KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT TOEGEPASTE WETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
AFDELING ESAT
K. Mercierlaan 94 — 3001 Leuven (Heverlee)

VARIABLE DIMENSION ALGORITHMS IN THE ANALYSIS OF NONLINEAR CIRCUITS AND NEURAL NETWORKS

Jury :
Prof. J. Berlamont, vice-dekaan, voorzitter
Prof. J. Vandewalle, promotor
Prof. H. De Man
Prof. B. De Moor
Prof. A. Haegemans
Prof. M. Hasler (EPFL, Lausanne)
Prof. D. Roose

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de toegepaste wetenschappen

door

Lieven VANDENBERGHE

U.D.C. 681.3*I2

April 1992

Voorwoord

De voltooiing van dit doctoraat biedt mij de gelegenheid om iedereen te bedanken die er op een of andere manier toe heeft bijgedragen. In de eerste plaats bedoel ik daarmee Prof. Vandewalle, die me meer dan vier jaar lang heeft gesteund. De invloed van mijn promotor op dit werk reikt ver, wat niemand zal ontgaan die vertrouwd is met zijn ruime wetenschappelijke interesse (onder andere in niet-lineaire netwerktheorie en diskrete optimisatie) en met zijn opvattingen over goed wetenschappelijk proza.

Ook wil ik hem samen met de andere leden van het leescomité, Prof. De Moor en Prof. Haegemans, bedanken voor de zorgvuldigheid waarmee ze deze tekst hebben doorgelezen. Met veel plezier denk ik terug aan mijn samenwerking met Bart De Moor, waaruit dit doctoraatswerk overigens is gegroeid.

De andere leden van de examenjury, Prof. Berlamont, Prof. De Man, Prof. Hasler en Prof. Roose, wens ik te bedanken voor hun bereidwillige medewerking, en, wat Martin Hasler betreft, vooral ook voor de belangstelling die hij steeds getoond heeft voor mijn werk, en voor zijn uitnodigingen om het op de ISCAS-conferenties voor te stellen.

Onze onderzoeksgroep SISTA is stilaan te talrijk geworden om iedereen op te sommen. Dit vermindert echter geenszins mijn sympathie tegenover alle huidige en voormalige leden, en mijn waardering voor de vriendschappelijke sfeer waarin ik al die jaren heb kunnen werken. Mijn gedachten gaan daarbij in het bijzonder naar de zeer schone uren in het gezelschap van Marc Moonen en Geert Schelfhout, mijn bureaugenoten met de langste staat van dienst. Verder wil ik in dit dankwoord ook Jan Swevers en Peter Kinget niet onvermeld laten.

Ik wil ook het personeel van het departement bedanken, en meer bepaald Ann Deforce, Ingrid Tokka en Johan Buelens, op wie ik nooit vergeefs een beroep heb gedaan.

Wat ik mijn ouders verschuldigd ben, is natuurlijk moeilijker onder woorden te brengen. Hen, en ook mijn broers en zussen, wil ik hier bedanken voor hun merkwaardig vertrouwen in het belang van mijn bezigheid en voor hun geringe nieuwsgierigheid naar de details ervan.

Tenslotte vermeld ik nog dat dit werk enkel tot stand is kunnen komen dank zij de steun van het Nationaal Fonds voor Wetenschappelijk Onderzoek.

Abstract

Variable dimension algorithms are a class of globally convergent methods for solving sets of nonlinear equations. The main purpose of the work reported here is to combine these algorithms with the qualitative results developed in nonlinear circuit theory during the last twenty years. The thesis contributes to the theory of nonlinear resistive circuits, not only by providing unified and constructive proofs of known existence and uniqueness results, but also by generalizing these results in several ways. Furthermore it is shown that variable dimension algorithms can be used as practically feasible and efficient methods, superior to comparable methods used for solving circuit equations. The criteria for convergence are given in a detailed topological form, taking into account the nature of the components. This also leads to an efficient method for finding multiple equilibrium points in neural networks.

Contents

Nederlandse samenvatting ix

List of symbols xxix

1 Introduction 1
- 1.1 Qualitative theory of nonlinear resistive circuits . . . 1
- 1.2 The continuous deformation principle . . . 3
- 1.3 The D.C.-operating point problem . . . 4
- 1.4 Piecewise-linear techniques in mathematical programming and numerical analysis . . . 7
- 1.5 Piecewise-linear analysis in circuit theory . . . 10
- 1.6 Connection with linear programming . . . 12
- 1.7 Chapter by chapter overview . . . 13
  - 1.7.1 Nonlinear resistive circuits . . . 13
  - 1.7.2 Neural networks . . . 15
  - 1.7.3 Harmonic balance . . . 16
  - 1.7.4 Appendices . . . 17

2 Variable dimension algorithms : basic principle and convergence criteria 19
- 2.1 Introduction . . . 19
- 2.2 The octahedral algorithm . . . 20
  - 2.2.1 Example . . . 20
  - 2.2.2 Problem formulation . . . 22
  - 2.2.3 General scheme . . . 23
  - 2.2.4 Well-definedness of the homotopy path . . . 26
  - 2.2.5 Convergence criterion . . . 28
  - 2.2.6 General formulation of the algorithm . . . 28
  - 2.2.7 The index of a solution found by the octahedral algorithm 29
- 2.3 Other types of variable dimension algorithms . . . 33
  - 2.3.1 The $2n$-ray algorithm . . . 33

  - 2.3.2 The $(3^n - 1)$-ray algorithm . . . 34
  - 2.3.3 The 2-ray algorithm . . . 35
- 2.4 Convergence in circuits with two-terminal elements . . . 38
- 2.5 Circuits with coupling among the resistors . . . 47
- 2.6 Circuits with ideal operational amplifiers . . . 50
- 2.7 Conclusion . . . 56

3 Numerical implementation 59
- 3.1 Introduction . . . 59
- 3.2 Tracing a one-dimensional path through a triangulated region . 59
  - 3.2.1 The reflected Freudenthal triangulation . . . 59
  - 3.2.2 Pivoting through a triangulated region . . . 61
  - 3.2.3 Lexicographical pivoting . . . 63
- 3.3 Application to the octahedral algorithm . . . 64
- 3.4 Pivoting operations on a modified label matrix. . . . 67
  - 3.4.1 The modified label matrix. . . . 67
  - 3.4.2 Acceleration by quasi-Newton steps . . . 69
  - 3.4.3 Exploitation of linear structure in the equations . . . 69
  - 3.4.4 Premultiplying with the inverse Jacobian matrix . . . 72
- 3.5 Computational results . . . 73
  - 3.5.1 The test problems in [132] . . . 73
  - 3.5.2 Test problems from [4] . . . 79
  - 3.5.3 Additional examples . . . 81
- 3.6 Conclusions . . . 82

4 DC-Analysis of active circuits 85
- 4.1 Introduction . . . 85
- 4.2 The nonparallel condition as a sufficient condition for existence of solutions . . . 86
  - 4.2.1 The antipodal property . . . 87
  - 4.2.2 The nonparallel condition . . . 87
- 4.3 Summary of the algorithm . . . 90
  - 4.3.1 The octahedral algorithm with reflections . . . 90
  - 4.3.2 Practical considerations . . . 91
- 4.4 Application of the nonparallel condition to nonlinear circuits . . 93
  - 4.4.1 Mappings of the form $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{g}(\mathbf{x}) + \mathbf{B}\mathbf{x} + \mathbf{c}$ . . . 93
  - 4.4.2 Circuits containing controlled sources . . . 98
  - 4.4.3 Circuits containing ideal operational amplifiers . . . 106
- 4.5 Computing multiple solutions . . . 108
  - 4.5.1 Method . . . 108

4.5.2 Examples 109
4.6 Conclusion 111

5 Nonlinear circuits and the complementarity problem 113
5.1 Introduction 113
5.2 Relation between variable dimension algorithms and Lemke's algorithm 114
5.2.1 The linear complementarity problem and Lemke's algorithm 114
5.2.2 The nonlinear complementarity problem 115
5.2.3 Equivalence between a set of nonlinear equations and the nonlinear complementarity problem 115
5.3 The complete solution of a piecewise-linear circuit 117
5.3.1 Introduction 117
5.3.2 A parametrization of piecewise-linear resistors 118
5.3.3 Formulation of circuit equations 121
5.3.4 The generalized linear complementarity problem 125
5.4 Conclusion 128

6 Applications to neural networks 131
6.1 Introduction 131
6.2 Interpretation in the state space 134
6.3 A path-following method for finding multiple equilibrium points 135
6.3.1 An equivalent problem 135
6.3.2 Basic principle of the algorithm 135
6.3.3 Degeneracy 143
6.3.4 Further properties 148
6.3.5 Numerical aspects 151
6.4 Stability of neural networks 152
6.4.1 The equilibria as stationary points of a variational inequality 154
6.4.2 Neural networks and equilibrium programming 154
6.4.3 Stability of networks with asymmetric coefficients 155
6.5 An iterative method for neural network design 157
6.5.1 Relaxation methods for linear inequalities 157
6.5.2 Application to cellular neural networks 159
6.6 Conclusions 163

7 Harmonic balance and variable dimension algorithms 165
7.1 Introduction 165

7.2 A variable dimension algorithm for the harmonic balance equations 167
7.3 Convergence conditions 171
7.4 Conclusion 174

8 Conclusions 175

A Triangulations 179
A.1 A few facts from nonnegative linear algebra 179
A.2 The octahedral triangulation 180
A.2.1 Representation rule 180
A.2.2 Replacement rule 181
A.3 The modified triangulation 182
A.3.1 Representation rule 182
A.3.2 Replacement rule 183

B Theorems of alternative 185
B.1 General form 185
B.2 Examples 186
B.3 Application to circuit theory 186

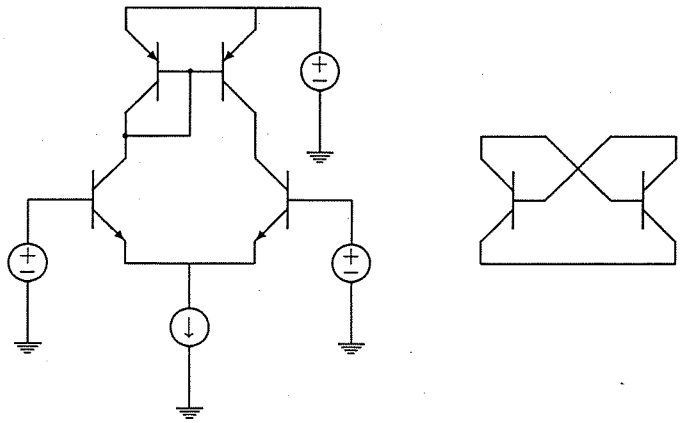
Nederlandse samenvatting

1 Inleiding

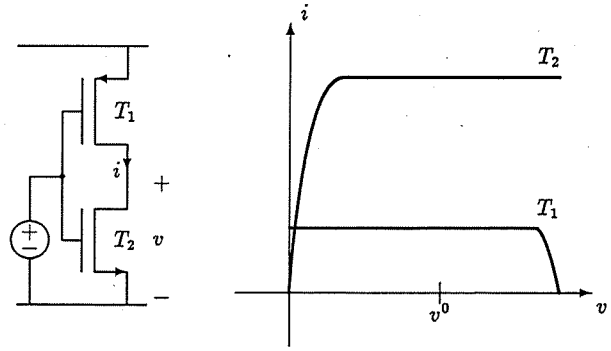
1.1 Situering binnen de niet-lineaire elektrische netwerktheorie

Niet-lineaire elektrische netwerkvergelijkingen vertonen een heel eigen structuur als gevolg van de lineaire en combinatorische aard van Kirchhoffs wetten. Veel onderzoek tijdens de laatste twintig jaar was erop gericht deze structuur te exploiteren in de hoop een algemene, kwalitatieve theorie voor niet-lineaire netwerken op te stellen. Voor resistieve netwerken is men daar grotendeels in geslaagd. Typische resultaten zijn de topologische criteria voor de uniciteit of het bestaan van oplossingen in resistieve netwerken. Het voorbeeld van fig. 1 illustreert de stelling van Nielsen en Willson [68] voor bipolaire-transistornetwerken. Dit circuit heeft een unieke oplossing ongeacht de waarde van de parameters omdat het onmogelijk herleid kan worden — door toepassing van een aantal bewerkingen gespecifieerd in de stelling — tot de feedback-structuur rechts op de figuur. Gelijkaardige criteria zijn later ook opgesteld voor circuits met afhankelijke bronnen [69], operationele versterkers [45] en negatieve weerstanden [71]. Ondanks hun grote algemeenheid zijn deze resultaten van minder onmiddellijk praktisch belang. Ze spreken zich enkel uit over het bestaan of de uniciteit van een oplossing, zonder aan te geven hoe die gevonden kan worden. Voor berekeningen dient men nog steeds zijn toevlucht te nemen tot de klassieke iteratieve methoden, die kunnen divergeren zelfs al kan men bewijzen dat er een oplossing bestaat. Het is b.v. treffend dat globale convergentie van de Newton-Raphsonmethode voor D.C.-analyse van bipolaire-transistornetwerken enkel bewezen is voor het geval met één enkele transistor [78].

Een van de hoofdconclusies van dit werk is dat het belang van de recente ontwikkelingen in de netwerktheorie verder reikt dan vaak wordt aangenomen. Figuur 2 toont met een eenvoudig voorbeeldje hoe een begrip als passiviteit nuttig kan zijn bij het zoeken naar D.C.-werkingspunten. De Newton-Raphsonmethode kan hier niet gestart worden in v^0 wegens de singulariteit



Figuur 1: Netwerk met een unieke oplossing. Feedback-structuur van twee transistoren.



Figuur 2: CMOS-invertor.

van de Jacobiaanmatrix. Op basis van de functiewaarden en de afgeleiden van beide karakteristieken in het punt v^0 is het inderdaad moeilijk om uit te maken of de oplossing links of rechts van v^0 zal liggen. Dit kan echter wel als we rekening houden met de passiviteit van beide transistoren : als v door nul gaat moet de stroom door T_2 ook negatief worden terwijl de stroom door T_1 positief blijft. Er moet dus minstens één oplossing liggen links van v^0, zelfs al zouden de karakteristieken niet monotoon zijn. Veel van wat hieronder gezegd wordt, kan men opvatten als een veralgemening van dit soort redenering naar hogere dimensies.

1.2 Geschiedenis van variabele-dimensiealgoritmen

De meeste methoden die hier besproken zullen worden, berusten op *continue vervorming*, een begrip dat in de differentiaaltopologie bekend staat als *homotopie*. Toegepast op een stelsel algebraïsche vergelijkingen houdt dit in dat het moeilijke probleem $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ eerst vervangen wordt door een triviaal probleem dat eenvoudig op te lossen is (b.v. een lineair stelsel). Dit triviale probleem wordt dan op een continue manier terug in het oorspronkelijke vervormd. Tijdens deze vervorming leggen de oplossingen een continu pad af dat numeriek gevolgd kan worden. Dit principe biedt enkele interessante voordelen : (1) globale convergentie (de convergentie hangt niet af van de nauwkeurigheid van de initiële schatting); (2) in sommige gevallen laten ze toe op een systematische manier meer dan één oplossing te vinden.

Dit soort methoden heeft de jongste decennia een indrukwekkende ontwikkeling gekend in de optimisatie en de econometrie [37]. Het vertrekpunt was het artikel van Scarf in 1967 [88] waarin voor het eerst een constructief bewijs werd gegeven voor de vastpuntstelling van Brouwer, een belangrijke existentiestelling uit de niet-lineaire analyse. Dit werd beschouwd als een belangrijke doorbraak en in de daaropvolgende jaren werd Scarfs algoritme verbeterd en vervangen door efficiëntere methoden. Aan het eind van deze ontwikkeling staan de variabele-dimensiealgoritmen die in dit proefschrift bestudeerd worden [53, 54, 55].

De geschiedenis van deze methoden gaat nog verder terug, eigenlijk tot het simplexalgoritme uit de lineaire programmering. Deze relatie is samengevat in tabel 1. De tabel toont een aantal problemen uit de netwerktheorie, telkens de equivalente wiskundige beschrijving, en het algoritme dat ermee correspondeert. Zoals uit de tabel blijkt, kan een circuit met enkel ideale diodes en onafhankelijke bronnen beschreven worden door een lineair programma [89], en dus met het simplexalgoritme in een eindig aantal stappen opgelost. Lemkes methode werd gepubliceerd in de jaren zestig en kan worden beschouwd als een veralgemening van het simplexalgoritme naar het probleem dat ontstaat wanneer ook weerstanden aan het circuit worden toegevoegd. Dit algoritme is

elektrisch netwerk	*wiskundige beschrijving*	*algoritme*
– onafhankelijke bronnen – ideale diodes	linear programma	simplexalgoritme
– onafhankelijke bronnen – ideale diodes – lineare weerstanden	linear complementariteitsprobleem	Lemkes algoritme
– onafhankelijke bronnen – ideale diodes – niet-lineare weerstanden	niet-lineair complementariteitsprobleem	Lemkes algoritme
– onafhankelijke bronnen – niet-lineaire weerstanden	$\mathbf{f}(\mathbf{x}) = \mathbf{0}$	variabele-dimensiealgoritmen

Tabel 1: Verband tussen enkele methoden uit de optimisatie en problemen uit de elektrische netwerktheorie.

gebaseerd op continue vervorming en het opmerkelijke van de methode lag dan ook in de globale convergentie ervan. Historisch was dit de belangrijkste stimulans om op zoek te gaan naar gelijkaardige, globaal convergente methoden voor het oplossen van stelsels niet-lineaire vergelijkingen. Dit leidde tot het reeds aangehaalde artikel van Scarf in 1967 en de hele ontwikkeling waarvan variabele-dimensiealgoritmen het eindpunt vormen.

1.3 Continue vervorming in de netwerktheorie

Het is de bedoeling variabele-dimensiealgoritmen te combineren met de boven beschreven, kwalitatieve resultaten uit de niet-lineaire netwerktheorie. Zoals zal worden aangetoond, leidt dit enerzijds tot constructieve bewijzen van vele belangrijke resultaten over resistieve netwerken; anderzijds resulteert het ook in praktisch bruikbare methoden voor berekening van oplossingen.

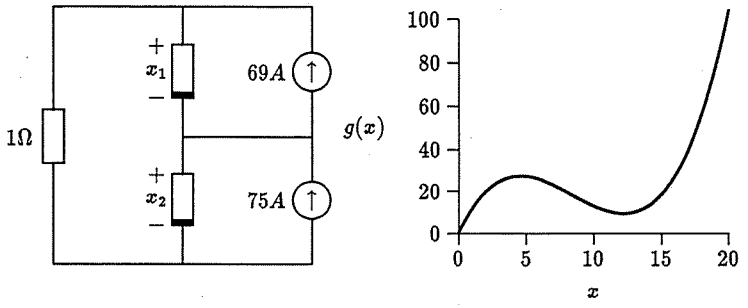
Voor zover ons bekend, worden variabele-dimensiealgoritmen hier voor het eerst op elektrische netwerken toegepast. Wel bestaan er enkele verwante, eveneens op continue vervorming gebaseerde methoden. Twee voorbeelden hiervan zijn *source-stepping* voor het oplossen van niet-lineaire D.C.-vergelijkingen, en de zgn. Newtonhomotopie gedefinieerd door

$$\mathbf{f}(\mathbf{x}) - \lambda \mathbf{f}(\mathbf{x}^0) = \mathbf{0}, \tag{1}$$

waarin $\mathbf{x}^0$ de startwaarde is en λ een parameter in het interval $[0,1]$. De keuze $\lambda = 1$, $\mathbf{x} = \mathbf{x}^0$ is het triviale probleem waarmee het algoritme start. Dan wordt het pad gedefinieerd door (1) verder gevolgd tot $\lambda = 0$. De Newtonhomotopie wordt toegepast in het *switching-parameter algorithm* van Chua en Ushida [16] en in het algoritme van Katzenelson [52, 72].

In sommige opzichten herinneren zowel de Newtonhomotopie als *source-stepping* aan oudere methoden uit de optimisatieliteratuur. Met die methoden hebben ze een aantal nadelen gemeen, die in de optimisatie de ontwikkeling van variabele-dimensiealgoritmen geïnspireerd hebben.

Zo leggen beide methoden toch nog beperkingen op aan het startpunt. Voor *source-stepping* is dat duidelijk, want alleen de oorsprong komt als startpunt in aanmerking. In de Newtonhomotopie legt men vaak als voorwaarde op dat de vergelijkingen *uniform passief* moeten zijn t.o.v. het startpunt $\mathbf{x}^0$, d.w.z. dat $(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}^0))^t.(\mathbf{x} - \mathbf{x}^0) > 0$, voor alle $\mathbf{x} \neq \mathbf{x}^0$. Hiermee kan worden vermeden dat het pad terugkeert naar het niveau $\lambda = 1$ voor het een oplossing heeft bereikt. Deze voorwaarde sluit echter ook bepaalde gebieden uit bij de keuze van het startpunt. Figuur 3 toont daar een voorbeeld van. Met de oorsprong als startpunt convergeert Katzenelsons algoritme zonder problemen naar de oplossing $(x_1, x_2) = (16.7, 17.1)$, maar het komt in een cyclus terecht als men het start in $(x_1, x_2) = (10, 10)$, alhoewel dit punt dichter bij de oplossing ligt dan de oorsprong. Een tweede voorbeeld is het stelsel van twee vergelijkingen



Figuur 3: Voorbeeld uit [72] met twee tunneldiodes en i-v–curven voor de diodes. De netwerkvergelijkingen zijn $g(x_1) + x_1 + x_2 = 69$ en $g(x_2) + x_1 + x_2 = 75$.

