf{A}}_C$ without subtracting matrix.

Round-off errors occur in all phases of the simulation. Also the transient simulation leads:

1. In general scaling the system $\mathbf{SCS}\mathbf{x}' + \mathbf{SGS}\mathbf{x} = \mathbf{SBs}$ with a diagonal non-singular matrix \mathbf{S} leads to non-symmetric matrices \mathbf{SCS} and or \mathbf{SGS} because IEEE arithmetic is only commutative but not associative. The same holds for the diode operator $\mathbf{dx} \mapsto \mathbf{SD}(\mathbf{dx})\mathbf{S}$. The round-off errors are small, but the Matlab backslash $\mathbf{A} \setminus \mathbf{b}$ solver is sensitive to these non-symmetric cases, leading to three times slower system-solution times. We solve this by explicitly symmetrizing \mathbf{SCS} and \mathbf{SGS} by replacing them by $\frac{1}{2}(\mathbf{SCS} + (\mathbf{SCS})^T)$ and $\frac{1}{2}(\mathbf{SGS} + (\mathbf{SGS})^T)$ respectively, where $\frac{1}{2}(\mathbf{SCS} + (\mathbf{SCS})^T)$ is guaranteed to be symmetric because IEEE arithmetic is commutative. However, this costs some computational time.
2. Advancing the time-step as $t := t + dt$ is round-off error sensitive but this matters only in the case of the C-select implementation, which also uses $t = t - dt$ when the current C-select solution is rejected and we set C-select := C. Also here, unfortunately, due to round-off, $t := t + dt$ followed by $t = t - dt$ does not need to yield to original value of t which can be and turned out to be an issue if the source functions are of heavy-side step nature (when the Matlab `square` function was used to emulate binary signals). This can be solved by using indices for t .

2.8 Summary

In this chapter, we emphasize that our circuits are incomplete which need to be fixed by removing zero crosses and G-regularization. This is needed to make the system of equations solvable. In addition, when there are voltage sources driven to the network, additional rows and columns are needed to describe unknown branch currents. The resulting system of equations would be very large for practical examples and takes a very long time to simulation or infeasible. To prevent adding rows and columns, we switch the system driven by voltage sources to system driven by current sources

(the current sources are branch currents derived from voltage sources). Besides, we mention that our system of interest is DAEs of index-1. Moreover, we provide the way to calculate the nullspace of \mathbf{C} and to determine the singularity of \mathbf{C} , \mathbf{G} and $\mathbf{G} + s\mathbf{C}$. Lastly, we describe the round-off errors and input errors that occur before and during transient simulation. We also provide the techniques to minimize the errors.

Chapter 3

Mathematical analysis

3.1 Introduction

This chapter examines the errors caused by moving capacitors for the to be presented SplitC and MoveC in Chapter 4, and DeleteC as an intermediate step of MoveC. More specifically, we examine for each method the related currents and see how much they could differ. We compute the net current differences between the original problem and MoveC problem, and between the original problem and SplitC problem:

$$I_p^{\text{orig}} - I_p^{\text{move}} = -\mathbf{1}_p^T \mathbf{C}_0(\mathbf{v}' - \hat{\mathbf{v}}') - \mathbf{1}_p^T \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}}),$$

$$I_p^{\text{orig}} - I_p^{\text{split}} = -\mathbf{1}_p^T \mathbf{C}_0(\mathbf{v}' - \hat{\mathbf{v}}') - \mathbf{1}_p^T \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}}) - \mathbf{1}_p^T \mathbf{C}_s \hat{\mathbf{v}}'.$$

The details about the formulations are described in the next section. We show that in the worst case $|I_p^{\text{orig}} - I_p^{\text{split}}|/|I_p^{\text{orig}}| = 1$, implying a 100% relative error. In particular, we show that MoveC problem maintains the capacitance between nets while SplitC does not. This leads to very large error for SplitC in the worst case scenario which is described latter in Chapter 4. In order to show how much net current I_p^{orig} differs from I_p^{move} , and I_p^{orig} versus I_p^{split} we first describe the net current difference in Section 3.2. The section defines net current for each problem and the net current differences between the three problems. For illustration purpose, a simple transmission lines circuit is first given and afterward yields the formulas for general RC circuits. Indeed, the net current is the currents via coupling capacitors between any two nets, see Definition 3.2.1 and the net current differences between the original problem and other problems are equal to the sum of all the currents via capacitors and resistors to ground. Thereafter we consider the differences $\|I_p^{\text{orig}} - I_p^{\text{move}}\|$ and $\|I_p^{\text{orig}} - I_p^{\text{split}}\|$ (on a per net-basis) in Section 3.3, using both the Euclidean norm and the infinity-norm. Also, an upper bound of $\frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt}$ for general RC circuits is provided. Since the estimates require also estimate of $\|\mathbf{v} - \hat{\mathbf{v}}\|$, the estimate of $\|\mathbf{v} - \hat{\mathbf{v}}\|$ and the analytical solutions of the three problems are calculated in Section 3.4. This section gives analytical solutions and an upper bound for $\|\mathbf{v} - \hat{\mathbf{v}}\|$ for general circuit with non-singular capacitance matrix. Addition estimates of $\|\mathbf{C}\|_\infty$, $\|\Delta\mathbf{C}\|_\infty$, $\|\mathbf{1}_p^T \mathbf{C}\|_\infty$ and $\|\mathbf{1}_p^T \Delta\mathbf{C}\|_\infty$ are presented in Section 3.5. First, $\|\mathbf{C}\|_\infty$ is estimated in Section 3.5.1. We also examine modifications of $\Delta\mathbf{C}$, for instance $\Delta\mathbf{C}_{\text{move}} = \mathbf{C}^{\text{orig}} - \mathbf{C}^{\text{move}}$ for MoveC and

$\Delta \mathbf{C}_{del}$ for DeleteC in Section 3.5.2 and $\Delta \mathbf{C}_{split}$ for SplitC in Section 3.5.3. Section 3.6 shows the exact solution for a transmission line class of problems, i.e., with m lines and n internal nodes for each lines.

3.2 The net current difference $I_p^{orig} - I_p^{move}$

For each $(i, j) \in E$ (set of edges) define $\mathbf{d}_{ij} = \mathbf{e}_i - \mathbf{e}_j \in \mathbb{R}^N$, where $\mathbf{e}_k \in \mathbb{R}^N$ is the k -th unit vector. Note that E can be split into $E = E_{cap} \cup E_{res} \cup \dots$, where e.g. E_{cap} and E_{res} are the set of edges related to capacitances and resistances, respectively. Assume that the components of \mathbf{G} (after exclusion of the reference node gnd) partition the set of nodes V into $V = V_1 \cup V_2 \cup \dots \cup V_P$ where V_p contains all nodes belong to component p , $p = 1, \dots, P$, with P is the number of components, and define $E_p = \{(i, j) : i, j \in V_p\}$, $p = 1, 2, \dots, P$ set of edges related to capacitances inside component p , $E_{p,q} = \{(i, j) : i \in V_p, j \in V_q, p \neq q\}$, $p, q = 1, 2, \dots, P$ is the set of edges related to coupling capacitances. Observe that by construction, $E_{p,q} = E_{q,p}$, for all p, q and that $E_p := E_{p,p}$, for all $p = 1, 2, \dots, P$. Because the reference node gnd has been removed, the capacitors to node gnd are related edges in $E_{p,0} = \{(i, gnd) : i \in V_p\}$, $p = 1, 2, \dots, P$ (and $E_{0,p} := E_{p,0}$ for all p). Because the components are those of \mathbf{G} , there are no resistors related to edges in $E_{p,q}$, i.e., $E_{p,q}$ only contains coupling capacitors edges. Define the edges related to all coupling capacitors by

$$E_{inter} = \bigcup_{\substack{p,q=1, \\ p \neq q}}^P E_{p,q}.$$

For our example in Fig. 3.1 we find $P = 2$, $E_1 = E_2 = \emptyset$, $E_{1,0} = \{(2, gnd), (3, gnd)\}$, $E_{2,0} = \{(5, gnd), (6, gnd)\}$ and $E_{inter} = E_{1,2} = \{(2, 5), (3, 6)\}$ with $V_1 = \{1, 2, 3\}$ and $V_2 = \{4, 5, 6\}$. To describe the net current, we first use the example in Fig. 3.1.

3.2.1 The net current difference of a transmission line circuit

We assume that the KCL corresponding to the reference node has been eliminated from the system of equations which cause capacitors to ground have associated edges in $E_{1,0}$ and $E_{2,0}$. This also means that E_{cap} and E_p do no longer contain edges related to capacitors to ground. Based on our system of equations (2.3.10) (assume the input sources are currents) we know (Kirchhoff's current law) that

$$\mathbf{C}\mathbf{v}' + \mathbf{G}\mathbf{v} = \mathbf{I}_{source} \quad (3.2.1)$$

where \mathbf{I}_{source} is the vector of input sources, for our example $\mathbf{I}_{source} = [s_1; 0; 0; s_2; 0; 0]$. For the nodal numbering 1, 2, ..., 6 in Fig. 3.1. We find

$$\mathbf{G} = \left[\begin{array}{ccc|ccc} g_{12} & -g_{12} & 0 & 0 & 0 & 0 \\ -g_{12} & g_{12} + g_{23} & -g_{23} & 0 & 0 & 0 \\ 0 & -g_{23} & g_{23} + g_3 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & g_{45} & -g_{45} & 0 \\ 0 & 0 & 0 & -g_{45} & g_{45} + g_{56} & -g_{56} \\ 0 & 0 & 0 & 0 & -g_{56} & g_{56} + g_6 \end{array} \right] = \left[\begin{array}{c|c} \mathbf{G}_{11} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{G}_{22} \end{array} \right], \quad (3.2.2)$$

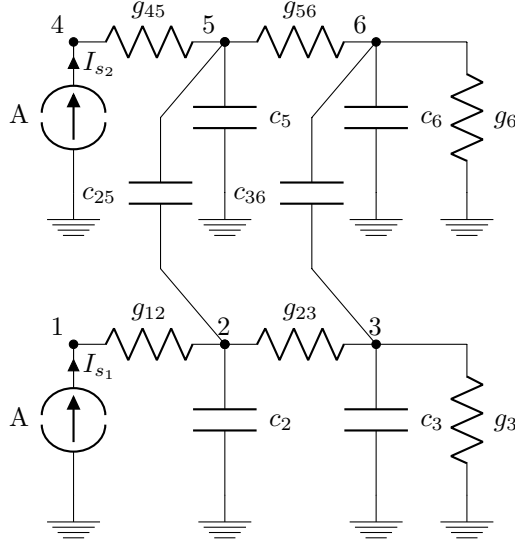


Figure 3.1: Two lines model.

$$\mathbf{C} = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_2 + c_{25} & 0 & 0 & -c_{25} & 0 \\ 0 & 0 & c_3 + c_{36} & 0 & 0 & -c_{36} \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -c_{25} & 0 & 0 & c_5 + c_{25} & 0 \\ 0 & 0 & -c_{36} & 0 & 0 & c_6 + c_{36} \end{array} \right] = \left[\begin{array}{c|c} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \hline \mathbf{C}_{21} & \mathbf{C}_{22} \end{array} \right], \quad (3.2.3)$$

$$\mathbf{I}_{\text{source}} = \mathbf{B}\mathbf{s}(t) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix}, \quad (3.2.4)$$

where instead of $\mathbf{s}(t)$, we sometimes write $\mathbf{u}(t)$. Our notation works as follows: c_{ij} is the capacitance of the capacitor between nodes i and j . Also, c_k is the capacitance between node k and gnd . Ditto for g_{ij} and g_k . Sometime we write $c(i, j)$ for c_{ij} . Nodal capacitors are denoted $\mathbf{C}(i, j)$ respectively $\mathbf{C}(k)$.

Now $\mathbf{C}\mathbf{v}'$ stands for the vector of currents, flowing out of nodes, each component being a sum of currents flowing through capacitors connect to the corresponding node. Thus

$$\mathbf{C} \frac{d\mathbf{v}}{dt} = \mathbf{I}_{\text{source}} - \mathbf{G}\mathbf{v}, \quad t > 0. \quad (3.2.5)$$

We first examine the term $\mathbf{C} \frac{d\mathbf{v}}{dt}$. Related to Fig. 3.1 we have:

$$\mathbf{C} = c_{25} \mathbf{d}_{25} \mathbf{d}_{25}^T + c_{36} \mathbf{d}_{36} \mathbf{d}_{36}^T + \mathbf{C}_0 \quad (3.2.6)$$

$$= c_{25} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 & -1 & 0 \end{bmatrix} + c_{36} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} + \mathbf{C}_0 \quad (3.2.7)$$

$$= c_{25} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + c_{36} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_6 \end{bmatrix} \quad (3.2.8)$$

$$= \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_2 + c_{25} & 0 & 0 & -c_{25} & 0 \\ 0 & 0 & c_3 + c_{36} & 0 & 0 & -c_{36} \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -c_{25} & 0 & 0 & c_5 + c_{25} & 0 \\ 0 & 0 & -c_{36} & 0 & 0 & c_6 + c_{36} \end{array} \right]. \quad (3.2.9)$$

where \mathbf{C}_0 is a diagonal matrix whose diagonal elements are capacitances of capacitors connected to reference node *gnd*. For this example $E_{\text{inter}} = \{(2, 5), (3, 6)\} = E_{1,2}$ (because capacitances to ground have been eliminated and net internal capacitances do not exist). Now look more closely at \mathbf{C} and the related two nets. The first net contains degrees of freedom $V_1 = \{1, 2, 3\}$ and the second net contains degrees of freedom $V_2 = \{4, 5, 6\}$. Based on (3.2.6) we find that for arbitrary $\mathbf{z} \in \mathbb{R}^N$ ($N = 6$)

$$\begin{aligned} \mathbf{C}\mathbf{z} &= c_{25} \mathbf{d}_{25} \mathbf{d}_{25}^T \mathbf{z} + c_{36} \mathbf{d}_{36} \mathbf{d}_{36}^T \mathbf{z} + \mathbf{C}_0 \mathbf{z} \\ &= c_{25} \mathbf{d}_{25} (z_2 - z_5) + c_{36} \mathbf{d}_{36} (z_3 - z_6) + \mathbf{C}_0 \mathbf{z} \\ &= c_{25} \mathbf{e}_2 (z_2 - z_5) + c_{36} \mathbf{e}_3 (z_3 - z_6) - c_{25} \mathbf{e}_5 (z_2 - z_5) - c_{36} \mathbf{e}_6 (z_3 - z_6) + \mathbf{C}_0 \mathbf{z} \\ &= \begin{bmatrix} 0 \\ c_{25}(z_2 - z_5) \\ c_{36}(z_3 - z_6) \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ c_{25}(z_2 - z_5) \\ c_{36}(z_3 - z_6) \end{bmatrix} + \mathbf{C}_0 \mathbf{z} \\ &= \sum_{(i,j) \in E_{1,2}} c(i,j) \mathbf{e}_i (z_i - z_j) - \sum_{(i,j) \in E_{1,2}} c(i,j) \mathbf{e}_j (z_i - z_j) + \mathbf{C}_0 \mathbf{z} \end{aligned}$$

Substituting $\mathbf{z} := \frac{d\mathbf{v}}{dt} = \left[\frac{dv_1}{dt}, \dots, \frac{dv_N}{dt} \right]$, we find

$$\mathbf{C} \frac{d\mathbf{v}}{dt} = \begin{bmatrix} 0 \\ c_{25} \frac{d(v_2 - v_5)}{dt} \\ c_{36} \frac{d(v_3 - v_6)}{dt} \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ c_{25} \frac{d(v_2 - v_5)}{dt} \\ c_{36} \frac{d(v_3 - v_6)}{dt} \end{bmatrix} + \mathbf{C}_0 \frac{d\mathbf{v}}{dt} = \begin{bmatrix} 0 \\ \iota_{25} \\ \iota_{36} \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \iota_{25} \\ \iota_{36} \end{bmatrix} + \mathbf{C}_0 \frac{d\mathbf{v}}{dt}, \quad (3.2.10)$$

using the branch constitutive equation for capacitor $i_{ij} = c_{ij} \frac{d(v_i - v_j)}{dt}$. Using (3.2.10) and the branch equation for resistor $i_{ij} = g_{ij}(v_i - v_j)$, equation (3.2.5) becomes

$$\begin{bmatrix} 0 \\ i_{25} \\ i_{36} \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ i_{25} \\ i_{36} \end{bmatrix} + \mathbf{C}_0 \frac{d\mathbf{v}}{dt} = \begin{bmatrix} s_1 \\ 0 \\ 0 \\ s_2 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} i_{12} \\ -i_{12} + i_{23} \\ -i_{23} \\ i_{45} \\ -i_{45} + i_{56} \\ -i_{56} \end{bmatrix} - \underbrace{\begin{bmatrix} 0 & & & & & \\ & 0 & & & & \\ & & g_3 & & & \\ & & & 0 & & \\ & & & & 0 & \\ & & & & & g_6 \end{bmatrix}}_{\mathbf{G}_0} \mathbf{v}, \quad t > 0, \Leftrightarrow$$

$$\begin{bmatrix} 0 \\ i_{25} \\ i_{36} \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ i_{25} \\ i_{36} \end{bmatrix} + \mathbf{C}_0 \frac{d\mathbf{v}}{dt} = \begin{bmatrix} s_1 \\ 0 \\ 0 \\ s_2 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} i_{12} \\ -i_{12} + i_{23} \\ -i_{23} + i_3 \\ i_{45} \\ -i_{45} + i_{56} \\ -i_{56} + i_6 \end{bmatrix}, \quad t > 0, \quad (3.2.11)$$

where i_3 and i_6 are the currents through g_3 respectively g_6 to ground and \mathbf{G}_0 is a diagonal matrix whose diagonal elements are conductances of conductors connected to reference node *gnd*. Define $\mathbf{1}_p \in \mathbb{R}^N, p = 1, \dots, P$ by

$$\mathbf{1}_p = \sum_{i \in V_p} \mathbf{e}_i$$

and note that $\sum_{p=1}^P \mathbf{1}_p = \mathbf{1} := [1, \dots, 1]^T$ because $V_1 \cup V_2 \cup \dots \cup V_P = V$. In our example for net 1, $V_1 = \{1, 2, 3\}$ i.e., a set of degree of freedom belongs to net 1, so

$$\mathbf{1}_1 = \mathbf{e}_1 + \mathbf{e}_2 + \mathbf{e}_3 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{1}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Then for (3.2.11)

$$\mathbf{1}_1^T \mathbf{C} \frac{d\mathbf{v}}{dt} = \mathbf{1}_1^T (\mathbf{I}_{\text{source}} - \mathbf{G}\mathbf{v}) \Leftrightarrow \quad (3.2.12)$$

$$\mathbf{1}_1^T \left(\begin{bmatrix} 0 \\ i_{25} \\ i_{36} \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ i_{25} \\ i_{36} \end{bmatrix} + \mathbf{C}_0 \frac{d\mathbf{v}}{dt} \right) = \mathbf{1}_1^T \left(\begin{bmatrix} s_1 \\ 0 \\ 0 \\ s_2 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} i_{12} \\ -i_{12} + i_{23} \\ -i_{23} + i_3 \\ i_{45} \\ -i_{45} + i_{56} \\ -i_{56} + i_6 \end{bmatrix} \right) \Leftrightarrow$$

$$i_{25} + i_{36} + c_2 \frac{dv_2}{dt} + c_3 \frac{dv_3}{dt} = s_1 - \mathbf{1}_1^T \mathbf{G}_0 \mathbf{v} = s_1 - i_3. \quad (3.2.13)$$

This is KCL for a cut-set (set of edges cut by a closed surface) that surrounds net 1. Thus

$$\begin{aligned} \mathbf{1}^T \mathbf{C} \frac{d\mathbf{v}}{dt} &= i_{25} + i_{36} + c_2 \frac{dv_2}{dt} + c_3 \frac{dv_3}{dt} - i_{25} - i_{36} + c_5 \frac{dv_5}{dt} + c_6 \frac{dv_6}{dt} \\ &= s_1 + s_2 - i_3 - i_6 = s_1 + s_2 - \mathbf{1}^T \mathbf{G}_0 \mathbf{v}. \end{aligned}$$

This is KCL for a cut-set that surrounds both net1 and net 2

For our example we move the capacitor related to edge (3, 6) to edge (2, 5) which results in one capacitor at (2, 5) with capacitance $\hat{c}_{25} = c_{25} + c_{36}$. We call the related capacitance matrix the MoveC matrix $\hat{\mathbf{C}}$

$$\hat{\mathbf{C}} = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_2 + \hat{c}_{25} & 0 & 0 & -\hat{c}_{25} & 0 \\ 0 & 0 & c_3 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\hat{c}_{25} & 0 & 0 & c_5 + \hat{c}_{25} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_6 \end{array} \right]. \quad (3.2.14)$$

Then the MoveC related Kirchhoff's laws are

$$\hat{\mathbf{C}} \hat{\mathbf{v}}' + \mathbf{G} \hat{\mathbf{v}} = \mathbf{I}_{\text{source}}$$

and similar to (3.2.12) we find

$$\begin{aligned} \mathbf{1}_1^T \hat{\mathbf{C}} \frac{d\hat{\mathbf{v}}}{dt} &= \mathbf{1}_1^T (\mathbf{I}_{\text{source}} - \mathbf{G} \hat{\mathbf{v}}) \Leftrightarrow \\ \mathbf{1}_1^T \left(\begin{bmatrix} 0 \\ \hat{i}_{25} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ \hat{i}_{25} \\ 0 \end{bmatrix} + \mathbf{C}_0 \frac{d\hat{\mathbf{v}}}{dt} \right) &= \mathbf{1}_1^T \left(\begin{bmatrix} s_1 \\ 0 \\ 0 \\ s_2 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} \hat{i}_{12} \\ -\hat{i}_{12} + \hat{i}_{23} \\ -\hat{i}_{23} + \hat{i}_3 \\ \hat{i}_{45} \\ -\hat{i}_{45} + \hat{i}_{56} \\ -\hat{i}_{56} + \hat{i}_6 \end{bmatrix} \right) \Leftrightarrow \\ \hat{i}_{25} + c_2 \frac{d\hat{v}_2}{dt} + c_3 \frac{d\hat{v}_3}{dt} &= s_1 - \hat{i}_3. \end{aligned} \quad (3.2.15)$$

Let I_p is the sum of currents flowing out of the net p through the coupling capacitors which have exactly one vertex in V_p . Equations (3.2.13) and (3.2.15) show that

$$\left\{ \underbrace{i_{25} + i_{36}}_{I_1^{\text{orig}}} + c_2 \frac{dv_2}{dt} + c_3 \frac{dv_3}{dt} = s_1 - i_3, \right. \quad (3.2.16a)$$

$$\left\{ \underbrace{\hat{i}_{25}}_{I_1^{\text{move}}} + c_2 \frac{d\hat{v}_2}{dt} + c_3 \frac{d\hat{v}_3}{dt} = s_1 - \hat{i}_3. \right. \quad (3.2.16b)$$

$$\Leftrightarrow \left\{ \begin{aligned} I_1^{\text{orig}} + \mathbf{1}_1^T \mathbf{C}_0 \frac{d\mathbf{v}}{dt} &= s_1 - \mathbf{1}_1^T \mathbf{G}_0 \mathbf{v}, \\ I_1^{\text{move}} + \mathbf{1}_1^T \mathbf{C}_0 \frac{d\hat{\mathbf{v}}}{dt} &= s_1 - \mathbf{1}_1^T \mathbf{G}_0 \hat{\mathbf{v}}. \end{aligned} \right. \quad (3.2.17a)$$

$$(3.2.17b)$$

$$(3.2.17a) - (3.2.17a) \Rightarrow I_1^{\text{orig}} - I_1^{\text{move}} + \mathbf{1}_1^T \mathbf{C}_0 \frac{d(\mathbf{v} - \widehat{\mathbf{v}})}{dt} = -\mathbf{1}_1^T \mathbf{G}_0 (\mathbf{v} - \widehat{\mathbf{v}}). \quad (3.2.18)$$

Note that (3.2.16a) and (3.2.16b) can be rewritten

$$\begin{cases} \mathbf{1}_1^T \widehat{\mathbf{C}} \frac{d\widehat{\mathbf{v}}}{dt} = s_1 - \mathbf{1}_1^T \mathbf{G}_0 \widehat{\mathbf{v}}, \\ \mathbf{1}_1^T \mathbf{C} \frac{d\mathbf{v}}{dt} = s_1 - \mathbf{1}_1^T \mathbf{G}_0 \mathbf{v}. \end{cases}$$

i.e., to estimate $I_1^{\text{orig}} - I_1^{\text{move}}$ we need to estimate $\frac{d(\mathbf{v} - \widehat{\mathbf{v}})}{dt}$ as well as $\mathbf{v} - \widehat{\mathbf{v}}$.

Now for error estimate (3.2.18), observe that our analysis for Fig. 3.1 is also valid when the two nets have internal capacitors because multiplication with $\mathbf{1}_p$ leads to zero capacitance sums (as it did for the net- p internal resistors).