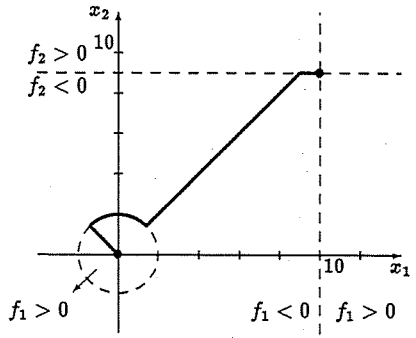
(zie fig. 4)

$$\begin{aligned} f_1(x_1, x_2) &= (x_1^2 + x_2^2 - 4)(x_1 - 10) \\ f_2(x_1, x_2) &= x_2 - 9. \end{aligned} \tag{2}$$

Hier kan het algoritme onmogelijk convergeren als het gestart wordt in de oorsprong, want uit de definitie (1) volgt dat het pad nooit de cirkel kan snijden. De figuur toont ook hoe dit geen probleem vormt voor het octaëdraal algoritme, het variabele-dimensiealgoritme dat in deze tekst vooral gebruikt wordt.

Deze beperkingen op het startpunt verlagen de efficiëntie, omdat de gebruiker meestal toch over een betere schatting van de oplossing beschikt. Het is bijvoorbeeld een aantrekkelijk idee om globaal convergente methoden te combineren met iteratieve methoden, die immers veel sneller zijn. Men zou dan eerst een iteratieve methode kunnen proberen en pas overschakelen op de homotopiemethode wanneer het resultaat van de iteratie begint te divergeren. In stuksgewijs lineaire analyse zou men op een analoge manier kunnen proberen de nauwkeurigheid van de benadering geleidelijk op te voeren door het algoritme enkele keren opnieuw te starten met een steeds fijnere verdeling. Dit kan alleen efficiënt gebeuren met methoden die in elk willekeurig punt kunnen starten, zoals het geval is met variabele-dimensiealgoritmen.

Variabele-dimensiealgoritmen zijn precies ontstaan met de bedoeling de efficiëntie van homotopiemethoden te verbeteren. Ze exploiteren het feit dat enkel het eindpunt van het pad nauwkeurig berekend moet worden, terwijl voor de rest een vrij ruwe benadering volstaat : ten eerste wordt de nauwkeurigheid van de stuksgewijs lineaire benadering verhoogd naarmate men nadert van de oplossing; daarnaast gaat men ook de dimensie van het probleem geleidelijk



Figuur 4: Domein van de functies (2). Pad gevolgd door het octaëdraal algoritme.

opvoeren, door een reeks problemen van lagere orde recursief op te lossen. Dit wordt verduidelijkt in hoofdstuk 2 van de tekst.

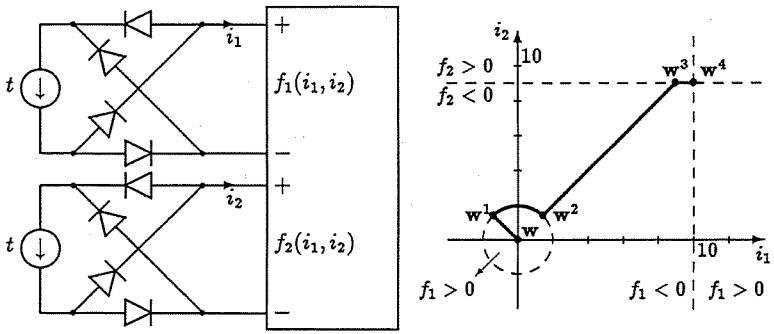
2 Principes en convergentiecriteria

In het tweede hoofdstuk worden de voornaamste variabele-dimensiealgoritmen beschreven aan de hand van een eenvoudig netwerkequivalent. Voor het zogenaamde *octaëdraal algoritme*, ingevoerd door Wright in ref. [128], verloopt dit als volgt. Veronderstellen we dat de afbeelding (2) de beschrijving is van een stroomafhankelijke tweepoort,

$$\begin{aligned} v_1 = f_1(i_1, i_2) &= (i_1^2 + i_2^2 - 4)(i_1 - 10), \\ v_2 = f_2(i_1, i_2) &= i_2 - 9, \end{aligned} \tag{3}$$

dan kan het gevolgde pad geïnterpreteerd worden alsof beide poorten afgesloten zijn met twee hulptakken zoals in fig. 5. Beide takken bestaan uit ideale diodes en onafhankelijke spanningsbronnen met waarde t. Hierin is t een parameter in het interval $[0, \infty)$. Zonder afbreuk aan de algemeenheid wordt hier verondersteld dat het algoritme gestart wordt in de oorsprong. Om te starten in een ander punt $\mathbf{x}^0$ wordt de afbeelding $\mathbf{f}(\cdot)$ gewoon vervangen door $\mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathbf{x} + \mathbf{x}^0)$.

De startwaarde voor t is $t = 0$. Beide takken gedragen zich dan als een open tak en de tweepoort bevindt zich in het punt $(i_1, i_2) = (0, 0)$. Fig. 5 toont de volledige oplossingenverzameling voor $t = [0, \infty)$, in dit geval een continu pad tussen de oorsprong en de oplossing $(i_1, i_2) = (10, 9)$. In het eindpunt geleiden alle diodes en vormen de hulptakken een kortsluiting.



Figuur 5: Het octaëdraal algoritme toegepast op de vergelijkingen (3) : hulptakken en het pad gevolgd door het algoritme.

Dit principe kan gemakkelijk veralgemeend worden naar hogere dimensies $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, $\mathbf{f} : R^n \to R^n$. De afbeelding $\mathbf{f}(\cdot)$ wordt dan geïnterpreteerd als de hybride beschrijving van een resistieve n-poort $\mathbf{y} = \mathbf{f}(\mathbf{x})$. Elke poort wordt afgesloten met een van de hulptakken afgebeeld in fig. 6. De verzameling van alle oplossingen van het op die manier gewijzigde netwerk voor verschillende waarden van t bestaat in het algemeen niet uit één enkele component zoals in bovenstaand voorbeeld. Generisch gelden echter de volgende eigenschappen [54].

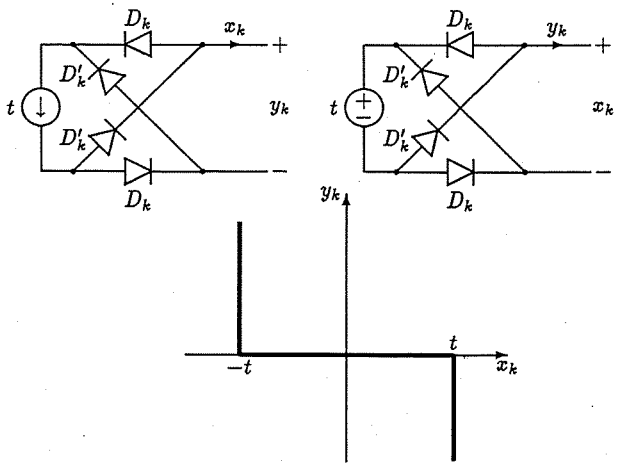
1. De oplossingenverzameling bestaat uit een aantal disjuncte componenten. Elke component is ofwel een lus ofwel een pad.

2. Een pad kan alleen eindpunten hebben van de vorm $(\mathbf{x}, \mathbf{y}) = (\mathbf{0}, \mathbf{f}(\mathbf{0}))$, of $(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^*, \mathbf{0})$ met $\mathbf{x}^*$ een nulpunt van $\mathbf{f}(\cdot)$.

3. Het punt $(\mathbf{x}, \mathbf{y}) = (\mathbf{0}, \mathbf{f}(\mathbf{0}))$ is het eindpunt van een pad.

Het algoritme vertrekt in een punt $(\mathbf{x}, \mathbf{y}) = (\mathbf{0}, \mathbf{f}(\mathbf{0}))$ en volgt het pad door dit punt. Het pad is ofwel onbegrensd en dan divergeert het algoritme ofwel leidt het tot een tweede eindpunt. Wegens de tweede eigenschap moet dit punt een nulpunt zijn van $\mathbf{f}(\cdot)$.

Deze vaststellingen herleiden het bewijzen van de convergentie van de methode tot een bewijs dat de oplossingenverzameling van het netwerk met hulptakken begrensd blijft voor alle waarden van t. Hier blijkt het netwerkequivalent van fig. 6 van veel meer dan louter didactisch nut te zijn. Dank zij deze



Figuur 6: Hulptakken gebruikt voor het afsluiten van de stroomafhankelijke poorten (links) en de spanningsafhankelijke poorten (rechts). Constitutieve relatie voor deze takken.

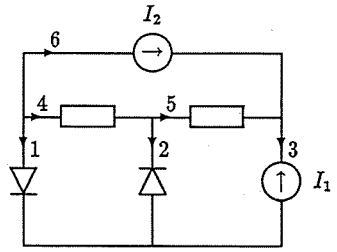
interpretatie kan de convergentie immers worden onderzocht in topologische termen, m.a.w. in functie van de karakteristieken van de componenten en van hun interconnectie. Ook gaat de keuze van de hulptakken minder willekeurig lijken : niet alleen herleiden ze zich voor de extreme waarden van $t = 0$ en $t = \infty$ tot kortsluitingen en open takken, ze gedragen zich ook steeds als passieve elementen, wat essentieel is voor de convergentie.

Figuur 7 illustreert de algemene gedachtengang die bij het bewijzen van convergentie gevolgd wordt.[1] Wanneer we het algoritme toepassen op de netwerkvergelijkingen,

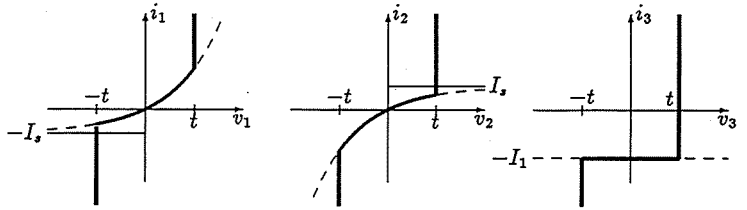
$$\begin{aligned} \hat{\imath}_1(v_1) + \hat{\imath}_4(v_1 - v_2) + I_2 &= 0, \\ \hat{\imath}_2(v_2) - \hat{\imath}_4(v_1 - v_2) + \hat{\imath}_5(v_2 - v_3) &= 0, \\ -I_1 - \hat{\imath}_5(v_2 - v_3) - I_2 &= 0, \end{aligned}$$

[1]Een karakteristiek $f(\rho)$ is

- van type B als $|f(\rho)|$ begrensd is als $|\rho| \to \infty$,
- van type H als ofwel $|f(\rho)|$ begrensd is als $\rho \to +\infty$, en $f(\rho) \to -\infty$ als $\rho \to -\infty$, ofwel $|f(\rho)|$ begrensd is als $\rho \to -\infty$, en $f(\rho) \to +\infty$ als $\rho \to +\infty$,
- van type U als $f(\rho) \to +\infty$ als $\rho \to +\infty$, en $f(\rho) \to -\infty$ als $\rho \to -\infty$.



Figuur 7: Voorbeeld met twee spanningsafhankelijke niet-lineaire weerstanden van type U en met twee exponentiële diodes.



Figuur 8: Wijzigingen van de weerstandskarakteristieken bij toepassing van het octaëdraal algoritme.

dan kunnen we de methode ook interpreteren alsof de karakteristieken van takken 1, 2 en 3 gewijzigd worden als in fig. 8. Met deze interpretatie kan men vrij eenvoudig bewijzen dat alle spanningen en stromen in het netwerk begrensd moeten blijven als $t \to \infty$. De spanningen in de verzameling sneden $\{1, 2, 3\}$ bijvoorbeeld kunnen niet onbeperkt stijgen want $v_1 \to \infty$, $v_2 \to \infty$, $v_3 \to \infty$ is in strijd met de stroomwet van Kirchhoff. Om dezelfde reden is $v_1 \to -\infty$, $v_2 \to -\infty$, $v_3 \to -\infty$ uitgesloten, en door deze gedachtengang te herhalen voor alle fundamentele verzamelingen sneden kan men aantonen dat alle spanningen begrensd moeten zijn. Dat ook alle stromen begrensd zijn, volgt uit een duale redenering. Dit laat niet alleen toe te besluiten dat het octaëdraal algoritme convergeert. Er is meteen ook op een constructieve manier mee bewezen dat er een oplossing bestaat ongeacht de waarde van de parameters in het circuit van fig. 7.

In hoofdstuk 2 worden deze convergentievoorwaarden algemeen geformuleerd en bewezen met behulp van stellingen uit de grafentheorie. De resultaten kunnen als volgt worden samengevat.

In een netwerk met niet-gekoppelde weerstanden, convergeert het algoritme als

- er geen verzamelingen sneden zijn met enkel spanningsafhankelijke takken van type B of van type H, met alle takken van type H in dezelfde zin georiënteerd.

- er geen lussen zijn met enkel stroomafhankelijke takken van type B of van type H met alle takken van type H in dezelfde zin georiënteerd.

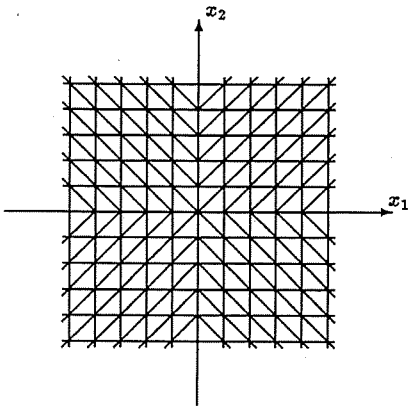
Deze voorwaarden staan in de literatuur bekend als algemene voorwaarden voor het bestaan van oplossingen of, met de nodige aanpassingen en indien de weerstanden monotoon zijn, ook voor uniciteit. Ze worden hier echter wel op een constructieve manier bewezen.

Verder worden in hoofdstuk 2 deze criteria ook uitgebreid naar netwerken met *gekoppelde* weerstanden. Hiertoe hebben we de definities van *type U*, *type H* en *type B* veralgemeend naar gekoppelde weerstanden. Het resultaat is een nieuw, topologisch criterium voor convergentie en dus ook voor bestaan van oplossingen in dergelijke circuits. Ook blijkt dat het algoritme toepasbaar is op de meeste circuits met transistoren, lineaire weerstanden en onafhankelijke bronnen.

Tenslotte worden de voorwaarden onderzocht voor convergentie in netwerken met ideale operationele versterkers, onafhankelijke bronnen en weerstanden van type U. Aangezien operationele versterkers *actieve* elementen zijn, zijn de convergentiecriteria veel strenger. In de literatuur worden operationele versterkers zelfs helemaal niet behandeld bij de bespreking van homotopiemethoden. Ook hier echter hebben we een zuiver topologisch criterium afgeleid, de afwezigheid van een *uniforme partiële oriëntatie*. Geïnterpreteerd als een constructief bewijs voor het bestaan van oplossingen, moet deze stelling gezien worden als een veralgemening van de resultaten van Hasler [45, 46], die immers veronderstelt dat alle weerstanden *monotoon* zijn. Bovendien moet opnieuw benadrukt worden dat de door ons gevolgde weg leidt tot *constructieve* bewijzen.

3 Numerieke implementatie

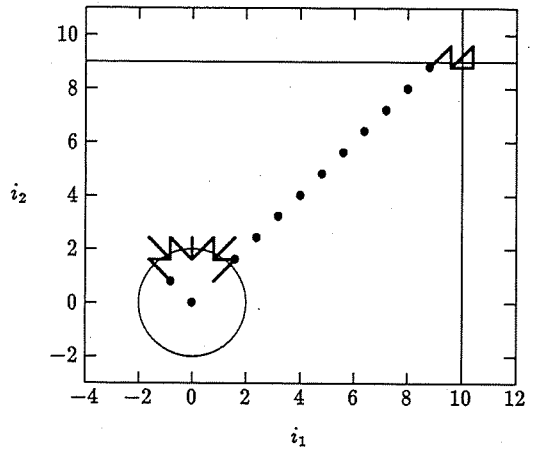
Terwijl in hoofdstuk 2 de nadruk ligt op het nut van het octaëdraal algoritme in constructieve bewijzen van existentie- of uniciteitsstellingen, staat in hoofdstuk 3 de numerieke implementatie centraal. Voor het numeriek volgen van een curve in een hogerdimensionale ruimte, kan men kiezen tussen twee groepen van methoden : klassieke continuatietechnieken en stuksgewijs lineaire methoden. In dit proefschrift wordt de tweede soort gebruikt om hun robuustheid en



Figuur 9: Een simplexverdeling van R^2.

omdat ze geen afgeleiden gebruiken. Ze zijn daarom ongevoelig voor singulariteiten in de Jacobiaanmatrix. Deze stuksgewijs lineaire methoden komen op het volgende neer. Het domein van $\mathbf{f}(\cdot)$ wordt verdeeld in simplexen (cf. fig. 9). Daarna wordt de afbeelding $\mathbf{f}(\cdot)$ stuksgewijs lineair benaderd door ze op elk simplex te vervangen door een affiene benadering. Een pad zoals in fig. 4 kan dan numeriek gevolgd worden door van het ene simplex naar het andere te bewegen, een bewerking die *pivoteren* wordt genoemd. Voor het voorbeeld van fig. 4 leidt dit tot fig. 10. Zonder bijkomende maatregelen kunnen simplextechnieken voor het volgen van curven echter heel traag uitvallen. De belangrijkste oorzaak hiervan is dat het aantal simplexen nodig om een eenheidskubus in R^n te verdelen stijgt als de faculteit van n, en de complexiteit van het algoritme *in het slechtste geval* is navenant. Gelukkig valt de gemiddelde efficiëntie veel beter uit, en kan men ze nog verder verhogen door de volgende maatregelen.

- Men kan een ruwe verdeling gebruiken tijdens de eerste stadia van de berekening, en ze geleidelijk verfijnen naarmate men nadert van de oplossing. In fig. 4 b.v. kan men de nauwkeurigheid van de oplossing verder verhogen door de oorsprong van de simplexverdeling te verplaatsen naar het midden van het eindsimplex, en het algoritme te herstarten met een fijnere roosterafstand. Zoals reeds benadrukt, is dit mogelijk omdat het octaëdraal algoritme in elk willekeurig punt gestart kan worden. Deze techniek vormt ook een verbetering ten opzichte van bijvoorbeeld de klassieke implementatie van Katzenelsons algoritme, waarbij een vaste verdeling door hypervlakken gebruikelijk is.

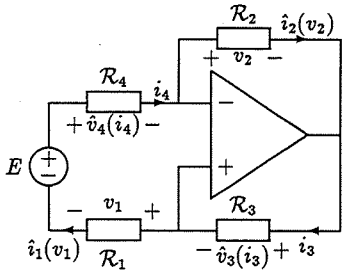


Figuur 10: De benadering van het pad van fig.4.

- De snelheid kan verder verhoogd worden door de dimensie van het probleem te reduceren en het algoritme zo veel mogelijk een lagerdimensionale ruimte te laten volgen. In de verdeling van fig. 9 bijvoorbeeld komt men het snelst vooruit langs de bissectrices in elk kwadrant, een eigenschap die door het octaëdraal algoritme wordt uitgebuit, zoals te zien is in fig. 10. Voor hogere dimensies wordt dit effect nog veel sterker.

- Typische netwerkvergelijkingen zijn sterk gestructureerd door de aanwezigheid van lineaire componenten, en door de combinatorische structuur van de bijhorende graaf. Als een gevolg hiervan is elke veranderlijke slechts met een beperkt aantal andere op een niet-lineaire manier gekoppeld en zal de stuksgewijs lineaire benadering lineair zijn op gebieden die verschillende aanliggende simplexen bevatten. Door elk van deze gebieden in een keer over te steken, kan het aantal pivots sterk gereduceerd worden.

- Het algoritme kan worden versneld door na elke cyclus een aantal quasi-Newtonstappen uit te voeren. Als deze iteratie divergeert vooraleer de gewenste nauwkeurigheid is bereikt, dan voert men een nieuwe cyclus van het octaëdraal algoritme uit, met een fijnere verdeling en met het beste resultaat van de quasi-Newtoniteratie als startpunt. Deze techniek combineert de robuustheid van een homotopiemethode met de asymptotische convergentiesnelheid van een iteratieve methode.

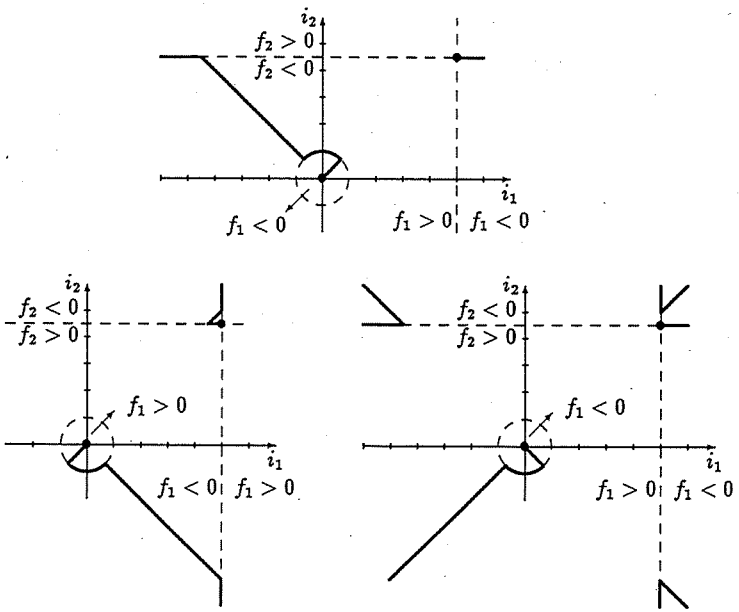


Figuur 11: Netwerk met vier niet-lineaire weerstanden en een ideale operationele versterker.

In hoofdstuk 3 wordt een volledige implementatie beschreven. Verder wordt er een reeks testproblemen opgesomd, die ons in staat stellen het octaëdraal algoritme te vergelijkingen met Katzenelsons algoritme zoals beschreven in [132]. Uit deze vergelijking kunnen we besluiten dat het octaëdraal algoritme sneller en algemener is dan Katzenelsons algoritme, wanneer beide algoritmen dezelfde simplexverdeling gebruiken. Verder wordt in [132] de structuur in de vergelijkingen geëxploiteerd door een rechthoekige in plaats van een simplexverdeling te gebruiken. Vergeleken met de methode uit hoofdstuk 3 is dit wel sneller, maar ook veel minder algemeen bruikbaar. Het is bijvoorbeeld niet toepasbaar op knooppuntsanalyse van transistornetwerken.

4 DC-analyse van actieve netwerken

De convergentievoorwaarden uit hoofdstuk 2 komen steeds neer op een of andere eis van asymptotische passiviteit. Samen met de passiviteit van de toegevoegde hulptakken laat dit toe te besluiten dat de oplossingenverzameling van het gewijzigde netwerk begrensd is en het algoritme convergeert. Die redenering gaat meestal niet op in het geval van actieve elementen zoals operationele versterkers en afhankelijke bronnen, zoals al gebleken is uit de strenge voorwaarden voor netwerken met ideale operationele versterkers in hoofdstuk 2. Het circuit van fig. 11 bijvoorbeeld voldoet niet aan die voorwaarden, maar toch bezit het een goed gedefinieerde oplossing voor de meeste keuzen van de parameters (b.v. als de weerstanden lineair zijn en $R_3R_4 \neq R_1R_2$). In hoofdstuk 5 wordt aangetoond dat een uitbreiding van het octaëdraal algoritme op basis van [119] wel in staat is een ruime klasse van actieve netwerken op te lossen. De redenering is geïllustreerd in fig 12. Hier werd uitgegaan van vergelijking (2), maar met het teken van $f_1(\cdot)$, $f_2(\cdot)$ of van beide omgekeerd, zodat



Figuur 12: Oplossingenverzameling van het gewijzigde netwerk uit figuur 5 als men het teken van $f_1(\cdot)$ of $f_2(\cdot)$ wijzigt.

de afbeelding niet meer aan de convergentievoorwaarden van het octaëdraal algoritme voldoet. We stellen vast dat de oorsprong en de oplossing nu telkens op verschillende takken liggen. Elk van deze takken is wel onbegrensd, maar tegenover elke onbegrensde tak ligt een andere, zodat het algoritme na een 'reflectie' verdergezet kan worden. In hoofdstuk 4 wordt formeel beschreven hoe dit kan gebeuren. De voornaamste bijdrage van dit hoofdstuk ligt in het afleiden van de precieze voorwaarden die men aan het netwerk moet opleggen om het te kunnen oplossen met dit uitgebreide algoritme.

De hoofdstelling (*Theorem 18*) formuleert deze voorwaarden wanneer de vergelijkingen de vorm hebben

$$\mathbf{Ag(x) + Bx + c = 0},$$

waarin $\mathbf{g}(\cdot)$ een diagonale afbeelding is. Opnieuw kan men dit bewijs van convergentie opvatten als een constructief bewijs van oplosbaarheid van deze

vergelijkingen. Als dusdanig vormt het een veralgemening van gelijkaardige stellingen van Sandberg en Willson [87] en Chua en Wang [17] in de volgende opzichten.

- Er wordt rekening gehouden met de sectoren waarin de componenten $g_k(\cdot)$ asymptotisch liggen als ze onbegrensd zijn. Dit verruimt de klasse van problemen waarop de stelling toepasbaar is aanzienlijk.

- Ook volgen enkele interessante resultaten als bijzondere gevallen uit de algemene stelling. Als alle componenten van $\mathbf{g}(\cdot)$ van type B zijn b.v., dan herleidt de voorwaarde zich tot de eis dat $\mathbf{B}$ inverteerbaar moet zijn. Dit dient vergeleken te worden met het octaëdraal algoritme zonder reflecties, dat in dit geval enkel zou convergeren als $\mathbf{B}$ positieve hoofdminoren heeft.

Verder wordt opnieuw het verband gelegd met de topologische criteria voor uniciteit van de oplossing in netwerken met afhankelijke bronnen of operationele versterkers. Voor wat betreft netwerken met afhankelijke bronnen, hebben we de resultaten van Nishi en Chua [69] op belangrijke punten veralgemeend : (1) we houden rekening met niet-lineaire afhankelijke bronnen, (2) niet-monotone karakteristieken, en (3) boven- en ondergrenzen op alle parameters. Bovendien bewijzen we ook hier eerst *existentie* op een constructieve manier, en leiden hieruit voorwaarden voor *uniciteit* af. Als een speciaal geval vinden we daarom, maar op een veel kortere manier bewezen, de stelling uit [69] terug. Op een gelijkaardige manier vormen de resultaten over netwerken met operationele versterkers een uitbreiding van [45, 46].

Tenslotte wordt erop gewezen dat de uitbreiding van het octaëdraal algoritme met reflecties ook gebruikt kan worden voor het opsporen van meer dan één oplossing.

5 Niet-lineaire netwerken en het complementariteitsprobleem

De bedoeling van dit hoofdstuk is tweevoudig. Enerzijds willen we erin duidelijk maken hoe variabele-dimensiealgoritmen gegroeid zijn uit Lemkes algoritme voor het lineaire complementariteitsprobleem, dat dan weer zelf zijn wortels heeft in het simplexalgoritme voor lineaire programma's.

Anderzijds wordt er ook beschreven hoe een veralgemeende vorm van het lineaire complementariteitsprobleem gebruikt kan worden voor het modelleren van stuksgewijs lineaire netwerken. Met het combinatorische algoritme uit [25] kunnen dan alle oplossingen berekend worden, zelfs wanneer ze continue of onbegrensde verzamelingen vormen.

Dit geeft ons ook de gelegenheid om een vergelijking te maken tussen twee verschillende werkwijzen om niet-lineaire netwerken op te lossen. Aan de ene kant staat het soort combinatorische methoden als in [25], die uitgaan van een stuksgewijs lineair model, en dan door een al dan niet impliciete opsomming van de gebieden erin slagen alle oplossingen te vinden. Aan de andere kant heeft men de homotopiemethoden zoals het octaëdraal algoritme, die alleen convergeren onder bepaalde globale voorwaarden, en ook geen garantie bieden alle oplossingen te zullen vinden. Combinatorische methoden zijn algemener omdat ze altijd toepasbaar zijn en alle oplossingen detecteren. Wegens de combinatorische explosie van de complexiteit zijn ze echter ook steeds beperkt tot problemen van bescheiden grootte. Ook mag men niet uit het oog verliezen dat deze methoden werken met een vaste stuksgewijs lineaire benadering die omwille van de efficiëntie dan nog vrij grof zal moeten zijn. Al vindt men dan ook alle oplossingen van het stuksgewijs lineaire netwerk, het is nooit uitgesloten dat men oplossingen van het oorspronkelijke probleem over het hoofd heeft gezien.

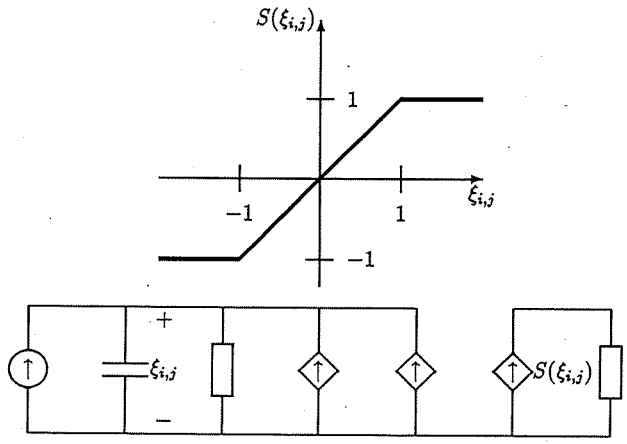
6 Toepassing op neurale netwerken

De toepassingen van variabele-dimensiealgoritmen en verwante methoden voor het complementariteitsprobleem zijn niet beperkt tot resistieve elektrische circuits. In hoofdstukken 6 en 7 worden daarom twee andere toepassingen besproken. Hoofdstuk 6 handelt over neurale netwerken en meer bepaald over de *cellulaire neurale netwerken* ingevoerd in [18, 19]. Deze systemen bestaan uit een groot aantal in een regelmatig rooster geplaatste cellen die lokaal met mekaar verbonden worden. Elke cel kan worden beschouwd als een klein niet-lineair netwerk bestaande uit een lineaire condensator, een niet-lineaire spanningsafhankelijke stroombron, en enkele lineaire resistieve elementen (zie fig. 13). De bedoeling is deze netwerken aan te wenden voor snelle parallelle signaalverwerking.

Het grootste deel van hoofdstuk 6 is gewijd aan een methode om evenwichtspunten te berekenen in cellulaire neurale netwerken. Dit is van belang omdat de goede werking van deze systemen gebaseerd is op niet-lineaire dynamica en op de aanwezigheid van verschillende evenwichtspunten. De evenwichtsvergelijkingen van een netwerk beschreven door

$$d\xi/dt = -\xi + \mathbf{A}S(\xi) + \mathbf{B}u + \chi, \qquad (4)$$

kunnen gemodelleerd worden als een zogenaamd lineair complementariteitsprobleem met boven- en ondergrenzen. Het algoritme uit hoofdstuk 6 is gebaseerd op dat van referentie [121], maar met een nieuwe uitbreiding voor het berekenen van verschillende oplossingen. Figuur 14 toont een voorbeeld. Hier komen

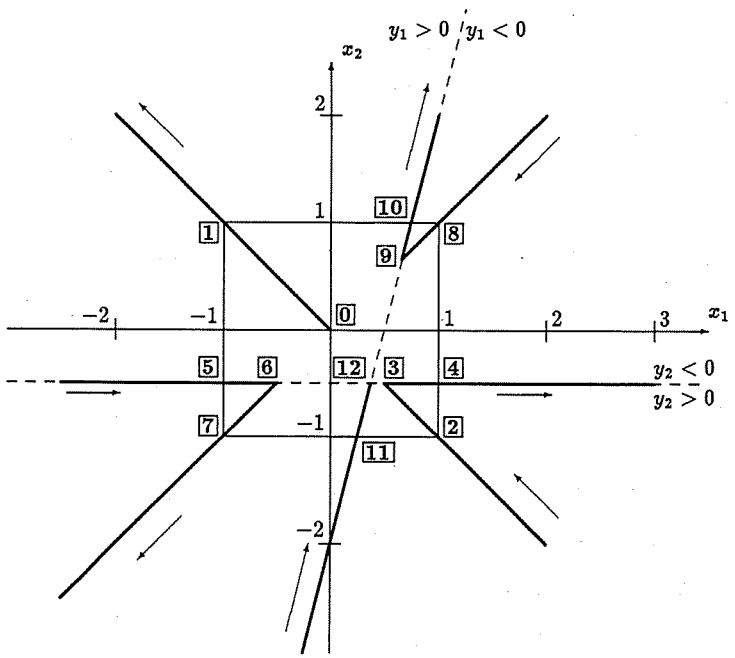


Figuur 13: Equivalent netwerk voor een cel in een cellulair neuraal netwerk. De afhankelijke bronnen in het linkerdeel verbinden de cel met zijn buren en met de externe ingang.

x_1 en x_2 overeen met de uitgangen van de niet-lineariteiten $S(\cdot)$. Het algoritme volgt een stuksgewijs lineair pad dat in de oorsprong vertrekt en gereflecteerd wordt telkens wanneer het divergeert. De evenwichtspunten vinden we in de intersectie van het pad met de rand van de eenheidskubus en eventueel ook in het eindpunt, op voorwaarde dat het binnen de kubus ligt. In dit voorbeeld vindt het algoritme alle negen oplossingen, wat in het algemeen echter niet gegarandeerd is. In hoofdstuk 6 worden verder uit dit algoritme enkele eigenschappen afgeleid over het verband tussen de coëfficiënten van het netwerk en het aantal oplossingen.

Verder bevat het hoofdstuk twee delen die min of meer los staan van de rest van het proefschrift. Een eerste stuk handelt over de dynamica van (4) wanneer de matrix $\mathbf{A}$ asymmetrisch is. Stabiliteit is dan niet meer gewaarborgd, in tegenstelling tot het symmetrische geval, waar men een eenvoudige kwadratische energiefunctie kan definiëren. In hoofdstuk 6 wordt bewezen dat de vergelijking (4) stabiel is als de diagonaalelementen van $\mathbf{A}$ positief zijn, en de matrix $\mathbf{I} - \mathbf{A}$ bovendien quasidominant is.

Tenslotte wordt een systematische manier beschreven om uitgaande van de ontwerpspecificaties de coëfficiënten van het netwerk te bepalen.



Figuur 14: Voorbeeld.

7 Harmonische balans

De regimeoplossing van niet-lineaire systemen kan berekend worden door de evenwichtsvergelijkingen te schrijven in het frequentiedomein en dit stelsel algebraïsche vergelijkingen, de zogenaamde harmonische balansvergelijkingen, op te lossen. Voor het oplossen van dit stelsel zijn in de literatuur allerhande methoden voorgesteld, waaronder ook globaal convergente methoden als *source stepping*. Het is daarom interessant te onderzoeken of variabeledimensiealgoritmen hiervoor ook bruikbaar zijn. In hoofdstuk 7 wordt een algoritme voorgesteld dat de structuur in de vergelijkingen exploiteert. De methode kan de bijdragen van verschillende orde na mekaar behandelen, beginnend met de fundamentele frequenties en de D.C.-bijdrage.

8 Besluit

Tijdens de jaren zeventig en tachtig is in de optimisatie en de econometrie een reeks efficiënte homotopiemethoden voor het oplossen van stelsels niet-lineaire vergelijkingen ontwikkeld, die superieur zijn aan gelijkaardige technieken in de netwerktheorie (Katzenelsons algoritme, source-stepping, e.a.). Ongeveer gelijktijdig is een kwalititatieve theorie voor niet-lineaire netwerken ontstaan, die op basis van de topologie van een circuit een uitspraak tracht te doen over bestaan of uniciteit van oplossingen. Als voornaamste bijdrage van dit proefschrift beschouwen we deze beide ontwikkelingen met mekaar gecombineerd te hebben en de implicaties ervan gedetailleerd te hebben uitgewerkt. Op meer specifieke bijdragen wordt aan het eind van elk hoofdstuk gewezen.

List of symbols

Mathematical notation and definitions

- Matrices and vectors are typed in boldface. If $\mathbf{A}$ is an $m \times n$-matrix, and if $\alpha \subseteq \{1,\ldots,m\}$ and $\beta \subseteq \{1,\ldots,n\}$, then $\mathbf{A}_{\alpha,\beta}$ denotes the matrix formed by the rows indexed by α and the columns indexed by β. Similarly $\mathbf{A}_{\cdot,\beta}$ and $\mathbf{A}_{\alpha,\cdot}$ denote the matrix formed by the columns indexed by β and the the matrix formed by the rows indexed by α, respectively.
  The transpose of a matrix $\mathbf{A}$ is written as $\mathbf{A}^t$.

- $\mathbf{e}^i$ is the i-th unit vector, $\mathbf{e}$ is the vector consisting of all ones, $\mathbf{I}$ is the identity matrix.

- $\|\mathbf{x}\|_\infty = \max_k |x_k|$ denotes the infinity norm of a vector $\mathbf{x}$.

- R is the set of real numbers, R^n the space of real n-vectors.

- $\mathbf{f}'$ is the Jacobian matrix of a mapping $\mathbf{f}(\cdot)$. It will always be clear from the context at which point the derivatives are taken.

- cvx σ is the convex hull of a set σ (cf. page 179)

- $\partial\sigma$ is the boundary of a set σ.

- If $\mathbf{s}$ is a sign vector (cf. p. 23), then $|\mathbf{s}|$ is the number of zero elements in $\mathbf{s}$. For all other vectors, $|\mathbf{x}|$ is the vector formed from $\mathbf{x}$ by taking the absolute value of each component.

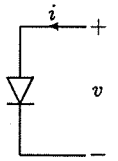
- Functions of *type U*, *type B*, *type H*, *type* H^+ and *type* H^- : cf. pp. 40, 94.

- P-matrix (matrix with positive principal minors), P_0-matrix (matrix with nonnegative principal minors), $\mathcal{W}_0$-matrix pairs : cf. p. 28, p. 94-95.

Circuit elements

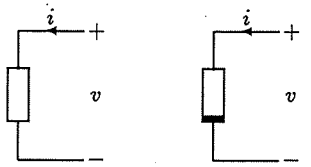
We include here a list of the most important symbols in order to avoid possible confusion. We also hope that this list will make the text more readable for people without engineering background.

Ideal diode : $i \geq 0,\ v \leq 0,\ i.v = 0.$
Exponential diode : $i = I_s(e^{v/v_t} - 1).$



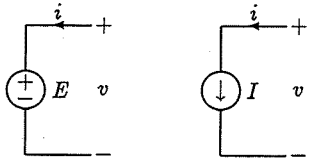
Represents an ideal diode unless explicitly stated.

Voltage-controlled resistor : $i = f(v).$
Current-controlled resistor : $v = g(i).$

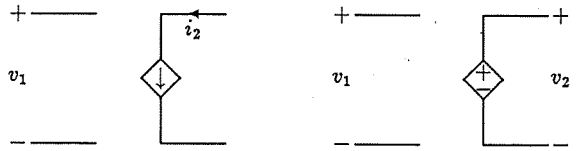


The first symbol represents either a linear resistor, or a general resistive element (linear or nonlinear).
The second symbol always represents a nonlinear resistor.

Independent voltage source : $v = E.$
Independent current source : $i = I.$



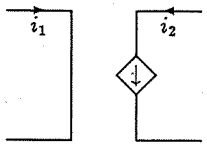
Voltage-controlled current source (VCCS) : $i_2 = f(v_1)$.
Voltage-controlled voltage source (VCVS) : $v_2 = g(v_1)$.



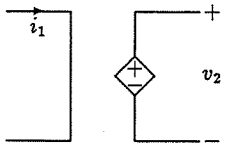
VCCS

VCVS

Current-controlled current source (CCCS) : $i_2 = f(i_1)$.
Current-controlled voltage source (CCVS) : $v_2 = g(i_1)$.

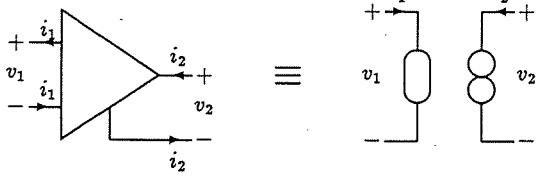


CCCS



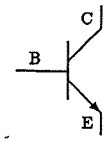
CCVS

Ideal operational amplifier : $i_1 = v_1 = 0$, i_2, v_2 arbitrary.
Nullator : $i_1 = v_1 = 0$.
Norator : i_2, v_2 arbitrary.

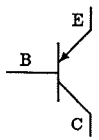


≡

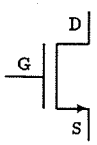
Transistors : cf. p. 48-49.



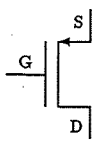
npn



pnp



nMOS



pMOS

Chapter 1

Introduction

1.1 Qualitative theory of nonlinear resistive circuits

During the past two decades the emphasis in circuit theory has gradually shifted from linear to nonlinear circuits. The interest in nonlinear circuits has obvious reasons. Every physical circuit is in fact nonlinear, and even circuits designed for linear operations, will behave nonlinearly in a wider range of excitation. Moreover, many essential functions (oscillation, modulation and demodulation, frequency multiplying, logic gates, etc.) can only be carried out using nonlinear elements. During these two decades a new field has emerged that aims at laying the theoretical foundations for an understanding of the qualitative behaviour of nonlinear networks. One of the most impressive achievements in this area was the establishment of a general qualitative theory for nonlinear resistive circuits. Building on earlier results [56, 127] a conclusive answer has been given to many important questions concerning existence and uniqueness of solutions in resistive circuits. As a typical example, Nielsen and Willson formulated in 1980 the necessary and sufficient conditions for a bipolar transistor circuit to have a unique D.C.-solution [68]. Applied to the circuit in fig. 1.1 [13], their result says that the circuit has a unique solution, because it is impossible to reduce it, by a sequence of operations described in the paper, to the feedback structure of two transistors. This type of topological criteria has later been formulated for circuits with controlled sources [69], ideal operational amplifiers [45] and negative resistors [71].

The results in this area have two basic features in common. First, they all aim at deciding about the uniqueness or existence of a solution on purely *topological* grounds. This forms a departure from the more traditional 'black

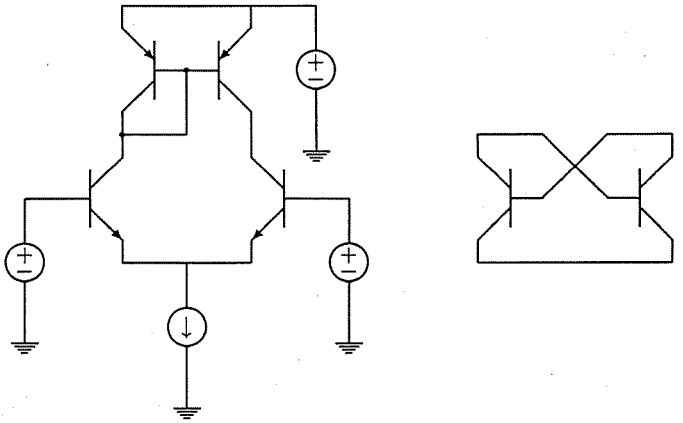


Figure 1.1: Bipolar circuit with a unique solution. Feedback structure of two transistors.

box'-approach of first writing circuit equations as $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ and then trying to invoke general theorems from nonlinear analysis. Only the exploitation of the very specific structure in the electrical circuit equations, imposed by the underlying graph and reflected in the linear and combinatorial nature of Kirchhoff's laws, could lead to results as powerful as those cited above.

A second striking fact in this development is that none of the results is proved constructively. The theorems assert existence or uniqueness of a solution but give no way to find it. This inevitably limits their applicability. For practical calculations one still has to resort to conventional iterative methods which may diverge even if a solution can be proved to exist.