3.2.2 The net current difference of general RC circuits

Definition 3.2.1. Given the problem related to an RC network

$$\mathbf{C}\mathbf{v}' + \mathbf{G}\mathbf{v} = \mathbf{I}_{\text{source}}$$

The net current I_p of net p at any given time t is defined to be

$$I_p(t) = \sum_{\substack{q=1, \\ q \neq p}}^P I_{p,q}(t),$$

where

$$I_{p,q}(t) = \sum_{(i,j) \in E_{p,q}} c_{ij}(v'_i(t) - v'_j(t)).$$

i.e., $I_p(t)$ is the sum of all currents via coupling capacitors between net p and nets q with capacitors connected to net p . Consider currents related to net p for the unmodified problem

$$\mathbf{1}_p^T \mathbf{C}\mathbf{v}' = \mathbf{1}_p^T (\mathbf{I}_{\text{source}} - \mathbf{G}\mathbf{v}) \Leftrightarrow$$

$$\begin{aligned} & \sum_{\substack{q=1, \\ q \neq p}}^P \sum_{(i,j) \in E_{p,q}} c_{ij}(v'_i - v'_j) + \sum_{(i,j) \in E_p} c_{ij}(v'_i - v'_j) + \sum_{(i,gnd) \in E_{p,0}} c_i v'_i \\ &= S_p - \sum_{(i,j) \in E_{res(p)}} g_{ij}(v_i - v_j) - \sum_{(i,gnd) \in E_{res(p,0)}} g_i v_i \Leftrightarrow \end{aligned} \quad (3.2.20)$$

$$I_p^{\text{orig}} + \sum_{(i,gnd) \in E_{p,0}} c_i v'_i = S_p - \sum_{(i,gnd) \in E_{res(p,0)}} g_i v_i, \quad (3.2.21)$$

where $S_p = \sum_{i \in V_p} s_i$, the total current sources applied to net p . In (3.2.20), the internal capacitance currents and resistor currents eliminate each others which leads

to (3.2.21). Denote $\hat{\mathbf{v}}$ solution of MoveC problem or SplitC problem. The net current for MoveC of the modified problem is

$$I_p^{\text{move}} = \sum_{q=1, q \neq p}^P \sum_{(i,j) \in E_{p,q}} \hat{c}_{ij}(\hat{v}_i' - \hat{v}_j') = S_p - \sum_{(i,gnd) \in E_{res(p,0)}} g_i \hat{v}_i - \sum_{(i,gnd) \in E_{p,0}} c_i \hat{v}_i'.$$

The net current difference for MoveC is

$$I_p^{\text{orig}} - I_p^{\text{move}} = \sum_{q=1, q \neq p}^P \sum_{(i,j) \in E_{p,q}} c_{ij}(v_i' - v_j') - \sum_{q=1, q \neq p}^P \sum_{(i,j) \in E_{p,q}} \hat{c}_{ij}(\hat{v}_i' - \hat{v}_j') \quad (3.2.22)$$

$$\stackrel{(3.2.20)}{=} - \sum_{(i,gnd) \in E_{res(p,0)}} g_i(v_i - \hat{v}_i) - \sum_{(i,gnd) \in E_{p,0}} c_i(v_i' - \hat{v}_i') \quad (3.2.23)$$

$$= -\mathbf{1}_p^T \mathbf{C}_0(\mathbf{v}' - \hat{\mathbf{v}}') - \mathbf{1}_p^T \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}}). \quad (3.2.24)$$

The current difference for SplitC is different as it adds capacitors to ground:

$$I_p^{\text{orig}} - I_p^{\text{split}} = - \sum_{(i,gnd) \in E_{res(p,0)}} g_i(v_i - \hat{v}_i) - \sum_{(i,gnd) \in E_{p,0}} c_i(v_i' - \hat{v}_i') - \sum_{(i,gnd) \in E_{p,0}} c_i \hat{v}_i' \\ = -\mathbf{1}_p^T \mathbf{C}_0(\mathbf{v}' - \hat{\mathbf{v}}') - \mathbf{1}_p^T \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}}) - \mathbf{1}_p^T \mathbf{C}_s \hat{\mathbf{v}}', \quad (3.2.25)$$

where \mathbf{C}_s is diagonal matrix created by splitting coupling capacitors to ground. However, in the worst case scenario, when we split all coupling capacitors, the relative net current difference becomes

$$\frac{|I_p^{\text{orig}} - I_p^{\text{split}}|}{I_p^{\text{orig}}} = \frac{\left| \sum_{q=1, q \neq p}^P \sum_{(i,j) \in E_{p,q}} c_{ij}(v_i' - v_j') - \sum_{q=1, q \neq p}^P \sum_{(i,j) \in E_{p,q}} \hat{c}_{ij}(\hat{v}_i' - \hat{v}_j') \right|}{\left| \sum_{q=1, q \neq p}^P \sum_{(i,j) \in E_{p,q}} c_{ij}(v_i' - v_j') \right|} = 1,$$

i.e., a 100% relative error ($I_p^{\text{split}} = 0$ because there are no coupling capacitors left, they are all split to ground). With this we want to emphasize that MoveC always maintains capacitance between nets even in the worst case scenario, which leads to small solution error.

3.3 Estimate for net current difference $|I_q^{\text{orig}} - I_q^{\text{move}}|$

For the error estimate we focus on the net current difference $I_q^{\text{orig}} - I_q^{\text{move}}$ which does not involve the term $\mathbf{1}_p^T \mathbf{C}_s \hat{\mathbf{v}}'$ of SplitC. This extra term indicate that SplitC generates large errors if there are larger currents through capacitors split to be connected to the ground. Based on (3.2.24), $I_1^{\text{orig}} - I_1^{\text{move}} = -\mathbf{1}_1^T \left(\mathbf{C}_0 \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} + \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}}) \right)$. The relative inter-component- q error is given by

$$\frac{|I_q^{\text{orig}} - I_q^{\text{move}}|}{|I_q^{\text{orig}}|} = \frac{|\mathbf{1}_q^T \mathbf{C}_0 \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} + \mathbf{1}_q^T \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}})|}{|S_q - \mathbf{1}_q^T \mathbf{C}_0 \frac{d\mathbf{v}}{dt} - \mathbf{1}_q^T \mathbf{G}_0 \mathbf{v}|}. \quad (3.3.1)$$

To obtain $\frac{|I_q^{\text{orig}} - I_q^{\text{move}}|}{|I_q^{\text{orig}}|} \leq \frac{a}{b}$, we need estimates of the form $|I_q^{\text{orig}} - I_q^{\text{move}}| \leq a$ (upper bound) and $|I_q^{\text{orig}}| \geq b$ (lower bound). For a lower bound for $|I_q^{\text{orig}}|$ use

$$I_q^{\text{orig}} = S_q - \underbrace{\left(\mathbf{1}_q^T \mathbf{C}_0 \frac{d\mathbf{v}}{dt} + \mathbf{1}_q^T \mathbf{G}_0 \mathbf{v} \right)}_{=:\Delta},$$

assume $|\Delta| < |S_q|$ and apply the inverse triangle inequality $|a - b| \geq ||a| - |b||$ for $a = S_q$ and $b = \Delta$, this results in

$$|I_q^{\text{orig}}| = |S_q - \Delta| \geq ||S_q| - |\Delta||$$

whence

$$\frac{1}{|I_q^{\text{orig}}|} \leq \frac{1}{|S_q| - |\Delta|} = \frac{1}{|S_q| - |\mathbf{1}_q^T \mathbf{C}_0 \frac{d\mathbf{v}}{dt} + \mathbf{1}_q^T \mathbf{G}_0 \mathbf{v}|}. \quad (3.3.2)$$

For an upper bound for $|I_q^{\text{orig}} - I_q^{\text{move}}|$ we will exploit the Euclidean or infinity norms.

3.3.1 An estimate for $|I_q^{\text{orig}} - I_q^{\text{move}}|$ using the 2-norm

There are two ways to exploit the Euclidean $\|\cdot\|_2$ norm. First,

$$|I_q^{\text{orig}} - I_q^{\text{move}}| = \left| \mathbf{1}_q^T \underbrace{\left(\mathbf{C}_0 \left(\frac{d\mathbf{v}}{dt} - \frac{d\hat{\mathbf{v}}}{dt} \right) + \mathbf{G}_0 (\mathbf{v} - \hat{\mathbf{v}}) \right)}_{=:\mathbf{w}} \right|. \quad (3.3.3)$$

To estimate the product $\mathbf{a}^T \mathbf{b}$ for $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ we can use Cauchy–Schwarz or the inner product-relation

$$\mathbf{a}^T \mathbf{b} = \|\mathbf{a}\|_2 \|\mathbf{b}\|_2 \cos(\angle(\mathbf{a}, \mathbf{b})) \leq \|\mathbf{a}\|_2 \|\mathbf{b}\|_2, \quad (3.3.4)$$

where $\angle(\mathbf{a}, \mathbf{b})$ the angle between two non-zero vectors \mathbf{a} and \mathbf{b} . Based on (3.3.4), (3.3.3) can be estimated above

$$|I_q^{\text{orig}} - I_q^{\text{move}}| = \|\mathbf{1}_q\|_2 \|\mathbf{w}\|_2 \cos(\langle \mathbf{1}_q, \mathbf{w} \rangle) \leq \|\mathbf{1}_q\|_2 \|\mathbf{w}\|_2$$

because $|\cos \phi| \leq 1$ for all angles ϕ , where

$$\|\mathbf{1}_q\|_2 = \left\| \sum_{i \in I_q} \mathbf{e}_i \right\|_2 = \sqrt{|V_q|}.$$

With $|V_q|$ the number of degrees belonging to component q .

Consider the term $\mathbf{C}_0 \frac{d\mathbf{v}}{dt}$ where \mathbf{C}_0 contains precisely one capacitor with capacitance c to ground connected to nodal voltage $v_1(t) = \sin(2\pi ft)$, see Fig. 3.2.

In that case $\mathbf{C}_0 \frac{d\mathbf{v}}{dt} = c \frac{dv_1}{dt} = 2\pi fc \cdot \cos(2\pi ft)$ will be large when fc is large (a different way to see this is to time-scale the system as in (2.3.9) which then becomes $(f\mathbf{C})\mathbf{x}'(s) + \mathbf{G}\mathbf{x}(s) = \mathbf{I}_{\text{source}}$ where now $v_1(s) = \sin(2\pi s)$). Hence, we assume that our circuits have capacitances to ground (much) smaller than their operating frequency.

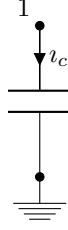


Figure 3.2: Circuit having a capacitance to ground.

Left to do is to estimate

$$\|\mathbf{w}\|_2 = \left\| \mathbf{C}_0 \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} + \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}}) \right\|_2 \leq \|\mathbf{C}_0\|_2 \left\| \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} \right\|_2 + \|\mathbf{G}_0\|_2 \|\mathbf{v} - \hat{\mathbf{v}}\|_2.$$

Thus

$$|I_q^{\text{orig}} - I_q^{\text{move}}| \leq \sqrt{|V_q|} \cdot \left(\|\mathbf{C}_0\|_2 \cdot \left\| \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} \right\|_2 + \|\mathbf{G}_0\|_2 \cdot \|\mathbf{v} - \hat{\mathbf{v}}\|_2 \right). \quad (3.3.5)$$

Applied for transmission lines example we have

$$|I_q^{\text{orig}} - I_q^{\text{move}}| \leq \sqrt{|V_q|} \cdot \left(\max\{c_2, c_3, c_5, c_6\} \cdot \left\| \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} \right\|_2 + \max\{g_3, g_6\} \cdot \|\mathbf{v} - \hat{\mathbf{v}}\|_2 \right).$$

Another estimate for $|I_q^{\text{orig}} - I_q^{\text{move}}|$, suppose $q = 1$ for transmission lines example, is

$$\begin{aligned} |I_1^{\text{orig}} - I_1^{\text{move}}| &= \left\| \mathbf{1}_1^T \mathbf{C}_0 \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} + \mathbf{1}_1^T \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}}) \right\|_2 \\ &= \left\| \begin{bmatrix} 0 & c_2 & c_3 & 0 & 0 & 0 \end{bmatrix} \cdot \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} + \begin{bmatrix} 0 & 0 & g_3 & 0 & 0 & 0 \end{bmatrix} \cdot (\mathbf{v} - \hat{\mathbf{v}}) \right\|_2 \\ &\leq \left\| \begin{bmatrix} 0 & c_2 & c_3 & 0 & 0 & 0 \end{bmatrix} \right\|_2 \cdot \left\| \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} \right\|_2 + \\ &\quad \left\| \begin{bmatrix} 0 & 0 & g_3 & 0 & 0 & 0 \end{bmatrix} \right\|_2 \cdot \|\mathbf{v} - \hat{\mathbf{v}}\|_2 \\ &= \sqrt{c_2^2 + c_3^2} \cdot \left\| \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} \right\|_2 + g_3 \|\mathbf{v} - \hat{\mathbf{v}}\|_2. \end{aligned}$$

For general RC circuits,

$$|I_p^{\text{orig}} - I_p^{\text{move}}| = \left\| \mathbf{1}_p^T \mathbf{C}_0 \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} + \mathbf{1}_p^T \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}}) \right\|_2 \quad (3.3.6)$$

$$\leq \sqrt{\sum_{i \in E_{p,0}} c_i^2} \cdot \left\| \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} \right\|_2 + \sqrt{\sum_{i \in E_{res(p,0)}} g_i^2} \|\mathbf{v} - \hat{\mathbf{v}}\|_2. \quad (3.3.7)$$

This leads $\mathbf{v} - \hat{\mathbf{v}}$ to be estimated, see Section 3.4. As an alternative to (3.3.6) we can

estimate:

$$\begin{aligned}
 |I_1^{\text{orig}} - I_1^{\text{move}}| &= \left\| \mathbf{1}_1^T \mathbf{C}_0 \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} + \mathbf{1}_1^T \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}}) \right\|_2 \\
 &= \left\| \begin{bmatrix} 0 & c_2 & c_3 & 0 & 0 & 0 \end{bmatrix} \cdot \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} + \begin{bmatrix} 0 & 0 & g_3 & 0 & 0 & 0 \end{bmatrix} \cdot (\mathbf{v} - \hat{\mathbf{v}}) \right\|_2 \\
 &\leq \left\| c_2 \frac{d(v_2 - \hat{v}_2)}{dt} + c_3 \frac{d(v_3 - \hat{v}_3)}{dt} + g_3(v_3 - \hat{v}_3) \right\|_2 \\
 &= \left| c_2 \frac{d(v_2 - \hat{v}_2)}{dt} + c_3 \frac{d(v_3 - \hat{v}_3)}{dt} + g_3(v_3 - \hat{v}_3) \right|.
 \end{aligned}$$

For net p of a general circuit,

$$\begin{aligned}
 |I_p^{\text{orig}} - I_p^{\text{move}}| &= \left\| \mathbf{1}_p^T \mathbf{C}_0 \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} + \mathbf{1}_p^T \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}}) \right\|_2 \\
 &\leq \left| \sum_{(i, \text{gnd}) \in E_{p,0}} c_i \frac{d(v_i - \hat{v}_i)}{dt} + \sum_{(i, \text{gnd}) \in E_{\text{res}(p,0)} } g_i(v_i - \hat{v}_i) \right|.
 \end{aligned}$$

3.3.2 An estimate for $|I_q^{\text{orig}} - I_q^{\text{move}}|$ using ∞ -norm

Based on (3.2.5) and assume $q = 1$

$$\begin{aligned}
 |I_1^{\text{orig}} - I_1^{\text{move}}| &= \left| \mathbf{1}_1^T \mathbf{C}_0 \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} + \mathbf{1}_1^T \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}}) \right| \\
 &= \left| \begin{bmatrix} 0 & c_2 & c_3 & 0 & 0 & 0 \end{bmatrix} \cdot \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} + \begin{bmatrix} 0 & 0 & g_3 & 0 & 0 & 0 \end{bmatrix} \cdot (\mathbf{v} - \hat{\mathbf{v}}) \right| \\
 &\leq \left\| \begin{bmatrix} 0 & c_2 & c_3 & 0 & 0 & 0 \end{bmatrix} \right\|_{\infty} \cdot \left\| \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} \right\|_{\infty} \\
 &\quad \left\| \begin{bmatrix} 0 & 0 & g_3 & 0 & 0 & 0 \end{bmatrix} \right\|_{\infty} \cdot \left\| (\mathbf{v} - \hat{\mathbf{v}}) \right\|_{\infty} \\
 &= (c_2 + c_3) \cdot \left\| \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} \right\|_{\infty} + g_3 \cdot \left\| (\mathbf{v} - \hat{\mathbf{v}}) \right\|_{\infty}
 \end{aligned}$$

and for general RC circuits

$$\begin{aligned}
 |I_p^{\text{orig}} - I_p^{\text{move}}| &= \left| \mathbf{1}_p^T \mathbf{C}_0 \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} + \mathbf{1}_p^T \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}}) \right| \\
 &\leq \sum_{(i, \text{gnd}) \in E_{p,0}} c_i \cdot \left\| \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} \right\|_{\infty} + \sum_{(i, \text{gnd}) \in E_{\text{res}(p,0)} } g_i \cdot \left\| (\mathbf{v} - \hat{\mathbf{v}}) \right\|_{\infty}
 \end{aligned}$$

using that $\|\cdot\|_{\infty}$ also is an associated (induced/ matrix norm).

3.3.3 An estimate of $\left\| \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} \right\|_2$

Next, we estimate $\left\| \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} \right\|_2$. Assume that there exists $\alpha > 0$ such that

$$\frac{dv_i(t)}{dt} \leq \alpha \quad \text{and} \quad \frac{d\hat{v}_i(t)}{dt} \leq \alpha \tag{3.3.8}$$

for all nodal voltages v_i and \hat{v}_i .

Bound (3.3.8) is valid if the derivatives of v_i and \hat{v}_i are all bounded in time, but the derivatives can be very large. Define the one-vector $\mathbf{1} = [1, \dots, 1]^T \in \mathbb{R}^N$, and define vector (in-)equalities such as $\mathbf{a} \leq \mathbf{b} \Leftrightarrow a_i \leq b_i, \forall i = 1, \dots, N$. Using these definitions one finds

$$\mathbf{0} \leq \left| \frac{d\mathbf{v}}{dt} \right| = \begin{bmatrix} \left| \frac{dv_1}{dt} \right| \\ \vdots \\ \left| \frac{dv_N}{dt} \right| \end{bmatrix} \leq \begin{bmatrix} \alpha \\ \vdots \\ \alpha \end{bmatrix} = \alpha \cdot \mathbf{1} \quad \text{and} \quad \mathbf{0} \leq \left| \frac{d\hat{\mathbf{v}}}{dt} \right| \leq \alpha \cdot \mathbf{1}.$$

Using the Euclidean norm, one finds

$$\left\| \frac{d\mathbf{v}}{dt} \right\|_2, \quad \left\| \frac{d\hat{\mathbf{v}}}{dt} \right\|_2 \leq \alpha \cdot \|\mathbf{1}\|_2 = \alpha\sqrt{N}$$

which implies

$$\underbrace{\left\| \frac{d\mathbf{v}}{dt} - \frac{d\hat{\mathbf{v}}}{dt} \right\|_2}_{\|\mathbf{u}\|_2} \leq \left\| \frac{d\mathbf{v}}{dt} \right\|_2 + \left\| \frac{d\hat{\mathbf{v}}}{dt} \right\|_2 \leq 2\alpha\sqrt{N}. \quad (3.3.9)$$

Using the infinity-norm and $\mathbf{0} < \mathbf{a} \leq \mathbf{b} \Rightarrow \|\mathbf{a}\|_p \leq \|\mathbf{b}\|_p$ for $p = \infty$ which leads to

$$\mathbf{0} \leq \left\| \frac{d\mathbf{v}}{dt} \right\|_\infty, \quad \left\| \frac{d\hat{\mathbf{v}}}{dt} \right\|_\infty \leq \alpha \quad \text{and} \quad \left\| \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} \right\|_\infty \leq 2\alpha. \quad (3.3.10)$$

Estimates (3.3.9) and (3.3.10) can be overestimates because $\frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt}$ could be small relative to the size of $\frac{d\mathbf{v}}{dt}$, $\frac{d\hat{\mathbf{v}}}{dt}$, for instance when $\hat{\mathbf{v}}(t) = \mathbf{v}(t) + \epsilon$ (a constant small error) and $\frac{d\mathbf{v}}{dt}$ is large.

3.4 An upper bound for $\|\mathbf{v} - \hat{\mathbf{v}}\|$

Without loss of generality, denote $\mathbf{v} = \mathbf{x}^o$ and $\hat{\mathbf{v}} = \mathbf{x}^m$. The solution of

$$\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{b}(t), \quad \mathbf{x}(0) = \mathbf{x}_0$$

is given by

$$\mathbf{x} = e^{\mathbf{A}t} \left[\int_0^t e^{-\mathbf{A}s} \mathbf{b}(s) ds + \mathbf{x}_0 \right], \quad \mathbf{x}_0 = \mathbf{I} \left[\int_0^0 \mathbf{b}(s) ds + \mathbf{x}_0 \right] = \mathbf{x}_0.$$

Now assume that the nodes are numbered as follows: first, the nodes to which only capacitors are connected, and next the ones to which only resistors are connected, then

$$\begin{aligned} \mathbf{C}\mathbf{x}'(t) + \mathbf{G}\mathbf{x}(t) + \mathbf{I}(t) &= \mathbf{0} \Leftrightarrow \\ \begin{bmatrix} \mathbf{C}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{x}'(t) + \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{bmatrix} \mathbf{x}(t) + \mathbf{I}(t) &= \mathbf{0} \end{aligned} \quad (3.4.1)$$

$\mathbf{x} = [\mathbf{x}_c; \mathbf{x}_r] = [\mathbf{x}_1; \mathbf{x}_2]$. Assume \mathbf{G} is non-singular. Because \mathbf{G} is SPD, \mathbf{G}_{22} is also SPD which implies that \mathbf{G}_{22} is non-singular. Now for our error analyzes, we will consider two cases, \mathbf{C}_{11} non-singular and \mathbf{C}_{11} singular.

Case 1: Assume \mathbf{C}_{11} non-singular (i.e. all nets of \mathbf{C}_{11} are non-singular, i.e. have a capacitor to ground). Then (3.4.1) is equivalent to

$$\begin{cases} \mathbf{C}_{11}\mathbf{x}'_1(t) + \mathbf{G}_{11}\mathbf{x}_1(t) + \mathbf{G}_{12}\mathbf{x}_2(t) + \mathbf{I}_1(t) = 0 \\ \mathbf{G}_{21}\mathbf{x}_1(t) + \mathbf{G}_{22}\mathbf{x}_2(t) + \mathbf{I}_2(t) = 0. \end{cases} \quad (3.4.2)$$

and implies

$$\begin{aligned} \begin{cases} \underbrace{\mathbf{C}_{11}}_{=: \mathbf{C}} \mathbf{x}'_1(t) + \underbrace{(\mathbf{G}_{11} - \mathbf{G}_{12}\mathbf{G}_{22}^{-1}\mathbf{G}_{21})}_{=: \mathbf{G} \text{ (if needed, is SPD)}} \mathbf{x}_1(t) + \underbrace{\mathbf{I}_1 - \mathbf{G}_{12}\mathbf{G}_{22}^{-1}\mathbf{I}_2}_{=: \mathbf{b}(t)} = 0, \\ \mathbf{x}_2(t) = -\mathbf{G}_{22}^{-1}(\mathbf{I}_2 + \mathbf{G}_{21}\mathbf{x}_1(t)) \end{cases} & \Leftrightarrow \\ \begin{cases} \mathbf{C}\mathbf{x}'_1(t) + \mathbf{G}\mathbf{x}_1(t) = \mathbf{b}(t), \\ \mathbf{x}_2(t) = -\mathbf{G}_{22}^{-1}(\mathbf{I}_2(t) + \mathbf{G}_{21}\mathbf{x}_1(t)) \end{cases} & \Leftrightarrow \\ \begin{cases} \mathbf{x}'_1(t) = -\mathbf{C}^{-1}\mathbf{G}\mathbf{x}_1(t) + \mathbf{C}^{-1}\mathbf{b}(t), \\ \mathbf{x}_2(t) = -\mathbf{G}_{22}^{-1}(\mathbf{I}_2(t) + \mathbf{G}_{21}\mathbf{x}_1(t)) \end{cases} \end{aligned}$$

which latter ODE holds for both original capacitance matrix \mathbf{C}_o and MoveC capacitance matrix \mathbf{C}_m , and defines

$$\mathbf{x}_1^o(t) = e^{-\mathbf{C}_o^{-1}\mathbf{G}t} \left(\int_0^t e^{\mathbf{C}_o^{-1}\mathbf{G}s} \mathbf{C}_o^{-1}\mathbf{b}(s) ds + \mathbf{x}_1(0) \right), \quad (3.4.3)$$

$$\mathbf{x}_1^m(t) = e^{-\mathbf{C}_m^{-1}\mathbf{G}t} \left(\int_0^t e^{\mathbf{C}_m^{-1}\mathbf{G}s} \mathbf{C}_m^{-1}\mathbf{b}(s) ds + \mathbf{x}_1(0) \right). \quad (3.4.4)$$

Assume $\mathbf{x}_1(0) = \mathbf{0}$. Applying a standard telescoping argument

$$\begin{aligned} \mathbf{x}_1^o(t) - \mathbf{x}_1^m(t) &= e^{-\mathbf{C}_o^{-1}\mathbf{G}t} \int_0^t e^{\mathbf{C}_o^{-1}\mathbf{G}s} \mathbf{C}_o^{-1}\mathbf{b}(s) ds - e^{-\mathbf{C}_m^{-1}\mathbf{G}t} \int_0^t e^{\mathbf{C}_m^{-1}\mathbf{G}s} \mathbf{C}_m^{-1}\mathbf{b}(s) ds \\ &= e^{-\mathbf{C}_o^{-1}\mathbf{G}t} \int_0^t e^{\mathbf{C}_o^{-1}\mathbf{G}s} \mathbf{C}_o^{-1}\mathbf{b}(s) ds - e^{-\mathbf{C}_m^{-1}\mathbf{G}t} \int_0^t e^{\mathbf{C}_o^{-1}\mathbf{G}s} \mathbf{C}_o^{-1}\mathbf{b}(s) ds \\ &\quad + e^{-\mathbf{C}_m^{-1}\mathbf{G}t} \int_0^t e^{\mathbf{C}_o^{-1}\mathbf{G}s} \mathbf{C}_o^{-1}\mathbf{b}(s) ds - e^{-\mathbf{C}_m^{-1}\mathbf{G}t} \int_0^t e^{\mathbf{C}_m^{-1}\mathbf{G}s} \mathbf{C}_m^{-1}\mathbf{b}(s) ds \\ &= \left(e^{-\mathbf{C}_o^{-1}\mathbf{G}t} - e^{-\mathbf{C}_m^{-1}\mathbf{G}t} \right) \int_0^t e^{\mathbf{C}_o^{-1}\mathbf{G}s} \mathbf{C}_o^{-1}\mathbf{b}(s) ds \\ &\quad + e^{-\mathbf{C}_m^{-1}\mathbf{G}t} \int_0^t \left(e^{\mathbf{C}_o^{-1}\mathbf{G}s} \mathbf{C}_o^{-1} - e^{\mathbf{C}_m^{-1}\mathbf{G}s} \mathbf{C}_m^{-1} \right) \mathbf{b}(s) ds. \end{aligned} \quad (3.4.5)$$

Thus, small $\|\mathbf{x}_1^o(t) - \mathbf{x}_1^m(t)\|$ can be obtained by having

$$\begin{cases} e^{-\mathbf{C}_o^{-1}\mathbf{G}t} - e^{-\mathbf{C}_m^{-1}\mathbf{G}t}, & (3.4.6a) \\ e^{\mathbf{C}_o^{-1}\mathbf{G}s} \mathbf{C}_o^{-1} - e^{\mathbf{C}_m^{-1}\mathbf{G}s} \mathbf{C}_m^{-1}. & (3.4.6b) \end{cases}$$

close to the zero matrix. We suspect that these matrices are close to zero if $\mathbf{C}_o - \mathbf{C}_m := \Delta \mathbf{C}$ is small. Thus we analyze (3.4.6a) as a function of $\Delta \mathbf{C}$

$$\begin{aligned}
 h_1(\Delta \mathbf{C}) &= e^{-\mathbf{C}_o^{-1} \mathbf{G} t} - e^{-(\mathbf{C}_o - \Delta \mathbf{C})^{-1} \mathbf{G} t} \\
 &= e^{-\mathbf{C}_o^{-1} \mathbf{G} t} - e^{-[\mathbf{C}_o(\mathbf{I} - \mathbf{C}_o^{-1} \Delta \mathbf{C})]^{-1} \mathbf{G} t} \\
 &= e^{-\mathbf{C}_o^{-1} \mathbf{G} t} - e^{-(\mathbf{I} - \mathbf{C}_o^{-1} \Delta \mathbf{C})^{-1} \mathbf{C}_o^{-1} \mathbf{G} t} \\
 &= \left(e^{\mathbf{I}} - e^{(\mathbf{I} - \mathbf{C}_o^{-1} \Delta \mathbf{C})^{-1}} \right) e^{-\mathbf{C}_o^{-1} \mathbf{G} t}.
 \end{aligned} \tag{3.4.7}$$

which latter factor is small if its exponent matrix is close to zero which holds if

$$\left(\underbrace{\mathbf{I} - \mathbf{C}_o^{-1} \Delta \mathbf{C}}_{=: \Delta} \right)^{-1} \approx \mathbf{I} \Leftrightarrow \mathbf{C}_o^{-1} \Delta \mathbf{C} \approx \mathbf{0}. \tag{3.4.8}$$

Therefore, to determine how small the error $\|\mathbf{x}^o - \mathbf{x}^m\|$ we need to control the quantity $\Delta = \mathbf{C}_o^{-1} \Delta \mathbf{C}$. To this end, one can use:

For $\|\Delta\| < 1$ we have

$$\begin{aligned}
 (\mathbf{I} - \Delta)(\mathbf{I} + \Delta + \Delta^2 + \dots + \Delta^n) &= \mathbf{I} - \Delta^{n+1} \\
 \Rightarrow (\mathbf{I} - \Delta) \sum_{i=0}^{\infty} \Delta^i &= \mathbf{I} \\
 \Rightarrow (\mathbf{I} - \Delta)^{-1} &= \sum_{i=0}^{\infty} \Delta^i \approx \mathbf{I} \\
 \Rightarrow \sum_{i=0}^{\infty} (\mathbf{C}_o^{-1} \Delta \mathbf{C})^i &\approx \mathbf{I} \quad \text{or} \quad \sum_{i=1}^{\infty} (\mathbf{C}_o^{-1} \Delta \mathbf{C})^i \approx \mathbf{0}.
 \end{aligned}$$

Next we analyze (3.4.6b):

$$\begin{aligned}
 h_2(\Delta \mathbf{C}) &= e^{\mathbf{C}_o^{-1} \mathbf{G} s} \mathbf{C}_o^{-1} - e^{\mathbf{C}_m^{-1} \mathbf{G} s} \mathbf{C}_m^{-1} \\
 &= e^{\mathbf{C}_o^{-1} \mathbf{G} s} \mathbf{C}_o^{-1} - e^{(\mathbf{C}_o - \Delta \mathbf{C})^{-1} \mathbf{G} s} (\mathbf{C}_o - \Delta \mathbf{C})^{-1} \\
 &= e^{\mathbf{C}_o^{-1} \mathbf{G} s} \mathbf{C}_o^{-1} - e^{(\mathbf{I} - \mathbf{C}_o^{-1} \Delta \mathbf{C})^{-1} \mathbf{C}_o^{-1} \mathbf{G} s} (\mathbf{I} - \mathbf{C}_o^{-1} \Delta \mathbf{C})^{-1} \mathbf{C}_o^{-1} \\
 &= e^{\mathbf{C}_o^{-1} \mathbf{G} s} \mathbf{C}_o^{-1} - e^{\mathbf{C}_o^{-1} \mathbf{G} s} e^{((\mathbf{I} - \mathbf{C}_o^{-1} \Delta \mathbf{C})^{-1} - \mathbf{I}) \mathbf{C}_o^{-1} \mathbf{G} s} (\mathbf{I} - \mathbf{C}_o^{-1} \Delta \mathbf{C})^{-1} \mathbf{C}_o^{-1} \\
 &= e^{\mathbf{C}_o^{-1} \mathbf{G} s} \left(\mathbf{I} - e^{((\mathbf{I} - \mathbf{C}_o^{-1} \Delta \mathbf{C})^{-1} - \mathbf{I}) \mathbf{C}_o^{-1} \mathbf{G} s} (\mathbf{I} - \mathbf{C}_o^{-1} \Delta \mathbf{C})^{-1} \right) \mathbf{C}_o^{-1}.
 \end{aligned}$$

is small if (3.4.8) holds.