One of the main conclusions of this text will be that the impact of the recent developments in nonlinear circuit theory is actually much larger and that a close connection exists between these concepts and the problem of numerically solving the circuit equations. Fig. 1.2 shows a simple example of how the circuit-theoretical concept of passivity can help to resolve problems in conventional methods for circuit simulation. If applied to the D.C.-equations of the circuit, the Newton-Raphson method will fail when started at v^0 due to the singularity of the Jacobian matrix. Strictly mathematically and with nothing but local information (function values and derivatives) about the voltage-current

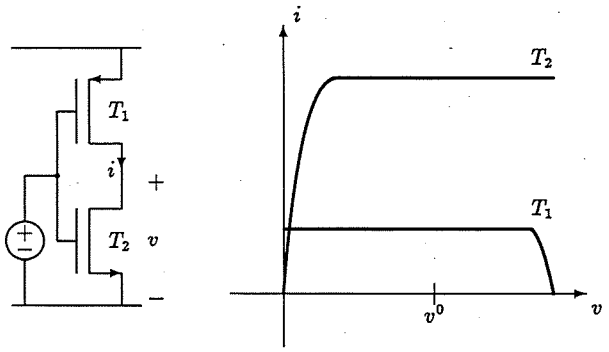


Figure 1.2: CMOS-inverter.

relations, there is indeed no reason why a solution should be located to the left of v^0 rather than to the right. If one takes the passivity of both transistors into account however, it becomes clear that a solution must be searched for to the left, even if the voltage-current relations might be non-monotonic : the current through T_2 must become negative if v goes through zero, while the current through T_1 must remain positive, so that at least one intersection must lie between $v = 0$ and $v = v^0$. A major part of this thesis deals with the generalization of this type of reasoning to higher dimensions.

1.2 The continuous deformation principle

The central idea in this thesis is the *continuous deformation principle.* It is a general idea applicable in many areas of nonlinear analysis and known in differential topology as the *homotopy principle.* Applied to a set of algebraic equations $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, it can be formulated as follows. First replace the difficult problem $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ by a trivial one, which is easy to solve (for example a linear system). Then continuously deform the trivial system into the original one and, during the deformation, trace the path followed by the solution numerically. Several examples of this idea will be encountered throughout the text. Continuous deformation methods have two important advantages. Unlike conventional iterative methods, their convergence does not depend on the proximity of the starting point to the actual solution. This is why they are called *globally convergent.* As a second asset, these methods can often be modified to enumerate several solutions in a systematic way.

Continuous deformation was a very popular research topic in mathematical programming during the seventies and early eighties. This activity was sparked off in 1967 by the publication of the first constructive proof of Brouwer's fixed point theorem [88]. This theorem states that a continuous function mapping a closed ball into itself has at least one fixed point (a point that satisfies $\mathbf{f}(\mathbf{x}) = \mathbf{x}$). Brouwer's theorem had always been an important tool in mathematical programming and operations research for proving existence of economic equilibrium points. Scarf's paper in 1967 made it possible not only to predict the existence of solutions to nonlinear problems, but also to compute them.

This breakthrough did not remain unnoticed, and during the years following Scarf's paper his work was extended and continued in many directions. At the theoretical side the link was made with concepts from differential topology such as homotopy, index and degree of a mapping [32]. At the numerical side, the fixed-point algorithm was quickly modified for related problems, and superseded by more practical and efficient methods. The reader is referred to [4, 37] for a survey of the evolution. Within this development, a class of algorithms called *variable dimension methods* appeared between 1980 and 1985 [53, 54, 55]. These methods are believed to rank among the most efficient and general continuous deformation algorithms and will be in the focus of attention of this thesis.

The main purpose of the work reported here, is to combine these algorithms with the qualitative results in circuit theory cited in the previous section. This leads to constructive proofs of most of the important results in the qualitative theory of nonlinear resistive circuits, and on the other hand to practical globally convergent algorithms for solving circuit equations.

1.3 The D.C.-operating point problem

Continuous deformation has also been around in circuit theory under several forms [52, 72, 16, 97, 98]. The best known example is *source stepping* to find D.C.-operating points in nonlinear circuits. Solving D.C. equations is a basic step in circuit simulation, even in transient analysis. It is well known that conventional iterative methods for solving nonlinear equations, such as the Newton-Raphson method, may have convergence problems if the initial guess is not sufficiently close to the solution. In a network with bipolar transistors and linear resistors for example, global convergence of Newton's method has been proved only in cases with just one transistor [78].

As an illustration of source stepping, consider the example of fig. 1.3 [22]. One starts with all independent sources turned off. This is a trivial problem

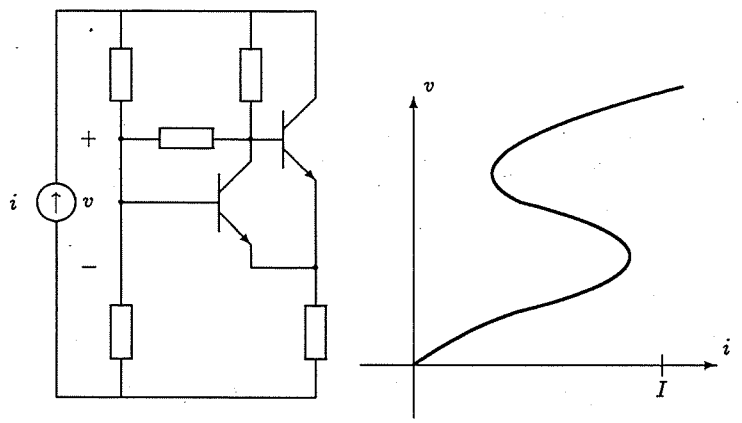


Figure 1.3: Example of source-stepping to find a solution with $i = I$.

since all node voltages will be zero. The source values are then adjusted and the solution vector computed during the process. A rudimentary algorithm would just increase the source values in fixed steps, and compute the solution at each step by the Newton-Raphson method using the previous result as starting point. In the example of the figure this would fail because of the turning point. Hence a more sophisticated path-following technique is needed.

In some respect, source stepping resembles the early homotopy methods in mathematical programming. One of its inconvenient properties is that it always starts at the origin, even if the user has a better guess of the solution. This is a source of inefficiency because the path will often be unnecessarily long. It also makes it difficult to combine source-stepping with iterative methods, by first trying an iterative method and then, in case of divergence, switching to source-stepping. In piecewise-linear analysis it is common practice to run the same algorithm several times with gradually finer subdivisions, and each time using the solution of one run as starting point for the next. This is why this type of restrictions on the starting point have been called *restart problems*.

A second inefficiency in source-stepping is that the algorithm solves a sequence of circuits with needlessly high accuracy. In path-following methods for solving nonlinear equations, one is only interested in an accurate estimate of the endpoint, while for the intermediate points a very rough approximation

is sufficient.

Variable dimension algorithms have been developed in order to overcome similar problems of earlier homotopy methods in mathematical programming. They have the capability of starting at every point, which makes it easy to restart the algorithm or to combine it with fast iterative methods. As a second advantage they start with a very rough approximation of the problem, not only by making a coarse piecewise-linear model, but also by approximating the problem by a lower dimensional one, as will become clear in the sequel.

Restart problems can also be encountered by the *Newton-homotopy*, another continuous deformation method well known in circuit theory : it has been used in the switching parameter algorithm of [16], its piecewise-linear version coincides with the (generalized) Katzenelson algorithm [52, 72] and it can also be shown to be equivalent to the global Newton method [35, 48]. Assume that the problem is to solve the system $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ where $\mathbf{f}(\cdot)$, a map from R^n to R^n, is given. If $\mathbf{x}^0$ is the initial guess, then the Newton-homotopy consists in continuously deforming the equations as

$$\mathbf{f}(\mathbf{x}) - \lambda \mathbf{f}(\mathbf{x}^0) = \mathbf{0}, \tag{1.1}$$

where λ is an extra variable lying between 0 and 1. The algorithm starts from the point $\mathbf{x} = \mathbf{x}^0$, $\lambda = 1$, and traces the path through this point, hoping that it will eventually hit the level $\lambda = 0$. Global convergence therefore requires two assumptions [72].

- A *coercivity assumption*, $\|\mathbf{f}(\mathbf{x})\| \to \infty$ if $\|\mathbf{x}\| \to \infty$, is sufficient to rule out divergence ($\|\mathbf{x}\| \to \infty$ before reaching $\lambda = 0$).

- If the initial point satisfies the condition that there is no second point $\mathbf{x}$ with $\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}^0)$, then the path cannot fold back to the level $\lambda = 1$. A typical condition is to require *uniform passivity* of the equations w.r.t. the starting point i.e., $(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}^0))^t(\mathbf{x} - \mathbf{x}^0) > 0$, $\forall \mathbf{x} \neq \mathbf{x}^0$.

This second assumption, however, relies on local properties and typically requires choosing a point $\mathbf{x}^0$ outside the region where the zeros are likely to lie.[1] The example of figure 1.4 illustrates this problem [132]. With the origin as starting point, the generalized Katzenelson algorithm finds the solution $(x_1, x_2) = (16.7, 17.1)$, but it cycles when started from $(10, 10)$ although this point lies closer to the solution than the origin.

[1] In piecewise-linear analysis for example, one chooses a starting point in an unbounded cell which contains a half-ray of the homotopy path. This excludes cycling.

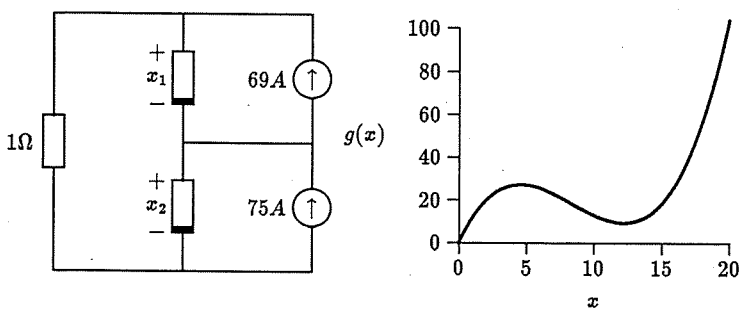


Figure 1.4: Twin tunnel-diode circuit from [72] and constitutive relation of the diodes. The circuit equations are $g(x_1)+x_1+x_2 = 69$ and $g(x_2)+x_1+x_2 = 75$.

As another example, take the equations $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ with (see fig. 1.5)

$$\begin{aligned} f_1(x_1,x_2) &= (x_1^2+x_2^2-4)(x_1-10) \\ f_2(x_1,x_2) &= x_2-9. \end{aligned} \tag{1.2}$$

Started at the origin, the Newton-homotopy will cycle, because the path defined by (1.1) can never cross the circle in fig. 1.5. The figure also shows how the *octahedral algorithm*, an algorithm that will be discussed in Chapter 2 and that can be started from an arbitrary point, succeeds in finding the solution $(x_1, x_2) = (10, 9)$.

1.4 Piecewise-linear techniques in mathematical programming and numerical analysis

The implementation of the idea of continuous deformation requires the availability of numerical techniques for following a nonlinear path. One can distinguish two classes of methods for this purpose : homotopy continuation methods [125] and piecewise-linear methods [4, 92]. A thorough comparison of both classes is not available in the literature, although one generally holds piecewise-linear methods to be more robust but slower than homotopy continuation methods. If properly implemented however, the difference in speed should be small [4], and, in addition, piecewise-linear methods have the advantage of not requiring derivatives.

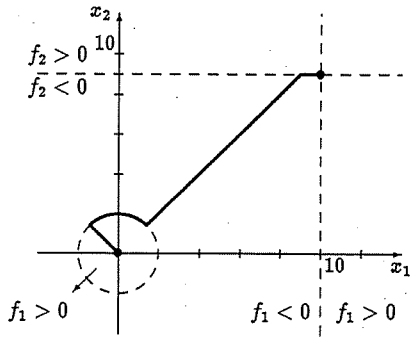


Figure 1.5: Zero levels of the functions (1.2). Path traced by the octahedral algorithm (bold line).

The technique used in this text is *simplicial pivoting* and belongs to the latter category. Assume that one wants to trace a path in R^n, implicitly defined as $\mathbf{g}^{-1}(\mathbf{0})$, with $\mathbf{g}(\cdot)$ a mapping from R^n to R^{n-1}. The basic idea in simplicial techniques is to triangulate R^n with simplices (see fig. 1.6) and to approximate the path by a piecewise-linear one defined as $\mathbf{G}^{-1}(\mathbf{0})$. Here $\mathbf{G}(\cdot)$ is the piecewise-linear mapping obtained from $\mathbf{g}(\cdot)$ by identifying $\mathbf{G}(\cdot)$ and $\mathbf{g}(\cdot)$ on the vertices of the triangulation and extending $\mathbf{G}(\cdot)$ affinely on each simplex. The path $\mathbf{G}^{-1}(\mathbf{0})$ can then be traced by moving from one simplex to an adjacent one in a process called *pivoting* (see fig. 1.7).

A straightforward implementation of simplicial techniques can be very inefficient, because the number of simplices (with unit size) needed to triangulate a unit cube in R^n grows as $n!$. In a worst case a pivoting algorithm would have to pass through all simplices before a solution is found, and the complexity would grow as $n!$. Fortunately it can be shown that in the average case there is no such explosion of complexity [79]. In addition, several measures can be taken to still improve the efficiency :

- One can work with a coarse subdivision in the first stages of the algorithm, and then gradually refine it. As mentioned above, this is the reason for the importance of restart methods.

- One can try to reduce the dimension of the problem and confine the path as long as possible to a lower-dimensional space. This will of course make the matrix manipulations easier, but it can also contribute to reduce the

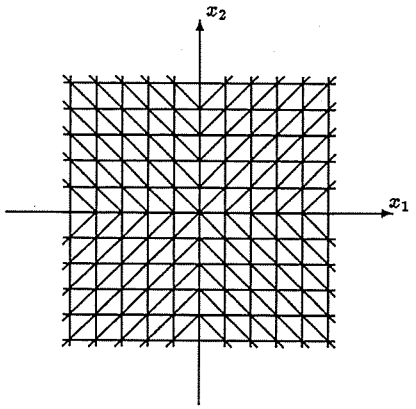


Figure 1.6: A triangulation of R^2 (the reflected Freudenthal triangulation).

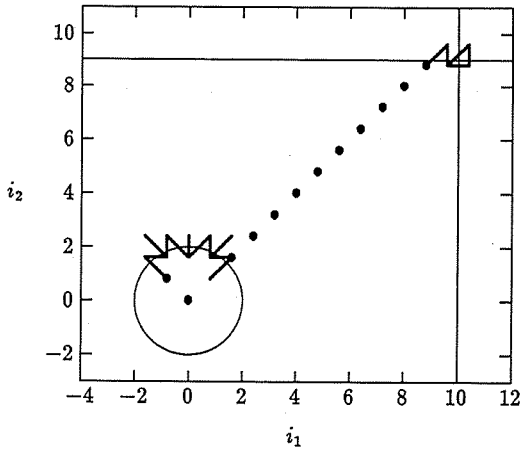


Figure 1.7: The approximation of the path shown in fig.1.5. The figure shows the entrance and the exit facets of the simplices of fig.1.6 through which the path passes.

number of pivots. In the triangulation of fig. 1.6 the directions in which one can make fastest progress are the diagonal rays emanating from the origin. This effect is exploited by the octahedral algorithm as can already be observed in the two-dimensional example of fig. 1.7, and its importance grows rapidly with growing dimension.

- Practical circuit equations are highly structured as a result of which the path $\mathbf{G}^{-1}(\mathbf{0})$ will actually be linear on the union of several adjacent simplices. These pieces of linearity can be traversed at little more than the cost of one single pivot.

- It is possible to accelerate the algorithm by combining it with a classical iterative method.

In this text the implementation of one particular representative of the class of variable dimension algorithms, called the *octahedral algorithm*, will be described. It will be shown how the incorporation of the above ideas leads to an efficient algorithm that combines the asymptotic speed of an iterative (quasi-Newton) method with the robustness of a homotopy algorithm. This typically leads to convergence as in fig. 1.8. The equations for this example are taken from reference [12] :

$$\begin{aligned} f_1(\mathbf{x}) &= 0.1x_1^3 - x_1^2 + 3x_1 - x_2 - x_3 - 2, \\ f_2(\mathbf{x}) &= 0.1x_2^3 - x_2^2 + 3x_2 - x_1 - x_4, \\ f_3(\mathbf{x}) &= x_3^3 + 2x_3^2 - x_2 - x_4 - 2, \\ f_4(\mathbf{x}) &= x_4^3 + 2x_4^2 - x_1 - x_3. \end{aligned}$$

Because of an unfortunate choice of the starting point ($x_1 = -8.33$, $x_2 = -8.33$, $x_3 = 1.17$, $x_4 = 8.33$), the convergence in fig 1.8 is slow until a neighbourhood is entered where the quasi-Newton steps converge. Note also that the error need not decrease monotonically, a feature typical for continuous deformation methods.

1.5 Piecewise-linear analysis in circuit theory

Piecewise-linear techniques are well known in circuit theory, be it in a form different from the simplicial algorithms in mathematical programming. The two favourite piecewise-linear models in circuit theory are piecewise-linear mappings on a subdivision by hyperplanes, and the linear complementarity problem.

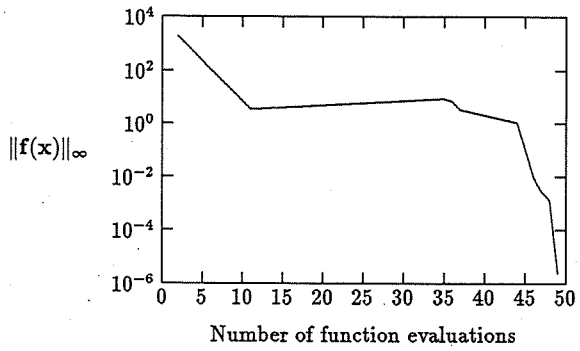


Figure 1.8: Typical convergence behaviour.

- *Subdivisions by hyperplanes* have mainly been used in connection with Katzenelson's algorithm [52, 72] or with algorithms to find all solutions of a set of equations [20, 50]. Here the domain of the mapping $\mathbf{f}(\cdot)$ is divided in polyhedral cells by a number of hyperplanes. On each cell C_i the mapping $\mathbf{f}(\cdot)$ is identified with an affine mapping : $\mathbf{x} \in C_i \Rightarrow \mathbf{f}(\mathbf{x}) = \mathbf{A}_i\mathbf{x} + \mathbf{b}_i$. To reduce the amount of data needed to represent the model, a canonical representation can be used as described in [21].

- In the *linear complementarity problem* the piecewise-linear mapping is modelled as

$$\mathbf{f}(\mathbf{x}) = \mathbf{M}\mathbf{x}^+ + \mathbf{N}\mathbf{x}^- + \mathbf{q},$$

  where $\mathbf{x}^+$ and $\mathbf{x}^-$ are defined as

$$x_k^+ = \max(0, x_k), \quad x_k^- = \max(0, -x_k), \quad k = 1, \ldots, n.$$

  The classical path-following algorithm for solving this type of equations is *Lemke's algorithm* [90, 99]. Iterative methods have been given in [99], and a method to find all the solutions in this thesis work (see also [100]).

These approaches to use piecewise-linear techniques for solving nonlinear equations have in common that the piecewise-linear model is fixed once and for all before the start of the algorithm. This implies that one must build an accurate model on the entire domain of the mapping. Simplicial techniques are much more flexible since they combine the modelling of the equations and the path-following process in one single algorithm. This combination should certainly be more efficient : one need not make a model in the regions of the

electrical circuit	*mathematical description*	*algorithm*
– independent sources – ideal diodes	linear program	simplex algorithm
– independent sources – ideal diodes – linear resistors	linear complementarity problem	Lemke's algorithm
– independent sources – ideal diodes – nonlinear resistors	nonlinear complementarity problem	Lemke's algorithm
– independent sources – nonlinear resistors	$\mathbf{f}(\mathbf{x}) = \mathbf{0}$	variable dimension algorithms

Table 1.1: Connection between several methods from mathematical programming and electrical circuit problems.

domain where the path does not pass through, and one can vary the accuracy according to the distance to the solution.

It is therefore remarkable that simplicial techniques hardly received any attention in circuit theory, with the exception of [12] and, more recently, the work of Yamamura and Horiuchi [132].

1.6 Connection with linear programming

The development of simplicial homotopy algorithms in the seventies, following the paper by Scarf, had its roots in the simplex algorithm for linear programming. The relationship is summarized in Table 1.1. It shows a number of problems in electrical circuit theory, the corresponding mathematical models and algorithms.

The main virtue of the simplex algorithm, at least theoretically speaking and disregarding its remarkable average case efficiency, is its finite convergence. The algorithm can be applied for solving electrical circuits containing ideal diodes and independent sources, since mathematically this problem is equivalent to a linear program [89].

Lemke's algorithm can be seen as an attempt to extend the simplex algo-

rithm for the more general problem that arises when linear resistors are added to the circuit. The circuit equations then take the form of a linear complementarity problem, and Lemke's algorithm offers finite convergence under certain convergence conditions.

Lemke's algorithm can easily be extended to the nonlinear case. The piecewise-linear path followed in Lemke's algorithm is then replaced by a nonlinear curve that can be traced by any of the available path-following methods (for instance by a simplicial method). Instead of *finite convergence* it is more appropriate to speak about *global convergence* in the nonlinear case.

The reason for placing variable dimension algorithms at the bottom of the table will become clear in Chapter 2, when a circuit-theoretical interpretation of the methods will be given. One possible interpretation of these algorithms is that they convert the problem $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ into a nonlinear complementarity problem by adding ideal diodes to the circuit and then apply Lemke's algorithm. This again results in a globally convergent algorithm. This is explained in more detail in Chapter 5.

The analogy between problems in mathematical programming and circuit theory has often been inspiring in both directions. For instance, we have already cited the application of Lemke's algorithm to the solution of piecewise-linear circuits in [90, 99]. Conversely, electrical circuits could be used as analog computers for solving optimization problems. This was first proposed by Dennis in 1959 [27], and later extended by Stern [89] for linear and quadratic programs and by Chua and Lin [15] for general nonlinear programming. The recent interest in neural networks has led to a revival of this idea [91]. In this thesis we will have ample opportunity for illustrating the interplay between mathematical programming and circuit theory.

1.7 Chapter by chapter overview

1.7.1 Nonlinear resistive circuits

This work contributes to the *theory* of nonlinear resistive circuits, not only by providing unified and constructive proofs of known existence and uniqueness results, but also by generalizing these results in several ways. Furthermore, it will be shown that variable dimension algorithms can be used as *practically feasible and efficient methods*, superior to comparable methods for solving circuit equations. The general strategy will be to exploit the structure of the problem as much as possible, an approach similar to the one taken in the qualitative theory of nonlinear circuits outlined in section 1.1. In order to arrive at useful results concerning existence or uniqueness of solutions in nonlinear

resistive circuits, researchers have had to go further than invoking general theorems pertaining to equations $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, and have had to take into account the linear and combinatorial constraints imposed by the underlying graph. Similar considerations hold for the application of general algorithms for solving sets of nonlinear equations. As will be exemplified here, there is a long way between writing down convergence conditions for problems in the form $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ and judging their validity in the specific class of problems occurring in circuit theory; or between discussing general aspects of the implementation, and deciding whether in practice structure can be exploited in order to make the algorithm efficient.

Chapters 2–5 cover the material on resistive circuits.

Chapter two introduces variable dimension algorithms by means of a simple circuit-theoretical equivalent. It is shown how these algorithms overcome the limitations of well-known homotopy methods in circuit theory (Katzenelson's algorithm and the method by Chao et al.). We then restrict the attention to the most important representative of the class of variable dimension algorithms, the octahedral algorithm. The theoretical foundation of the principle is given and the link is made with degree and index theory. We discuss the convergence conditions, which include virtually all circuits with eventually passive resistors (including common transistor circuits) under a mild interconnection condition, and a restricted class of op-amp circuits and circuits with controlled sources.

Chapter three deals with the numerical implementation of the octahedral algorithm. The implementation uses simplicial pivoting and lexicographical perturbation. It is shown how the algorithm can be accelerated by combining it with quasi-Newton steps, and by exploiting the separable structure in the circuit equations. The chapter concludes with a list of test examples, and a comparison with Katzenelson's algorithm and homotopy continuation methods.

Chapter four contains an extension of the algorithm to process active circuits and, in particular, circuits with controlled sources or ideal operational amplifiers. These elements are usually excluded from the discussions of homotopy methods. This again leads to construcive proofs of several new results concerning existence of solutions. We also present a modification of the octahedral algorithm for finding multiple solutions.

Chapter five places variable dimension algorithms in the more general context of complementarity theory. In a first part, the connection of variable dimension algorithms with Lemke's algorithm, as indicated in table 1.1, is worked out. The second part of the chapter discusses the application of linear complementarity for the complete analysis of piecewise-linear circuits. A

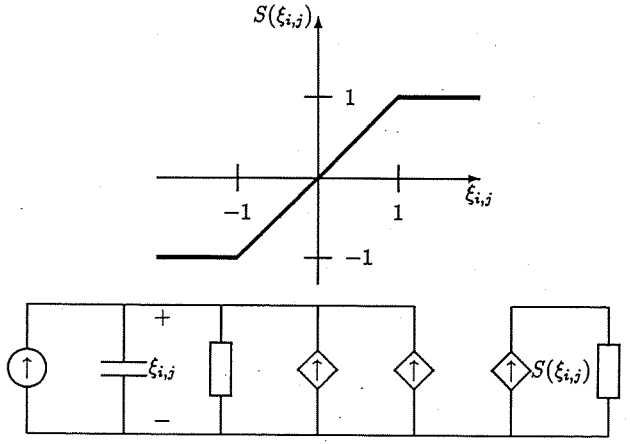


Figure 1.9: Equivalent circuit of one cell in a cellular neural network. The controlled sources in the left part connect the cell to its neighbours and to the external input.

comparison between this approach and variable dimension algorithms is also included.

1.7.2 Neural networks

The application of variable dimension algorithms and complementary pivoting in circuit theory is not restricted to resistive ciruits. In Chapters 6 and 7 two other areas of application are explored.

Chapter six is devoted to neural networks, and more specifically to the recent class of cellular neural networks. Cellular neural networks are built by a local interconnection of a large number of regularly spaced cells. Each cell is a small nonlinear circuit, consisting of a linear capacitor, a nonlinear voltage-controlled current source, and a few linear resistive elements (see figure 1.9). The purpose of these systems is to provide high-speed parallel signal processing [18, 19].

The main part of Chapter six is concerned with the problem of finding equilibrium points in cellular neural networks. This is important since the operation of cellular neural networks depends crucially on nonlinear dynamics and on the existence of multiple equilibrium points. This problem can be modelled as a linear complementarity problem with upper and lower bounds, and it can be solved by an algorithm which is very close to the octahedral

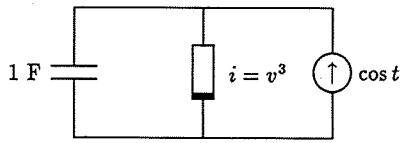


Figure 1.10: Example.

algorithm.

The chapter also contains a theorem on the stability of the networks, and a method to design a cellular neural network from a set of specifications.

1.7.3 Harmonic balance

Harmonic balance is a method for steady-state analysis of nonlinear circuits. By formulating the circuit equations in the frequency domain, the problem of solving a set of ordinary differential equations is reduced to a problem of solving a set of algebraic equations. The simple circuit of fig 1.10 for example is described by the differential equation

$$\dot{x} + x^3 = \cos t.$$

A steady-state solution with period 2π must have the form

$$x(t) = \sum_{k=1}^{\infty} (a_k \cos kt + b_k \sin kt), \tag{1.3}$$

with $a_k = b_k = 0, \quad k = 2, 4 \ldots$. To make the problem tractable, one truncates the Fourier series e.g. after $k = 3$:

$$x(t) = a_1 \cos t + a_3 \cos 3t + b_1 \sin t + b_3 \sin 3t.$$

Then x^3 can be expressed as

$$x^3 = \sum_{k=0}^{4} (c_{2k+1} \cos(2k+1)t + d_{2k+1} \sin(2k+1)t),$$

where the coefficients c_{2k+1} and d_{2k+1} can be calculated from a_1, a_3, b_1 and b_3. With this notation, equation (1.3) can be written as

$$b_1 + c_1(a_1, b_1, a_3, b_3) \;=\; 1,$$

$$
\begin{aligned}
-a_1 + d_1(a_1, b_1, a_3, b_3) &= 0, \\
3b_3 + c_3(a_1, b_1, a_3, b_3) &= 0 \\
-3a_3 + d_3(a_1, b_1, a_3, b_3) &= 0.
\end{aligned}
$$

Higher order contributions are neglected. These equations are called the harmonic balance equations and they can be solved by any algorithm for nonlinear equations, including continuous deformation methods.

In **Chapter seven**, a new variable dimension algorithm is discussed briefly as an interesting alternative to solve the harmonic balance equations. The method is globally convergent and processes the equations of different orders one after the other. Since harmonic balance typically leads to very large systems of equations, such a strategy can be expected to increase the efficiency considerably.

1.7.4 Appendices

Some further preliminaries are provided in appendices A and B.

Appendix A contains a few prerequisites from nonnegative linear algebra (the concepts of simplex, cell and triangulation). It also lists a flow chart of the octahedral triangulation and of its modification used in Chapter seven.

Appendix B gives a unified treatment of the theorems of alternative on the solvability of systems of linear inequalities. These theorems are used at several places in the text.

Chapter 2

Variable dimension algorithms : basic principle and convergence criteria

2.1 Introduction

An attractive idea for a globally convergent method to solve $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, $\mathbf{f}(\cdot) : R^n \rightarrow R^n$ was proposed by Chao *et al.* in references [11] and [59]. It is based on the fact that the solutions of the $n-1$ equations

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0, \\ f_2(x_1, x_2, \ldots, x_n) &= 0, \\ &\vdots \\ f_{n-1}(x_1, x_2, \ldots, x_n) &= 0, \end{aligned} \tag{2.1}$$

generically form a union of disjoint paths and loops. The method then goes as follows.

- First find a solution $\mathbf{x}^0$ of (2.1). Make sure that this point does not lie on a loop of the solution set of these equations.
- Trace the component of the solution set of (2.1) that contains $\mathbf{x}^0$ until a point is reached with $f_n = 0$.

Variable dimension algorithms can be seen as a generalization of this idea and overcome two immediate problems.

- The problem of finding a solution of (2.1) is solved by recursively solving problems of dimension 1 to $n-1$. The important insight enabling global convergence is that the dimension can also decrease in the course of the algorithm.
- Chao's method breaks down when $\mathbf{x}^0$ lies on a loop of the solution set of the first $n-1$ equations. This can cause restart problems similar to the ones discussed in Section 1.3. In piecewise-linear analysis the phenomenon can be avoided by locating $\mathbf{x}^0$ in an unbounded region of the subdivision, but in general this is inefficient. It means starting outside the region of interest, and makes it difficult to use a previous approximation as starting point for a search with a finer subdivision. Variable dimension algorithms circumvent this problem as will be illustrated in Example 2.2.1.

In the first part of this chapter (Sections 2.2 and 2.3) the basic idea of variable dimension algorithms will be introduced by means of a circuit-theoretical equivalent. The availability of such an equivalent will be crucial in a later stage, because it allows for the use of results from circuit and graph theory. A second part (Sections 2.4, 2.5 and 2.6) formulates the conditions for convergence in topological terms and derives a number of existence theorems from the basic principle. Convergence will be shown to be closely related to the asymptotic behaviour of the components and to the concept of *eventual passivity*.

2.2 The octahedral algorithm

This thesis will mainly deal with one specific type of variable dimension algorithm, called the *octahedral algorithm*.[1] The algorithm was published in reference [128] but will be introduced here via a circuit-theoretical interpretation. Let us first look at a two-dimensional example.

2.2.1 Example

Assume $\mathbf{f}(\cdot)$ in (1.2) is the description of a current-controlled 2-port (see figure 2.1) :

$$\begin{aligned} v_1 = f_1(i_1, i_2) &= (i_1^2 + i_2^2 - 4)(i_1 - 10), \\ v_2 = f_2(i_1, i_2) &= i_2 - 9, \end{aligned} \tag{2.2}$$

and that a solution is sought with $v_1 = v_2 = 0$. The origin is taken as a starting point. In figure 2.1 the ports are terminated by two auxiliary branches, built up from ideal diodes and from independent current sources with adjustable and positive value t. Initially, t is zero, all diodes are blocking and $i_1 = i_2 = 0$. The

[1] The origin of this name will become clear in Section 3.3.

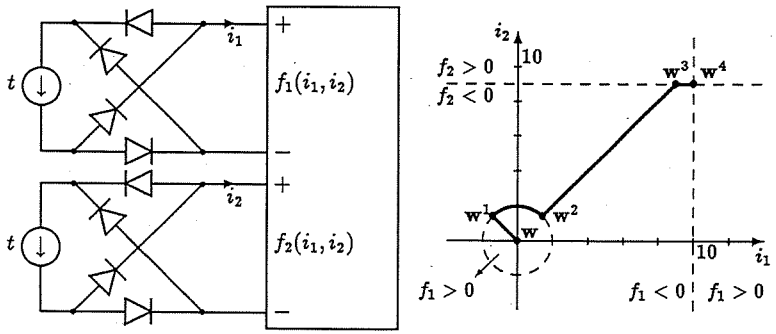


Figure 2.1: The octahedral algorithm applied to the equations (2.2) : auxiliary branches and path traced by the algorithm.

algorithm attempts to increase the parameter t until all diodes conduct and the two ports are short-circuited. The path followed is shown in (i_1, i_2)-space in figure 2.1.

From $\mathbf{w}$ to $\mathbf{w}^1$: When $t = 0$, both currents i_1 and i_2 are zero and we are in point $\mathbf{w} = \mathbf{0}$. In this point $v_1 = f_1(\mathbf{w}) > 0$ and $v_2 = f_2(\mathbf{w}) < 0$. Upon increasing t, a current $\pm t$ must start to flow through each port. Owing to the diode orientation, i_2 will be positive and equal to t, whereas i_1 will be negative and equal to $-t$. So we move along the ray from $\mathbf{w}$ to $\mathbf{w}^1$. It is important to note that the direction to choose at $\mathbf{w}$ is uniquely determined by the signs of $f_1(\mathbf{w})$ and $f_2(\mathbf{w})$. This uniqueness implies that the starting point can never be reached from another side and thus that cycling is excluded.

From $\mathbf{w}^1$ to $\mathbf{w}^2$: At $\mathbf{w}^1$ the voltage v_1 becomes zero while i_1 is still negative. Owing to the diode constraints the sign of v_1 cannot reverse unless the current i_1 changes sign. So it is impossible to travel further along the ray from $\mathbf{w}$ to $\mathbf{w}^1$. At $\mathbf{w}^1$, however, the constraint $-i_1 = i_2 = t$ changes to $i_2 = t$, $-t \leq i_1 \leq t$, and we can proceed along the upper half of the circle until $i_1 = t$ at $\mathbf{w}^2$.

From $\mathbf{w}^2$ to $\mathbf{w}^3$: Current i_1 has become equal to t so one can let v_1 become negative and move along the ray from $\mathbf{w}^2$ to $\mathbf{w}^3$.

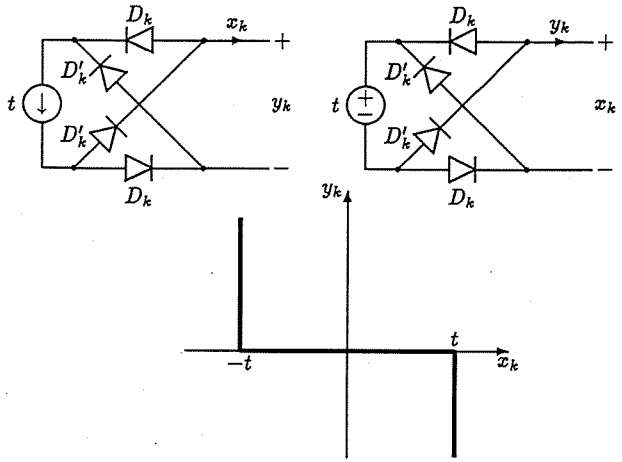


Figure 2.2: Auxiliary branches to be appended to current-controlled ports (left) and voltage-controlled ports (right). Constitutive relation for these branches.

From $\mathbf{w}^3$ to $\mathbf{w}^4$: At $\mathbf{w}^3$ the situation is similar to point $\mathbf{w}^1$ but now v_2 has become zero. We keep $i_1 = t$ and $-t \leq i_2 \leq t$, and move towards the solution $\mathbf{w}^4$. It can easily be seen that the path must stop at this point.

Note that the parameter t does not increase monotonically during the algorithm. This is no problem because the algorithm traces a one-dimensional path, so no ambiguity exists as to which direction to choose.

2.2.2 Problem formulation

The algorithm can be applied to several types of circuit analysis by inserting the branches of figure 2.2 at appropriate places in a circuit.[2] This will be the subject of Sections 2.4–2.6. Here the system $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ will be treated as a black-box description of an n-port as follows.

A resistive n-port is given with hybrid representation

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \tag{2.3}$$

[2]The presence of ideal diodes in the auxiliary branches is no coincidence, as will be shown in Chapter 5.

where $\mathbf{x}, \mathbf{y} \in R^n$ are dual port variables : x_k is a port current (port voltage) and y_k a port voltage (port current) if port k is current-controlled (voltage-controlled). For ease of notation, the algorithm will be assumed to start at the origin. If a better initial guess $\mathbf{x}^0$ is known, it can be taken into account by replacing $\mathbf{f}(\cdot)$ by the mapping $\mathbf{g}(\cdot)$ defined as $\mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathbf{x} + \mathbf{x}^0)$.

2.2.3 General scheme

The generalization of the circuit in figure 2.1 for the problem of (2.3) is depicted in figure 2.2. One of both types of auxiliary branches is connected to each port k, depending on whether the port is voltage- or current-controlled. Parameter t is a nonnegative number and is initially set to zero. Then all diodes at the current-controlled ports are blocking, all diodes at the voltage-controlled ports are conducting, and the circuit operates at $\mathbf{x} = \mathbf{0}$. During the algorithm the parameter t will be increased until the diodes at the current-controlled ports are conducting and the diodes at the voltage-controlled ports are blocking. Then the current-controlled ports are short-circuited, the voltage-controlled ports are open-circuited and a zero point of system (2.3) has been found.

To work out this idea, it is useful to represent the path followed by the algorithm geometrically in the $(\mathbf{x}, \mathbf{y})$-space R^{2n}. The path is part of the intersection of two manifolds : $\mathcal{F} = \{(\mathbf{x}, \mathbf{y}) \mid \mathbf{y} = \mathbf{f}(\mathbf{x})\}$, and $\mathcal{Z} = \{(\mathbf{x}, \mathbf{y}) \mid \mathbf{x}$ and $\mathbf{y}$ are compatible with the branches of figure 2.2 for some $0 \leq t < \infty\}$. A formal description of $\mathcal{Z}$ can be obtained by introducing the definition of sign vector : a **sign vector** $\mathbf{s}$ is a nonzero vector with components -1, 0 or $+1$, and indicates the *state* of the diode branches. If port k is a current-controlled port then

- $s_k = +1$ if diodes D'_k in figure 2.2 are on and diodes D_k are off,
- $s_k = -1$ if diodes D_k in figure 2.2 are on and diodes D'_k are off,
- $s_k = 0$ if all diodes at port k are on.

If port k is a voltage-controlled port then

- $s_k = +1$ if diodes D_k in figure 2.2 are on and diodes D'_k are off,
- $s_k = -1$ if diodes D'_k in figure 2.2 are on and diodes D_k are off,
- $s_k = 0$ if all diodes at port k are off.

The number of zero elements in $\mathbf{s}$ will be written as $|\mathbf{s}|$.

For each sign vector $\mathbf{s}$, sets $X(\mathbf{s})$, $Y(\mathbf{s})$ and $Z(\mathbf{s})$ are defined as

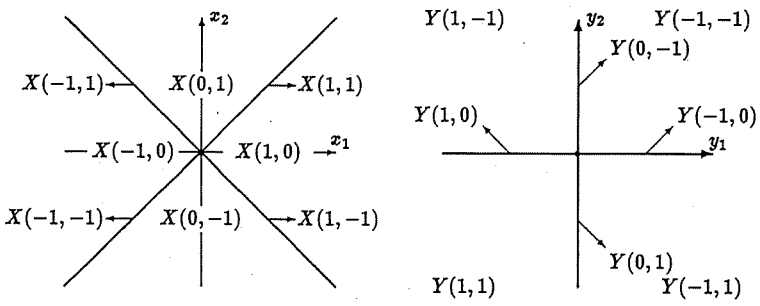


Figure 2.3: The sets $X(\mathbf{s})$ and $Y(\mathbf{s})$ for $n = 2$.

- $X(\mathbf{s}) = \{\mathbf{x} \in R^n \mid s_k x_k = \|\mathbf{x}\|_\infty \text{ if } s_k \neq 0\}$,
- $Y(\mathbf{s}) = \{\mathbf{y} \in R^n \mid y_k = 0 \text{ if } s_k = 0, \ s_k y_k \leq 0 \text{ if } s_k \neq 0\}$,
- $Z(\mathbf{s}) = X(\mathbf{s}) \times Y(\mathbf{s})$.

The sets $X(\mathbf{s})$ and $Y(\mathbf{s})$ are illustrated in figure 2.3. It can be verified that for a given state of the diodes in figure 2.2, a pair $(\mathbf{x}, \mathbf{y})$ is admissible only when it belongs to $Z(\mathbf{s})$. The value of t corresponding to an admissible point $\mathbf{x}$ equals $\|\mathbf{x}\|_\infty$. The set $\mathcal{Z}$ is the union of all sets $Z(\mathbf{s})$.

From figure 2.3 one can observe that the dimension of cells $X(\mathbf{s})$ and $Y(\mathbf{s})$ depends on the number of zero elements $|\mathbf{s}|$ in the sign vector $\mathbf{s}$, but that the sum of $\dim X(\mathbf{s})$ and $\dim Y(\mathbf{s})$ is equal to three for all $\mathbf{s}$. More generally, one has that

$$\dim X(\mathbf{s}) = |\mathbf{s}| + 1, \quad \dim Y(\mathbf{s}) = n - |\mathbf{s}|, \quad \dim Z(\mathbf{s}) = n + 1.$$

Example 2.2.1 continued. Figure 2.4 shows qualitatively the path in $\mathbf{x}$- and $\mathbf{y}$-space. The algorithm starts with $\mathbf{x}$ in the origin and $\mathbf{y}$ in the n-dimensional cell $Y(-1,1)$, then it traces a path through cells $X(\mathbf{s})$ and $Y(\mathbf{s})$ of varying dimension until a point is reached with $\mathbf{x}$ in an n-dimensional cell $X(1,0)$ and with $\mathbf{y}$ in the origin. *This switching between cells of varying dimension in* $\mathbf{x}$- *and* $\mathbf{y}$-*space is the essence of the notion variable dimension algorithm.*

From figure 2.4 one can also observe that each time the path moves from one cell $Z(\mathbf{s})$ into another, only one component of $\mathbf{s}$ changes (from 0 to ± 1 or vice versa). In general, if adjacent cells are defined as cells that meet in common n-faces, then the following can be shown [54] :

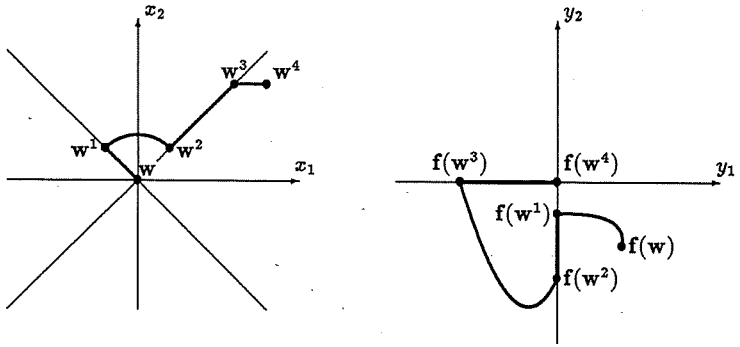


Figure 2.4: The path of figure 2.1 in x- and y-space (the figure only shows the qualitative behaviour).