Employing Mathematica, we compute the analytical solution for the original problem derived from the circuit in Fig. 3.1 and the MoveC problem. For simplicity, assume that $c_{25} = c_i = 1$, $g_{ij} = g_i = 1$ for all $1 \leq i, j \leq 6$ and $c_{36} = c$. $\mathbf{x}_r = [x_1; x_4]$, $\mathbf{x}_c = [x_2; x_5; x_3; x_6]$.

$$\mathbf{C}_o = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & 0 & 0 \\ 0 & 0 & c+1 & -c \\ 0 & 0 & -c & c+1 \end{bmatrix}, \quad \mathbf{G}_{11} = \begin{bmatrix} 2 & 0 & -1 & 0 \\ 0 & 2 & 0 & -1 \\ -1 & 0 & 2 & 0 \\ 0 & -1 & 0 & 2 \end{bmatrix}$$

$$\mathbf{G}_{12} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \mathbf{G}_{21}^T, \quad \mathbf{G}_{22} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad \mathbf{s}(t) = \begin{bmatrix} \cos(2\pi t) \\ a \cdot \cos(2\pi t) \end{bmatrix}. \quad (3.4.9)$$

To be studied for error estimate is $e^{\pm \mathbf{C}_o^{-1} \mathbf{G} t}$ where \mathbf{C}_o is diagonalizable. Then for $c = 1$, using Mathematica we find that

$$\mathbf{C}_o^{-1} \mathbf{G} = \begin{bmatrix} 1 & \frac{1}{2} & -\frac{2}{3} & -\frac{1}{3} \\ \frac{1}{2} & 1 & -\frac{1}{3} & -\frac{2}{3} \\ -\frac{2}{3} & -\frac{1}{3} & \frac{4}{3} & \frac{2}{3} \\ -\frac{1}{3} & -\frac{2}{3} & \frac{2}{3} & \frac{4}{3} \end{bmatrix} = \mathbf{W} \mathbf{D} \mathbf{W}^{-1}$$

where

$$\mathbf{W} = \begin{bmatrix} \frac{1}{4}(1 - \sqrt{17}) & \frac{1}{4}(\sqrt{17} - 1) & \frac{1}{4}(1 + \sqrt{17}) & \frac{1}{4}(-1 - \sqrt{17}) \\ \frac{1}{4}(1 - \sqrt{17}) & \frac{1}{4}(1 - \sqrt{17}) & \frac{1}{4}(1 + \sqrt{17}) & \frac{1}{4}(1 + \sqrt{17}) \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

$$\mathbf{D} = \begin{bmatrix} \frac{1}{4}(7 + \sqrt{17}) & 0 & 0 & 0 \\ 0 & \frac{1}{12}(7 + \sqrt{17}) & 0 & 0 \\ 0 & 0 & \frac{1}{4}(7 - \sqrt{17}) & 0 \\ 0 & 0 & 0 & \frac{1}{12}(7 - \sqrt{17}) \end{bmatrix}$$

and $e^{\pm \mathbf{C}_o^{-1} \mathbf{G} t} = \mathbf{W} e^{\pm \mathbf{D} t} \mathbf{W}^{-1}$. Furthermore, compute

$$\mathbf{x}^o(t) = e^{-\mathbf{C}_o^{-1} \mathbf{G} t} \left(\int_0^t e^{\mathbf{C}_o^{-1} \mathbf{G} s} \mathbf{C}_o^{-1} \mathbf{b}(s) ds + \mathbf{x}(0) \right)$$

where for $a = 0$ in (3.4.9) we find

$$\begin{aligned} \mathbf{C}_o^{-1} \mathbf{b}(s) &= \mathbf{C}_o^{-1} \mathbf{G}_{12} \mathbf{G}_{22}^{-1} \mathbf{s}(t) \\ &= \begin{bmatrix} -\frac{1}{3} & -\frac{1}{6} \\ -\frac{1}{6} & -\frac{1}{3} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{s}(t). \end{aligned}$$

For \mathbf{C}_m , (see (3.2.14)), after permutation we get

$$\mathbf{C}_m = \begin{bmatrix} 2 + c & -1 - c & 0 & 0 \\ -1 - c & 2 + c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and $\mathbf{C}_m^{-1} \mathbf{G} = \mathbf{W}_m \mathbf{D}_m \mathbf{W}_m^{-1}$ where ($c = 1$ is the capacitance of the capacitor c_{36} to be moved)

$$\mathbf{W}_m = \begin{bmatrix} \frac{1}{4}(1 - \sqrt{17}) & \frac{1}{20}(3\sqrt{41} - 17) & \frac{1}{4}(1 + \sqrt{17}) & \frac{1}{20}(-17 - 3\sqrt{41}) \\ \frac{1}{4}(1 - \sqrt{17}) & \frac{1}{20}(17 - 3\sqrt{41}) & \frac{1}{4}(1 + \sqrt{17}) & \frac{1}{20}(17 + 3\sqrt{41}) \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

$$\mathbf{D}_m = \begin{bmatrix} \frac{1}{4}(7 + \sqrt{17}) & 0 & 0 & 0 \\ 0 & \frac{1}{20}(23 + 3\sqrt{41}) & 0 & 0 \\ 0 & 0 & \frac{1}{4}(7 - \sqrt{17}) & 0 \\ 0 & 0 & 0 & \frac{1}{20}(23 - 3\sqrt{41}) \end{bmatrix}$$

and for $a = 0$,

$$\begin{aligned} \mathbf{C}_m^{-1}\mathbf{b}(s) &= \mathbf{C}_m^{-1}\mathbf{G}_{12}\mathbf{G}_{22}^{-1}\mathbf{s}(t) \\ &= \begin{bmatrix} -\frac{3}{10} & -\frac{1}{5} \\ -\frac{1}{5} & -\frac{3}{10} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{s}(t). \end{aligned}$$

The solutions \mathbf{x}^o and \mathbf{x}^m are plotted in Fig. 3.3 for $c = 1$ and in Fig. 3.5 for $c = 0.25$:

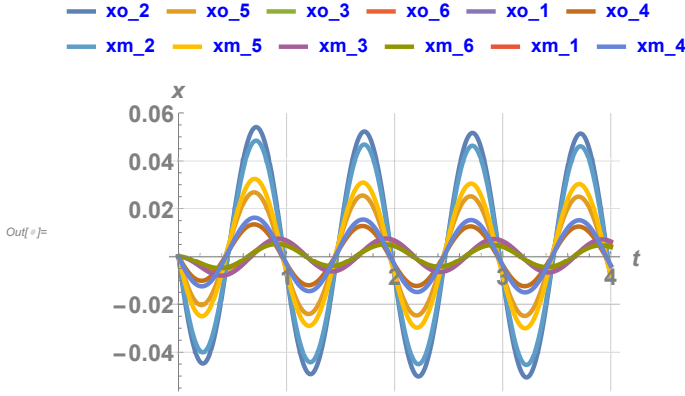


Figure 3.3: Analytical solution of \mathbf{x}^o and \mathbf{x}^m respectively, $c = 1$.

We observe that if the move capacitor c_{36} is small, the error $|\mathbf{x}^o - \mathbf{x}^m|$ is small. Finally, the inverse of \mathbf{C}_m can be determined analytically as well, as follows. Consider $\mathbf{C}_o^{-1}\Delta\mathbf{C}$. First, assume we move precisely 1 capacitor then

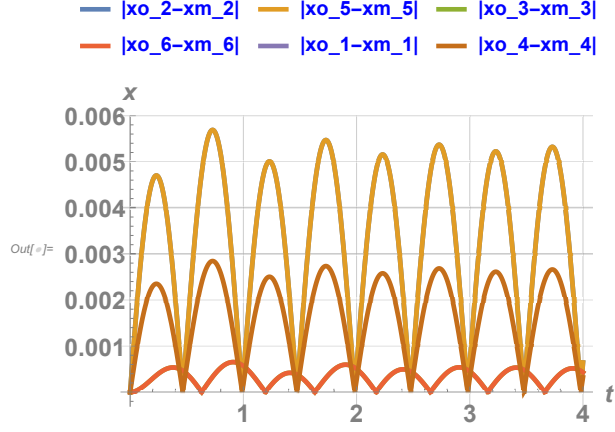
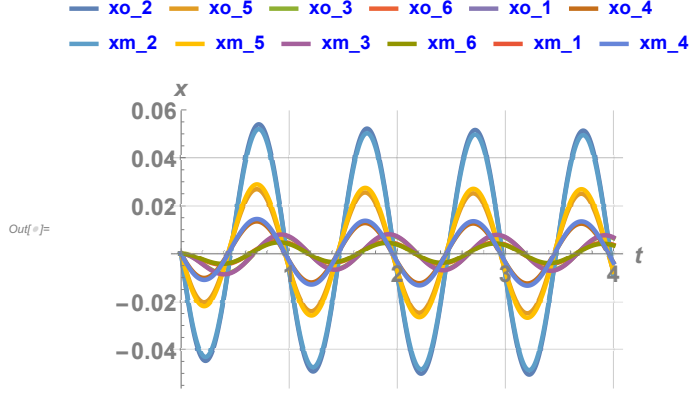
$$\Delta\mathbf{C} = -\underbrace{\mathbf{d}_{36}c_{36}\mathbf{d}_{36}^T}_{\mathbf{u}\mathbf{u}^T} + \underbrace{\mathbf{d}_{25}c_{36}\mathbf{d}_{25}^T}_{\mathbf{v}\mathbf{v}^T},$$

two rank-1 updates $\mathbf{C}_m = \mathbf{C}_o - \Delta\mathbf{C}$

$$\begin{aligned} \Rightarrow \mathbf{C}_m^{-1} &= (\mathbf{C}_o - \Delta\mathbf{C})^{-1} =: (\mathbf{C}_o + \mathbf{u}\mathbf{u}^T - \mathbf{v}\mathbf{v}^T)^{-1} \Leftrightarrow \\ \mathbf{C}_m^{-1} &= \underbrace{(\mathbf{C}_o + \mathbf{u}\mathbf{u}^T)}_{\mathbf{B}} - \mathbf{v}\mathbf{v}^T)^{-1}. \end{aligned}$$

Using Woodbury formula, then

$$(\mathbf{B} - \mathbf{v}\mathbf{v}^T)^{-1} = \mathbf{B}^{-1} + \frac{\mathbf{B}^{-1}\mathbf{v}\mathbf{v}^T\mathbf{B}^{-1}}{1 + \mathbf{v}^T\mathbf{B}^{-1}\mathbf{v}}$$


 Figure 3.4: Absolute error of \mathbf{x}^o and \mathbf{x}^m , $c = 1$.

 Figure 3.5: Analytical solution of \mathbf{x}^o and \mathbf{x}^m respectively, $c = 0.25$.

where

$$\mathbf{B}^{-1} = \mathbf{C}_o^{-1} - \frac{\mathbf{C}_o^{-1} \mathbf{u} \mathbf{u}^T \mathbf{C}_o^{-1}}{1 + \mathbf{u}^T \mathbf{C}_o^{-1} \mathbf{u}}.$$

Thus

$$\mathbf{C}_m^{-1} = \mathbf{C}_o^{-1} - \frac{\mathbf{C}_o^{-1} \mathbf{u} \mathbf{u}^T \mathbf{C}_o^{-1}}{1 + \mathbf{u}^T \mathbf{C}_o^{-1} \mathbf{u}} + \frac{\left(\mathbf{C}_o^{-1} - \frac{\mathbf{C}_o^{-1} \mathbf{u} \mathbf{u}^T \mathbf{C}_o^{-1}}{1 + \mathbf{u}^T \mathbf{C}_o^{-1} \mathbf{u}} \right) \mathbf{v} \mathbf{v}^T \left(\mathbf{C}_o^{-1} - \frac{\mathbf{C}_o^{-1} \mathbf{u} \mathbf{u}^T \mathbf{C}_o^{-1}}{1 + \mathbf{u}^T \mathbf{C}_o^{-1} \mathbf{u}} \right)}{1 + \mathbf{v}^T \left(\mathbf{C}_o^{-1} - \frac{\mathbf{C}_o^{-1} \mathbf{u} \mathbf{u}^T \mathbf{C}_o^{-1}}{1 + \mathbf{u}^T \mathbf{C}_o^{-1} \mathbf{u}} \right) \mathbf{v}}.$$

The inverse of \mathbf{C}_o is easy to compute because it is a block Toeplitz matrix, see section 3.6. Summarized, we find that for case 1:

Corollary 3.4.1.

$$\|\mathbf{v}(t) - \widehat{\mathbf{v}}(t)\| = \|\mathbf{x}^o(t) - \mathbf{x}^m(t)\| \leq c(t) \|\Delta \mathbf{C}\| \quad (3.4.10)$$

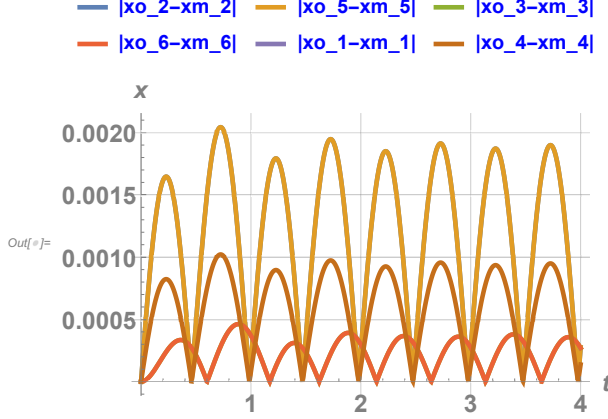


Figure 3.6: Absolute error of \mathbf{x}^o and \mathbf{x}^m , $c = 0.25$.

if all involved sources are smooth enough. More precisely, the antiderivatives in (3.4.3) and (3.4.4) should exist in the sense of Riemann or Lebesgue integration. Hence, the integrals/antiderivatives of $s \rightarrow e^{C_o^{-1}Gs} C_o^{-1}\mathbf{b}(s)$, $s \rightarrow e^{C_m^{-1}Gs} C_m^{-1}\mathbf{b}(s)$ should exist, which is definitely the case for continuous $s \rightarrow e^{C_o^{-1}Gs} C_o^{-1}\mathbf{b}(s)$ which implies for continuous $s \rightarrow \mathbf{b}(s)$. Essentially, binary (non-differentiable) periodic sources (such as the Matlab "square" function) induce an element $\mathbf{b} \in (L^1([0, T]))^N$ which implies that $t \rightarrow \int_0^t e^{C_o^{-1}Gs} C_o^{-1}\mathbf{b}(s)ds$, $t \in [0, T]$ an element of $W^{1,1}([0, T])^N$ using Lebesgue integration, where

$$W^{k,p}(\Omega) = \{u \in L^p(\Omega) : D^\alpha u \in L^p(\Omega), \forall |\alpha| \leq k\}.$$

Case 2: For our simplest error analysis case, we demand that both \mathbf{C}_{move} and \mathbf{C}_{orig} are non-singular. This may not be automatically the case. Assume $V_1 = \{1, 3\}$ and $V_2 = \{2, 4\}$ i.e., nodes 1 and 3 belong to component 1 of G or G -net 1, and respectively. Also, c_b is in component 1 of C and c_a is to be moved capacitor. There are two cases:

1. c_a is also in component 1 of C . C -net 1 contains nodes $\{1, 2, 3, 4, 5\}$. The resulting C -net is the same.
2. c_a is in component 2 of C , C -net 1 contains nodes $\{1, 2\}$ (Fig. 3.7) and C -net 2 contains nodes $\{3, 4, 5\}$ (Fig. 3.8). This can create extra C -nets (Fig. 3.10). For example:

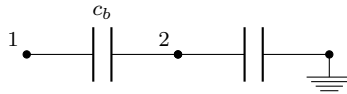


Figure 3.7: C -net 1.

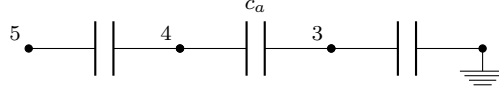


Figure 3.8: C-net 2.

Move c_a onto c_b :

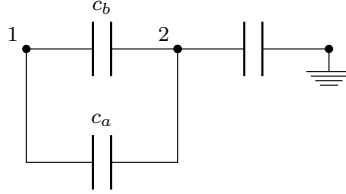


Figure 3.9: C-net 1 is non-singular.



Figure 3.10: C-net 2: move and one of components is singular.

We can rewrite equation (3.4.1) into:

$$\begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{x}'(t) + \mathbf{G}\mathbf{x}(t) = \mathbf{b}(t). \quad (3.4.11)$$

Assume \mathbf{C} is singular. Because it is symmetric there exists $\mathbf{Q}^{-1} = \mathbf{Q}^T$ such that $\mathbf{Q}^{-1}\mathbf{C}\mathbf{Q} = \begin{bmatrix} \Lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ which, setting $\mathbf{B} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$, induces a splitting different from (3.4.11):

$$\begin{aligned} \mathbf{B}^{-1} \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{B}\mathbf{y}'(t) + \mathbf{B}^{-1}\mathbf{G}\mathbf{B}\mathbf{y}(t) &= \mathbf{B}^{-1}\mathbf{b}(t) \Leftrightarrow \\ \begin{bmatrix} \Lambda & \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{y}'(t) + \mathbf{B}^{-1}\mathbf{G}\mathbf{B}\mathbf{y}(t) &= \mathbf{B}^{-1}\mathbf{b}(t) \quad \mathbf{E} := \mathbf{B}^{-1}\mathbf{G}\mathbf{B} \\ \begin{bmatrix} \Lambda & \mathbf{0} \end{bmatrix} \mathbf{y}'(t) + \begin{bmatrix} \mathbf{E}_{11} & \mathbf{E}_{12} \\ \mathbf{E}_{21} & \mathbf{E}_{22} \end{bmatrix} \mathbf{y}(t) &= \begin{bmatrix} \mathbf{s}_1(t) \\ \mathbf{s}_2(t) \end{bmatrix} \end{aligned}$$

where one needs \mathbf{E}_{22} to be non-singular, which is the case because \mathbf{G} is positive definite thus $\mathbf{E} = \mathbf{B}^{-1}\mathbf{G}\mathbf{B} = \mathbf{B}^T\mathbf{G}\mathbf{B}$ is also positive definite $(\mathbf{B}^{-1}\mathbf{G}\mathbf{B}, \mathbf{y}) = (\mathbf{G}\mathbf{B}\mathbf{y}, \mathbf{B}\mathbf{y}) = (\mathbf{G}\mathbf{z}, \mathbf{z}) > 0$ if $\mathbf{y} \neq \mathbf{0}$ which implies \mathbf{E}_{22} positive definite, i.e., \mathbf{E}_{22} is non-singular. Thus we find

$$\begin{cases} \Lambda_o \mathbf{y}'_1(t) = (\mathbf{E}_{12}\mathbf{E}_{22}^{-1}\mathbf{E}_{21} - \mathbf{E}_{11})\mathbf{y}_1(t) - \mathbf{E}_{12}\mathbf{E}_{22}^{-1}\mathbf{s}_2(t) + \mathbf{s}_1(t), \\ \Lambda_m \hat{\mathbf{y}}'_1(t) = (\hat{\mathbf{E}}_{12}\hat{\mathbf{E}}_{22}^{-1}\hat{\mathbf{E}}_{21} - \hat{\mathbf{E}}_{11})\hat{\mathbf{y}}_1(t) - \hat{\mathbf{E}}_{12}\hat{\mathbf{E}}_{22}^{-1}\mathbf{s}_2(t) + \mathbf{s}_1(t) \end{cases}$$

Observe $\dim(\ker(\mathbf{C}_o)) = \dim(\ker(\mathbf{C}_m))$ is needed to ensure we solve an identical amount of ODEs for both the original and MoveC problem. To conclude: we have provided error estimates which will be small if $\mathbf{C}_o^{-1}\Delta\mathbf{C} \approx \mathbf{0}$ (see (3.4.8)). Therefore, Section 3.5 will estimate $\|\mathbf{C}\|_\infty$ and $\|\Delta\mathbf{C}\|_\infty$.

3.5 Estimates for $\|\mathbf{C}\|_\infty$, $\|\Delta\mathbf{C}\|_\infty$, $\|\mathbf{1}_p^T \mathbf{C}\|_\infty$ and $\|\mathbf{1}_p^T \Delta\mathbf{C}\|_\infty$

Earlier versions of the error estimates for the net current difference (3.2.24), (3.2.25) and the voltage difference (3.4.5) require $\|\mathbf{1}_p^T \mathbf{C}\|_\infty$ and $\|\mathbf{1}_p^T \Delta\mathbf{C}\|_\infty$. That is why these estimates are presented as well. We have \mathbf{C} symmetric positive definite, thus $\|\mathbf{C}\|_2 = \rho(\mathbf{C})$, the spectral radius of \mathbf{C} and $\|\mathbf{C}^{-1}\|_2 = \frac{1}{\lambda_{\min}(\mathbf{C})}$ where $\lambda_{\min}(\mathbf{C})$ is not known. A lower bound for $\|\mathbf{C}^{-1}\|$ can be obtained using that 2-norm and ∞ -norm are sub-multiplicative, i.e., $\|\mathbf{I}\| = \|\mathbf{C}\mathbf{C}^{-1}\| \leq \|\mathbf{C}\| \|\mathbf{C}^{-1}\| \Rightarrow \|\mathbf{C}^{-1}\| \geq \frac{1}{\|\mathbf{C}\|}$, i.e., $\|\mathbf{C}^{-1}\|_2 \geq \frac{1}{\rho(\mathbf{C})}$, which is a very large bound.

3.5.1 The value of $\|\mathbf{C}\|_\infty$

By definition, the infinity norm of a matrix $\mathbf{C} \in \mathbb{R}^{N \times N}$ is

$$\|\mathbf{C}\|_\infty = \max_{i=1,\dots,N} \underbrace{\sum_{j=1}^N |c_{ij}|}_{=:s_i}$$

Thus, to calculate $\|\mathbf{C}\|_\infty$ one must determine its absolute row sums $s_i, 1 \leq i \leq N$. Each non-zero row in \mathbf{C} is related to a node $1 \leq i \leq N$ with at least one connected capacitor. If only one capacitor is connected to node i , the row has only two non-zero entries as shown in Fig. 3.11

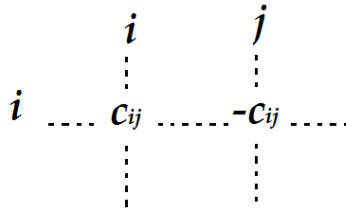


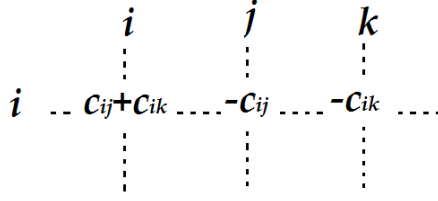
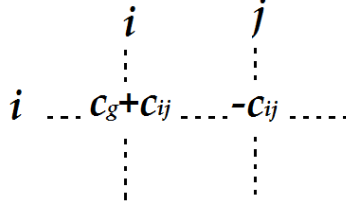
Figure 3.11: Two non-zero entries at row i .

which leads to $s_i = |c_{ij}| + |-c_{ij}| = 2c_{ij}$.

If two capacitors are connected to node i , the row has three non-zero entries as shown in Fig. 3.12

and $s_i = |c_{ij} + c_{ik}| + |-c_{ij}| + |-c_{ik}| = 2(c_{ij} + c_{ik})$.

Assuming that E_{cap} also contains the edges related to capacitors connected to the reference node. If $(i, g) \in E_{\text{cap}}$ and g is the reference node, then column and row g will be eliminated from \mathbf{C} . Thus if row i would have only one capacitor connected to a non-reference node (see Fig. 3.13) one finds $s_i = |c_g + c_{ij}| + |-c_{ij}| = 2c_{ij} + c_g$.


 Figure 3.12: Three none-zero entries at row i .

 Figure 3.13: Node i has one capacitor connected to the reference node g and one capacitor connected to a non-reference node j .