- If $0 < |s| < n-1$, then the cells $Z(\tilde{s})$ with $|\tilde{s}| = |s| - 1$ and $\tilde{s}_k = s_k$ if $s_k \neq 0$, and the cells $Z(\tilde{s})$ with $|\tilde{s}| = |s| + 1$ and $\tilde{s}_k = s_k$ if $\tilde{s}_k \neq 0$, are adjacent to $Z(s)$. Each n-face of $Z(s)$ can be written as $Z(s) \cap Z(\tilde{s})$ with $Z(\tilde{s})$ adjacent to $Z(s)$.

- If $|s| = 0$, then the cells $Z(\tilde{s})$ with $|\tilde{s}| = |s| + 1$ and $\tilde{s}_k = s_k$ if $\tilde{s}_k \neq 0$ are adjacent to $Z(s)$. Each n-face of $Z(s)$ can be written as $Z(s) \cap Z(\tilde{s})$ with $Z(\tilde{s})$ adjacent to $Z(s)$ or as $\{\mathbf{0}\} \times Y(s)$ (which is part of the boundary of $\mathcal{Z}$ and therefore belongs to only one cell $Z(s)$).

- If $|s| = n-1$, then the cells $Z(\tilde{s})$ with $|\tilde{s}| = |s| - 1$ and $\tilde{s}_k = s_k$ if $s_k \neq 0$ are adjacent to $Z(s)$. Each n-face of $Z(s)$ can be written as $Z(s) \cap Z(\tilde{s})$ with $Z(\tilde{s})$ adjacent to $Z(s)$ or as $X(s) \times \{\mathbf{0}\}$ (which is part of the boundary of $\mathcal{Z}$ as well).

The boundary of $\mathcal{Z}$, consists of two parts [54] : $\partial\mathcal{Z} = (\{\mathbf{0}\} \times R^n) \cup (R^n \times \{\mathbf{0}\})$. This is also clear from figure 2.2. Part $\{\mathbf{0}\} \times R^n$ corresponds to the situation where all diodes at current-controlled ports (voltage-controlled ports) are blocking (conducting), and part $R^n \times \{\mathbf{0}\}$ to the situation where all diodes at current-controlled ports (voltage-controlled ports) are conducting (blocking).

With these definitions, the geometrical representation can be completed as follows. Generically, one can expect the intersection of $\mathcal{Z}$ and $\mathcal{F}$ to be a one-dimensional manifold, i.e. a collection of disjoint loops and paths. Exactly one of these paths intersects part $\{\mathbf{0}\} \times R^n$ of the boundary of $\mathcal{Z}$ i.e., in the

starting point $(\mathbf{0}, \mathbf{f}(\mathbf{0}))$ of the algorithm. The path through this point traverses a number of cells $Z(s)$ until it either goes to infinity or reaches the other part $R^n \times \{\mathbf{0}\}$ of the boundary of $\mathcal{Z}$. In the latter case, the endpoint satisfies $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ and is a solution to the problem.

In other words, success of the algorithm will depend on two points that will be discussed in the next two sections.

- Section 2.2.4 describes under what conditions the solution set $\mathcal{F} \cap \mathcal{Z}$ is well defined in the sense that the the component containing the starting point $(\mathbf{0}, \mathbf{f}(\mathbf{0}))$ is a one-dimensional path that can be traced numerically. This will turn out to be a merely technical point of no real practical concern.

- Once this well-definedness of the path can be taken for granted, convergence of the algorithm follows from boundedness of the solution set $\mathcal{F} \cap \mathcal{Z}$. Section 2.2.5 will show how this can be established from circuit-theoretical considerations such as eventual passivity of the circuit.

2.2.4 Well-definedness of the homotopy path

The fundamental theorem that underlies the algorithm can be formulated as follows [53, 54, 55].

Theorem 1 *If the regularity assumptions 1-4 below are valid, then $\mathcal{Z} \cap \mathcal{F}$ is the union of disjoint piecewise differentiable paths and loops such that*

1. *No loop intersects $\partial\mathcal{Z}$.*

2. *A point $(\mathbf{x}, \mathbf{y}) \in \mathcal{F} \cap \mathcal{Z}$ is an endpoint of a path iff $(\mathbf{x}, \mathbf{y}) \in \partial\mathcal{Z}$.*

3. *Every open or semiclosed path is unbounded.*

Regularity assumption 1 *The mapping $\mathbf{f}(\cdot)$ is continuously differentiable.*

Regularity assumption 2 *For each sign vector s one has at all points of $\mathcal{F} \cap Z(s)$ that* $\operatorname{rank} \begin{pmatrix} \mathbf{f}'_{\alpha,\alpha} & \mathbf{f}'_{\alpha,\cdot} s \end{pmatrix} = |s|$, *where $\alpha = \{k \mid s_k = 0\}$.*

Regularity assumption 3 *For each sign vector s one has the following two properties :*

1. *The set $\mathcal{F} \cap Z(s)$ intersects no faces of $Z(s)$ with dimension less than n,*

2. *If* $\tilde{\mathbf{s}}$ *is a sign vector with* $|\tilde{\mathbf{s}}| = |\mathbf{s}| + 1$ *and with* $\tilde{s}_k = s_k$ *for all* $\tilde{s}_k \neq 0$ *then one has at all points of* $\mathcal{F} \cap \mathcal{Z}$ *on the* n*-face* $Z(\mathbf{s}) \cap Z(\tilde{\mathbf{s}})$ *that* rank $\begin{pmatrix} \mathbf{f}'_{\tilde{\alpha},\alpha} & \mathbf{f}'_{\tilde{\alpha},\cdot}\mathbf{s} \end{pmatrix} = |\tilde{\mathbf{s}}|$, *with* $\alpha = \{k \mid s_k = 0\}$ *and* $\tilde{\alpha} = \{k \mid \tilde{s}_k = 0\}$.

Regularity assumption 4 *At all solutions to* $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ *one has* rank $\mathbf{f}' = n$.

An important observation is that assumptions 2–4 hold *generically* in the sense that one can always find an arbitrarily small perturbation of the problem $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ to make these assumptions hold. A vector $\mathbf{c}$ is called a *regular value* if regularity assumptions 2–4 hold with $\mathcal{F}$ redefined as $\mathcal{F} = \{(\mathbf{x}, \mathbf{y}) \mid \mathbf{f}(\mathbf{x}) - \mathbf{y} = \mathbf{c}\}$, and with $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ in assumption 4 replaced by $\mathbf{f}(\mathbf{x}) = \mathbf{c}$, and a *critical value* otherwise. It follows from Sard's theorem in differential topology [44] that the set of critical values has measure zero. In other words, if $\mathbf{0}$ happens to be a critical value, it can be replaced by an arbitrarily small regular value $\mathbf{c}$.

The proof of Theorem 1 will not be given here but the basic idea is as follows. Because of assumptions 1 and 2 one can apply the implicit function theorem [73] in each cell $Z(\mathbf{s})$ to show that locally the solution set is one-dimensional and differentiable. Assumptions 3 and 4 ensure that the path will be transversal [44] to the faces of $Z(\mathbf{s})$ it crosses.

The above remarks imply that, to some extent, the regularity assumptions can be dispensed with.

- For practical computations, continuity of $\mathbf{f}(\cdot)$ is sufficient when piecewise-linear techniques are used for approximating the path. Also whenever a degenerate point is encountered where assumptions 2–4 do not hold, this can be resolved by an infinitesimally small perturbation.

- In Sections 2.4–2.6 the algorithm will be used as a constructive tool for deriving existence results. The criteria on $\mathbf{f}(\cdot)$ will always be stated in terms of the problem $\mathbf{f}(\mathbf{x}) = \mathbf{c}$ for arbitrary $\mathbf{c}$ (for resistive circuits this amounts to inserting arbitrary current sources by soldering iron entry and arbitrary voltage sources by pliers entry). Because regularity assumptions 2–4 only hold generically, the best thing one could hope to prove by means of the algorithm is existence of a solution to the problem $\mathbf{f}(\mathbf{x}) = \mathbf{c}$ for almost all values of $\mathbf{c}$. However, if a solution exists for *almost all* values of $\mathbf{c}$ then there must be one for *all* values of $\mathbf{c}$.[3]

[3]This follows from the fact that there is a neighbourhood V of each regular value $\mathbf{c}$ such that $\#\mathbf{f}^{-1}(\mathbf{c}) = \#\mathbf{f}^{-1}(\mathbf{y})$ for all $\mathbf{y} \in V$. The regular values are the values $\mathbf{c}$ for which either $\mathbf{f}^{-1}(\mathbf{c})$ is empty, or $\mathbf{f}'$ is of full rank in all points of $\mathbf{f}^{-1}(\mathbf{c})$ [66].

2.2.5 Convergence criterion

According to Theorem 1, provided the regularity assumptions hold, the path traced by the algorithm must either end at a solution or diverge to infinity. A sufficient condition to exclude divergence is therefore to show boundedness of $\mathcal{F} \cap \mathcal{Z}$ as in the next theorem.

Theorem 2 *If there exists a number* $R > 0$ *such that* $\|\mathbf{x}\| > R$ *implies that* $\max_k x_k f_k(\mathbf{x}) > 0$ *then the solution set* $\mathcal{F} \cap \mathcal{Z}$ *is bounded.*

Proof. The meaning of the above condition is that, as $\|\mathbf{x}\|$ goes to infinity, at least one of the ports of (2.3) must dissipate power. Operation at such a point would contradict the power balance because of the passivity of the auxiliary branches in figure 2.2. Hence $\mathcal{F} \cap \mathcal{Z}$ is bounded. □

Example. A typical example of the condition above is the equation used by Sandberg and Willson in their study of transistor networks [127] :

$$\mathbf{f}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) + \mathbf{A}\mathbf{x} - \mathbf{b}, \tag{2.4}$$

where $\mathbf{g}(\cdot)$ is a diagonal mapping satisfying $g_k(x_k) \to \infty$ if $x_k \to \infty$ and $g_k(x_k) \to -\infty$ if $x_k \to -\infty$, and where $\mathbf{A}$ is a P_0 matrix [33]. The condition in Theorem 2 then follows from the well known property [33]

$$\forall \mathbf{x} \neq \mathbf{0}, \exists k \ : \ x_k \neq 0 \text{ and } x_k.(\mathbf{A}\mathbf{x})_k \geq 0,$$

if $\mathbf{A}$ is a P_0-matrix. The octahedral algorithm applied to (2.4) is therefore guaranteed to converge (for almost all values of $\mathbf{b}$), and this proves the existence of at least one zero point of the mapping (2.4) (for all values of $\mathbf{b}$).

2.2.6 General formulation of the algorithm

In summary of the above sections, a formal description of the algorithm will now be given. We assume that Theorem 1 holds, otherwise the equations should be perturbed as will be explained later.

Start

Let $\mathbf{x}^* := \mathbf{0}$, $\mathbf{y}^* := \mathbf{f}(\mathbf{0})$, and for $k = 1, \ldots, n$, $s_k := 1$ if $y_k^* \leq 0$ and $s_k := -1$ if $y_k^* > 0$. Then $(\mathbf{x}^*, \mathbf{y}^*)$ lies on an n-face of $Z(\mathbf{s})$ and belongs to $\mathcal{Z} \cap \mathcal{F}$. Denote the component of $\mathcal{Z} \cap \mathcal{F}$ that contains $(\mathbf{x}^*, \mathbf{y}^*)$ with S.

Repeat

Continue the path S through $Z(\mathrm{s})$ from $(\mathbf{x}^*, \mathbf{y}^*)$ onwards until another point $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ on the boundary of $Z(\mathrm{s})$ is reached. Inside $Z(\mathrm{s})$, the path S is determined by

(i) A set of n equations $\mathbf{f}(\mathbf{x}) - \mathbf{y} = \mathbf{0}$. Let $\alpha = \{k \mid s_k = 0\}$ and $\beta = \{k \mid s_k \neq 0\}$. Because $\mathbf{y}_\alpha = \mathbf{0}$ and $\mathbf{x}_\beta = \|\mathbf{x}\|_\infty . \mathbf{s}_\beta$, the number of unknowns is $n+1$: $(\mathbf{x}_\alpha, \mathbf{y}_\beta, t)$ with $t = \|\mathbf{x}\|_\infty$.

(ii) A set of inequalities :

- $x_k \leq \|\mathbf{x}\|_\infty$ if $s_k = 0$,
- $x_k \geq -\|\mathbf{x}\|_\infty$ if $s_k = 0$,
- $s_k y_k \leq 0$ if $s_k \neq 0$.

At the point $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ where S hits an n-face of $Z(\mathrm{s})$, exactly one of the constraints in (ii) holds with equality. This determines the next piece of $\mathcal{Z}$ to be entered :

- if $\tilde{x}_k = \|\tilde{\mathbf{x}}\|_\infty$, and $s_k = 0$, then set $s_k := 1$,
- if $\tilde{x}_k = -\|\tilde{\mathbf{x}}\|_\infty$, and $s_k = 0$, then set $s_k := -1$,
- if $\tilde{y}_k = 0$, and $s_k \neq 0$, set $s_k := 0$.

Take $\mathbf{x}^* := \tilde{\mathbf{x}}$ and $\mathbf{y}^* := \tilde{\mathbf{y}}$.

until $\mathbf{s} = \mathbf{0}$
Stop : $\mathbf{y}^* = \mathbf{f}(\mathbf{x}^*) = \mathbf{0}$.

In Chapter 3 it will be described how the algorithm can be carried out numerically.

2.2.7 The index of a solution found by the octahedral algorithm

According to Theorem 1 the solution set $\mathcal{F} \cap \mathcal{Z}$ consists of a number of disjoint paths and loops. In this section an orientation will be assigned to each path in $\mathcal{F} \cap \mathcal{Z}$ and in this way important degree properties will be derived. The results will be useful for proving uniqueness of solutions.

The discussion here is based on the following lemma from [36].

Lemma 1 *Suppose* $\mathbf{H}$ *is a* C^1 *mapping from* R^{n+1} *to* R^n. *Let* $\mathbf{H}'$ *be the* $n \times (n+1)$ *Jacobian matrix with respect to the* $n+1$ *variables* $x_1, \ldots, x_{n+1}$ *and let* $\mathbf{H}'_{_j}$ *be the* $n \times n$ *matrix formed from* $\mathbf{H}'$ *by deleting the column* j.

Assume $\mathbf{H}'$ *has rank* n *at each point of* $\mathbf{H}^{-1}(\mathbf{0})$. *Let* $\mathbf{x}^0$ *be a point in* $\mathbf{H}^{-1}(\mathbf{0})$. *Then the solution* $\mathbf{x}(p)$ *to the set of ordinary differential equations*

$$\dot{x}_i = \gamma(-1)^{i+1} \det \mathbf{H}'_{_i} \quad i = 1, \ldots, n \tag{2.5}$$

with starting point $\mathbf{x}^0$ *determines a component of* $\mathbf{H}^{-1}(\mathbf{0})$. *Here* $\dot{x}_i = dx_i/dp$ *and the scalar* $\gamma = +1$ *or* -1 *depending on the orientation chosen.*

From this it is straightforward to derive a set of differential equations that describe $\mathcal{F} \cap \mathcal{Z}$ within each cell $Z(\mathrm{s})$. If $\alpha = \{k \mid s_k = 0\}$ and $\beta = \{k \mid s_k \neq 0\}$, then $\mathbf{x}_\beta = \mathbf{s}_\beta . \|\mathbf{x}\|_\infty$ and $\mathbf{y}_\alpha = \mathbf{0}$. The set $\mathcal{F} \cap Z(\mathrm{s})$ is then defined by a set of n equations $\mathbf{f}(\mathbf{x}) - \mathbf{y} = \mathbf{0}$ in $n+1$ variables $\mathbf{x}_\alpha, \mathbf{y}_\beta$ and $t = \|\mathbf{x}\|_\infty$. Denote $\mathbf{f}(\mathbf{x}) - \mathbf{y}$ as a mapping $\mathbf{h}(z_1, \ldots, z_{n+1})$ with $\mathbf{z}_\alpha = \mathbf{x}_\alpha$, $\mathbf{z}_\beta = \mathbf{y}_\beta$ and $z_{n+1} = t$. The Jacobian matrix of $\mathbf{h}(\cdot)$ is the $n \times (n+1)$-matrix $\mathbf{h}'$ with

$$\mathbf{h}'_{\cdot,\alpha} = \mathbf{f}'_{\cdot,\alpha}\,, \qquad \mathbf{h}'_{\cdot,\beta} = -\mathbf{I}_{\cdot,\beta}\,, \qquad \mathbf{h}'_{\cdot,n+1} = \mathbf{f}'\mathbf{s}\,,$$

where $\mathbf{I}$ is the $n \times n$ identity matrix. Application of (2.5) yields

$$\dot{x}_k = -\gamma \det \begin{pmatrix} \mathbf{f}'_{\alpha\setminus k,\alpha\setminus k} & \mathbf{f}'_{\alpha\setminus k,\cdot}\mathbf{s} \\ \mathbf{f}'_{k,\alpha\setminus k} & \mathbf{f}'_{k,\cdot}\mathbf{s} \end{pmatrix}, \quad k \in \alpha \tag{2.6}$$

$$\dot{y}_k = \gamma \det \begin{pmatrix} \mathbf{f}'_{\alpha,\alpha} & \mathbf{f}'_{\alpha,\cdot}\mathbf{s} \\ \mathbf{f}'_{k,\alpha} & \mathbf{f}'_{k,\cdot}\mathbf{s} \end{pmatrix}, \quad k \in \beta \tag{2.7}$$

$$\dot{t} = \gamma \det \left(\mathbf{f}'_{\alpha,\alpha}\right) \tag{2.8}$$

with $\gamma = \pm 1$ depending on the orientation of the path. If $\alpha = \emptyset$, then $\det \mathbf{f}'_{\alpha,\alpha} = 1$ by definition. Equation (2.8) demonstrates that the principal minors of $\mathbf{f}'$ determine whether the parameter t increases or decreases.

Definition 1 *Assume* $\mathbf{f}(\cdot)$ *is a differentiable mapping and that* $\det \mathbf{f}'$ *is nonsingular at all solutions to* $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. *Then each point* $(\mathbf{x}, \mathbf{y})$ *of* $\mathcal{F} \cap \mathcal{Z}$ *on the boundary of* $\mathcal{Z}$ *has the following index.*

- *The index of* $(\mathbf{0}, \mathbf{f}(\mathbf{0}))$ *is* -1.
- *The index of a point* $(\mathbf{x}, \mathbf{0})$ *with* $\mathbf{x}$ *a zero of* $\mathbf{f}(\cdot)$ *is defined as* sign $\det \mathbf{f}'$.

The definition of index can be extended to continuous, nondifferentiable mappings, and to isolated solutions where the Jacobian matrix is singular [17].

Theorem 3 *Assume that the regularity assumptions 1-4 hold. Let* S *be a component of* $\mathcal{F} \cap \mathcal{Z}$. *If* S *is a bounded path then both endpoints must have opposite index.*

Proof. Call the endpoints $(\mathbf{x}^1, \mathbf{y}^1)$ and $(\mathbf{x}^2, \mathbf{y}^2)$ and assume the first one has negative index. It will be shown that if the path S is oriented from $(\mathbf{x}^1, \mathbf{y}^1)$ onwards, then it must end at a solution with index $+1$. First we verify that with this orientation of S, the scalar γ must be $+1$ in all cells $Z(\mathbf{s})$ traversed by S. The proof of this statement is by induction.

- If $(\mathbf{x}^1, \mathbf{y}^1)$ is the point $(\mathbf{0}, \mathbf{f}(\mathbf{0}))$, then the state of the first cell encountered has no zero elements. Equation (2.8) becomes $\dot{t} = \gamma$. We know that initially $\dot{t} > 0$, thus $\gamma = +1$.

- If $(\mathbf{x}^1, \mathbf{y}^1)$ is a point on $R^n \times \{\mathbf{0}\}$, then the state $\mathbf{s}$ of the first cell encountered has $n-1$ zero elements. Let j be the index with $s_j \neq 0$. Equations (2.7) become

$$\dot{y}_j = \gamma \det \begin{pmatrix} \mathbf{f}'_{\alpha,\alpha} & \mathbf{f}'_{\alpha,\cdot}\mathbf{s} \\ \mathbf{f}'_{j,\alpha} & \mathbf{f}'_{j,\cdot}\mathbf{s} \end{pmatrix} = \gamma s_j \det \begin{pmatrix} \mathbf{f}'_{\alpha,\alpha} & \mathbf{f}'_{\alpha,j} \\ \mathbf{f}'_{j,\alpha} & \mathbf{f}'_{j,j} \end{pmatrix}.$$

  The determinant is nonzero by regularity assumption 4 and negative because the index of $(\mathbf{x}^1, \mathbf{y}^1)$ is -1. Inside $Z(\mathbf{s})$ we have $s_j y_j \leq 0$ while at the boundary point $y_j = 0$. Hence $s_j \dot{y}_j < 0$ at $(\mathbf{x}^1, \mathbf{y}^1)$ as the path moves into $Z(\mathbf{s})$ and γ must be $+1$.

This proves that we have $\gamma = +1$ in the first cell encountered. The induction step assumes that the path enters a cell $Z(\tilde{\mathbf{s}})$ through an n-face $Z(\mathbf{s}) \cap Z(\tilde{\mathbf{s}})$ and that inside $Z(\mathbf{s})$ it is described by (2.6)–(2.8) with $\gamma = +1$.

- Assume $|\tilde{\mathbf{s}}| = |\mathbf{s}| + 1$, $\tilde{s}_k = s_k$ if $\tilde{s}_k \neq 0$ and $\tilde{s}_j \neq s_j$. Inside $Z(\mathbf{s})$ we have $s_j y_j \leq 0$ while on the boundary $y_j = 0$. From (2.7) it follows that

$$s_j \det \begin{pmatrix} \mathbf{f}'_{\alpha,\alpha} & \mathbf{f}'_{\alpha,\cdot}\mathbf{s} \\ \mathbf{f}'_{j,\alpha} & \mathbf{f}'_{j,\cdot}\mathbf{s} \end{pmatrix} > 0 \tag{2.9}$$

  at the boundary point (the determinant is nonzero owing to regularity assumption 3). Equation (2.9) can be rewritten as

$$\det \begin{pmatrix} \mathbf{f}'_{\alpha,\alpha} & \mathbf{f}'_{\alpha,j} \\ \mathbf{f}'_{j,\alpha} & \mathbf{f}'_{j,j} \end{pmatrix} + s_j \det \begin{pmatrix} \mathbf{f}'_{\alpha,\alpha} & \mathbf{f}'_{\alpha,\cdot}\tilde{\mathbf{s}} \\ \mathbf{f}'_{j,\alpha} & \mathbf{f}'_{j,\cdot}\tilde{\mathbf{s}} \end{pmatrix} > 0.$$

  We have that $\dot{t} - s_j \dot{x}_j > 0$ as the path enters $Z(\tilde{\mathbf{s}})$ and combined with (2.6),(2.8) this implies that $\gamma = +1$ in $Z(\tilde{\mathbf{s}})$ as well.

- The other case $|\tilde{\mathbf{s}}| = |\mathbf{s}| - 1$ can be proved similarly.

Now, if $\gamma = +1$ in all cells $Z(\mathbf{s})$ traversed by the path, then the index of the endpoint $(\mathbf{x}^2, \mathbf{y}^2)$ must be $+1$ as can be seen as follows. The endpoint lies on the boundary of a cell $Z(\mathbf{s})$ with $|\mathbf{s}| = n - 1$. Assume $s_j \neq 0$. Then $s_j y_j \leq 0$ inside $Z(\mathbf{s})$ and $y_j = 0$ on the boundary. Hence $s_j \dot{y}_j > 0$ at the endpoint and from (2.7) it follows that

$$\det \begin{pmatrix} \mathbf{f}'_{\alpha,\alpha} & \mathbf{f}'_{\alpha,j} \\ \mathbf{f}'_{j,\alpha} & \mathbf{f}'_{j,j} \end{pmatrix} > 0.$$

In the same way it can be shown that if $(\mathbf{x}^1, \mathbf{y}^1)$ has index $+1$, then the other endpoint must have index -1. □

This theorem has the following consequences.

1. If the path traced by the octahedral algorithm converges, it must end at a zero of $\mathbf{f}(\cdot)$ with index $+1$.

2. If the solution set $\mathcal{F} \cap \mathcal{Z}$ is bounded, the number of zeros of $\mathbf{f}(\cdot)$ must be odd and the degree of $\mathbf{f}(\cdot)$ (defined as the sum of the indices of all zeros) is $+1$.

3. As mentioned before, the algorithm can be seen as a constructive way to prove existence of solutions : provided the regularity assumptions hold, it is sufficient to show boundedness of $\mathcal{F} \cap \mathcal{Z}$ in order to establish existence of at least one solution. If, in addition, one can show that all solutions have index $+1$, then the problem can have only one solution.

Example. We again take equation (2.4) but now add the assumption that each component of $\mathbf{g}(\cdot)$ is strictly increasing. From Section 2.2.5 we already know that (2.4) has at least one zero point. Assume it has two zeros $\mathbf{x}^1$ and $\mathbf{x}^2$. Then one can find a strictly positive diagonal matrix $\mathbf{D}$ such that $\mathbf{D}(\mathbf{x}^1 - \mathbf{x}^2) = \mathbf{g}(\mathbf{x}^1) - \mathbf{g}(\mathbf{x}^2)$ and a vector $\bar{\mathbf{b}}$ such that both $\mathbf{x}^1$ and $\mathbf{x}^2$ are zeros of the mapping $\mathbf{D}\mathbf{x} + \mathbf{A}\mathbf{x} - \bar{\mathbf{b}}$. This affine mapping satisfies the same properties as (2.4) and therefore has a solution for almost all values of $\bar{\mathbf{b}}$. This is only possible if $\det(\mathbf{D} + \mathbf{A}) \neq 0$, and hence $\mathbf{x}^1$ and $\mathbf{x}^2$ must coincide. These considerations lead to the classical result that (2.4) with $\mathbf{g}(\cdot)$ strictly increasing and mapping R^n onto itself, and with $\mathbf{A}$ a P_0 matrix has a unique solution [84].

2.3 Other types of variable dimension algorithms

The octahedral algorithm is just one of a number of algorithms that share the same variable dimension feature. These algorithms are often classified according to the number of one-dimensional rays along which the path can leave the origin because for general purpose applications, the larger the number of rays, the shorter the path can be expected to be. The octahedral algorithm for example has 2^n rays (one in the middle of each orthant). The purpose of this section is to show that for the other variable dimension algorithms a similar circuit equivalent can be derived by combining the same elements of figure 2.2.

2.3.1 The $2n$-ray algorithm

This method [121] is dual to the octahedral algorithm. We take the same auxiliary branches as in figure 2.2 but append the left ones to the voltage-controlled ports and the right ones to the current-controlled ports (i.e. interchange x_k and y_k in figure 2.2). Initially $t = \infty$, all diodes at current-controlled (voltage-controlled) ports are blocking (conducting) and the circuit operates at $\mathbf{x} = \mathbf{0}$. The algorithm tries to decrease t until $t = 0$ and $\mathbf{y} = \mathbf{0}$. The description is dual to the octahedral algorithm with the definitions of $X(\mathrm{s})$ and $Y(\mathrm{s})$ interchanged. If port k is a current-controlled port (voltage-controlled port) then

- $s_k = +1$ if diodes D'_k are on (off) and diodes D_k are off (on),
- $s_k = -1$ if diodes D_k are on (off) and diodes D'_k are off (on),
- $s_k = 0$ if all diodes are off (on).

When the diodes are in state s, a point $(\mathbf{x}, \mathbf{y})$ is admissible if it belongs to $Z(\mathrm{s})$ defined as follows (see figure 2.5) :

- $X(\mathrm{s}) = \{\mathbf{x} \in R^n \mid x_k = 0 \text{ if } s_k = 0, s_k x_k \geq 0 \text{ if } s_k \neq 0\}$
- $Y(\mathrm{s}) = \{\mathbf{y} \in R^n \mid s_k y_k = -\|\mathbf{y}\|_\infty, \text{ if } s_k \neq 0\}$,
- $Z(\mathrm{s}) = X(\mathrm{s}) \times Y(\mathrm{s})$.

No detailed analysis will be undertaken here but one can verify that the path can leave the origin along $2n$ rays (each coordinate axis in both directions).

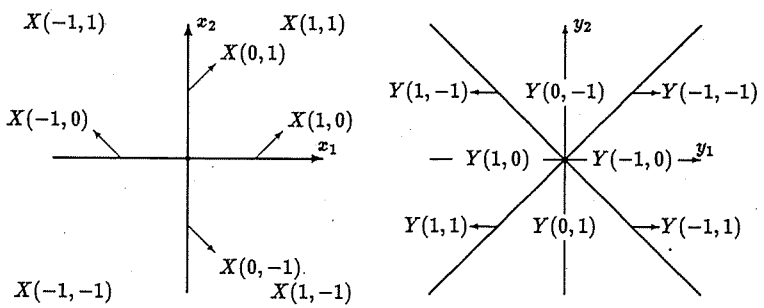


Figure 2.5: The sets $X(\mathrm{s})$ and $Y(\mathrm{s})$ for the $2n$-ray algorithm.

2.3.2 The $(3^n - 1)$-ray algorithm

This is a combination of the two previous schemes [55]. Figure 2.6 shows the auxiliary branches for current-controlled ports (left) and voltage-controlled ports (right). The scheme is parametrized by a number γ with $0 < \gamma < 1/n$. The limiting cases $\gamma = 0$ and $\gamma = 1/n$ correspond to the 2^n- and the $2n$-ray algorithm, respectively. As in the octahedral algorithm, t is increased from zero to infinity. The states of the diodes will be characterized by two sign vectors s and t with $t_k = s_k$ if $t_k \neq 0$: if port k is current-controlled (voltage-controlled), then

- $s_k = +1$ if diodes D_k are off (on) and diodes D'_k on (off),
- $s_k = -1$ if diodes D_k are on (off) and diodes D'_k off (on),
- $s_k = 0$ if diodes D_k and D'_k are off (on),
- $t_k = 0$ if diode D''_k is on (off),
- $t_k = s_k \neq 0$ if diode D''_k is off (on).

A point $(\mathbf{x}, \mathbf{y})$ is admissible in state (s, t) if it belongs to $Z(\mathrm{s}, \mathrm{t})$ defined as follows :

- $X(\mathrm{s}, \mathrm{t}) = \{\mathbf{x} \in R^n \mid \exists t \geq 0 : s_k x_k = t \text{ if } t_k = s_k \neq 0;\ 0 \leq s_k x_k \leq t \text{ if } t_k = 0 \neq s_k;\ x_k = 0 \text{ if } t_k = s_k = 0\}$.

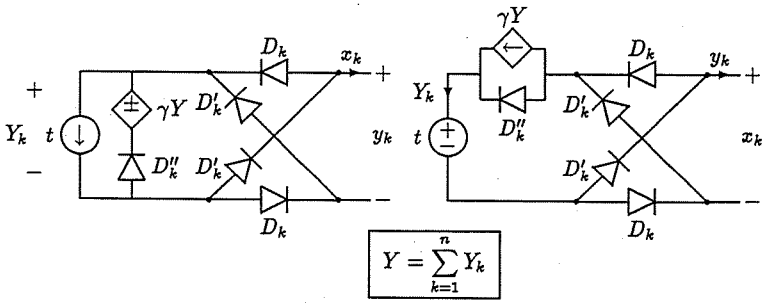


Figure 2.6: Auxiliary branches for the $3^n - 1$ algorithm (γ is a parameter between 0 and $1/n$).

- $Y(\mathbf{s},\mathbf{t}) = \{\mathbf{y} \in R^n \mid \exists Y \geq 0 : Y = -\sum t_k y_k + |\mathbf{t}|\gamma Y$ and $-s_k.y_k \geq \gamma.Y$ if $t_k = s_k \neq 0$; $s_k.y_k = -\gamma.Y$ if $t_k = 0 \neq s_k$; $|y_k| \leq \gamma.Y$ if $s_k = t_k = 0\}$,
- $Z(\mathbf{s},\mathbf{t}) = X(\mathbf{s},\mathbf{t}) \times Y(\mathbf{s},\mathbf{t})$.

As before $|\mathbf{t}|$ denotes the number of zero elements in the vector $\mathbf{t}$. Figure 2.7 illustrates this idea for $n = 2$ and $\gamma = 1/3$.

2.3.3 The 2-ray algorithm

This algorithm was independently described by Yamamoto [130, 131] and Saigal [83]. It is the recursive extension of the idea of Chao, Liu and Pan described in Section 2.1. Figure 2.8 shows the auxiliary branches. Here a different parameter t_k is used at each port and these parameters are adjusted one after the other. The state of the diodes is characterized by an index $k \in \{1,\ldots,n\}$ and a number $\alpha \in \{-1,+1\}$:

- all diodes at a current-controlled (voltage-controlled) port j are conducting (blocking) if $j < k$,
- all diodes at a current-controlled (voltage-controlled) port j are blocking (conducting) if $j > k$,
- if $\alpha = +1$ and port k is current-controlled (voltage-controlled), then diodes D'_k are on (off) and diodes D_k are off (on),

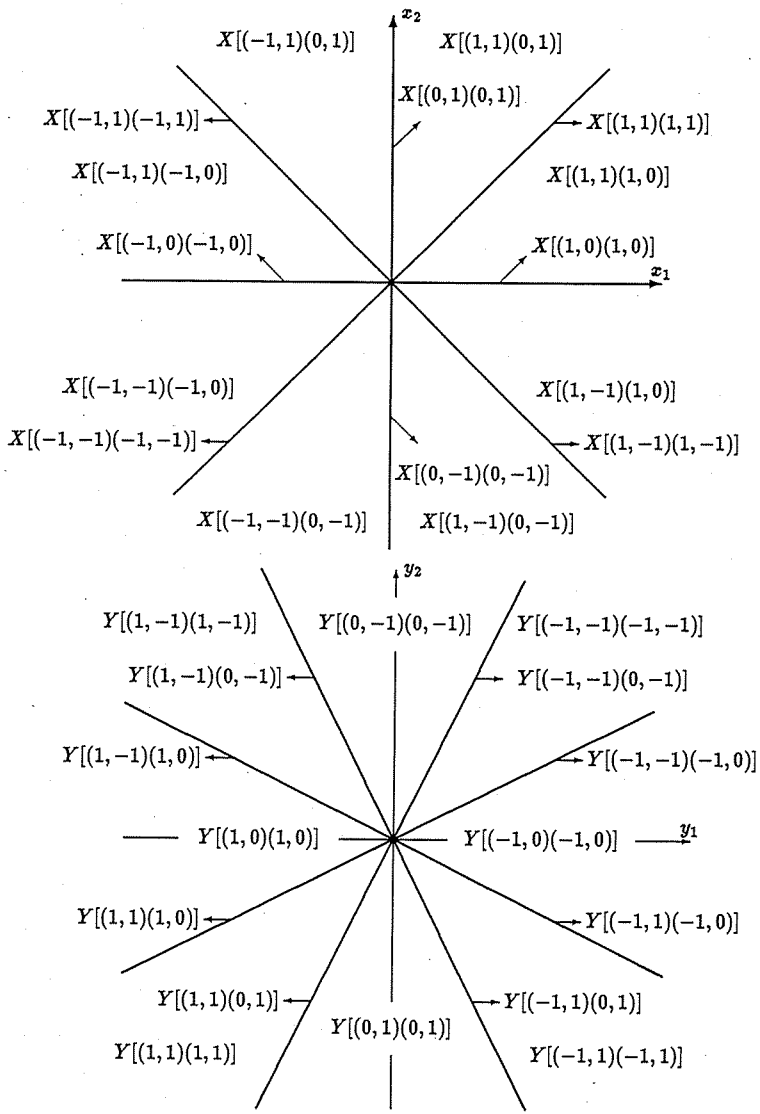


Figure 2.7: The sets $X(\mathbf{s},\mathbf{t})$ and $Y(\mathbf{s},\mathbf{t})$ in the (3^n-1)-ray algorithm for $n = 2$.

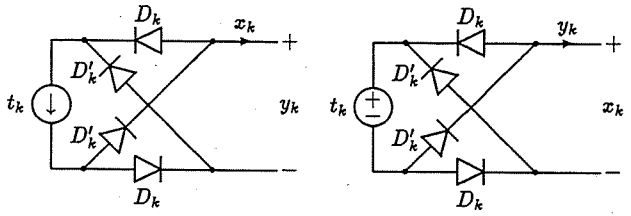


Figure 2.8: Auxiliary branches for the 2-ray algorithm.

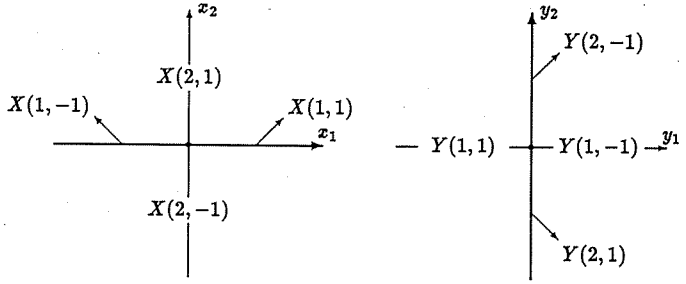


Figure 2.9: The sets $X(k,\alpha)$ and $Y(k,\alpha)$ in the 2-ray algorithm for $n=2$.

- if $\alpha = -1$ and port k is current-controlled (voltage-controlled), then diodes D_k are on (off) and diodes D'_k are off (on).

The admissible points are then characterized by (see figure 2.9)

- $X(k,\alpha) = \{\mathbf{x} \in R^n \mid \alpha x_k \geq 0,\ x_j = 0 \text{ if } j > k\}$,
- $Y(k,\alpha) = \{\mathbf{y} \in R^n \mid \alpha y_k \leq 0,\ y_j = 0 \text{ if } j < k\}$,
- $Z(k,\alpha) = X(k,\alpha) \times Y(k,\alpha)$.

Figure 2.10 shows the result of this algorithm when applied to the example of (1.2). The path can be followed from breakpoint to breakpoint, in a similar way as in Section 2.2.1. The difference is that now one keeps $t_2 = 0$ as long as $f_1 \neq 0$, and only increases t_2 if $f_1 = 0$.

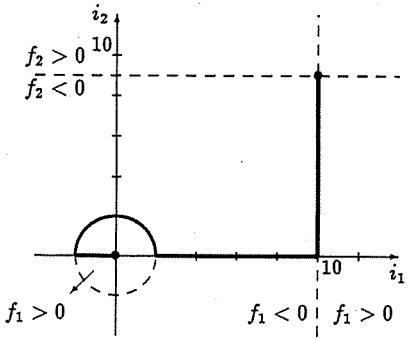


Figure 2.10: The path followed by the 2-ray algorithm when applied to the equations (1.2).

2.4 Convergence in circuits with two-terminal elements

The octahedral algorithm can be applied to most types of circuit analysis by inserting the auxiliary branches of figure 2.2 at appropriate places in the circuit. We take (modified) node analysis as an example.

Node analysis. Here the variable x_k is the voltage at the k-th node and $y_k = f_k(\mathbf{x})$ is the sum of all currents from node k to other nodes in the network. Application of the algorithm would then amount to inserting an auxiliary branch between each node and the ground as in figure 2.11. Independent voltage sources may be added in series with each of these branches to account for an initial guess different from zero.

Modified node analysis. Assume one has a current-controlled resistor $v = \hat{v}(i)$ connected between nodes j and k. This can be handled by modifying the branch as in figure 2.12 and adding the equation $\hat{v}(\tilde{x}) + x_j - x_k = 0$, where $\tilde{x}$ is the branch current.

A similar approach can be given for other common types of circuit analysis. As follows from the discussion in Section 2.2, the question concerning convergence of the algorithm amounts to checking whether the solution set of a circuit modified in the way of figs. 2.11 or 2.12 is bounded for $t \in [0,\infty)$. This can be shown for a broad class of circuits as will now be done.

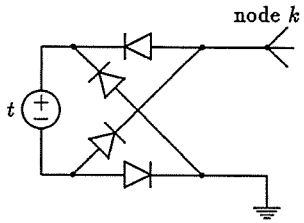


Figure 2.11: Modification of the circuit corresponding to the equation for node k in nodal analysis.

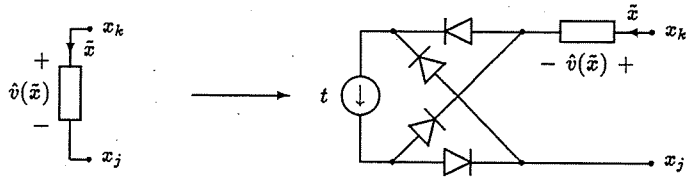


Figure 2.12: Modification corresponding to the branch equation of a current-controlled resistor.

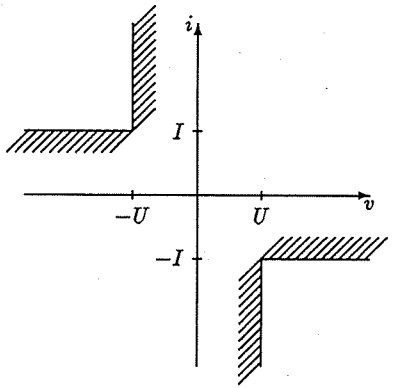


Figure 2.13: The characteristic of a weakly active resistor lies in the shaded region.

In this section we consider networks containing only two-terminal resistors, either current-controlled with all currents admissible or voltage-controlled with all voltages admissible, and with continuously differentiable characteristics. The following definitions will be needed.

Definition 2 [46] *A resistor is weakly active if one can find two numbers* $0 \leq U < \infty$ *and* $0 \leq I < \infty$, *such that there are no admissible points* (v, i) *with* $v > U$ *and* $i < -I$, *or with* $v < -U$ *and* $i > I$.

In other words, the characteristic of a weakly active resistor lies in the shaded area of figure 2.13. The power produced by a weakly active resistor with voltage v and current i is bounded by $\max\{|v|.I, |i|.U\}$.

Definition 3 [30] *A weakly active resistor with characteristic* $f(\rho)$ *is*

of **type U** *if* $f(\rho) \to \infty$ *as* $\rho \to \infty$ *and* $f(\rho) \to -\infty$ *as* $\rho \to -\infty$,

of **type H** *if either* $f(\rho) \to \infty$ *as* $\rho \to \infty$ *and* $|f(\rho)| < M$ *for some* M *as* $\rho \to -\infty$, *or* $f(\rho) \to -\infty$ *as* $\rho \to -\infty$ *and* $|f(\rho)| < M$ *for some* M *as* $\rho \to \infty$,

of **type B** *if* $|f(\rho)| < M$ *for some* M *as* $|\rho| \to \infty$.