Thus in general one obtains

$$s_i = 2 \sum_{\substack{(i,j) \in E_{\text{cap}} \\ j \text{ non-reference node}}} c_{ij} + \sum_{\substack{(i,j) \in E_{\text{cap}} \\ j \text{ reference node}}} c_{ij},$$

and thus

$$\|\mathbf{C}\|_{\infty} = \max_{i=1,\dots,N} \left\{ \sum_{\substack{(i,j) \in E_{\text{cap}} \\ j \text{ non-reference node}}} 2c_{ij} + \sum_{\substack{(i,j) \in E_{\text{cap}} \\ j \text{ reference node}}} c_{ij} \right\}.$$

3.5.2 The value of $\|\Delta \mathbf{C}_{\text{move}}\|_{\infty}$ and $\|\Delta \mathbf{C}_{\text{del}}\|_{\infty}$

Assume that for each pair of net numbers $1 \leq p, q \leq P, p \neq q$, there are two special nodes (i_{pq}, j_{pq}) which form one edge $m(p, q) := (i_{pq}, j_{pq})$ called the destination edge. We also assume that capacitors to ground cannot be moved (as the related ground nodes have been eliminated).

We intend to move the capacitor between nodes i and j with $i \in V_p$ (node i in net p) and $j \in V_q$ (node j in net q), $p \neq q$. The destination is between nodes i_{pq} and j_{pq} , $i_{pq} \neq i$ and $j_{pq} \neq j$. For $\alpha = 0, 1$ ($\alpha = 0$: create $\Delta \mathbf{C}_{\text{del}}$; $\alpha = 1$: create $\Delta \mathbf{C}_{\text{move}}$) one obtains

$$\Delta \mathbf{C}_{\text{del/move}} = \sum_{\substack{p,q \\ p \neq q}} \left(\underbrace{\sum_{\substack{(i,j) \in E_{p,q}, \\ (i,j) \neq m(p,q), \\ c_{ij} < \tau_{\text{cap}}}} -c_{ij} \mathbf{d}_{ij} \mathbf{d}_{ij}^T}_{\text{delete}} + \alpha \underbrace{\sum_{\substack{(i,j) \in E_{p,q}, \\ (i,j) \neq m(p,q), \\ c_{ij} < \tau_{\text{cap}}}} c_{ij} \mathbf{d}_{m(p,q)} \mathbf{d}_{m(p,q)}^T}_{\text{add}} \right) \quad (3.5.1)$$

Note that (3.5.1) contains two terms. In further examinations we will omit the condition $c_{ij} < \epsilon$ for simplicity of notation. For the first term of (3.5.1), each edge which will be moved i.e., each $(i, j) \in E_{p,q}$, $(i, j) \neq m(p, q)$ first is deleted which leads to a row and related row sum (see Fig. 3.14)

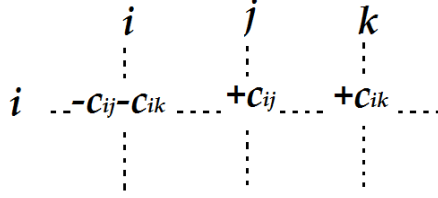


Figure 3.14: Delete capacitors c_{ij} and c_{ik} .

and

$$s_i = \sum_{j=1}^N |c_{ij}| = |-c_{ij} - c_{ik}| + |c_{ij}| + |c_{ik}| = 2(c_{ij} + c_{ik}).$$

The second term of (3.5.1) accumulates all capacitances at the destination edge(s) $m(p, q)$ (see Fig. 3.15) and has a typical row and related row sum $s_{i_{pq}} = 2\alpha(c_{ij} + c_{ik})$.

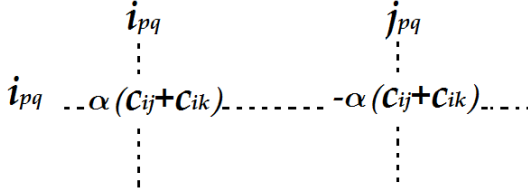


Figure 3.15: Accumulate capacitors c_{ij} and c_{ik} at the destination edge (i_{pq}, j_{pq}) .

Summarized we find that for the first term:

$$s_i = \sum_{\substack{p,q \\ p \neq q}} \sum_{\substack{(i,j) \in E_{(p,q)}, \\ (i,j) \neq m(p,q)}} 2c_{ij}$$

and for the second term, for the sake of simplicity we assume that no two move edges are connected

$$s_{i_{pq}} = \alpha \sum_{\substack{(i,j) \in E_{(p,q)}, \\ (i,j) \neq m(p,q)}} 2c_{ij}$$

which combines leads to

$$\|\Delta \mathbf{C}_{\text{move}}\|_{\infty} = \max \left\{ \max_{\substack{p,q \\ p \neq q}} \left\{ \sum_{\substack{(i,j) \in E_{(p,q)}, \\ (i,j) \neq m(p,q)}} 2c_{ij} \right\}, \max_{\substack{p,q \\ p \neq q}} \left\{ \alpha \sum_{\substack{(i,j) \in E_{(p,q)}, \\ (i,j) \neq m(p,q)}} 2c_{ij} \right\} \right\}. \quad (3.5.2)$$

The expression (3.5.2) cannot be simplified. To see this, let the circuit be (see Fig. 3.16)

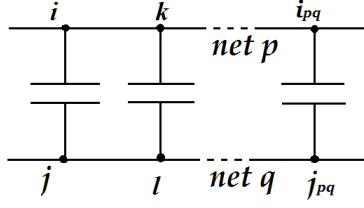


Figure 3.16: Move/deleted capacitors. Nodes have only one capacitor connected to.

If c_{ij}, c_{kl} are moved to $c(i_{pq}, j_{pq})$ then the maximum is attained in the second term of (3.5.2)

$$\begin{aligned} \|\Delta \mathbf{C}\|_{\infty} &= \max \left\{ \max\{2c_{ij}, 2c_{kl}\}, \max_{p,q} \{2\alpha(c_{ij} + c_{kl})\} \right\} \\ &= \max_{p,q} \{2\alpha(c_{ij} + c_{kl})\}. \end{aligned}$$

However, when there are more than one capacitor connected to a node to be moved/deleted (see Fig. 3.17) the maximum is attained in the first term of (3.5.2)

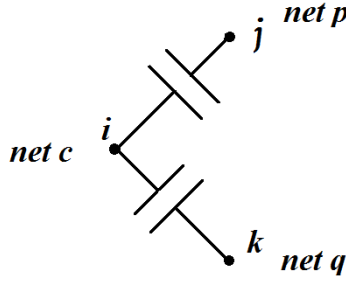


Figure 3.17: Move/deleted capacitors. Nodes have capacitors connected to.

$$\|\Delta \mathbf{C}_{\text{move}}\|_{\infty} = \|\Delta \mathbf{C}_{\text{del}}\|_{\infty} = \max\{2c_{ij} + 2c_{ik}, \max\{2\alpha c_{ij} + 2\alpha c_{ik}\}\},$$

with $i \in \text{net } c, j \in \text{net } p, k \in \text{net } q$.

3.5.3 The value of $\|\Delta \mathbf{C}_{\text{split}}\|_{\infty}$

Here we consider $\|\Delta \mathbf{C}_{\text{split}}\|_{\infty}$ for $\Delta \mathbf{C}_{\text{split}}$ related to $\mathbf{C}_{\text{split}} = \mathbf{C} + \Delta \mathbf{C}_{\text{split}}$. Let E_{split} be the set of edges related to the to be split capacitors. c_{ij} and c_{ik} with $i \in I_p, j \in I_q, k \in I_c, p \neq q \neq c$ are made to connected to the reference node (Fig. 3.18).

We say that c_{ij} and c_{ik} are made to connected to the reference node because the matrix representation of a capacitor connected to the reference node is a diagonal entry in \mathbf{C} . Splitting coupling capacitors c_{ij} and c_{ik} results in four capacitors connected to the reference nodes

$$\mathbf{C}_{\text{split}} = \begin{bmatrix} c_{ij} + c_{ik} & 0 & 0 \\ 0 & c_{ij} & 0 \\ 0 & 0 & c_{ik} \end{bmatrix}, \text{ with } \mathbf{C} = \begin{bmatrix} c_{ij} + c_{ik} & -c_{ij} & -c_{ik} \\ -c_{ij} & c_{ij} & 0 \\ -c_{ik} & 0 & c_{ik} \end{bmatrix}.$$

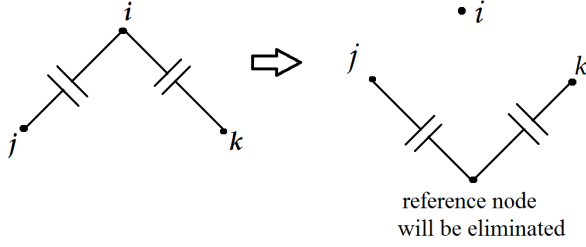


Figure 3.18: Split coupling capacitors to the reference node.

We get for two to be removed/split capacitors with nodes i, j, k in arbitrary nets that

$$\Delta \mathbf{C}_{\text{split}} = \begin{bmatrix} 0 & c_{ij} & c_{ik} \\ c_{ij} & 0 & 0 \\ c_{ik} & 0 & 0 \end{bmatrix},$$

all edges or components in netlist between net p and q . Herein, netlist is a connectivity description of an electronic circuit, see [10, 45]. In the general case one finds

$$\|\Delta \mathbf{C}_{\text{split}}\|_{\infty} = \max_{i \text{ with } (i, \cdot) \in E_{\text{split}}} \left\{ \sum_{(i, j) \in E_{\text{split}}} c_{ij} \right\},$$

i.e., $\|\Delta \mathbf{C}\|_{\infty}$ is the maximum total connected capacitance over all nodes $(i \text{ with } (i, \cdot) \in E_{\text{split}})$.

3.5.4 Calculation of $\|\mathbf{1}_k^T \mathbf{C}\|_{\infty}$

Now each non-zero row in \mathbf{C} is related to a capacitor, either interior to the net or on its boundary. Boundary (inter-net) capacitors are related to nodes $(i, j) \in E_{p,q}$ with $p \neq q$. Same net capacitors related to $(i, j) \in E_c$ (see Fig. 3.19) do not contribute to $\mathbf{1}_c^T \mathbf{C}$.

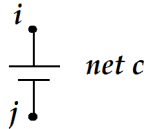


Figure 3.19: Same net capacitor.

$$i, j \in I_c \Rightarrow \mathbf{1}_c^T \left(c_{ij} \mathbf{d}_{ij} \mathbf{d}_{ij}^T \right) = \mathbf{0}^T \in \mathbb{R}^{1 \times N}, \text{ because}$$

$$\begin{aligned}
 \mathbf{1}_c^T (c_{ij} \mathbf{d}_{ij} \mathbf{d}_{ij}^T) &= \left(\sum_{k \in I_c} \mathbf{e}_k \right)^T (c_{ij} \mathbf{d}_{ij} \mathbf{d}_{ij}^T) \\
 &= \left(\sum_{k \in I_c} \mathbf{e}_k \right)^T c_{ij} (\mathbf{e}_i - \mathbf{e}_j) (\mathbf{e}_i - \mathbf{e}_j)^T \\
 &= (\mathbf{e}_i + \mathbf{e}_j)^T c_{ij} (\mathbf{e}_i - \mathbf{e}_j) (\mathbf{e}_i - \mathbf{e}_j)^T = \mathbf{0}^T.
 \end{aligned}$$

since $(\mathbf{e}_i^T + \mathbf{e}_j^T)(\mathbf{e}_i - \mathbf{e}_j) = 1 - 0 + 0 - 1 = 0$.

Consider node i with all of its connected capacitors, here with $(i, j), (i, k) \in E_{bnd}$ (Fig. 3.20)

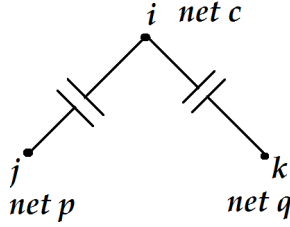


Figure 3.20: Node i connected to nodes j and k via capacitors.

which contributes these entries to \mathbf{C} :

$$\begin{array}{ccccc}
 c_{ij} + c_{ik} & \dots & -c_{ij} & \dots & -c_{ik} & \leftarrow \text{row } i \text{ in net } c \\
 \vdots & \ddots & \vdots & \vdots & \vdots & \\
 -c_{ij} & \dots & c_{ij} & \dots & \vdots & \leftarrow \text{row } j \text{ in net } p \\
 \vdots & \vdots & \vdots & \ddots & \vdots & \\
 -c_{ik} & \dots & \dots & \dots & c_{ik} & \leftarrow \text{row } k \text{ in net } q
 \end{array} \tag{3.5.3}$$

and this node leaves only one row with nonzeros $c_{ij} + c_{ik}, -c_{ij}, -c_{ik}$. The part of the matrix shown in (3.5.3) is obtained by

$$\begin{aligned}
 \underbrace{\left(\sum_{l \in I_c} \mathbf{e}_l \right)^T}_{\mathbf{1}_c^T} [c_{ij} \mathbf{d}_{ij} \mathbf{d}_{ij}^T + c_{ik} \mathbf{d}_{ik} \mathbf{d}_{ik}^T] &= \mathbf{e}_i^T [c_{ij} \mathbf{d}_{ij} \mathbf{d}_{ij}^T + c_{ik} \mathbf{d}_{ik} \mathbf{d}_{ik}^T] \\
 &= c_{ij} \underbrace{\mathbf{e}_i^T (\mathbf{e}_i - \mathbf{e}_j)}_{=1} \mathbf{d}_{ij}^T + c_{ij} \underbrace{\mathbf{e}_i^T (\mathbf{e}_i - \mathbf{e}_k)}_{=1} \mathbf{d}_{ik}^T \\
 &= c_{ij} \mathbf{d}_{ij}^T + c_{ik} \mathbf{d}_{ik}^T \\
 &= [0, \dots, \overset{i}{\underset{\downarrow}{c_{ij}}}, c_{ik}, 0, \dots, 0, \overset{j}{\underset{\downarrow}{-c_{ij}}}, 0, \dots, 0, \overset{k}{\underset{\downarrow}{c_{ik}}}, 0, \dots, 0],
 \end{aligned}$$

so

$$\sum_{l=1}^N c_{il} = 2 \sum_{\substack{i \in I_c, l \in I_d \\ c \neq d}} c_{il}. \quad (3.5.4)$$

To be taken into account is that grounded nodes are eliminated

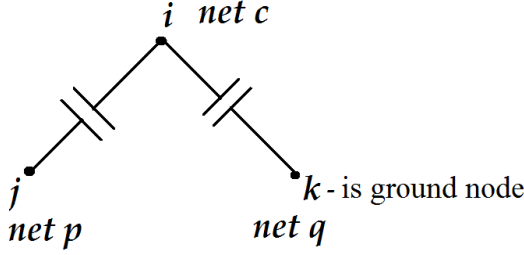


Figure 3.21: A part of a circuit with reference node k .

If k is a ground node (in net q) (Fig. 3.21) then only $+c_{ik}$ occurs the diagonal of \mathbf{C} (at location ii), and $-c_{ik}$ does not occur. Together with (3.5.4) this leads to

$$\sum_{l=1}^N c_{il} = 2 \sum_{\substack{i \in I_c, l \in I_d \\ c \neq d}} c_{il} + \sum_{\substack{i \in I_c, l \in I_{gnd} \\ c \neq gnd}} c_{il}$$

All combined one gets (now assume $(i, \cdot) \in E_k$, above assumed $(i, \cdot) \in E_c$):

$$\left\| \mathbf{1}_k^T \mathbf{C} \right\|_{\infty} = \max_{(i, \cdot) \in E_{kq}} \left\{ 2 \sum_{\substack{i \in I_k, l \in I_q \\ k \neq q}} c_{ij} + \sum_{\substack{i \in I_k, l \in I_{gnd} \\ k \neq gnd}} c_{ij} \right\}.$$

3.5.5 Calculation of $\left\| \mathbf{1}_q^T \Delta \mathbf{C}_{\text{move/del}} \right\|_{\infty}$ and $\left\| \mathbf{1}_q^T \Delta \mathbf{C}_{\text{split}} \right\|_{\infty}$

We calculate $\left\| \mathbf{1}_q^T \Delta \mathbf{C} \right\|_{\infty}$ for the case that $\Delta \mathbf{C}$ is related to $\mathbf{C}_{\text{move}}/\mathbf{C}_{\text{del}}$ (is one case, α dependent) and the case that $\Delta \mathbf{C}$ is related to $\mathbf{C}_{\text{split}}$.

Calculation of $\left\| \mathbf{1}_q^T \Delta \mathbf{C}_{\text{move/del}} \right\|_{\infty}$

First, let $\Delta \mathbf{C}$ be related to $\mathbf{C}_{\text{move}}/\mathbf{C}_{\text{del}}$. Again, since ground nodes are eliminated, capacitors to ground cannot be moved, and are assumed are not deleted. Multi-

ply (3.5.1) with $\mathbf{1}_p^T$ leads to two terms, the first being

$$\begin{aligned}
 \mathbf{1}_p^T \Delta \mathbf{C}_{\text{move/del}} &= \left(\sum_{l \in I_p} \mathbf{e}_l \right)^T \left[\sum_{\substack{(i,j) \in E_{\text{move}} \cap E_{p,q} \\ p \neq q}} -c_{ij} \mathbf{d}_{ij} \mathbf{d}_{ij}^T \right] \\
 &\stackrel{=}{=} \sum_{j \neq I_p \forall j} \sum_{i \in I_p} \mathbf{e}_i^T \left[\sum_{\substack{(i,j) \in E_{\text{move}} \cap E_{p,q} \\ p \neq q}} -c_{ij} \mathbf{d}_{ij} \mathbf{d}_{ij}^T \right] \\
 &= \sum_{\substack{(i,j) \in E_{\text{move}} \cap E_{p,q} \\ p \neq q \\ i \in I_p}} -c_{ij} \underbrace{\mathbf{e}_i^T \mathbf{d}_{ij}}_{=1} \mathbf{d}_{ij}^T \\
 &= - \sum_{\substack{(i,j) \in E_{\text{move}} \cap E_{p,q} \\ p \neq q \\ i \in I_p}} c_{ij} \mathbf{d}_{ij}^T.
 \end{aligned}$$

The second term is

$$\begin{aligned}
 \mathbf{1}_p^T \Delta \mathbf{C}_{\text{move}} &= \left(\sum_{l \in I_p} \mathbf{e}_l \right) \alpha \sum_{\substack{(i,j) \in E_{\text{move}} \cap E_{p,q} \\ p \neq q \\ (i_{pq}, j_{pq}) \notin E_{\text{move}} \\ (i_{pq}, j_{pq}) \in E_{pq}}} c_{ij} \mathbf{d}_{i_{pq}, j_{pq}} \mathbf{d}_{i_{pq}, j_{pq}}^T \\
 &= \alpha \sum_{\substack{(i,j) \in E_{\text{move}} \cap E_{p,q} \\ p \neq q \\ i_{pq} \in I_p \\ (i_{pq}, j_{pq}) \notin E_{\text{move}} \\ (i_{pq}, j_{pq}) \in E_{pq}}} c_{ij} \underbrace{\mathbf{e}_{i_{pq}}^T \mathbf{d}_{i_{pq}, j_{pq}}}_{=1} \mathbf{d}_{i_{pq}, j_{pq}}^T \\
 &= \alpha \sum_{\substack{(i,j) \in E_{\text{move}} \cap E_{p,q} \\ i_{pq} \in I_p \\ (i_{pq}, j_{pq}) \notin E_{\text{move}} \\ (i_{pq}, j_{pq}) \in E_{pq}}} c_{ij} \mathbf{d}_{i_{pq}, j_{pq}}^T.
 \end{aligned}$$

The last two results combined leads to

$$\begin{aligned}
 \|\mathbf{1}_k^T \Delta \mathbf{C}_{\text{move/del}}\|_\infty &= \max \left\{ \max_{\substack{(i,\cdot) \in E_{\text{move}} \cap E_{kq} \\ i \neq i_{kq}}} \left\{ \sum_{(i,j) \in E_{kq}} 2c_{ij} \right\}, \right. \\
 &\quad \left. \max_{(i_{kq}, \cdot) \in E_{kq}} \left\{ \alpha \sum_{(i_{kq}, j) \in E_{kq}} 2c_{i_{kq}j} \right\} \right\} \quad (3.5.5)
 \end{aligned}$$

where the last term

$$\alpha \sum_{(i_{kq}, j) \in E_{kq}} 2c_{i_{kq}j} = \alpha \sum_{(i_{kq}, j_{kq}) \in E_{kq}} 2c_{i_{kq}j_{kq}}.$$

Both inside maxima (3.5.5) are implicit maxima over nets $q \neq k$, coupled to net k .

Calculation of $\|\mathbf{1}_k^T \Delta \mathbf{C}_{\text{split}}\|_\infty$

$$\begin{aligned}
 \mathbf{1}_p^T \Delta \mathbf{C}_{\text{split}} &= \left(\sum_{l \in I_p} \mathbf{e}_l \right)^T \left[\sum_{\substack{(i,j) \in E_{\text{split}} \cap E_{p,q} \\ p \neq q}} c_{ij} (\mathbf{e}_i \mathbf{e}_j^T + \mathbf{e}_j \mathbf{e}_i^T) \right] \\
 &\stackrel{\text{}}{=} \sum_{j \neq I_p \forall j \ i \in I_p} \mathbf{e}_i^T \left[\sum_{\substack{(i,j) \in E_{\text{split}} \cap E_{p,q} \\ p \neq q}} c_{ij} (\mathbf{e}_i \mathbf{e}_j^T + \mathbf{e}_j \mathbf{e}_i^T) \right] \\
 &= \sum_{\substack{(i,j) \in E_{\text{split}} \cap E_{p,q} \\ p \neq q \\ i \in I_p}} c_{ij} \underbrace{\mathbf{e}_i^T \mathbf{e}_i}_{=1} \mathbf{e}_j^T \\
 &= \sum_{\substack{(i,j) \in E_{\text{split}} \cap E_{p,q} \\ p \neq q \\ i \in I_p}} c_{ij} \mathbf{e}_j^T
 \end{aligned}$$

thus

$$\|\mathbf{1}_k^T \Delta \mathbf{C}_{\text{split}}\|_\infty = \max_{i \in V_k} \left\{ \max_{\substack{(i, \cdot) \in E_{\text{split}} \cap E_{kq} \\ i \neq i_{kq}}} \left\{ \sum_{(i,j) \in E_{kq}} c_{ij} \right\} \right\}.$$

3.6 An example system $\mathbf{C}\mathbf{x}' + \mathbf{G}\mathbf{x} = \mathbf{b}$ with exact solution

In the previous section we found an expression for the solution of our DAE/ODE, using general system of ODE solutions

$$\mathbf{x} = e^{\mathbf{A}t} \left(\int_0^t e^{-\mathbf{A}s} \mathbf{b}(s) + \mathbf{x}(0) \right), \quad t > 0 \quad (3.6.1)$$

which enable us to establish error estimates.

This section will work out in detail, for a transmission line class of problems how the solution of (3.6.1) depends on the resistance and capacitance values. Except for giving more insight into the structure of \mathbf{x} of (3.6.1) this solution is also useful for the verification of the correct implementation of our circuit simulation.

Definition 3.6.1. Let $\mathbf{A} \in \mathbb{R}^{n \times m}$ and $\mathbf{B} \in \mathbb{R}^{p \times q}$ then

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1m}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2m}\mathbf{B} \\ \vdots & & \vdots & \\ a_{n1}\mathbf{B} & a_{n2}\mathbf{B} & \dots & a_{nm}\mathbf{B} \end{bmatrix}$$

is called Kronecker product of \mathbf{A} and \mathbf{B} .

Assume our transmission line problem is that in Fig. 3.1 with m lines with each n internal nodes, i.e., each with $n - 1$ internal resistors of $r > 0$ Ohm. Assume all capacitors are $c > 0$ Farad. Assume nodes are numbered left-right and bottom-top. This leads to $m \times m$ block matrices with $n \times n$ blocks:

$$\mathbf{G} = \begin{bmatrix} \mathbf{A}_n & & & \\ & \mathbf{A}_n & & \\ & & \ddots & \\ & & & \mathbf{A}_n \end{bmatrix} = \mathbf{I}_m \otimes \mathbf{A}_n \in \mathbb{R}^{mn \times mn}$$

and

$$\mathbf{C} = \begin{bmatrix} b_0 \mathbf{I}_n & b_1 \mathbf{I}_n & & \\ b_{-1} \mathbf{I}_n & b_0 \mathbf{I}_n & \ddots & \\ & \ddots & \ddots & b_1 \mathbf{I}_n \\ & & b_{-1} \mathbf{I}_n & b_0 \mathbf{I}_n \end{bmatrix} = \mathbf{B}_m \otimes \mathbf{I}_n \in \mathbb{R}^{mn \times mn}$$

with

$$\mathbf{A}_n = \begin{bmatrix} a_0 & a_1 & & \\ a_{-1} & a_0 & \ddots & \\ & \ddots & \ddots & a_1 \\ & & a_{-1} & a_0 \end{bmatrix}, \quad \mathbf{B}_m = \begin{bmatrix} b_0 & b_1 & & \\ b_{-1} & b_0 & \ddots & \\ & \ddots & \ddots & b_1 \\ & & b_{-1} & b_0 \end{bmatrix}$$

and

$$a_{-1} = -1, a_0 = 2, a_1 = -1; \quad b_{-1} = -1, b_0 = 2, b_1 = -1 \quad (3.6.2)$$

(for our application \mathbf{C} and \mathbf{G} are symmetric, i.e., $a_{-1} = a_1$ and $b_{-1} = b_1$) which are Toeplitz matrix with eigenvalues

$$\begin{aligned} \lambda_p(\mathbf{A}_n) &= \left\{ a_0 + \left(a_1 \sqrt{\frac{a_{-1}}{a_1}} + a_{-1} \left(\sqrt{\frac{a_{-1}}{a_1}} \right)^{-1} \right) \cos\left(\frac{p\pi}{n+1}\right) \right\}_{p=1}^n \\ &= \left\{ 2 - 2 \cos\left(\frac{p\pi}{n+1}\right) \right\}_{p=1}^n, \end{aligned} \quad (3.6.3)$$

$$\begin{aligned} \mu_q(\mathbf{B}_m) &= \left\{ b_0 + \left(b_1 \sqrt{\frac{b_{-1}}{b_1}} + b_{-1} \left(\sqrt{\frac{b_{-1}}{b_1}} \right)^{-1} \right) \cos\left(\frac{q\pi}{m+1}\right) \right\}_{q=1}^m \\ &= \left\{ 2 - 2 \cos\left(\frac{q\pi}{m+1}\right) \right\}_{q=1}^m. \end{aligned} \quad (3.6.4)$$

These are the eigenvectors of \mathbf{A}_n and \mathbf{B}_m :

$$\begin{aligned} \mathbf{s}_p &= \left(\left(\frac{a_{-1}}{a_1} \right)^{i/2} \sin\left(\frac{p\pi}{n+1}\right) \right)_{i=1}^n \in \mathbb{R}^n, & \mathbf{S}_n &= [\mathbf{s}_1, \dots, \mathbf{s}_n] \in \mathbb{R}^{n \times n} \\ \mathbf{t}_q &= \left(\left(\frac{b_{-1}}{b_1} \right)^{i/2} \sin\left(\frac{q\pi}{m+1}\right) \right)_{i=1}^m \in \mathbb{R}^m, & \mathbf{T}_m &= [\mathbf{t}_1, \dots, \mathbf{t}_m] \in \mathbb{R}^{m \times m}. \end{aligned}$$

With these results define:

$$\begin{aligned}
 \sigma(\mathbf{G}) &= \{\lambda_p(\mathbf{A}_n)\}_{p=1,\dots,n} \\
 \sigma(\mathbf{C}) &= \{\mu_q(\mathbf{B}_m)\}_{q=1,\dots,m} \\
 \mathbf{H} &= \text{diag}(\underbrace{\lambda_1, \dots, \lambda_n}_{\text{repeat } m \text{ times}}, \dots) = \mathbf{I}_m \otimes \lambda \in \mathbb{R}^{mn \times mn} \\
 \mathbf{D} &= \text{diag}(\underbrace{\mu_1, \dots, \mu_1}_{n \text{ times}}, \dots, \underbrace{\mu_m, \dots, \mu_m}_{n \text{ times}}) = \mu \otimes \mathbf{I}_n \in \mathbb{R}^{mn \times mn} \\
 \mathbf{V} &= \mathbf{I}_m \otimes \mathbf{S}_n \in \mathbb{R}^{mn \times mn} \\
 \mathbf{W} &= \mathbf{T}_m \otimes \mathbf{I}_n \in \mathbb{R}^{mn \times mn},
 \end{aligned}$$

where

$$\begin{aligned}
 \lambda &= \text{diag}(\lambda_1(\mathbf{A}_n), \dots, \lambda_n(\mathbf{A}_n)) \\
 \mu &= \text{diag}(\mu_1(\mathbf{B}_n), \dots, \mu_m(\mathbf{B}_m))
 \end{aligned}$$

Observations:

1. operator \mathbf{G} is the standard 3-point FDM discretization of $-\partial_x^2$
2. operator \mathbf{C} is the standard 3-point FDM discretization of $-\partial_y^2$
3. operator

$$\mathbf{F} := c\mathbf{C} + \mathbf{G}/r = \mathbf{G}/r + c\mathbf{C} = \mathbf{A}_n/r \oplus c\mathbf{B}_m$$

is the standard 5-point FDM discretization of

$$-\frac{1}{r}\partial_x^2 - c\partial_y^2,$$

a uniform tensor grid anisotropic diffusion operator.

Matrices \mathbf{C} and \mathbf{G} are both “block-diagonal”: Let p be the permutation which renumbers nodes first bottom-top and next left-right, i.e., in Matlab notation:

$$\mathbf{p} = ((0:(m-1))*n) + (0:(n-1)) + 1; \mathbf{p} = \mathbf{p}(:)$$

Example: Standard node numbers for $m = 3$ and $n = 4$

9	10	11	12
5	6	7	8
1	2	3	4

and permuted $p = [1 \ 5 \ 9 \ 2 \ 6 \ 10 \ 3 \ 7 \ 11 \ 4 \ 8 \ 12]$ node numbers:

3	6	9	12
2	5	8	11
1	4	7	10

Define permutation matrix

$$\mathbf{P} = [\mathbf{e}_{p(1)}, \dots, \mathbf{e}_{p(mn)}] \in \mathbb{R}^{mn \times mn}.$$

Then

$$\mathbf{P}^T \mathbf{C} \mathbf{P} = \begin{bmatrix} \mathbf{B}_m & & & \\ & \mathbf{B}_m & & \\ & & \ddots & \\ & & & \mathbf{B}_m \end{bmatrix} = \mathbf{I}_n \otimes \mathbf{B}_m \in \mathbb{R}^{mn \times mn}$$

is an $n \times n$ block diagonal matrix with $m \times m$ blocks.

Lemma 3.6.2. Assume that $\text{sign}(a_{-1}) = \text{sign}(a_1)$ and $\text{sign}(b_{-1}) = \text{sign}(b_1)$ (which holds due to (3.6.2)). Then eigendecompositions exist

1. $\mathbf{C} = \mathbf{W}\mathbf{D}\mathbf{W}^T$ and
2. $\mathbf{G} = \mathbf{V}\mathbf{H}\mathbf{V}^T$
3. $\mathbf{F} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$

with

- \mathbf{D}, \mathbf{H} and $\mathbf{\Lambda}$ diagonal and
- \mathbf{W}, \mathbf{V} and \mathbf{U} orthogonal ($\mathbf{W}^T\mathbf{W} = \mathbf{I}$, etc.)

Proof. \mathbf{C}, \mathbf{G} and \mathbf{F} are real-valued symmetric block-Toeplitz matrices with eigenvalues (3.6.3) and (3.6.4). □

Theorem 3.6.3. For all positive integers m and n

1. $\mathbf{W}^{-1}\mathbf{G}\mathbf{W} = \mathbf{G}$, $\mathbf{V}^{-1}\mathbf{C}\mathbf{V} = \mathbf{C}$
2. $\mathbf{D}\mathbf{V} = \mathbf{V}\mathbf{D}$, $\mathbf{H}\mathbf{W} = \mathbf{W}\mathbf{H}$
3. $\mathbf{W}\mathbf{V} = \mathbf{U}$ and \mathbf{U} is orthogonal
4. $\mathbf{C}\mathbf{G} = \mathbf{G}\mathbf{C}$, \mathbf{C} and \mathbf{G} commute (not used below)

and if in addition $n = m$ (not likely for a transmission line example) also:

5. $\mathbf{P}^T\mathbf{C}\mathbf{P} = \mathbf{G}$
6. $\mathbf{P}^T\mathbf{W}\mathbf{P} = \mathbf{V}$.

Proof. One can use the properties of the Kronecker product:

- $\mathbf{I}_m \otimes \mathbf{I}_n = \mathbf{I}_{mn}$ for all natural n, m
- $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{A}\mathbf{C}) \otimes (\mathbf{B}\mathbf{D})$ for all compatible $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and \mathbf{D}
- $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$.

Thus for instance:

$$\mathbf{W}\mathbf{V} = (\mathbf{T}_m \otimes \mathbf{I}_n)(\mathbf{I}_m \otimes \mathbf{S}_n) = (\mathbf{T}_m\mathbf{I}_m) \otimes (\mathbf{I}_n\mathbf{S}_n) = \mathbf{T}_m \otimes \mathbf{S}_n.$$

See [28] for many more properties of the Kronecker product. For instance, if $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ and $\mathbf{B}\mathbf{y} = \mu\mathbf{y}$ then

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{x} \otimes \mathbf{y}) = \lambda\mu \cdot (\mathbf{x} \otimes \mathbf{y})$$

which shows that the eigenpairs of \mathbf{F} are

$$\left(\frac{c}{r}\lambda_p(\mathbf{A}_n)\mu_q(\mathbf{B}_m), \mathbf{t}_q \otimes \mathbf{s}_p\right)$$

for all $p = 1, \dots, n$ and $q = 1, \dots, m$.

1. To see that $\mathbf{W}^{-1}\mathbf{G}\mathbf{W} = \mathbf{G}$ holds:

$$\begin{aligned}
 \mathbf{W}^{-1}\mathbf{G}\mathbf{W} &\stackrel{?}{=} \mathbf{G} \Leftrightarrow \\
 \mathbf{G}\mathbf{W} &\stackrel{?}{=} \mathbf{W}\mathbf{G} \Leftrightarrow \\
 \underbrace{\mathbf{V}\mathbf{H}\mathbf{V}^{-1}}_{\mathbf{G}}\mathbf{W} &\stackrel{?}{=} \mathbf{W}\underbrace{\mathbf{V}\mathbf{H}\mathbf{V}^{-1}}_{\mathbf{G}} \Leftrightarrow \\
 (\mathbf{I}_m \otimes \mathbf{S}_n) \underbrace{(\mathbf{I}_m \otimes \lambda)}_{\mathbf{H}} (\mathbf{I}_m \otimes \mathbf{S}_n)^{-1} (\mathbf{T}_m \otimes \mathbf{I}_n) &\stackrel{?}{=} (\mathbf{T}_m \otimes \mathbf{I}_n) (\mathbf{I}_m \otimes \mathbf{S}_n) \underbrace{(\mathbf{I}_m \otimes \lambda)}_{\mathbf{H}} (\mathbf{I}_m \otimes \mathbf{S}_n)^{-1} \Leftrightarrow \\
 (\mathbf{I}_m \mathbf{I}_m \mathbf{I}_m \mathbf{T}_m) \otimes (\mathbf{S}_n \lambda \mathbf{S}_n^{-1} \mathbf{I}_n) &\stackrel{?}{=} (\mathbf{T}_m \mathbf{I}_m \mathbf{I}_m \mathbf{I}_m) \otimes (\mathbf{I}_n \mathbf{S}_n \lambda \mathbf{S}_n^{-1}) \Leftrightarrow \\
 \mathbf{T}_m \otimes (\mathbf{S}_n \lambda \mathbf{S}_n^{-1}) &= \mathbf{T}_m \otimes (\mathbf{S}_n \lambda \mathbf{S}_n^{-1}). \tag{3.6.5}
 \end{aligned}$$

$\mathbf{V}^{-1}\mathbf{C}\mathbf{V} = \mathbf{C}$ can be similarly shown.

2. $\mathbf{D}\mathbf{V} = \mathbf{V}\mathbf{D}, \quad \mathbf{H}\mathbf{W} = \mathbf{W}\mathbf{H}$

$$\mathbf{D}\mathbf{V} = (\mu \otimes \mathbf{I}_n)(\mathbf{I}_m \otimes \mathbf{S}_n) = (\mu \mathbf{I}_m) \otimes (\mathbf{I}_n \mathbf{S}_n) = (\mathbf{I}_m \mu) \otimes (\mathbf{S}_n \mathbf{I}_n) = (\mathbf{I}_m \otimes \mathbf{S}_n)(\mu \otimes \mathbf{I}_n) = \mathbf{V}\mathbf{D}$$

$$\mathbf{H}\mathbf{W} = (\mathbf{I}_m \otimes \lambda)(\mathbf{T}_m \otimes \mathbf{I}_n) = (\mathbf{I}_m \mathbf{T}_m) \otimes (\lambda \mathbf{I}_n) = (\mathbf{T}_m \mathbf{I}_m) \otimes (\mathbf{I}_m \lambda) = (\mathbf{T}_m \otimes \mathbf{I}_n)(\mathbf{I}_m \otimes \lambda) = \mathbf{W}\mathbf{H}.$$

3. $\mathbf{W}\mathbf{V} = \mathbf{U}$

$$\mathbf{W}\mathbf{V} = (\mathbf{T}_m \otimes \mathbf{I}_n)(\mathbf{I}_m \otimes \mathbf{S}_n) = (\mathbf{T}_m \mathbf{I}_m) \otimes (\mathbf{I}_n \mathbf{S}_n) = \mathbf{T}_m \otimes \mathbf{S}_n = \mathbf{U}.$$

Also \mathbf{U} is orthogonal because

$$\mathbf{U}^T \mathbf{U} = (\mathbf{T}_m \otimes \mathbf{S}_n)^T (\mathbf{T}_m \otimes \mathbf{S}_n) = (\mathbf{T}_m^T \mathbf{T}_m) \otimes (\mathbf{S}_n^T \mathbf{S}_n) = \mathbf{I}_m \otimes \mathbf{I}_n = \mathbf{I}_{mn}$$

(because \mathbf{G} and \mathbf{C} are symmetric, their eigenvectors are orthogonal, i.e., $\mathbf{T}_m^T \mathbf{T}_m = \mathbf{I}_m$ and $\mathbf{S}_n^T \mathbf{S}_n = \mathbf{I}_n$)

4. $\mathbf{C}\mathbf{G} = \mathbf{G}\mathbf{C}$

$$\mathbf{C}\mathbf{G} = (\mathbf{B}_m \otimes \mathbf{I}_n)(\mathbf{I}_m \otimes \mathbf{A}_n) = \mathbf{B}_m \otimes \mathbf{A}_n$$

$$\mathbf{G}\mathbf{C} = (\mathbf{I}_m \otimes \mathbf{A}_n)(\mathbf{B}_m \otimes \mathbf{I}_n) = \mathbf{B}_m \otimes \mathbf{A}_n.$$

5. $\mathbf{P}^T \mathbf{C}\mathbf{P} \stackrel{?}{=} \mathbf{G}$ if $m = n$ with $\mathbf{G} = \mathbf{I}_m \otimes \mathbf{A}_n$

$$\mathbf{P}^T \mathbf{C}\mathbf{P} = \mathbf{P}^T (\mathbf{B}_m \otimes \mathbf{I}_n) \mathbf{P} = \mathbf{I}_n \otimes \mathbf{B}_m$$

when $m = n$,

$$\mathbf{P}^T \mathbf{C}\mathbf{P} = \mathbf{I}_n \otimes \mathbf{B}_n, \quad \mathbf{G} = \mathbf{I}_n \otimes \mathbf{A}_n$$

thus

$$\mathbf{P}^T \mathbf{C}\mathbf{P} = \mathbf{G} \text{ iff } \mathbf{A}_n = \mathbf{B}_n$$

which is true by initial assumption of coefficients $a_o = b_o = 2, a_1 = b_1 = -1$ and $a_{-1} = b_{-1} = -1$. Therefore, $\mathbf{P}^T \mathbf{C}\mathbf{P} = \mathbf{G}$ when $m = n$.

6. $\mathbf{P}^T \mathbf{W} \mathbf{P} \stackrel{?}{=} \mathbf{V}$ if $m = n$ with $\mathbf{V} = \mathbf{I}_m \otimes \mathbf{S}_n$

$$\mathbf{P}^T \mathbf{W} \mathbf{P} = \mathbf{P}^T (\mathbf{T}_m \otimes \mathbf{I}_n) \mathbf{P} = \mathbf{I}_n \otimes \mathbf{T}_m$$

when $m = n$,

$$\mathbf{P}^T \mathbf{W} \mathbf{P} = \mathbf{I}_n \otimes \mathbf{T}_n, \quad \mathbf{V} = \mathbf{I}_n \otimes \mathbf{S}_n$$

because $\mathbf{A}_n = \mathbf{B}_n$ therefore they have the same set of eigenvectors $\mathbf{T}_n = \mathbf{S}_n$.
Thus $\mathbf{P}^T \mathbf{W} \mathbf{P} = \mathbf{V}$.

□

Based on properties 1 – 4 in Theorem 3.6.3

$$\begin{aligned} \mathbf{C}^{-1} \mathbf{G} &= \mathbf{W} \mathbf{D}^{-1} \mathbf{W}^{-1} \mathbf{G} \\ &\stackrel{\mathbf{C} = \mathbf{W} \mathbf{D} \mathbf{W}^{-1}}{=} \mathbf{W} \mathbf{D}^{-1} \mathbf{W}^{-1} \mathbf{G} \mathbf{W} \mathbf{W}^{-1} \\ &= \mathbf{W} \mathbf{D}^{-1} \mathbf{G} \mathbf{W}^{-1} \\ &\stackrel{\mathbf{W}^{-1} \mathbf{G} \mathbf{W} = \mathbf{G}}{=} \mathbf{W} \mathbf{D}^{-1} \mathbf{V} \mathbf{H} \mathbf{V}^{-1} \mathbf{W}^{-1} \\ &\stackrel{\mathbf{G} = \mathbf{V} \mathbf{H} \mathbf{V}^{-1}}{=} \mathbf{W} \mathbf{D}^{-1} \mathbf{V} \mathbf{H} \mathbf{V}^{-1} \mathbf{W}^{-1} \\ &\stackrel{\mathbf{D} \mathbf{V} = \mathbf{V} \mathbf{D}}{\Longleftrightarrow} \mathbf{W} \mathbf{V} \mathbf{D}^{-1} \mathbf{H} \mathbf{V}^{-1} \mathbf{W}^{-1} \\ &\stackrel{\mathbf{W} \mathbf{V} = \mathbf{U}}{\Longleftrightarrow} \mathbf{U} \mathbf{D}^{-1} \mathbf{H} \mathbf{U}^{-1} \end{aligned}$$

whence $(c\mathbf{C})^{-1}(\mathbf{G}/r) = \mathbf{U}(1/(cr)\mathbf{D}^{-1}\mathbf{H})\mathbf{U}^{-1}$ and

$$\begin{aligned} c\mathbf{C}\mathbf{x}' + (1/r)\mathbf{G}\mathbf{x} &= \mathbf{b} \Longleftrightarrow \\ \mathbf{x}' + \mathbf{U}(1/(cr)\mathbf{D}^{-1}\mathbf{H})\mathbf{U}^{-1}\mathbf{x} &= c^{-1}\mathbf{C}^{-1}\mathbf{b}. \end{aligned}$$

For the homogeneous case one finds (set $\mathbf{b} = \mathbf{0}$)

$$\begin{aligned} \mathbf{x}' &= -\mathbf{U}(1/(cr)\mathbf{D}^{-1}\mathbf{H})\mathbf{U}^{-1}\mathbf{x} \\ &= \mathbf{U}(-1/(cr)\mathbf{D}^{-1}\mathbf{H})\mathbf{U}^{-1}\mathbf{x} \Longleftrightarrow \\ \mathbf{x}(t) &= \left[\exp\left(t \cdot \mathbf{U}(-1/(rc))\mathbf{D}^{-1}\mathbf{H})\mathbf{U}^{-1}\right) \right] \mathbf{x}(0) \\ &= \mathbf{U} \left[\exp(-(t/(rc))\mathbf{D}^{-1}\mathbf{H}) \right] \mathbf{U}^{-1} \mathbf{x}(0) \\ \mathbf{x}(t) &= \mathbf{U} \mathbf{E}(t) \mathbf{U}^{-1} \mathbf{x}(0) \end{aligned}$$

where diagonal matrix

$$\begin{aligned} \mathbf{E}(t) &= \exp(-(t/(rc))\mathbf{D}^{-1}\mathbf{H}) \\ &= \text{diag}(e^{-\frac{t}{rc} \cdot \frac{h_{ii}}{d_{ii}}}) \end{aligned}$$

for $i = 1, \dots, mn$ is stable when $\text{diag}(\mathbf{D}), \text{diag}(\mathbf{H}) > 0$ because $rc > 0$ by definition. This hold for our example because \mathbf{A}_n and \mathbf{B}_m are positive definite which implies eigenvalues of \mathbf{D} and \mathbf{H} are positive. Also observe that for a transmission line example m is small and bounded whence $\mu_q(\mathbf{B})$ stays well away from zero for all q .

Now use Duhamel's formula when the system is not homogenous:

$$\begin{aligned} \mathbf{x}'(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{f}(t) \Longleftrightarrow \\ \mathbf{x}(t) &= \exp(t\mathbf{A})\mathbf{x}(0) + \int_0^t \exp((t-s)\mathbf{A})\mathbf{f}(s) ds \\ &= \exp(t\mathbf{A}) \left(\mathbf{x}(0) + \int_0^t \exp(-s\mathbf{A})\mathbf{f}(s) ds \right) \end{aligned} \tag{3.6.6}$$

which in our case

$$\begin{aligned}
 \mathbf{x}'(t) &= \mathbf{A}\mathbf{x}(t) + c^{-1}\mathbf{C}^{-1}\mathbf{b}(t) \iff \\
 \mathbf{x}(t) &= \exp(t\mathbf{A})\mathbf{x}(0) + c^{-1} \int_0^t \exp((t-s)\mathbf{A})\mathbf{C}^{-1}\mathbf{b}(s) ds \\
 &= \mathbf{U}\mathbf{E}(t)\mathbf{U}^{-1}\mathbf{x}(0) + c^{-1}\mathbf{U} \int_0^t \mathbf{E}(t-s)\mathbf{U}^{-1}\mathbf{C}^{-1}\mathbf{b}(s) ds
 \end{aligned} \tag{3.6.7}$$

For the transmission line, we put a signal on node 1 only, i.e.,

$$\mathbf{b}(t) = \mathbf{e}_1 \cdot A \cdot (1 - \cos(2\pi \cdot f \cdot t))/2. \tag{3.6.8}$$

which is such that $\mathbf{b}(0) = \mathbf{0}$ and $\mathbf{b}'(0) = \mathbf{0}$, which seems compatible with $\mathbf{x}_0 = \mathbf{0}$.

Observe that a single non-zero source at node 1 is equivalent to a Dirichlet boundary condition for \mathbf{F} which is zero everywhere except in a single point (patch) which makes the transmission line problem dissimilar to standard convection-diffusion boundary value problems.

Further, observe that the choice of \mathbf{b} in (3.6.8) **does not** correspond to a physical realistic transmission line as this choice essentially grounds the (eliminated) neighbor nodes at the start and end of a all transmission lines.

Thus, for our case (assume for now $\mathbf{x}(0) = \mathbf{0}$)

$$\begin{aligned}
 \mathbf{x}(t) &= \exp(t\mathbf{A})\mathbf{x}(0) + c^{-1} \int_0^t \exp((t-s)\mathbf{A})\mathbf{C}^{-1}\mathbf{b}(s) ds \\
 &= c^{-1}\mathbf{U} \int_0^t \mathbf{E}(t-s)\mathbf{U}^{-1}\mathbf{C}^{-1}\mathbf{b}(s) ds
 \end{aligned}$$

One has

$$\begin{aligned}
 \mathbf{C}\mathbf{U} &= \underbrace{\mathbf{W}\mathbf{D}\mathbf{W}^{-1}}_{\mathbf{C}} \underbrace{\mathbf{W}\mathbf{V}}_{\mathbf{U}} = \mathbf{W}\mathbf{D}\mathbf{V} = \mathbf{W}\mathbf{V}\mathbf{D} = \mathbf{U}\mathbf{D} \\
 (\mathbf{C}\mathbf{U})^{-1} &= \mathbf{U}^{-1}\mathbf{C}^{-1} = (\mathbf{U}\mathbf{D})^{-1} = \mathbf{D}^{-1}\mathbf{U}^{-1}
 \end{aligned}$$

Thus (3.6.7) becomes

$$\begin{aligned}
 \mathbf{x}(t) &= c^{-1}\mathbf{U} \int_0^t \mathbf{E}(t-s)\mathbf{D}^{-1}\mathbf{U}^{-1}\mathbf{b}(s) ds \\
 &= c^{-1}\mathbf{U} \int_0^t \mathbf{E}_\mathbf{D}(t-s)\mathbf{U}^{-1}\mathbf{b}(s) ds
 \end{aligned}$$

where

$$\begin{aligned}
 \mathbf{E}_\mathbf{D}(t-s) &= \exp(-((t-s)/(rc))\mathbf{D}^{-1}\mathbf{H}) \cdot \mathbf{D}^{-1} \\
 &= \text{diag}\left(\frac{1}{d_{ii}} \cdot e^{-\frac{(t-s)}{rc} \cdot \frac{h_{ii}}{d_{ii}}}\right)
 \end{aligned}$$

Let $\mathbf{U} = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \dots \quad \mathbf{u}_{mn}]$, i.e., $\mathbf{u}_i, i = 1, \dots, mn$ columns of \mathbf{U} orthogonal. Compute

$$\mathbf{U}\mathbf{E}_\mathbf{D}(t-s) = \left[\mathbf{u}_1 \cdot e^{-\frac{(t-s)}{rc} \cdot \frac{h_{11}}{d_{11}}} \cdot d_{11}^{-1} \quad \dots \quad \mathbf{u}_{mn} \cdot e^{-\frac{(t-s)}{rc} \cdot \frac{h_{mn,mn}}{d_{mn,mn}}} \cdot d_{mn,mn}^{-1} \right]$$

$$\begin{aligned}
 \mathbf{U}^{-1}\mathbf{b}(s) &= [\mathbf{u}_1^T \quad \dots \quad \mathbf{u}_{mn}^T] \begin{bmatrix} b_1(s) \\ \vdots \\ b_{mn}(s) \end{bmatrix} \\
 &= [\mathbf{u}_1^T b_1(s) + \dots + \mathbf{u}_{mn}^T b_{mn}(s)] , \\
 &= \begin{bmatrix} c_1(s) \\ \vdots \\ c_{mn}(s) \end{bmatrix}
 \end{aligned}$$

which shows that

$$\begin{aligned}
 \mathbf{U}\mathbf{E}_D(t-s)\mathbf{U}^{-1}\mathbf{b}(s) &= \left[\mathbf{u}_1 \cdot e^{-\frac{(t-s)}{rc} \cdot \frac{h_{11}}{d_{11}}} \cdot d_{11}^{-1} \cdot c_1(s) + \dots + \right. \\
 &\quad \left. \mathbf{u}_{mn} \cdot e^{-\frac{(t-s)}{rc} \cdot \frac{h_{mn,mn}}{d_{mn,mn}}} \cdot d_{mn,mn}^{-1} \cdot c_{mn}(s) \right]
 \end{aligned}$$

and

$$\begin{aligned}
 \mathbf{x}(t) &= c^{-1} \left[\mathbf{u}_1 d_{11}^{-1} \int_0^t e^{-\frac{(t-s)}{rc} \cdot \frac{h_{11}}{d_{11}}} \cdot c_1(s) ds + \dots + \right. \\
 &\quad \left. \mathbf{u}_{mn} d_{mn,mn}^{-1} \int_0^t e^{-\frac{(t-s)}{rc} \cdot \frac{h_{mn,mn}}{d_{mn,mn}}} \cdot c_{mn}(s) ds \right]
 \end{aligned}$$

Now for some non-zero sinusoidal inputs $b_i(s), i = 1, \dots, mn$ (most inputs are zero) one can compute $\mathbf{U}^{-1}\mathbf{b}(s) = \mathbf{U}^T\mathbf{b}(s)$ using expressions

$$\begin{aligned}
 \int e^{cx} \sin(bx) dx &= \frac{e^{cx}}{c^2 + b^2} [c \cdot \sin(bx) - b \cdot \cos(bx)] \\
 \int e^{cx} \cos(bx) dx &= \frac{e^{cx}}{c^2 + b^2} [c \cdot \cos(bx) + b \cdot \sin(bx)] ,
 \end{aligned}$$

and use the result to validate the circuit simulation.

3.7 Conclusion

The estimates for the current differences in Sections 3.2 – 3.5 are used in Section 4.3 on the error analysis for MoveC and SplitC. We underline the net current differences between the original problem and MoveC problem, and between the original problem and SplitC problem:

$$\begin{aligned}
 I_p^{\text{orig}} - I_p^{\text{move}} &= -\mathbf{1}_p^T \mathbf{C}_0(\mathbf{v}' - \hat{\mathbf{v}}') - \mathbf{1}_p^T \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}}) , \\
 I_p^{\text{orig}} - I_p^{\text{split}} &= -\mathbf{1}_p^T \mathbf{C}_0(\mathbf{v}' - \hat{\mathbf{v}}') - \mathbf{1}_p^T \mathbf{G}_0(\mathbf{v} - \hat{\mathbf{v}}) - \mathbf{1}_p^T \mathbf{C}_s \hat{\mathbf{v}}' .
 \end{aligned}$$

and that in the worst case $|I_p^{\text{orig}} - I_p^{\text{split}}|/|I_p^{\text{orig}}|$ can provide 100% relative error because of losing many coupling capacitors. In addition, we estimate the difference $\mathbf{v} - \hat{\mathbf{v}}$ for general circuit for general circuit with non-singular capacitance matrix in Section 3.4. Addition estimates of $\|\mathbf{C}\|_\infty$, $\|\Delta\mathbf{C}\|_\infty$, $\|\mathbf{1}_p^T \mathbf{C}\|_\infty$ and $\|\mathbf{1}_p^T \Delta\mathbf{C}\|_\infty$ are presented in Section 3.5 for related capacitance matrices of the three problems. Finally, Section 3.6 shows the exact solution for a transmission line class of problems, i.e., with m lines and n internal nodes for each lines.

Chapter 4

SplitC and MoveC: splitting and moving coupling capacitors

4.1 Introduction

In this chapter, we introduce MoveC method to deal with parasitic/coupling capacitors of RC networks. MoveC can be used for any circuit containing RCc parasitics to further reduce the number of parasitic capacitors after any netlist reduction. The approach improves the sparsity of the capacitance matrix by moving many small coupling capacitors to specific coupling capacitors. MoveC method reduces the number of capacitors of RC networks but maintains the same network size. Although MoveC reduces the amount of capacitors, it does preserve the total network capacitance and the net-to-net coupling capacitance, which is essential for preserving accuracy. Numerical results demonstrate that MoveC leads to sparse systems, faster transient simulation and accurate results.

The chapter is organized as follows. In Section 4.2, we introduce MoveC method. It involves a specific (**G**-net) node ordering and algorithm. In Section 4.3, theoretical proofs and electrical reasoning for MoveC error is given within a given threshold. Numerical experiments of realistic netlists are shown in Section 4.4. In Section 4.5 we draw conclusions.

4.2 The MoveC method

We will first introduce some basic concepts. Let $G = (V, E)$ be an undirected graph, $V = \{v_1, v_2, \dots, v_n\}$ be the set of vertices or nodes and $E \in \{(v_i, v_j) : v_i, v_j \in V\}$ be the set of edges. A graph is called connected if there is a path between every pair of nodes in the graph. A connected component [8, 39] of a graph G , by definition of [37], “is a maximal connected subgraph of G . A graph G that is not connected has two or more connected components that are disjoint and have G as their union”.

For MoveC, it is essential to identify the coupling capacitors. The connected components of G can, electrically speaking, be interpreted as nets. Then $C(i, j)$ is a coupling capacitor if $\text{net}(i) \neq \text{net}(j)$, i.e., the net containing node i is not identical to the net containing node j . For the sake of simplicity we assume that \mathbf{G} is block partitioned in $P \times P$ blocks where the nodes related to each block (component/net) $1 \leq p \leq P$ are mutually strongly connected. For $\mathbf{A} \in \mathbb{R}^{N \times N}$, we define $S(\mathbf{A}) = \{(i, j) : a_{ij} \neq 0\}$ (we will later describe how to achieve this situation). Let $\mathbf{C}_{p|q}$ is matrix with \mathbf{C}_{pq} non-zero block and zero elements elsewhere. For instance, if \mathbf{C} is partitioned into 2×2 blocks, we have

$$\mathbf{C} = \left[\begin{array}{c|c} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \hline \mathbf{C}_{21} & \mathbf{C}_{22} \end{array} \right], \quad \mathbf{C}_{1|2} = \left[\begin{array}{c|c} \mathbf{0} & \mathbf{C}_{12} \\ \hline \mathbf{0} & \mathbf{0} \end{array} \right].$$

Observe that $\mathbf{C}_{p|q}$ is not a block-part of \mathbf{C} (which would be smaller in size). However, block \mathbf{C}_{pq} has the same amount (and values of) non-zeros as $\mathbf{C}_{p|q}$.

The algorithm steps are as follows:

In phase 1 - Determine coupling capacitors:

Lines 1 and 2 Usually, the *gnd* node is known, and if not, it can be identified by the node which has the largest number of non-zero elements in both \mathcal{G} and \mathcal{C} . Power node(s) *pows* is/are node(s) whose number of neighbors in the matrices \mathcal{G} and \mathcal{C} is much larger than others (but not the *gnd*). After determining the *gnd* and *pows*, we delete the corresponding rows and columns.

Line 3 Reorder the conductance matrix \mathcal{G} by connected components. The reordered conductance matrix $\mathbf{G} = \mathcal{G}(\mathbf{P}, \mathbf{P})$ will be a block-diagonal matrix.

Line 4 Reorder the capacitance matrix \mathcal{C} with the same permutation \mathbf{P} into $\mathbf{C} = \mathcal{C}(\mathbf{P}, \mathbf{P})$. The reordered capacitance matrix \mathbf{C} contains diagonal and off-diagonal blocks. Off-diagonal-blocks are related to coupling capacitances. For example, off-diagonal-block \mathbf{C}_{12} of matrix \mathbf{C} contains the coupling capacitors between nets 1 and 2.

In phase 2 - Determine the move nodes:

Line 7 In case *argmax* is not unique, we pick the first one encountered. Note that "the first one encountered" likely depends on the order of the node numbers (still free to choose within each \mathbf{G} -net).

In phase 3 - move elements:

Lines 12-14 Capacitors $\mathbf{C}(i, j)$ which satisfy the moving condition $c(i, j) \leq \epsilon$ will be moved between nodes $k_{(p,q)}$ and $l_{(p,q)}$.

In phase 4 - Reordering:

Lines 19 and 20 Finally, $\mathbf{G}, \hat{\mathbf{C}}$ are reordered by \mathbf{P}^{-1} , i.e., compute $\mathbf{G}(\mathbf{P}^{-1}, \mathbf{P}^{-1})$ and $\hat{\mathbf{C}}(\mathbf{P}^{-1}, \mathbf{P}^{-1})$ (having the original order). Finally, *gnd* and *pows* are inserted again.

Algorithm 2 MoveC algorithm

Input: \mathcal{G}, \mathcal{C} , $0 < \epsilon \ll 1$, terminal nodes and nodemap

Output: Simplified $\hat{\mathcal{C}}$ and $\hat{\mathbf{C}}$

Phase 1 - Determine coupling capacitors

- 1: Find *gnd* and *power node(s)*
 - 2: Mark and temporarily remove *gnd* and *power node(s)* from \mathcal{G}, \mathcal{C}
 - 3: Permute \mathcal{G} by connected components of \mathcal{G} ; $\mathbf{G} \leftarrow \mathcal{G}(\mathbf{P}, \mathbf{P})$
 - 4: Permute \mathcal{C} by the same connected-component permutation of \mathcal{G} ; $\mathbf{C} \leftarrow \mathcal{C}(\mathbf{P}, \mathbf{P})$
Coupling capacitors are on off-diagonal blocks of \mathbf{C}
-

Phase 2 - Determine move-nodes

- 5: **for** $p = 1 : P$ **do** $\triangleright P$ is the number of components
 - 6: **for** $q = 1 : P$ **do**
 - 7: $(k_{(p,q)}, l_{(p,q)}) := \arg \max_{(i,j) \in S(\mathbf{C}_{p|q})} \{c(i,j)\}$ $\triangleright k, l \in \{1, \dots, N\}, 1 \leq p, q \leq P$.
 - 8: **end for**
 - 9: **end for**
-

Phase 3 - Move elements

- 10: **for** $i = 1 : N$ **do**
 - 11: **for** $j = 1 : N$ **do**
 - 12: **if** $(i, j) \in S(\mathbf{C}_{p|q})$ & $c(i, j) \leq \epsilon$ **then**
 - 13: $\mathbf{C} - = c_{ij} \mathbf{d}_{ij} \mathbf{d}_{ij}^T$ (remove)
 - 14: $\mathbf{C} + = c_{ij} \mathbf{d}_{k(p,q)l(p,q)} \mathbf{d}_{k(p,q)l(p,q)}^T$ (add)
 - 15: **end if**
 - 16: **end for**
 - 17: **end for**
 - 18: $\mathbf{C}_{\text{move}} := \hat{\mathbf{C}} := \mathbf{C}$.
-

Phase 4 - Reordering

- 19: Reorder $\mathbf{G}, \hat{\mathbf{C}}$ by the inverse permutation of \mathcal{G} to get simplified $\hat{\mathbf{C}} \leftarrow \hat{\mathbf{C}}(\mathbf{P}^{-1}, \mathbf{P}^{-1})$
and $\mathbf{G} \leftarrow \mathbf{G}(\mathbf{P}^{-1}, \mathbf{P}^{-1})$
 - 20: Put back the *gnd* and *power node(s)* in $\hat{\mathbf{C}}$ and \mathbf{G} to get $\hat{\mathcal{C}}$ and \mathcal{G}
-

Remark: We do not move coupling capacitors connected to *gnd* or *pows*. Moving the small capacitors makes $\hat{\mathbf{C}}$ sparser than \mathbf{C} . Instead of the maximum coupling capacitance criterion in step 7 we can employ a **time constant [40] strategy**: For every two diagonal blocks, we find the maximum time constant of each block, for instance, the maximum time constant of net p is $\tau_p = \max_{\text{net}(i)=p} \left\{ \frac{\text{diag}(\mathbf{C}(i,i))}{\text{diag}(\mathbf{G}(i,i))} \right\}$ and the maximum time constant of net q is $\tau_q = \max_{\text{net}(j)=q} \left\{ \frac{\text{diag}(\mathbf{C}(j,j))}{\text{diag}(\mathbf{G}(j,j))} \right\}$. Then with $k = \arg \max_{\text{net}(i)=p} \left\{ \frac{\text{diag}(\mathbf{C}(i,i))}{\text{diag}(\mathbf{G}(i,i))} \right\}$ and $l = \arg \max_{\text{net}(j)=q} \left\{ \frac{\text{diag}(\mathbf{C}(j,j))}{\text{diag}(\mathbf{G}(j,j))} \right\}$ the capacitor to move to is $\mathbf{C}(k, l)$. This is motivated heuristically by the idea that moving coupling capacitors to between two nodes (with high RC time constant) of two nets will lead to small solution errors. To avoid that we add additional non-zero elements into the system matrix \mathbf{C} , we only consider already existing capacitors between nets as candidates. However, this criterion is likely to move small capacitors to edges (i, j) which previously had no related capacitors, i.e., they could increase the density of \mathbf{C} . The criterion shown in step 7 of Algorithm 2 is **maximum coupling capacitance**: Capacitor to move to is the maximum coupling capacitance per each coupling net (or off-diagonal block in \mathbf{C}). With this strategy we do not create additional capacitors. Besides, adding small capacitances to the maximum capacitance of off-diagonal-block seems reasonable since we do not significantly change the netlist.

For the sake of an example, focus on the circuit in Fig. 3.1. The permutation \mathbf{P} to reorder the nodes into \mathbf{G} -nets turn out to be the identity, so we focus on the system $\mathbf{C}\mathbf{x}'(t) + \mathbf{G}\mathbf{x}(t) = \mathbf{B}\mathbf{s}(t)$ with \mathbf{G} , \mathbf{C} and \mathbf{B} as in equations (3.2.2), (3.2.3) and (3.2.4). Now, suppose that c_{25} is the selected coupling capacitor between net 1 and net 2 to move to. If c_{36} satisfies the condition for moving it, we move c_{36} to c_{25} between node 2 and 5, to get $\sum c = c_{25} + c_{36}$ in (4.2.1). Simultaneously, diagonal elements will be changed respectively to the changes in the off-diagonal-block elements.

Note that MoveC does not make any change to the \mathbf{G} matrix. \mathbf{G} is only used to determine the connected components (or nets) which are used to determine the coupling capacitances in \mathbf{C} . The MoveC system related to the original system (3.2.1) based on Algorithm 2 is:

$$\hat{\mathbf{C}}\hat{\mathbf{x}}'(t) + \mathbf{G}\hat{\mathbf{x}}(t) = \mathbf{B}\mathbf{s}(t)$$

where $\hat{\mathbf{x}} = \mathbf{x} + \delta\mathbf{x}$, and the MoveC matrix $\hat{\mathbf{C}} = \mathbf{C}_m = \mathbf{C} - \Delta\mathbf{C}$ is given by:

$$\hat{\mathbf{C}} = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_2 + \sum c & 0 & 0 & -\sum c & 0 \\ 0 & 0 & c_3 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\sum c & 0 & 0 & c_5 + \sum c & 0 \\ 0 & 0 & 0 & 0 & 0 & c_6 \end{array} \right] = \left[\begin{array}{c|c} \hat{\mathbf{C}}_{11} & \hat{\mathbf{C}}_{12} \\ \hline \hat{\mathbf{C}}_{21} & \hat{\mathbf{C}}_{22} \end{array} \right] \quad (4.2.1)$$

with $\sum c = c_{25} + c_{36}$ and

$$\Delta \mathbf{C} = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -c_{36} & 0 & 0 & c_{36} & 0 \\ 0 & 0 & c_{36} & 0 & 0 & -c_{36} \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_{36} & 0 & 0 & -c_{36} & 0 \\ 0 & 0 & -c_{36} & 0 & 0 & c_{36} \end{array} \right]. \quad (4.2.2)$$

Observe that $\Delta \mathbf{C} \in \mathbb{R}^{6 \times 6}$ is symmetric, positive semi-definite. Note that the entries in $\Delta \mathbf{C}$ are a function of threshold and MoveC strategy.

4.3 Error analysis for global and inter-net currents

We compute the current between two nets for MoveC and SplitC to see which one gives the larger error compared to the original. The inter-net current for MoveC has already been computed in Section 3.2, so here we focus on the inter-net current for SplitC. Consider the two transmission line example (Fig. 3.1) which contains two nets connected to each other via coupling capacitors. For our example we split the capacitors related to edge (3, 6) and edge (2, 5) which results in capacitors to ground. We call the related capacitance matrix the SplitC matrix $\hat{\mathbf{C}}$,

$$\hat{\mathbf{C}} = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_2 + c_{25} & 0 & 0 & 0 & 0 \\ 0 & 0 & c_3 + c_{36} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_5 + c_{25} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_6 + c_{36} \end{array} \right].$$

Then the SplitC related Kirchhoff's laws induced equations are

$$\hat{\mathbf{C}} \hat{\mathbf{v}}' + \mathbf{G} \hat{\mathbf{v}} = \mathbf{I}_{source}$$

and similar to (3.2.13) we find

$$\begin{aligned} \mathbf{1}_1^T \hat{\mathbf{C}} \frac{d\hat{\mathbf{v}}}{dt} &= \mathbf{1}_1^T (\mathbf{I}_{source} - \mathbf{G} \hat{\mathbf{v}}) \Leftrightarrow \\ \mathbf{1}_1^T \hat{\mathbf{C}} \frac{d\hat{\mathbf{v}}}{dt} &= \mathbf{1}_1^T \left(\begin{bmatrix} s_1 \\ 0 \\ 0 \\ s_2 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} \hat{v}_{12} \\ -\hat{v}_{12} + \hat{v}_{23} \\ -\hat{v}_{23} + \hat{v}_3 \\ \hat{v}_{45} \\ -\hat{v}_{45} + \hat{v}_{56} \\ -\hat{v}_{56} + \hat{v}_6 \end{bmatrix} \right) \Leftrightarrow \quad (4.3.1) \\ (c_2 + c_{25}) \frac{d\hat{v}_2}{dt} + (c_3 + c_{36}) \frac{d\hat{v}_3}{dt} &= s_1 - \hat{v}_3. \end{aligned}$$

Let I_p be the current over all edges which have exactly one vertex in V_p . The (3.2.13) and (4.3.1) show that

$$\left\{ \underbrace{i_{25} + i_{36}}_{I_1^{\text{orig}}} + c_2 \frac{dv_2}{dt} + c_3 \frac{dv_3}{dt} = s_1 - i_3, \right. \quad (4.3.2a)$$

$$\left\{ \underbrace{0}_{I_1^{\text{split}}} + (c_2 + c_{25}) \frac{d\hat{v}_2}{dt} + (c_3 + c_{36}) \frac{d\hat{v}_3}{dt} = s_1 - \hat{i}_3 \right. \quad (4.3.2b)$$

$$\Leftrightarrow \left\{ I_1^{\text{orig}} + \mathbf{1}_1^T \mathbf{C}_0 \frac{d\mathbf{v}}{dt} = s_1 - \mathbf{1}_1^T \mathbf{G}_0 \mathbf{v}, \right. \quad (4.3.3a)$$

$$\left\{ I_1^{\text{split}} + \mathbf{1}_1^T (\mathbf{C}_0 + \mathbf{C}_s) \frac{d\hat{\mathbf{v}}}{dt} = s_1 - \mathbf{1}_1^T \mathbf{G}_0 \hat{\mathbf{v}}, \right. \quad (4.3.3b)$$

remind that \mathbf{C}_s is diagonal matrix created by splitting coupling capacitors to ground.

$$(4.3.3a) - (4.3.3b) \Rightarrow I_1^{\text{orig}} - I_1^{\text{split}} + \mathbf{1}_1^T \mathbf{C}_0 \frac{d\mathbf{v} - d\hat{\mathbf{v}}}{dt} - \mathbf{1}_1^T \mathbf{C}_s \frac{d\hat{\mathbf{v}}}{dt} = -\mathbf{1}_1^T \mathbf{G}_0 (\mathbf{v} - \hat{\mathbf{v}}).$$

Note that (4.3.2a) and (4.3.2b) can be rewritten

$$\begin{cases} \mathbf{1}_1^T \hat{\mathbf{C}} \frac{d\hat{\mathbf{v}}}{dt} = s_1 - \mathbf{1}_1^T \mathbf{G}_0 \hat{\mathbf{v}}, \\ \mathbf{1}_1^T \mathbf{C} \frac{d\mathbf{v}}{dt} = s_1 - \mathbf{1}_1^T \mathbf{G}_0 \mathbf{v}. \end{cases}$$

To get an estimate for the relative error, we also have to estimate I_1^{orig} , which is given in (3.3.2). Thus we find for net q :

$$\frac{|I_q^{\text{orig}} - I_q^{\text{split}}|}{|I_q^{\text{orig}}|} = \frac{\left| -\mathbf{1}_q^T \mathbf{C}_0 \frac{d(\mathbf{v} - \hat{\mathbf{v}})}{dt} + \mathbf{1}_1^T \mathbf{C}_s \frac{d\hat{\mathbf{v}}}{dt} - \mathbf{1}_1^T \mathbf{G}_0 (\mathbf{v} - \hat{\mathbf{v}}) \right|}{\left| S_q - \mathbf{1}_q^T \mathbf{C}_0 \frac{d\mathbf{v}}{dt} - \mathbf{1}_q^T \mathbf{G}_0 \mathbf{v} \right|}.$$

just as for the MoveC in (3.3.1), but now $\hat{\mathbf{v}}$ is related to SplitC, so the error will be different. For MoveC, the net currents are preserved since the coupling capacitance doesn't change after MoveC.

4.3.1 Global error in term of net current of MoveC/SplitC

The original problem after permutation based on \mathbf{G} -nets leads to a block-matrix problem, here, for the sake of simplicity show for our example in Fig. 3.1:

$$\begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}'_1 \\ \mathbf{x}'_2 \end{bmatrix} + \begin{bmatrix} \mathbf{G}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{s}_1(t) \\ \mathbf{s}_2(t) \end{bmatrix} \Leftrightarrow \begin{cases} \mathbf{C}_{11} \mathbf{x}'_1 = \mathbf{s}_1(t) - \mathbf{G}_{11} \mathbf{x}_1(t) - \mathbf{C}_{12} \mathbf{x}'_2 \\ \mathbf{C}_{22} \mathbf{x}'_2 = \mathbf{s}_2(t) - \mathbf{G}_{22} \mathbf{x}_2(t) - \mathbf{C}_{21} \mathbf{x}'_1. \end{cases}$$

SplitC, with the assumption that we remove all coupling capacitors, gives

$$\begin{bmatrix} \tilde{\mathbf{C}}_{11} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{C}}_{22} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}'_1 \\ \tilde{\mathbf{x}}'_2 \end{bmatrix} + \begin{bmatrix} \mathbf{G}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}'_1 \\ \tilde{\mathbf{x}}'_2 \end{bmatrix} = \begin{bmatrix} \mathbf{s}_1(t) \\ \mathbf{s}_2(t) \end{bmatrix} \Leftrightarrow$$

$$\begin{cases} \tilde{\mathbf{C}}_{11}\tilde{\mathbf{x}}'_1 = \mathbf{s}_1(t) - \mathbf{G}_{11}\tilde{\mathbf{x}}_1(t) \\ \tilde{\mathbf{C}}_{22}\tilde{\mathbf{x}}'_2 = \mathbf{s}_2(t) - \mathbf{G}_{22}\tilde{\mathbf{x}}_2(t). \end{cases} \quad (4.3.5)$$

SplitC decouples coupling capacitors, therefore, in (4.3.5) we completely lose the currents between the nets. Mathematically speaking, the system of equations (4.3.5) is a decoupled problem since \mathbf{x}_1 does not interact with \mathbf{x}_2 ; \mathbf{x}_1 and \mathbf{x}_2 can be solved for independently. In practice, for large \mathbf{G} and \mathbf{C} , the loss of the inter-net currents may result in large errors.

Thus SplitC may cause errors (up to 100%) when the source in one net get decoupled from sources in a (previously) connected net (when \mathbf{s}_1 or \mathbf{s}_2 is close (or equal) to zero). MoveC, assuming that small coupling capacitors are moved into one coupling capacitor, gives

$$\begin{bmatrix} \hat{\mathbf{C}}_{11} & \hat{\mathbf{c}}_{12} \\ \hat{\mathbf{c}}_{21} & \hat{\mathbf{C}}_{22} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}'_1 \\ \hat{\mathbf{x}}'_2 \end{bmatrix} + \begin{bmatrix} \mathbf{G}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{22} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{s}_1(t) \\ \mathbf{s}_2(t) \end{bmatrix} \Leftrightarrow$$

$$\begin{cases} \hat{\mathbf{C}}_{11}\hat{\mathbf{x}}'_1 = \mathbf{s}_1(t) - \mathbf{G}_{11}\hat{\mathbf{x}}_1(t) - \hat{\mathbf{c}}_{12}\hat{\mathbf{x}}'_2, \\ \hat{\mathbf{C}}_{22}\hat{\mathbf{x}}'_2 = \mathbf{s}_2(t) - \mathbf{G}_{22}\hat{\mathbf{x}}_2(t) - \hat{\mathbf{c}}_{21}\hat{\mathbf{x}}'_1, \end{cases}$$

where $\hat{\mathbf{c}}_{12}$ is a matrix with only one non-zero entry.

Potential error estimates and Elmore delay

For the circuit in Fig. 3.1 with $\Delta\mathbf{C}$ in (4.2.2), using (3.4.10) we find that the error decreases to zero for $c_{36} \rightarrow 0$ (in finite time). For the sake of illustration, consider (3.5.1) applied to our example in Fig. 3.1.

Definition 4.3.1. Let \mathbf{e}_K and $\mathbf{e}_M \in \mathbb{R}^N$ be the K -th and M -th unit vectors, respectively, and define

$$g_M = \mathbf{G}(:, M) = \begin{bmatrix} g(1, M) \\ \vdots \\ g(N_{MG}, M) \end{bmatrix}, \quad c_M = \mathbf{C}(:, M) = \begin{bmatrix} c(1, M) \\ \vdots \\ c(N_{MC}, M) \end{bmatrix};$$

where $c(., M)$ and $g(., M)$ are the capacities respectively conductivities connected to node M . Define:

$$G_M = \|\mathbf{g}_M\| = \sum_{i=1}^{MG} g(i, M),$$

$$C_M = \|\mathbf{c}_M\| = \sum_{i=1}^{MC} c(i, M),$$

$$R_{MK} = (\mathbf{e}_M - \mathbf{e}_K)^T \mathbf{G}^{-1} (\mathbf{e}_M - \mathbf{e}_K)$$

where MC is the amount of capacities connected to node M and MG is the amount of conductivities connected to node M . R_{MK} is the path resistance between nodes M and K [36].

An alternative strategy for MoveC, suppose e_r and τ be the bounds of the voltage error and delay error (Elmore delay [38]), respectively. We move a coupling capacitor $c(i, M)$ from node M to node K if

- $\frac{c_{iM}}{hG_M + C_M} < \frac{e_r}{2}$
- $\frac{c_{iM}}{hG_K + C_K} < \frac{e_r}{2}$
- $R_{MK}C_K < \tau$

The result of this strategy will be discussed later on.

General local error formula

Let C_M and G_M be defined as in Definition 4.3.1. Suppose the local error at node M is bounded by e_r , i.e., $\left| \frac{x_M - \hat{x}_M}{x_M} \right| < e_r$, we can remove $c(i, M)$ connected to node M if

$$\frac{c(i, M)}{hG_M + C_M} < \frac{e_r}{2}$$

Similarly, with $\left| \frac{x_K - \hat{x}_K}{x_K} \right| < e_r$, we can add c_{iM} to node K if

$$\frac{c(i, M)}{hG_K + C_K} < \frac{e_r}{2}.$$

For SplitC, with the error bound e_r , we can split $c(i, M)$ if

$$\frac{c(i, M)}{hG_M + C_M} < e_r.$$

Therefore the condition for removing a capacitor is sharper than that of splitting a capacitor

$$\frac{c(i, M)}{hG_M + C_M} < \frac{e_r}{2} < e_r.$$

In order words, there will be more coupling capacitors removed by the condition of SplitC than the condition of MoveC. Locally, removing a coupling capacitor may cause larger error than when we split a coupling capacitor to the ground. Nevertheless, practical experiments show that the result of MoveC with removing and adding step is more accurate than SplitC. In the next section we show why MoveC is more accurate than SplitC.

An electrical explanation for MoveC

The electrical reason why MoveC does not introduce large error is that in a resistive net, all node-voltages usually change at the same time and with the same rate (except for nodes with high RC time constant). In addition, we have $i = C \frac{dv}{dt}$ and with the realistic assumption that $\frac{dv}{dt}$ is almost the same for all pairs of nodes, we would expect that the total amount of current via capacitive coupling remains the same, no matter

where we put the coupling capacitors in a net as long as we put them somewhere between the nets. On the other hand, if we split, the error can rapidly increase because we start to change the current flowing between the nets, by decreasing the total amount of coupling capacitors.

The global error analysis below will show how errors accumulate during MoveC and SplitC, and proof the reasoning above.

4.3.2 The delay error

In the previous sections we have made the assumption of worst case on voltage error. In this section we will show what we have from this assumption and what it means in electrical reasoning. In the worst case, we assume:

1. The Elmore delay from v_a to v_b is small $R_{ab}C_b < \tau$, $R_{ab} = (\mathbf{e}_a - \mathbf{e}_b)^T \mathbf{G}^{-1} (\mathbf{e}_a - \mathbf{e}_b)$ is the path resistance between nodes a and b (see [36]). In other words, the change in node-voltage with respect to time t at every node is almost the same, i.e.,

$$\frac{dv_a}{dt} \approx \frac{dv_b}{dt}$$

2. The current via each node can be bounded above by

$$i_c = c \frac{d(v_i - v_j)}{dt} \Big|_{t=t_n} \leq c \frac{2V_{dd}}{dt} \quad \text{and} \quad i_r = (v_i - v_j)g_{ij} \leq 2V_{dd}g_{ij},$$

where V_{dd} the amplitude of the signal and dt is minimum time step.

With the assumption of worst case, we have seen that MoveC always gives more accurate results than SplitC. However, MoveC may fail if the above assumption does not hold. Electrically speaking, the time for current flowing from node a through resistor(s), capacitor(s) (including charging and discharging) to node b will have a difference, called delay. The Elmore delay [38] from node a to b is small or large depending on whether the RC time constant from a to b is small or not, i.e., $[R_{eff} \times C]_{a \text{ to } b} < \tau$. Suppose two node-voltages in the same net do not change with the same rate: $\frac{dv_a}{dt} \gg \frac{dv_b}{dt}$, i.e., the change in voltage at node a is much larger than the change in voltage at node b . As a result, it breaks the first assumption of worst case and may lead to large error in MoveC. With this observation, however, we can set locally the additional condition $[R_{eff} \times C]_{a \text{ to } b} < \tau$ for moving a capacitor from a to b to preserve local delay error within τ .

Loosely speaking, indeed locally (looking at one node) the error can be relatively large, but globally the impact on the delay over the full net is averaged out, as we will see in the numerical result, even if we move off nodes with relatively high RC time constant.

4.4 Numerical results

In this section, we present some results to demonstrate MoveC's performance. MoveC is implemented in MATLAB [31] 8.1 (R2013a). To find the connected components of \mathcal{G} , mentioned in the section 2, we used the Matlab function `components` which is in

Boost Graph Library [19].

Fig. 4.1 plots the sparsity structures of the original \mathcal{G}, \mathcal{C} and the reordered \mathbf{G}, \mathbf{C} matrices. The original \mathcal{G}, \mathcal{C} matrices are reordered by the connected components to find the coupling capacitors between nets. Fig. 4.2 shows the sparsity structure of $\hat{\mathbf{C}}$ (after MoveC) with the number of nonzero elements in $\hat{\mathbf{C}}$ is reduced significantly compared to \mathbf{C} .

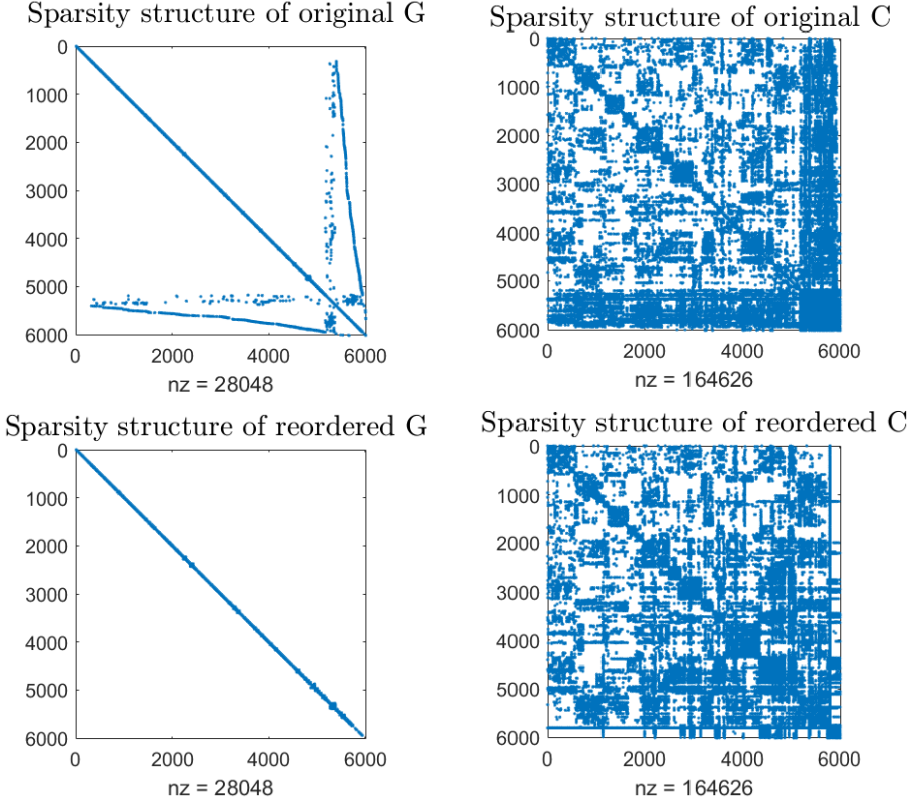


Figure 4.1: Sparsity structures of the original \mathcal{G}, \mathcal{C} and reordered \mathbf{G}, \mathbf{C} .

Table 4.1: Numerical results of MoveC, moving c_{ij} if $c_{ij} < \epsilon$.

Netlist	ϵ	Type	#C	Sim. Time (s)	Error	#Er. meas.
1. N = 32,376; #R = 53,915; #MOS = 11,386	-	Orig.	265,089	11,073	3.4%	10/5331
	10^{-17}	MoveC	102,577	4645	3.4%	10/5331
		Red. rate	61.3%	2.4X		
	10^{-16}	MoveC	89,959	4545	3.5%	10/5331
		Red. rate	66.1%	2.4X		
	-	Orig.	307,015	47,782	4.1ps	0/1
2. N = 47,358; #R = 82,801; #MOS = 11,987	10^{-17}	MoveC	157,944	8400	2.3ps	0/1
		Red. rate	48.55%	5.6X		
	10^{-16}	MoveC	130,752	8303	2ps	0/1
		Red. rate	57.41%	5.8X		
	-	Orig.	307,015	47,782	4.1ps	0/1
	-	Orig.	307,015	47,782	4.1ps	0/1

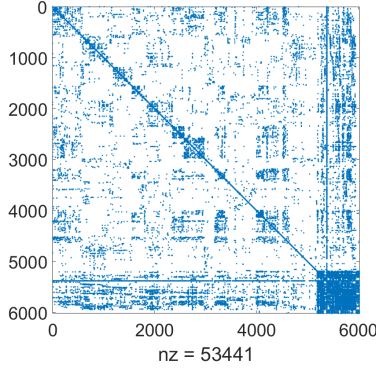

 Figure 4.2: Sparsity structure of $\hat{\mathbf{C}}$

 Table 4.2: Numerical results of SplitC, splitting c_{ij} if $c_{ij} < \epsilon$.

Netlist	ϵ	Type	#C	Sim. Time (s)	Error	#Er. meas.
1. N = 32,376; #R = 53,915; #MOS = 11,386	-	Orig.	265,089	11,703	3.4%	10/5331
	10^{-17}	SplitC	64,476	1981	2.5%	10/5331
		Red. rate	75.68%	5.6X		
	10^{-16}	SplitC	25,761	1669	63.6%	465/5331
		Red. rate	90.28%	6.6X		
	-	Orig.	307,015	47,782	4.1ps	0/1
2. N = 47,358; #R = 82,801; #MOS = 11,987	10^{-17}	SplitC	113,985	6378	0.11ps	0/1
		Red. rate	62.87%	7.5X		
	10^{-16}	SplitC	59,630	5364	9.12ps	1/1
		Red. rate	80.58%	8.9X		
	-	Orig.	307,015	47,782	4.1ps	0/1
	-	Orig.	307,015	47,782	4.1ps	0/1

 Table 4.3: MoveC's results with criteria $c_{iM}/(hG_M + C_M) < e_r$ and $c_{iM} < 10^{-14}$.

Netlist	e_r	Type	#C	Sim. Time (s)	Error	#Er. meas.
1. N = 32,376; #R = 53,915; #MOS = 11,386	-	Orig.	265,089	11,073	3.4%	10/5331
	0.1%	MoveC	98,255	5010	3.4%	10/5331
		Red. rate	63%	2.2X		
	1%	MoveC	89,484	4012	3%	10/5331
		Red. rate	66.2%	2.8X		
	-	Orig.	307,015	47,782	4.1ps	0/1
2. N = 47,358; #R = 82,801; #MOS = 11,987	0.1%	MoveC	130,243	8590	3ps	0/1
		Red. rate	57.6%	5.6X		
	1%	MoveC	119,033	8414	1.5ps	0/1
		Red. rate	61.2%	5.7X		
	-	Orig.	307,015	47,782	4.1ps	0/1
	-	Orig.	307,015	47,782	4.1ps	0/1

Table 4.1 shows some MoveC's results for two multi-terminal netlists extracted from real chip designs compared to the results obtained by SplitC, see Table 4.2. We introduce SplitC as a simplification in which we completely replace every coupling capacitors, that are smaller than a given threshold ϵ , by two capacitors to ground, i.e., if $|\mathbf{C}(a, b)| = c \leq \epsilon$ then $|\mathbf{C}(a, b)| = 0$, $|\mathbf{C}(a, 0)| = |\mathbf{C}(b, 0)| = c$. In the table 4.1, each row consists of a netlist with its information such as the number of nodes N , the number of resistors #R and the number of MOS devices #MOS. For each netlist, at each threshold ϵ , the sparsity structure before and after simplification are noted by the number of capacitors #C. The reduction rates (Red. rate) are

shown for the corresponding columns. For example, the percentage reduction in $\#C$ is $\frac{(\#C_{\text{Orig}} - \#C_{\text{MoveC}}) \times 100}{\#C_{\text{Orig}}}$, the Red. rate in simulation time (Sim. Time) is $\frac{\text{Sim. Time}_{\text{Orig}}}{\text{Sim. Time}_{\text{MoveC}}}$. The "Error" displays either the maximum relative error (%) or maximum absolute error (picosecond) of all measurements. The number of error measurements ($\#Er.$ meas.) is the number of measurements whose relative error exceeds 2% or whose absolute error exceeds 10 ps. For instance in the seventh column of table 4.1, the first netlist has 5331 important measurements and 10 of which exceed 2% relative error. The second netlist has only one important measurement which does not exceed 10ps. The absolute error and the relative error are computed between the measurements of the original and the simplified circuit.

As shown in tables 4.1 and 4.2, for the two netlists, both MoveC and SplitC give a reduction up to more than 50% for the number of capacitors ($\#C$) and the speedup up to more than 2X. SplitC reduces much more the number of capacitors and hence gives faster transient simulation than MoveC. However, with higher threshold SplitC gives larger error and more error measurements. For MoveC, by experiments even with higher threshold, the maximum errors do not exceed 5% for relative error and 10ps for absolute error. Also, there are constantly the same 10 sensitive measurements exceeding 2% error. We can even increase the threshold ϵ for more speedup. However, by experiments these netlists with bigger thresholds, we didn't obtain much speedup since the number of coupling capacitors below 10^{-15} does not vary greatly from the number of coupling capacitors below 10^{-16} .

Table 4.3 shows results of another strategy to remove coupling capacitors. The strategy is based on the local error e_r (when removing one capacitor) to remove a capacitor. The results confirm our electrical reasoning that even if the local delay error is large, the global impact on the delay over the full net is averaged out because we preserve the total coupling capacitances between nets. For the two netlists, MoveC has more or less the same speedup, reduction and error as results shown in table 4.1.

Fig.4.3 shows voltage outputs of SplitC, MoveC and the original problem at a specific node. SplitC with threshold 10^{-16} completely fails since the difference toward the reference is big (up to 15ps). The differences of the other outputs toward the reference are about 3.6ps (smaller than the time step used for integration) so they are acceptable.

Numerical results of MoveC for several other realistic netlists are listed in table 4.4. MoveC gives speedups of more than 2X while preserving the default accuracy criteria (typically within 2% for relative error and 10ps for absolute error).

4.5 Conclusion

MoveC is an efficient method to speedup transient simulation of RC interconnect circuit after usual reduction, without accuracy loss theoretical and electrical arguments for MoveC's performance are given and confirmed by numerical experiments. MoveC is applicable in a robust way, even for sensitive netlists, because the error is controlled by a single threshold. Speedup up of a factor 2 and more is reported without accuracy loss.

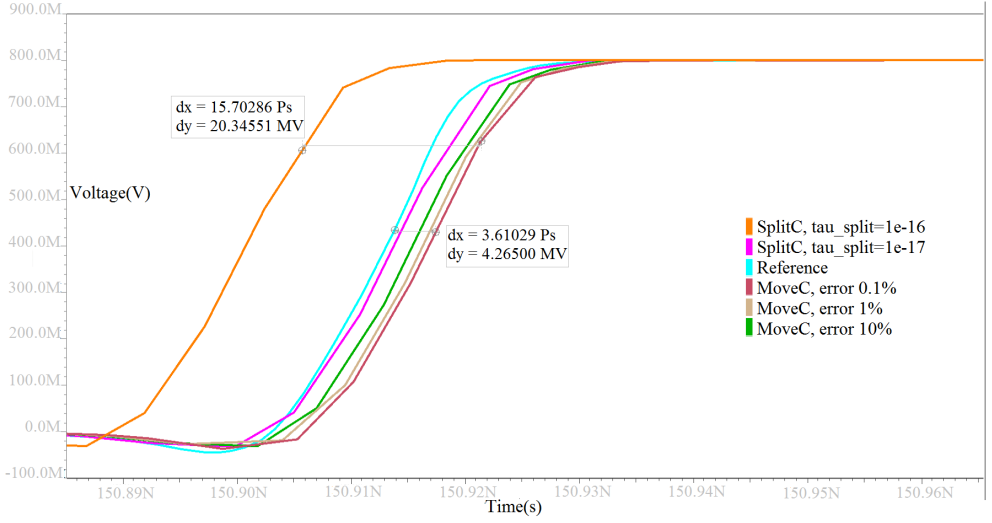


Figure 4.3: Voltage outputs of SplitC, MoveC and the original problem. From left to right: SplitC with thresholds 10^{-16} (orange), reference (cyan), SplitC 10^{-17} (magenta), MoveC with errors 10% (green), 1% (beige) and 0.1% (pink)

Table 4.4: MoveC's results for realistic netlists

Netlist	CPU orig (10 ³ s)	CPU MoveC (10 ³ s)	CPU orig/MoveC	#MOS	#R	#Cc orig	#Cc MoveC	#Cgnd orig	#Cgnd MoveC	#nodes
1	3.1	1.2	2.7	9931	28937	169979	47088	111765	33129	20096
2	2.7	1	2.7	14022	45262	213455	62282	267263	72085	26647
3	45.4	18	2.6	1378	13042	21686	1425	33397	520	4117
4	3.1	1.2	2.5	14022	44232	218787	64182	257853	68848	26141
5	2.1	0.9	2.4	11592	36373	163806	51603	206753	58640	22413
6	3.9	1.7	2.3	14738	42228	371545	115810	465626	11434	41037
7	3.6	1.6	2.2	12101	39223	186655	53430	155182	48213	27400
8	2.6	1.2	2.2	9749	27259	153837	44655	109312	33023	19374
9	2	0.9	2.2	11592	36765	158440	49731	216539	61837	22904
10	1.8	0.9	2.1	12912	39953	172202	53581	230962	67757	24642
11	1.5	0.7	2.1	11386	40980	125222	41798	259429	73272	26205
12	2.1	1	2.1	6078	47140	120555	16563	35250	6649	15179
13	5.6	2.7	2.1	33003	100412	481604	137469	538431	149993	67400
14	2.8	1.4	2.1	9943	27888	160323	46310	106832	32671	19472
15	5.4	2.6	2.0	21151	77119	255544	79584	466503	132509	51844
16	3.1	1.5	2.0	19356	72722	212347	60998	461258	129681	49064
17	1.8	0.9	2.0	12912	40250	177529	55720	221254	64686	24360
18	1.7	0.9	1.9	13816	53717	159604	50399	333251	91944	33063
19	2.1	1.1	1.9	13816	52887	164481	52147	329359	89180	32428
20	1.4	0.8	1.9	11386	41607	120601	40032	265313	76241	26784
21	4.2	2.3	1.9	33543	102488	481408	136388	555321	158290	69803
22	0.09	0.05	1.8	3338	3999	24774	7689	147715	24512	4615
23	2.4	1.3	1.8	9749	29627	151136	43727	114714	34678	21528
24	2.3	1.3	1.8	23803	230617	322137	76170	988350	107638	163078
25	3	1.7	1.7	28265	93212	434697	164168	243583	115358	50243
26	2.2	1.3	1.7	11987	54699	135393	44253	406094	106707	41079
27	55.6	32.4	1.7	24544	61534	245899	76964	350392	26158	59008
28	0.115	0.07	1.7	2931	7936	63380	20771	31344	7461	13419
29	1.5	0.9	1.7	23973	120555	169935	30614	312869	26338	45508
30	1.4	0.9	1.6	12706	54930	98390	34479	351536	114433	32023
31	0.6	0.4	1.6	11276	29569	52413	28323	297733	76763	20975
32	1.6	1	1.6	11915	55834	89412	28388	396054	127104	41420
33	2	1.3	1.6	11141	41567	128350	38310	222190	66739	27501

Chapter 5

SelectC: switching capacitors off and on at demand

5.1 Introduction

In this chapter, we propose an efficient method for the case of RC networks with dense capacitive coupling. The basic idea is to deactivate small coupling capacitors between distinct resistor nets for the time(s) that their currents is negligible. In particular, coupling capacitor currents at the current time step are used to decide which coupling capacitors will be active in the next step of the transient simulation. Thus, for many time steps, sparser capacitance matrices are obtained and replace the original capacitance matrix in the system matrix, thus providing faster transient simulation. For several experiments to be described later, the method gives promising results.

The chapter is presented as follows. In Section 5.2 we introduce the SelectC method. Section 5.3 addresses error estimate issues. Numerical results are showed in Section 5.4. Finally, in Section 5.5 we provide conclusions.

5.2 The SelectC method

Because SelectC [9] deals with coupling capacitors (the capacitors between distinct **G**-components), it is vital to identify those capacitors. This is done by reordering **C** by **G**-components as described in the first procedure of algorithm 3. To explain the idea of SelectC, let c_{ij} be a capacitor between nodes i and j of distinct nets. The current i_c passing through c_{ij} (direction $i \rightarrow j$) is

$$i_c = c_{ij} \cdot \frac{d(x_i - x_j)}{dt} \approx c_{ij} \cdot \left(\frac{x_i(t_{n+1}) - x_i(t_n)}{t_{n+1} - t_n} - \frac{x_j(t_{n+1}) - x_j(t_n)}{t_{n+1} - t_n} \right) = \tilde{i}_c. \quad (5.2.1)$$

When i_c is negligible, i.e., $i_c \leq \tau_{\text{sel}}$, then c_{ij} is said to be inactive (here we assume that cumulative effects are neglected) and can be deactivated in the capacitance matrix. In other words, SelectC is the method of selecting active/inactive coupling capacitors based on the currents going through them. The inactive capacitors are coupling

capacitors whose node voltages barely change from the previous time step to the current one. To ensure the voltage derivative part is minor and the i_c value is not affected by the c_{ij} value, the possibly inactive capacitors c_{ij} to be removed should have capacitance smaller than a chosen constant τ_{cap} . Eventually, a capacitance is removed if both

$$\begin{cases} c_{ij} \leq \tau_{\text{cap}} \\ \tilde{i}_c \leq \tau_{\text{sel}}, \end{cases} \quad (5.2.2)$$

where τ_{sel} and τ_{cap} are user-defined thresholds. Algorithm 3 summarizes the entire flow of SelectC method.

Algorithm 3 SelectC

```

1: procedure USE THE G-COMPONENTS TO DETERMINE THE COUPLING CAPACITORS
2:    $[comps, \sim] = \text{components}(\mathbf{G})$ 
3:    $[\sim, \mathbf{P}] = \text{sort}(comps)$ 
4:    $\mathbf{G} = \mathbf{G}(\mathbf{P}, \mathbf{P})$ 
5:    $\mathbf{C} = \mathbf{C}(\mathbf{P}, \mathbf{P})$ 
6: end procedure
7: procedure DYNAMIC SELECTION OF COUPLING CAPACITORS
8:    $n = 0, \quad \mathbf{C}_{\text{sel}}^{(n)} := \mathbf{C}, \quad \mathbf{x}(t_n)$  is DC solution
9:   while  $t_n < t_{\text{end}}$  do
10:      $n = n + 1$ 
11:      $k = 0; \quad \mathbf{x}_{n-1}^{(k)} = \mathbf{x}(t_{n-1}) = \mathbf{x}_{n-1}$ 
12:     while  $\|\mathbf{f}(\mathbf{x}_{n-1}^{(k)})\|_{\infty} > \varepsilon$  do  $\triangleright$  Newton iteration
13:        $\mathbf{x}_{n-1}^{(k+1)} = \mathbf{x}_{n-1}^{(k)} - \mathbf{J}_f^{-1} \mathbf{f}(\mathbf{x}_{n-1}^{(k)})$ 
14:        $k = k + 1$ 
15:     end while
16:      $\mathbf{x}(t_n) = \mathbf{x}_{n-1}^{(m)}$   $\triangleright \|\mathbf{f}(\mathbf{x}_{n-1}^{(m)})\|_{\infty} < \varepsilon$ 
17:     if  $\|\mathbf{dx}(t_{n-1})\|_{\infty} := \|\mathbf{x}(t_n) - \mathbf{x}(t_{n-1})\|_{\infty} \geq \tau_{dx}$  &  $\mathbf{C}_{\text{sel}}^{(n-1)} \neq \mathbf{C}$  then
18:        $\mathbf{C}_{\text{sel}}^{(n-1)} := \mathbf{C}$ 
19:        $n = n - 1$ 
20:     else
21:       build  $\mathbf{C}_{\text{sel}}^{(n)}$  satisfying (5.2.2)
22:     end if
23:   end while
24: end procedure

```

The condition of $\|\mathbf{dx}(t_{n-1})\|_{\infty} \geq \tau_{dx}$ and $\mathbf{C}_{\text{sel}}^{(n-1)} \neq \mathbf{C}$ is used to back up the solution $\mathbf{x}(t_n)$ in case there is a large difference between the two consecutive vector of solutions $\mathbf{x}(t_n) - \mathbf{x}(t_{n-1}) = \mathbf{dx}(t_{n-1})$. τ_{dx} is an absolute value as time scaling the linear version of (2.2.4) by $s := f \cdot t$ (frequency f) reduces (2.2.4) to $\mathbf{f}(\mathbf{x}(s)) = f \cdot \mathbf{C}\mathbf{x}'(s) + \mathbf{G}\mathbf{x}(s) + \mathbf{B}\mathbf{u}(s)$, $s \in (0, f \cdot T]$ where instead of $\mathbf{x}'(t) = O(f)$ time the input signal, $\mathbf{x}'(s) = O(1)$ times inputs signal. This implies that with or without time scaling $\mathbf{x}(t_{n-1}) - \mathbf{x}(t_n) = dt \cdot O(f)$ and $\mathbf{x}(s_{n-1}) - \mathbf{x}(s_n) = dt \cdot 1$ where $dt \cdot O(f) = ds$.

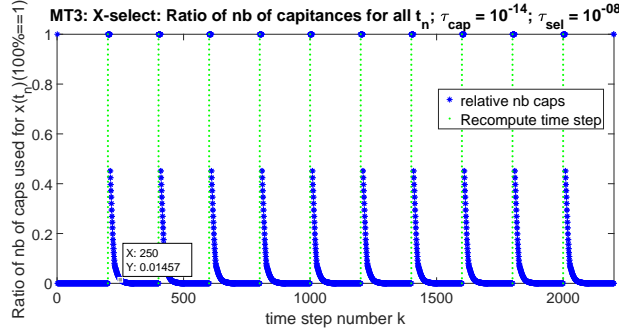


Figure 5.1: The ratio of the number of active capacitors during transient simulation (star-dotted line). The time steps where the recalculation with all capacitors is required (dotted line). The time steps correlate with the simulation interval $[0, 5.5] \times 10^{-10}$.

5.3 SelectC error estimate

We have found no tangible manner to estimate an upper bound for the error $\mathbf{x}_{original} - \mathbf{x}_{select}$. The reason being that Duhamel’s formula (see (3.6.6)) only applies in the case that the system of ODE’s related to our DAE is of the form $\mathbf{x}' = \mathbf{A}\mathbf{x} + \mathbf{b}$ with the time-independent constant coefficient \mathbf{A} matrix. However, for SelectC, the coefficient of \mathbf{C} are time-dependent (capacitors are deactivated and reactivated), i.e., $\mathbf{A} = -\mathbf{C}_{select}\mathbf{G}$ depends on time.

5.4 Numerical results

In this section, we present results of SelectC for large linear RC networks derived from realistic designs of very-large-scale integration (VLSI) chips (netlists 1 and 2 in Table 5.1) and a self-created nonlinear network (netlist 3, Fig. 5.5). For the linear networks the square voltage inputs $2\text{-square}(2\pi ft)$ are injected to one external per net and for nonlinear network, the input square($2\pi ft$) is driven to the first external of the first net. The error $\varepsilon = 10^{-8}$ is chosen for condition of Newton iteration. SelectC is implemented in MATLAB ver. 9.4(R2018a). The method requires \mathbf{G}, \mathbf{C} to be reordered by \mathbf{G} -components, as mentioned in Algorithm 3. We employed the MATLAB function `components` which is in the Boost Graph Library [19]. The numerical integration is Backward Euler with constant increment.

Figure 5.1 shows the ratio of the number of active capacitors during the transient simulation (of netlist 1) when \mathbf{C}_{sel} is used. While full transient simulation requires full \mathbf{C} elements, SelectC uses full \mathbf{C} elements only 10 times (to recompute the time steps) when the condition of $\|\mathbf{dx}(t_n)\|_\infty \geq \tau_{dx}$ and $\mathbf{C}_{sel}^{(n)} \neq \mathbf{C}$ meets (because the signal suddenly rises from 0 to 2 and falls from 2 to 0, see Fig. 5.3 *the top figure* for the input signal - plotted in blue line). For other time steps, for instance at time step $t_n = t_{250}$, \mathbf{C}_{sel} is sparser than \mathbf{C} (Fig. 5.2).

Table 5.1 shows some SelectC’s results for two multi-terminal netlists extracted from real chip designs and a self-created nonlinear netlist compared to the results

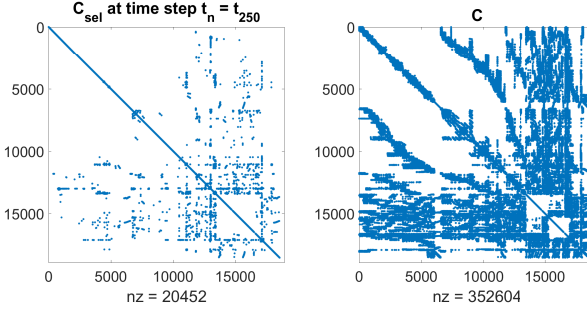


Figure 5.2: \mathbf{C}_{sel} (left) at time step $t_n = t_{250}$ is much sparser than the original (re-ordered) \mathbf{C} (right).

obtained by their original problem. Herein, N denotes the number of nodes, τ_{cap} and τ_{sel} are thresholds for selecting inactive capacitors in (5.2.2), $\tau_{dx} = 1$ (linear case) and $\tau_{dx} = 0.1$ (non-linear case) indicate the condition of re-computing the time step with dramatic change in \mathbf{dx} , and **Avg#cap** stands for the mean number of the active capacitors during the transient simulation. Indeed, for the original problem **Avg#cap** is the number of capacitors in \mathbf{C} network and is computed by the MATLAB command `nnz(triu(C,1))`, i.e., the number of non-zero elements of the strictly upper triangular of \mathbf{C} . For the SelectC system, **Avg#cap** stands for the division of the sum of the amount of the active capacitors per time step over the total number of the time steps. The reduction rates (**Red. rate**) are shown for the corresponding columns. For instance, the percentage reduction in **Avg#cap** is $\frac{(Avg\#cap_{orig} - Avg\#cap_{sel}) \cdot 100}{Avg\#cap_{orig}}$, the **Red. rate** in simulation time (**Sim. Time**) is $\frac{Sim.Time_{orig}}{Sim.Time_{sel}}$. Finally, **Rel. Error** displays the maximum relative error (in voltage) of all variables and is computed by $\max_{i=1,2,\dots,N} \{\|x_i - x_i^{C_{sel}}\|_{\infty}\} / \max_{i=1,2,\dots,N} \{\|x_i\|_{\infty}\}$ with $\|x_i\|_{\infty} = \max_{k=1,2,\dots,n_T} \{|x_i(t_k)|\}$, n_T is the number of time steps. **Time Newt. Iter.** and **Time Lin. Solver** are time needed for Newton iteration (without linear solving) and for solving the linear system inside, respectively.

The transient simulation time strongly depends on the sparsity of the resulting matrices (Table 5.1). Obviously, SelectC can reduce about 90% of the average density of the original \mathbf{C} , leading to faster transient simulation time (at maximum by a factor of 7 in netlist 1). Additionally, the error is acceptable for $\tau_{sel} = 10^{-8}$ Amperes(A) (for instance in netlist 1 Fig. 5.3). For $\tau_{sel} = 10^{-4}$ (A), especially in netlist 2 we gain more speed up, however, the accuracy is not acceptable (Fig. 5.4). Note that the Time Lin. Solver of the SelectC problem is faster than that of the original problem (up to 8.9X). However, considering the transient simulation time, the speedup is only up to 7X since there needs the computation effort at \mathbf{C}_{sel} construction at every time step.

Finally focus at the non-linear circuit in Fig. 5.5, more specifically focus on the sole capacitors (call it \mathbf{C}) connected to the source (first capacitor on the very left in bottom to top direction). As mentioned in **capacitors connected to voltage sources** on page 23, capacitors connected to voltage sources are moved to the right

Table 5.1: Numerical results of SelectC. Netlists 1 and 2 are linear, netlist 3 is non-linear.

Netlist	$\tau_{\text{cap}}(F)$	$\tau_{\text{sel}}(A)$	Type	Avg#cap	Sim. Time (s)	Rel. Error (V)	Time Lin. Solver (s)	Time Newt. Iter. (s)
1. N = 18,927	1e-14	-	Orig.	168,309	619	-	588	31
		1e-8	SelectC	15,090	105	6e-4	77	11
			Red. rate	91%	5.9X		7.6X	2.8X
		1e-4	SelectC	17,756	88.5	3.1e-2	66	10
			Red. rate	89.5%	7X		8.9X	3.1X
2. N = 6,887	1e-14	-	Orig.	14,161	53	-	48	6
		1e-8	SelectC	2,102	23	3.1e-4	15	5
			Red. rate	85%	2.3X		3.2X	1.2X
		1e-4	SelectC	611	11.6	6e-2	7	4
			Red. rate	94.7%	4.6X		6.9X	1.5X
3. N = 32	1e-14	-	Orig.	23	7.7	-	0.4	4
		1e-8	SelectC	8	7.1	3.3e-4	0.2	3.3
			Red. rate	65.2%	1.1X		2X	1.2X
		1e-4	SelectC	3	6.7	4.7e-4	0.2	3.2
			Red. rate	86.9%	1.15X		2X	1.25X

hand side (since voltage sources are eliminated), i.e., the contribution of C ends up in the right hand side. Hence, possible de-activation and or reactivation change the right hand side of our system and does lead to large errors. Therefore, as indicated in Chapter 2 Section 2.3, C should not be deactivated/reactivated or moved.

5.5 Conclusions and outlook

The SelectC technique provides faster transient simulations of RC networks up to the factor of 7. The method works nicely with signals having constant period of times, for instance trapezoidal, pulse and/or square signals for problems with many C-parasitic. To be investigated are the reliability and automatic detection/generation of the method (more precisely, given an error tolerance, which value of τ_{sel} and τ_{cap} we should choose, and vice versa), and also the application of SelectC for general circuits including nonlinear elements.

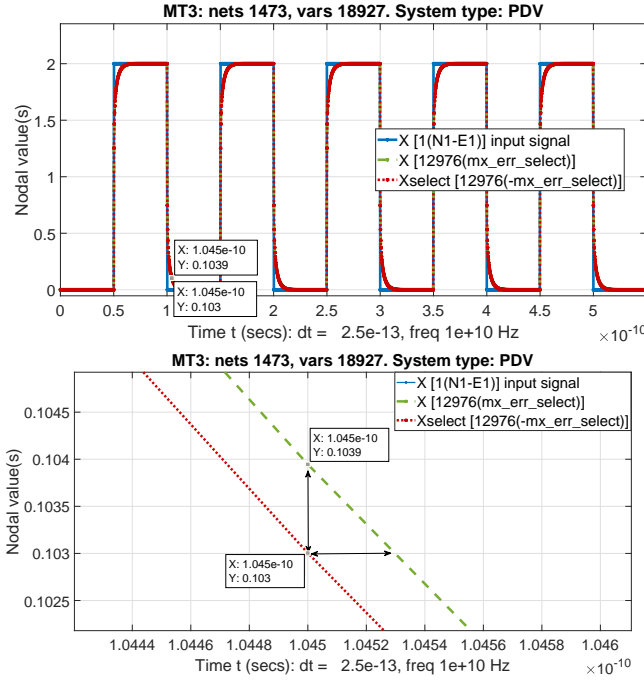


Figure 5.3: *The top figure:* Transient simulation of the original system versus SelectC system (netlist 1), the output 12976 gives maximum absolute error, inputs are square signals injected to one external/net, $\tau_{\text{sel}} = 10^{-8}(\text{A})$. *The bottom figure* shows the zoom in of the two marked points (from the top figure) shown the difference (about 10^{-3}) between the original problem and the modified one. The delay error (horizontal line) is about 0.03 (picosecond) and is also acceptable.

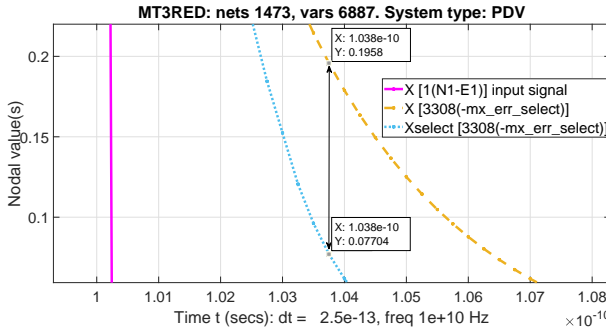


Figure 5.4: The zoom in of transient simulation of the original system versus SelectC system (netlist 2), the output 3308 gives maximum absolute error, inputs are square signals injected to one external/net, $\tau_{\text{sel}} = 10^{-4}(\text{A})$. The two marked points shown the large difference (about 10^{-1}) between the original problem and the modified one.

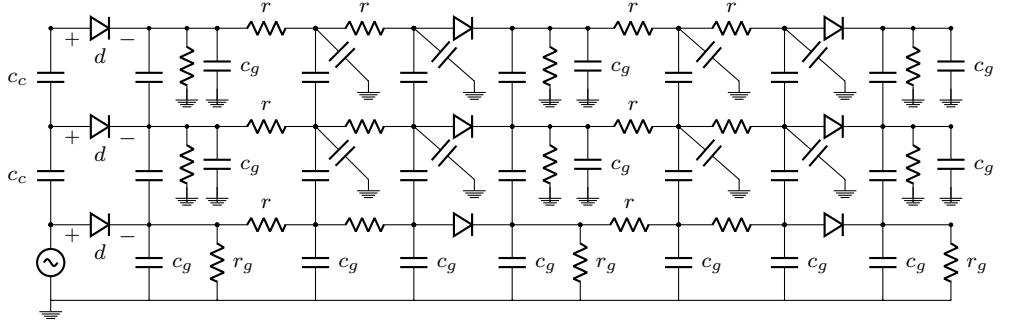


Figure 5.5: An academical nonlinear network. Coupling capacitor $c_c = 10^{-15}(\text{F})$, capacitor connected to ground $c_g = 10^{-16}(\text{F})$, $r = 10^{-4}(\Omega)$, $r_g = 10^4(\Omega)$ and diode d with $I_{\text{sat}} = 1.23e - 9(\text{A})$, $\eta = 1.73$ and $V_T = 26e - 3(\text{V})$.

Chapter 6

Conclusions and Recommendations

6.1 Conclusions

In this dissertation, we presented new numerical methods to speedup the transient simulation of dense parasitic RC networks. In particular for networks derived from real designs of very-large-scale integration (VLSI) chips with parasitic capacitors. These new methods focus on the parasitic capacitors.

First, Chapter 4 presents MoveC which moves parasitic capacitances. Typically, parasitic capacitors connect two distinct resistor singly connected components. MoveC determines (for each two distinct components) the largest coupling capacitor between them and next moves coupling capacitances (between the same components) which are smaller than a default threshold to the largest coupling capacitance. We mention "one of the largest coupling capacitances" because there might be capacitances with the same maximum value. The criteria of moving coupling capacitors can be changed by the user, for instance, one can choose larger or smaller threshold τ_{move} for moving small coupling capacitors, for which coupling capacitors under τ_{move} will be moved to other place. Note that we should not create extra coupling capacitances because results can become sub-optimal. For certain networks (from industry) a factor 2 speedup was found. Of course in case with few coupling capacitors, speedup (as expected) is minimal.

We also mentioned SplitC in this chapter to compare to MoveC. For SplitC, each small coupling capacitor which is smaller than a default threshold is replaced by two capacitors, one from each vertex to the reference node. The capacitance value of each capacitor to reference node equals that of the removed coupling capacitor. Numerical experiments show that using SplitC speeds up the process but also can result in large simulation errors. This happens when the coupling capacitances between two nets are under threshold and are removed resulting in two decoupled components.

Chapter 5 introduced SelectC as another method to reduce simulation time. At each time step of the transient simulation, SelectC select active coupling capacitors to be included in the simulation at the next time step. An active coupling capacitor

is a coupling capacitor whose current i_c is large enough, i.e., $i_c > \tau_{sel}$. This way, we solve sparser problem at each time step because we use sparser \mathbf{C}_{sel} . The speedup of SelectC can reach up to a factor 7 with acceptable error. In general, the speedup of these three methods is high for dense parasitic capacitors.

To show the simulation speed improvement, simulations were run for all of SplitC, MoveC and SelectC in a Matlab environment. To run these simulations we needed to apply sources (current or voltage). The precise position of where the sources are applied influences the speedup factors, but cannot be extracted from the data from the (industrial) networks.

6.2 Recommendations

In the future, it might be useful to investigate the following. For MoveC, one could also change the condition of where to move small coupling capacitors to, for example, instead of moving to the capacitor with maximum capacitance, one can move to other maximal capacitances with the same value. By this way, we can avoid adding too many small coupling capacitors to only one maximum capacitor which might result in a very large capacitor between nets.

For SplitC, we can improve SplitC by only splitting small coupling capacitors $< \tau_{split}$ and keep at least one coupling capacitor between two nets. This will avoid that SplitC can disconnect nets.

All presented methods can be extended to the non-linear case, with promising results, shown by solving an academic non-linear circuit with SelectC in this thesis.

The last point we would like to mention is the determination of a strategy for positioning of the sources. As mentioned in the previous section, the results of the simulation are affected by the position of the sources. One proposed strategy for positioning the sources is to inject a source to one potential vertex at a time and run the full simulation. The source vertex where we obtain the maximum number of non-zero nodal voltages and the smallest voltage errors be appointed as the source position.

Appendix

We supply the code in Mathematica for research in Chapter 3, section 3.4. The code provide the results and Figures 3.3, 3.4, 3.5, 3.6,

```

ClearAll
a = 0;
st = {Cos[2*Pi*s], a*Cos[2*Pi*s]};
b = G12.Inverse[G22].st;
G11 = {{2, 0, -1, 0}, {0, 2, 0, -1}, {-1, 0, 2, 0}, {0, -1, 0, 2}};
G12 = {{-1, 0}, {0, -1}, {0, 0}, {0, 0}};
G21 = Transpose[G12];
G22 = {{2, 0}, {0, 2}};
G = G11 - G12.Inverse[G22].G21;
c =  $\frac{1}{4}$ ; (*c=1;*)
(*orig problem*)
C11 = {{2, -1, 0, 0}, {-1, 2, 0, 0}, {0, 0, 1+c, -c}, {0, 0, -c, 1+c}}
CinvG = Inverse[C11].G
VCinvG = Transpose[Eigenvalues[CinvG]]
DCinvG = Eigenvalues[CinvG]
Cinvb = Inverse[C11].b
TimeConstrained[xc = MatrixExp[-CinvG*t].Integrate[MatrixExp[CinvG*s].Cinvb, {s, 0, t}],
Infinity];
xr = -Inverse[G22].(st + G21.xc);
(*moveC problem*)
C11 = {{2+c*1, -1-c*1, 0, 0}, {-1-c*1, 2+c*1, 0, 0}, {0, 0, 1, 0}, {0, 0, 0, 1}}
CinvG = Inverse[C11].G
VCinvG = Transpose[Eigenvalues[CinvG]]
DCinvG = Eigenvalues[CinvG]
Cinvb = Inverse[C11].b
TimeConstrained[xcm = MatrixExp[-CinvG*t].Integrate[MatrixExp[CinvG*s].Cinvb, {s, 0, t}],
Infinity];
xrm = -Inverse[G22].(st + G21.xcm);
(*plot xo,xm*)
Plot[{xc, xr, xcm, xrm}, {t, 0, 4},
PlotLegends ->
Placed[{"xo_2", "xo_5", "xo_3", "xo_6", "xo_1", "xo_4", "xm_2", "xm_5", "xm_3", "xm_6",
"xm_1", "xm_4"}, Above], AxesLabel -> {t, x}, LabelStyle -> Directive[Blue, Bold],
PlotStyle -> {Thickness[0.01]}, TicksStyle -> Directive[FontSize -> 16],
AxesStyle -> Directive[Gray, FontSize -> 16], GridLines -> Automatic]
rxr = Abs[xr - xrm];
xcxcm = Abs[xc - xcm];
Plot[{xcxcm, rxr}, {t, 0, 4},
PlotLegends ->
Placed[{"|xo_2-xm_2|", "|xo_5-xm_5|", "|xo_3-xm_3|", "|xo_6-xm_6|", "|xo_1-xm_1|",
"|xo_4-xm_4|"}, Above], AxesLabel -> {t, x}, LabelStyle -> Directive[Blue, Bold],
PlotStyle -> {Thickness[0.01]}, TicksStyle -> Directive[FontSize -> 16],
AxesStyle -> Directive[Gray, FontSize -> 16], GridLines -> Automatic]
Clear[a, t, c]

```

Summary

Numerical methods for accelerating transient simulation of dense parasitic RC networks

The growth in the complexity of modern Integrated Circuit (IC) design makes the design and manufacturing process more and more complex. Circuit simulation is necessary since it helps to discover undesirable behaviors prior to manufacturing and deployment, and thus minimize the possibility of producing faulty devices, saving time and money. Due to the ever denser packed components on the ICs, interconnects and parasitic effects have become more dominant. It is crucial to take these density induced effects into account. This requires the simulation of very large scale electrical networks, with a very large number of electrical and parasitic elements. Related simulation is usually time-consuming or infeasible for standard circuit simulation tools. This thesis focuses on speeding up the transient RC network simulation with parasitic capacitors. The considered speed up / acceleration methods are: (1) SplitC, (2) MoveC and (3) SelectC.

All three methods affect the matrix $D\mathbf{F}(\mathbf{x})$ in the Newton update $D\mathbf{F}(\mathbf{x})d\mathbf{x} = -\mathbf{F}(\mathbf{x})$ with $D\mathbf{F}(\mathbf{x}) = s\mathbf{C} + \mathbf{G}$, where \mathbf{C} represents the capacitance matrix related to the unaccelerated or accelerated method, and \mathbf{G} represents the conductance matrix. A circuit with fewer capacitors implies fewer non-zero entries in \mathbf{C} . The time it takes to solve the linear system in general depends on the amount of non-zeros in $s\mathbf{C} + \mathbf{G}$, i.e., in \mathbf{C} and in \mathbf{G} . To speed up the simulation time our three methods mentioned above focus on the reduction of non-zeros in \mathbf{C} , i.e., on the (re-)removal of capacitances or even on the dynamic capacitor selection per time-step.

SplitC is the method of, electrically speaking, replacing a coupling capacitor between two nodes by two coupling capacitors with the same capacitance from each node to ground. Mathematically speaking, two non-zero entries from \mathbf{C} are removed for each split coupling capacitor. Note that only coupling capacitors which are smaller than a chosen threshold are split. Thus, if we use a large threshold, there is a high probability that all coupling capacitors between nets are removed, resulting in disconnection between nets and hence accuracy loss. Therefore, to maintain the connections between nets (at least one coupling capacitor between every two nets), there is an additional option for SplitC. The option is to keep the maximum capacitances per two nets regardless of the threshold value.

MoveC is developed to improve on SplitC. There are two MoveC approaches. In the first approach, for every two distinct connected \mathbf{G} components, we select a maximum-capacitance coupling capacitor, after which we move all other coupling capacitances

(below a threshold) to join (add up to) these selected maximum-capacitance coupling capacitors. The second approach is identical except that we select the maximum-RC value capacitor instead of the maximum-capacitance capacitors. Moving other coupling capacitors (below a threshold) to join (add up to) the selected inter-component capacitors causes fewer non-zero entries in \mathbf{C} . Thus speed up the operation \mathbf{F} in the Newton's iteration.

Regarding SelectC, for each step t_n in the transient simulation, selects the active capacitors (above a threshold) to constitute the \mathbf{C}_n capacitance matrix at that time step t_n (i.e. $s\mathbf{C} + \mathbf{G}$ is replaced by $s\mathbf{C}_n + \mathbf{G}$). A coupling capacitor is active if the current passing through it is approximately zero. Because the SelectC matrix uses a potentially different capacitance matrix per time-step, it is not possible to use off-shelf ODE simulators such as Matlab's ODE toolbox (ode23t, ode15s, etc.). We therefore wrote a Matlab research prototype time integrator based on a simple trapezoid time integration which can handle the dynamic selection of capacitors. Matlab was chosen because it facilitates rapid prototyping, but even in Matlab a fast enough prototype requires specialized vectorized code for about any operation. Important issue is the assessment of the improved speed of Cselect. We present Matlab timings which compare the solution time for the full system against the solution time for the Cselect approach. To this end our simulator aims to provide the fastest possible Matlab implementation of the time integrator. Of interest to the project's participant was the potential improved speed of the Cselect approach in the industrial's partner commercially available circuit simulation software, but providing these timings was deemed to be out of scope (would mean a full C++ implementation in ELDO). Working with matrix representation rather than a graph representation of the networks (based on the provided matrix input) caused extra numerical round-off issues to pop up in many parts of the simulator. All of these issues are addressed.

Numerical experiments are shown to verify the performance for each proposed method. In general, we conclude that the reductions lead to fewer non-zeros in \mathbf{C} which speeds up the transient simulation without "too much" loss of accuracy in voltages and delays, depending on the industrial / academical examples at hand.

Curriculum Vitae

T. K. Nhung Dang was born on November 17, 1991 in Tayninh, Vietnam. She obtained a Bachelor degree (2013) in Mathematics & Computer Science at University of Science, Vietnam National University – Ho Chi Minh City (VNU-HCM), Vietnam. In 2014, she received the Master II degree in Applied Mathematics at Université Francois-Rabelais, Tours, France. From July 2015, she started her Ph.D. project at Eindhoven University of Technology in Mathematics and Computer Science department, under the supervision of Prof. Dr. Wil Schilders and Dr. Joseph Maubach (from TU/e, Netherlands), and Dr. Pascal Bolcato and Dr. Joost Rommes (from Mentor Graphics, Grenoble, France). The results of her research are presented in this thesis.

Acknowledgements

First of all, I would like to thank my promotor Wil Schilders for the opportunity to increase my interest and expertise in scientific research. Thank you for your guidance, support and recommendation to many international and national conferences/workshops where can meet and discuss with people having the same interest as well as improved my confidence.

I'm extremely grateful to my co-promotors Joseph Maubach and Joost Rommes. You were always available for advice, inspiration, and constructive ideas. Thank you for willing to review and discuss my manuscripts. Your comments and suggestions were valuable to me. I always enjoyed our collaboration. Thanks to Dr. Pascal Bolcato for many discussions.

I very much appreciate to the members of the committee, Prof. W.H.A Schilders, Dr. J.M.L. Maubach, Dr. J. Rommes, Prof. G. Ciuprina, Dr. M.E. Hochstenbach, Prof. C. Vuik, and Dr. N.P. van der Meijs. Your suggestions and comments helped me to improve the chapters of this thesis.

Special thanks to Enna, secretary of our CASA group, for helping me with official documents and housing every time I moved back to Eindhoven from Grenoble, France.

Many thanks to all colleagues at the Mathematics and Computer Science Department and the members of CASA group in particular. Thanks to my colleagues and friends in Mentor Graphics as well.

I thank my parents, my younger brother Trung Dang and my soulmate Khoa Nguyen for watching and supporting me all the time.

Bibliography

- [1] A. Antoulas. Approximation of large-scale dynamical systems: An overview. *IFAC Proceedings Volumes*, 37(11):19–28, 2004.
- [2] ASIVA14 project: Analog SIMulation and Variability Analysis for 14nm designs. www.itn-asiva14.eu.
- [3] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.
- [4] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, 1995.
- [5] M. Celik, L. Pileggi, and A. Odabasioglu. *IC Interconnect Analysis*. Kluwer Academic Publishers, New York, 2002.
- [6] S. Chowdhry, H. Krendl, and A. A. Linninger. Symbolic numeric index analysis algorithm for differential algebraic equations. *Industrial & Engineering Chemistry Research*, 43(14):3886–3894, 2004.
- [7] L. O. Chua and P.-M. Lin. *Computer-aided analysis of electronic circuits: algorithms and computational techniques*. Prentice-Hall series in electrical and computer engineering. Prentice-Hall, Englewood Cliffs, N.J, 1975.
- [8] T. H. Cormen, editor. *Introduction to Algorithms*. MIT Press, Cambridge, Mass, 3rd edition, 2009.
- [9] T. K. N. Dang, J. M. L. Maubach, J. Rommes, P. Bolcato, and W. H. A. Schilders. Fast transient simulation of RC circuits with dense capacitive coupling. In *Scientific Computing in Electrical Engineering - SCEE 2018, Taormina, Italy, 2018*, Taormina, Sicily, Italy, 2018.
- [10] G. De Luca. *Advanced Numerical Methods for Simulating Non-autonomous Periodic Strongly Nonlinear Circuits*. PhD thesis, Eindhoven University of Technology, Eindhoven, 2018.
- [11] Eldo Platform. https://www.mentor.com/products/ic_nanometer_design/analog-mixed-signal-verification/eldo-platform.

- [12] P. Elias. Extracting circuit models for large RC interconnections that are accurate up to a predefined signal frequency. In *Extracting circuit models for large RC interconnections that are accurate up to a predefined signal frequency*, pages 764–769, 1996.
- [13] D. Estvez Schwarz and C. Tischendorf. Structural analysis of electric circuits and consequences for MNA. *International Journal of Circuit Theory and Applications*, 28(2):131–162, 2000.
- [14] P. Feldmann. Model order reduction techniques for linear systems with large numbers of terminals. *Proceedings IEEE Comput. Soc Design, Automation and Test in Europe Conference and Exhibition*, pages 944–947, 2004.
- [15] P. Feldmann and R. Freund. Efficient linear circuit analysis by Pade approximation via the Lanczos process. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(5):639–649, 1995.
- [16] R. Freund. SPRIM: structure-preserving reduced-order interconnect macromodeling. In *IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004.*, pages 80–87, San Jose, CA, USA, 2004. IEEE.
- [17] C. W. Gear. Differential-algebraic equation index transformations. *SIAM Journal on Scientific and Statistical Computing*, 9(1):39–47, 1988.
- [18] C. W. Gear. Differential algebraic equations, indices, and integral algebraic equations. *SIAM Journal on Numerical Analysis*, 27(6):1527–1534, 1990.
- [19] D. Gleich. MatlabBGL: A Matlab Graph Library. <https://www.cs.purdue.edu/homes/dgleich/packages/matlabbg1/>.
- [20] M. Günther, U. Feldmann, and J. ter Maten. Modelling and Discretization of Circuit Problems. In *Handbook of Numerical Analysis*, volume 13, pages 523–659. Elsevier, 2005.
- [21] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins studies in the mathematical sciences. Johns Hopkins University Press, Baltimore, 3rd edition, 1996.
- [22] E. J. Grimme. *Krylov Projection Methods for Model Reduction*. Ph.D. dissertation, Univ. Illinois UrbanaChampaign, Dept. Elect. Eng., Champaign, IL, USA, 1997.
- [23] G. Hachtel, R. Brayton, and F. Gustavson. The sparse tableau approach to network analysis and design. *IEEE Transactions on Circuit Theory*, 18(1):101–113, 1971.
- [24] I. N. Herstein. *Topics In Algebra*. Wiley, New York, 2nd edition, 1975.
- [25] C.-W. Ho, A. Ruehli, and P. Brennan. The modified nodal approach to network analysis. *IEEE Transactions on Circuits and Systems*, 22(6):504–509, 1975.

- [26] R. Ionutiu, J. Rommes, and A. C. Antoulas. Passivity-preserving model reduction using dominant spectral-zero interpolation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27(12):2250–2263, 2008.
- [27] R. Ionutiu, J. Rommes, and W. H. A. Schilders. SparseRC: Sparsity preserving model reduction for rc circuits with many terminals. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(12):1828–1841, 2011.
- [28] S. Kathrin. On the Kronecker product. <https://www.math.uwaterloo.ca/~hwolkowi/henry/reports/kronthesisschaecke04.pdf>, 2013.
- [29] P. Liu, S. X.-D. Tan, B. Yan, and B. McGaughy. An efficient terminal and model order reduction algorithm. *Integration*, 41(2):210–218, 2008.
- [30] S. Lungten, W. Schilders, and J. Maubach. Sparse block factorization of saddle point matrices. *Linear Algebra and its Applications*, 502:214–242, 2016.
- [31] MATLAB. version 8.1 (R2013a). <http://www.mathworks.com>.
- [32] F. N. Najm. *Circuit Simulation*. Wiley, Hoboken, N.J, 2010.
- [33] A. Odabasioglu, M. Celik, and L. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(8):645–654, 1998.
- [34] D. Oyaró and P. Triverio. TurboMOR-RC: An efficient model order reduction technique for RC networks with many ports. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(10):1695–1706, 2016.
- [35] T. Reis and T. Stykel. PABTEC: Passivity-preserving balanced truncation for electrical circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(9):1354–1367, 2010.
- [36] J. Rommes and W. H. A. Schilders. Efficient methods for large resistor networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(1):28–39, 2010.
- [37] K. H. Rosen. *Discrete Mathematics and Its Applications*. McGraw-Hill, New York, 7th edition, 2012.
- [38] R. Rutenbar. Electrical timing issues: The Elmore Delay Model. <https://www.ece.cmu.edu/~ee760/760docs/lec18.pdf>, 2001.
- [39] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2nd edition, 2003.
- [40] B. N. Sheehan. Realizable reduction of RC networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(8):1393–1407, 2007.

- [41] J. M. S. Silva, J. F. Villena, P. Flores, and L. M. Silveira. Outstanding issues in model order reduction. In *Scientific Computing in Electrical Engineering, Berlin, Heidelberg*, volume 11, pages 139–152, Berlin, Heidelberg, 2007.
- [42] B. Simeon. Lecture: Differential-Algebraic Equations (DAEs) (Differential-algebraische Gleichungen), 2012. Vorlesung Wintersemester 2012/13, TU Kaiserslautern, Fachbereich Mathematik.
- [43] A. K. Tyagi. *Speeding up Rare-event Simulations In Electronic Circuit Design By Using Surrogate Models*. PhD thesis, Eindhoven University of Technology, Eindhoven, 2018.
- [44] A. Verhoeven. *Redundancy Reduction of IC Models by Multirate Time-integration and Model Order Reduction*. Ph.D. dissertation, Eindhoven University of Technology, Eindhoven, 2008.
- [45] Wikipedia: Netlist. <https://en.wikipedia.org/wiki/Netlist>. Online.