Independent voltage (current) sources are treated as weakly active current-controlled (voltage-controlled) resistors of type B. We will discuss convergence of the octahedral algorithm when applied to the fundamental cutset equations obtained as follows (see e.g. [89]). Pick an arbitrary tree in the network and partition the branches in four classes :

1. the voltage-controlled tree resistors, with characteristics $\mathbf{i}_1 = \hat{\mathbf{i}}_1(\mathbf{v}_1)$,
2. the current-controlled tree resistors, with characteristics $\mathbf{v}_2 = \hat{\mathbf{v}}_2(\mathbf{i}_2)$,
3. the voltage-controlled cotree resistors, with characteristics $\mathbf{i}_3 = \hat{\mathbf{i}}_3(\mathbf{v}_3)$,
4. the current-controlled cotree resistors, with characteristics $\mathbf{v}_4 = \hat{\mathbf{v}}_4(\mathbf{i}_4)$,

and write the fundamental cutset matrix as

$$\begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{Q}_{13} & \mathbf{Q}_{14} \\ \mathbf{0} & \mathbf{1} & \mathbf{Q}_{23} & \mathbf{Q}_{24} \end{pmatrix}. \end{array}$$

Then $\mathbf{v}_1, \mathbf{v}_2, \mathbf{i}_2$ and $\mathbf{i}_4$ can be taken as basic variables for the equations

$$\begin{aligned} \mathbf{0} &= \hat{\mathbf{i}}_1(\mathbf{v}_1) + \mathbf{Q}_{13}\hat{\mathbf{i}}_3(\mathbf{Q}_{13}^t\mathbf{v}_1 + \mathbf{Q}_{23}^t\mathbf{v}_2) + \mathbf{Q}_{14}\mathbf{i}_4 && (2.10)\\ \mathbf{0} &= \mathbf{i}_2 + \mathbf{Q}_{23}\hat{\mathbf{i}}_3(\mathbf{Q}_{13}^t\mathbf{v}_1 + \mathbf{Q}_{23}^t\mathbf{v}_2) + \mathbf{Q}_{24}\mathbf{i}_4 && (2.11)\\ \mathbf{0} &= \hat{\mathbf{v}}_2(\mathbf{i}_2) - \mathbf{v}_2 && (2.12)\\ \mathbf{0} &= \hat{\mathbf{v}}_4(\mathbf{i}_4) - \mathbf{Q}_{14}^t\mathbf{v}_1 - \mathbf{Q}_{24}^t\mathbf{v}_2 && (2.13) \end{aligned}$$

The octahedral algorithm can be applied to these equations by modifying the branches of the circuit as in figure 2.14. If each composite branch of figure 2.14 is treated as a single branch, this corresponds to modifying the resistor characteristics as in figure 2.15.

Example. Applied to the circuit of figure 2.16 one gets the following equations for the tree $\{1,2,3\}$ (assuming that the diodes are of the exponential type and the resistors 4 and 5 are voltage-controlled of type U) :

$$\begin{aligned} \hat{\imath}_1(v_1) + \hat{\imath}_4(v_1 - v_2) + I_2 &= 0,\\ \hat{\imath}_2(v_2) - \hat{\imath}_4(v_1 - v_2) + \hat{\imath}_5(v_2 - v_3) &= 0,\\ -I_1 - \hat{\imath}_5(v_2 - v_3) - I_2 &= 0. \end{aligned}$$

The modifications of fig. 2.15 lead to fig. 2.17. In order to prove convergence one has to show that the branch currents and voltages in this modified circuit

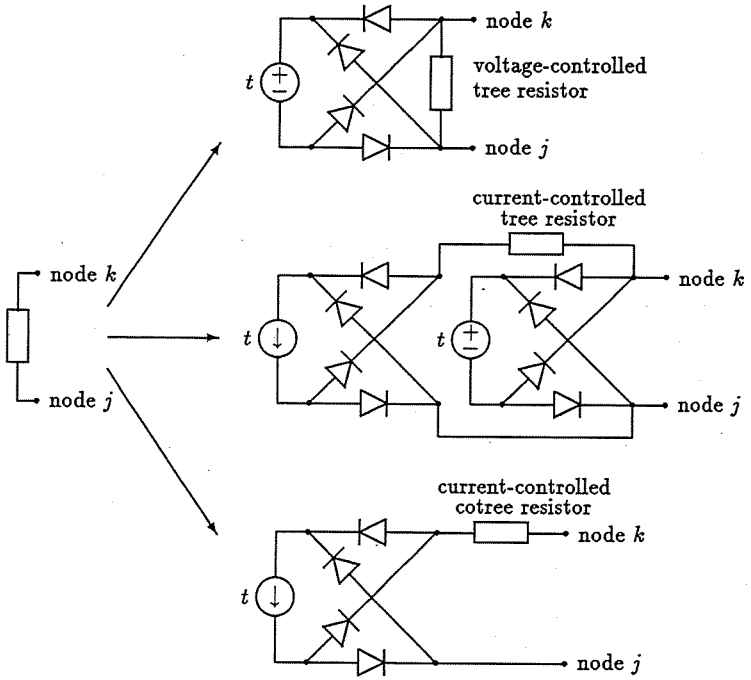
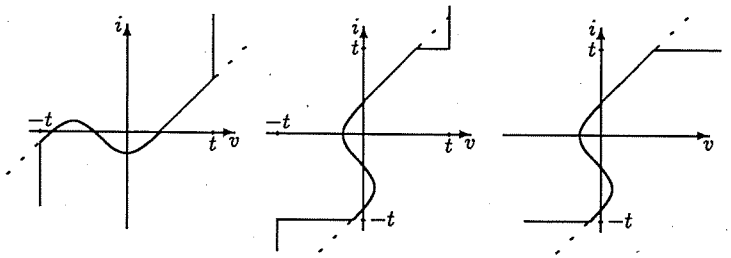


Figure 2.14: Modification of the circuit for solving the cut-set equations (2.10)-(2.13). The voltage-controlled cotree resistors are not modified.



v.c. tree resistor c.c. tree resistor c.c. cotree resistor

Figure 2.15: Original characteristic (dashed lines) and modified characteristics (solid line) for the composite resistors of figure 2.14.

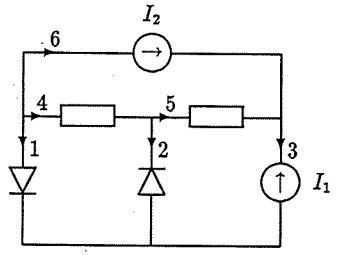


Figure 2.16: Example with two voltage-controlled nonlinear resistors and two exponential diodes.

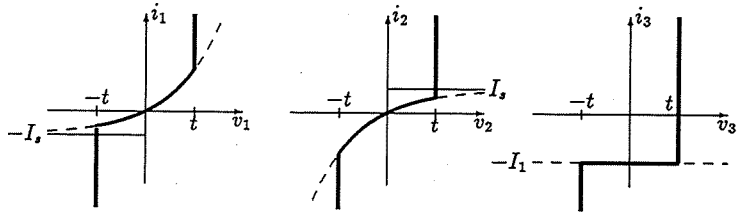


Figure 2.17: Modifications of the resistor characteristics in fig. 2.16.

remain bounded for all values of t. Assume for instance that the voltage across the cutset {1,2,3} grows unboundedly i.e.,

$$v_1 \to \infty, \quad v_2 \to \infty, \quad v_3 \to \infty. \tag{2.14}$$

Then from figure 2.17 one can observe that i_2 and i_3 are bounded below ($i_2 > 0$ and $i_3 \geq -I_1$) while i_1 increases unboundedly. This is impossible since Kirchhoff's current law requires that $i_1+i_2+i_3 = 0$, and hence the assumption (2.14) leads to a contradiction. A similar argument holds for the opposite orientation $v_1 \to -\infty$, $v_2 \to -\infty$ and $v_3 \to -\infty$, and for each of the other cutsets in the circuit. This shows that all voltages in the circuit must remain bounded. A dual reasoning holds for the branch currents.

If the circuit topology is changed by reversing the diode in branch 1, then (2.14) becomes perfectly possible. Currents i_1 and i_2 are both bounded below and this lower bound approaches I_s, while $i_3 \geq -I_1$. Convergence will now depend on the actual values of I_s and I_1 : if $I_1 \geq 2I_s$, then one can have (2.14) without violating Kirchhoff's current law (but in that case no solution exists). If $I_1 < 2I_s$, the solution set will again be bounded and the algorithm would converge. Hence we notice that convergence sometimes depends on the numerical values of the circuit parameters. In the analysis below such quantitative information will not be taken into account, mainly for reasons of clarity. The conditions derived will exclude the circuit of fig 2.16 (with both diodes directed similarly) because of the presence of a cutset composed of a voltage-controlled resistor of type B, and of two similarly directed voltage-controlled of type H.

The following theorem formulates the general form of this condition. It turns out to be a well known condition for existence of a solution to the equations (2.10)–(2.13) (see [17, 46]).

Theorem 4 *Suppose the circuit is composed of weakly active resistors, either voltage-controlled with all voltages admissible or current-controlled with all currents admissible. Suppose the following two topological conditions are satisfied :*

1. *There is no cutset exclusively composed of voltage-controlled resistors of type B and similarly directed voltage-controlled resistors of type H.*
2. *There is no loop exclusively composed of current-controlled resistors of type B and similarly directed current-controlled resistors of type H.*

Then all branch voltages and branch currents in the circuit modified as in figure 2.14 remain bounded for $0 \leq t < \infty$. *Hence the circuit has at least*

one solution and provided the regularity assumptions 1–4 hold, the octahedral algorithm converges.

Proof. Let b be the number of branches in the circuit and denote by S_t the set of all operating points $(\mathbf{v},\mathbf{i}) \in R^{2b}$ of the modified circuit in figure 2.14 for some fixed value of t, and by S the union of all $S_t, t \in [0,\infty)$. We have to prove that S is bounded. The reasoning is similar to the proof of Theorem 3.4.14 in [46].

First note that, since the parameter t is the infinity norm of the unknowns in eqs. (2.10)–(2.13), each set S_t must be bounded and S can only be unbounded as t increases to infinity.

The proof is based on the following observations.

- At each operating point of the circuit one has a uniform cutset (uniform with respect to the voltages) of branches with voltages of at least $(1/b).\|\mathbf{v}\|_\infty$. This is an immediate consequence of the coloured branch theorem (see Appendix B), if the branches with voltage greater than or equal to $(1/b).\|\mathbf{v}\|_\infty$ are directed as their voltage and coloured green, and the other branches are coloured red. Since a uniform loop of green and red branches through the branch with voltage $\|\mathbf{v}\|_\infty$ would violate Kirchoff's voltage law, there must be a uniform cutset of green branches. Because of the first topological condition in the theorem, this cutset must contain a current-controlled resistor or a voltage-controlled resistor of type U, or a voltage-controlled resistor of type H with its direction of unbounded current coinciding with the orientation of the cutset. It is easy to verify that the (i,v)-characteristics of each of these three element types, possibly modified in one of the ways shown in figure 2.14, have the property that

  $$\forall M > 0 \quad \exists T, R > 0 \ : \ i > M \text{ whenever } v > R \text{ and } t > T, \qquad (2.15)$$

  for any orientation of the current-controlled or the voltage-controlled type U resistors, and for the voltage-controlled type H resistors if they are oriented in the sense of unbounded current. To sum up, at every operating point one can find a branch with voltage at least $(1/b).\|\mathbf{v}\|_\infty$, and if this branch is oriented in the direction of positive voltage, its characteristic has property (2.15).

- By a dual reasoning one can show that it is always possible to find a branch with current at least $(1/b).\|\mathbf{i}\|_\infty$, and if this branch is oriented in the direction of positive current, its characteristic has the property that

  $$\forall M > 0 \quad \exists T, R > 0 \ : \ v > M \text{ whenever } i > R \text{ and } t > T. \qquad (2.16)$$

Now, assume S is unbounded, then we can distinguish three cases each of which will be reduced to a contradiction :

case 1. The voltages and the currents grow unboundedly, i.e.

$$\forall M, N > 0 \ : \ \exists (\mathbf{v},\mathbf{i}) \in S \text{ with } \|\mathbf{v}\|_\infty > M, \ \ \|\mathbf{i}\|_\infty > N.$$

In this case one can find a point $(\mathbf{v},\mathbf{i}) \in S$, such that there is a branch k (with property (2.15)) with $v_k.i_k > 0$, $|v_k| \geq (1/b).\|\mathbf{v}\|_\infty$ and $|i_k| > I.b^2$ and a branch j (with property (2.16)) with $v_j.i_j > 0$, $|i_j| \geq (1/b).\|\mathbf{i}\|_\infty$ and $|v_j| > U.b^2$, where U and I are the constants appearing in Definition 2 (without loss of generality we can assume that all resistors are characterized by the same I and U). The power dissipated by these branches is greater than $b.\|\mathbf{v}\|_\infty.I$ and $b.\|\mathbf{i}\|_\infty.U$ resp. and this contradicts the power balance since the power delivered by all other branches together cannot exceed $(b-1).\max\{I.\|\mathbf{v}\|_\infty, U.\|\mathbf{i}\|_\infty\}$.

case 2. The voltages grow unboundedly while the currents remain bounded, i.e.

$$\exists N > 0 \ : \ \forall M > 0 \ : \ \exists (\mathbf{v},\mathbf{i}) \in S \text{ with } \|\mathbf{v}\|_\infty > M \text{ and } \|\mathbf{i}\|_\infty < N.$$

Find a point $(\mathbf{v},\mathbf{i}) \in S$ with $\|\mathbf{v}\|_\infty > N.U/I$, $\|\mathbf{i}\|_\infty < N$ and such that there is a branch k with $v_k.i_k > 0$, $|v_k| > (1/b).\|\mathbf{v}\|_\infty$ and $|i_k| > I.b^2$. The power dissipated by this branch is greater than $b.\|\mathbf{v}\|_\infty.I$. This contradicts the power balance since the power delivered by all other branches together cannot exceed $(b-1).\max\{I.\|\mathbf{v}\|_\infty, U.\|\mathbf{i}\|_\infty\} = (b-1).I.\|\mathbf{v}\|_\infty$.

case 3. The currents grow unboundedly while the voltages remain bounded, i.e.

$$\exists N > 0 \ : \ \forall M > 0 \ : \ \exists (\mathbf{v},\mathbf{i}) \in S \text{ with } \|\mathbf{i}\|_\infty > M \text{ and } \|\mathbf{v}\|_\infty < N.$$

This case is dual to the second. □

The connection between the octahedral algorithm and degree theory, as explained in Section 2.2.7, immediately yields a sufficient condition for uniqueness of the solution.

Theorem 5 [30] *Suppose a circuit as in Theorem 4 is given, but, in addition, the resistors have increasing characteristics and the following two topological conditions are satisfied :*

1. *Every cutset made of voltage-controlled resistors contains at least one strictly increasing resistor.*

2. *Every loop made of current-controlled resistors contains at least one strictly increasing resistor.*

Then the network has a unique solution.

Proof. By Theorem 4 there is at least one solution. Assume the circuit has two solutions $(\mathbf{v}^1, \mathbf{i}^1)$ and $(\mathbf{v}^2, \mathbf{i}^2)$. Then we can replace each strictly increasing resistor by a strictly increasing affine resistor, and each increasing resistor by an increasing affine resistor, and obtain a linear circuit that still possesses the solutions $(\mathbf{v}^1, \mathbf{i}^1)$ and $(\mathbf{v}^2, \mathbf{i}^2)$. Because of the two topological conditions required by this theorem, this linear circuit satisfies the requirements for Theorem 4 and therefore its network equations must have degree $+1$. Hence it can only have one solution and $(\mathbf{v}^1, \mathbf{i}^1)$ and $(\mathbf{v}^2, \mathbf{i}^2)$ must coincide. □

2.5 Circuits with coupling among the resistors

It is straightforward to extend the analysis of eqs. (2.10)–(2.13) and figure 2.14 to include coupling among the resistors. In this section it will be investigated how this affects the convergence of the algorithm. In particular, we aim at proving that the algorithm works for circuits with transistors, positive resistors and independent sources provided a number of quite unrestrictive conditions are met. To this end a generalization of the definitions of the previous section is needed. In this section we call resistor each set of branches with a joint constitutive characteristic $\mathbf{f}(\boldsymbol{\rho})$ where ρ_k and $f_k(\boldsymbol{\rho})$ are dual branch variables.

Definition 4 *A resistor with branch currents* $\mathbf{i}$ *and branch voltages* $\mathbf{v}$ *is weakly active if one can find two numbers* $0 \leq U < \infty$ *and* $0 \leq I < \infty$ *such that the power delivered by the resistor is bounded by* $b.\max\{\|\mathbf{v}\|_\infty.I, \|\mathbf{i}\|_\infty.U\}$, *where* b *is the number of branches of the resistor.*

All passive resistors (for instance transistors) satisfy the above definition. All resistors obtained by placing independent voltage sources (current sources) in series (in parallel) with the branches of a weakly active resistor are also weakly active.

Definition 5 *Assume a weakly active resistor is given with constitutive relation* $\mathbf{f}(\boldsymbol{\rho})$ *and that* $\mathbf{f}(\cdot)$ *is continuous with all values of* $\boldsymbol{\rho}$ *admissible. We say*

that the branches of the resistor can be partitioned in branches of type U, H or B if one can find an orientation of the type H branches s.t.

1. $\|\boldsymbol{\rho}\|_* \to \infty \Rightarrow (\boldsymbol{\rho}^t.\mathbf{f}(\boldsymbol{\rho}))/\|\boldsymbol{\rho}\|_* \to \infty$,
   where $\|\boldsymbol{\rho}\|_*$ *is defined as* $\max_{k\in U, j\in H}\{|\rho_k|, \rho_j\}$,

2. $\|\mathbf{f}(\boldsymbol{\rho})\|_\infty \to \infty \Rightarrow (\boldsymbol{\rho}^t.\mathbf{f}(\boldsymbol{\rho}))/\|\mathbf{f}(\boldsymbol{\rho})\|_\infty \to \infty$.

Example. The second condition in Definition 5 is an important one. For instance it excludes ideal transformers from the analysis below. For an ideal transformer one has $\boldsymbol{\rho} = (v_1, i_2)^t$ and $\mathbf{f}(\boldsymbol{\rho}) = (-ni_2, nv_1)^t$ and therefore $\boldsymbol{\rho}^t.\mathbf{f}(\boldsymbol{\rho}) = 0$ identically.

Example. The passive Ebers-Moll model for a bipolar transistor is given by

$$\begin{pmatrix} i_1 \\ i_2 \end{pmatrix} = \begin{pmatrix} f_1(v_1, v_2) \\ f_2(v_1, v_2) \end{pmatrix} = \begin{pmatrix} 1 & -\alpha_r \\ -\alpha_f & 1 \end{pmatrix} \begin{pmatrix} m_1[\exp(n_1.v_1) - 1] \\ m_2[\exp(n_2.v_2) - 1] \end{pmatrix},$$

with $m_k.n_k > 0, k = 1, 2$ and $0 < \alpha_r, \alpha_f < 1$. Here the first branch is directed from base to emitter and the second from base to collector. It is well known that this model is passive if $\alpha_r \leq m_1/m_2 \leq 1/\alpha_f$ and $\alpha_r \leq n_1/n_2 \leq 1/\alpha_f$ [41]. Using these inequalities, one can prove that both branches are of type H. We consider only the *npn*-case.

1. The first part of Definition 5 follows from the analysis in [86] : write $\mathbf{v}^t\mathbf{f}(\mathbf{v})$ as

   $$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}^t \begin{pmatrix} 1 & -\alpha_r \\ -\alpha_f & 1 \end{pmatrix} \begin{pmatrix} m_1 g(n_1.v_1) \\ m_2 g(n_2.v_2) \end{pmatrix}$$

   where $g(x) = e^x - 1$ and distinguish three cases :

   $v_1 \geq 0$ *and* $v_2 \geq 0$: Assume $v_2 = \lambda v_1$ for some $0 \leq \lambda \leq 1$. Then

   $$\mathbf{v}^t.\mathbf{f}(\mathbf{v}) = v_1[m_1(1 - \lambda\alpha_f)g(n_1v_1) + m_2(\lambda - \alpha_r)g(n_2\lambda v_1)].$$

   If $\lambda > \alpha_r$ we have $\mathbf{v}^t.\mathbf{f}(\mathbf{v})/v_1 \to \infty$ as $v_1 \to \infty$. If $\lambda \leq \alpha_r$, we must have $g(n_2\lambda v_1) \leq g(n_1v_1)$ because $\alpha_r \leq n_1/n_2$ and hence

   $$\begin{aligned} \mathbf{v}^t.\mathbf{f}(\mathbf{v}) &\geq v_1(m_1 - m_1\lambda\alpha_f + m_2\lambda - m_2\alpha_r)g(n_1v_1) \\ &\geq v_1[(m_1 - m_2\alpha_r) + \lambda(m_2 - m_1\alpha_f)].g(n_1v_1). \end{aligned}$$

   Since $\alpha_r \leq m_1/m_2 \leq 1/\alpha_f$ with at least one strict inequality [41], one can conclude that $\mathbf{v}^t.\mathbf{f}(\mathbf{v})/v_1 \to \infty$ as $v_1 \to \infty$. The case $v_2 \geq v_1$ is analogous.

$v_1 \geq 0$ *and* $v_2 < 0$: Here

$$\begin{aligned} \mathbf{v}^t.\mathbf{f}(\mathbf{v}) &= (m_1 v_1 - \alpha_f m_1 v_2) g(n_1 v_1) + (m_2 v_2 - \alpha_r m_2 v_1) g(n_2 v_2) \\ &\geq v_1 m_1 g(n_1 v_1) - \alpha_r m_2 v_1 g(n_2 v_2) \end{aligned}$$

and hence $\mathbf{v}^t.\mathbf{f}(\mathbf{v})/v_1 \to \infty$ as $v_1 \to \infty$.

$v_2 \geq 0$ *and* $v_1 < 0$: . Similar to the previous case.

2. The second part of Definition 5 follows from the property that

$$\forall \mathbf{c} \in R^2 \ : \ \exists M \ : \ \mathbf{v}^t\mathbf{f}(\mathbf{v}) \geq \mathbf{c}^t\mathbf{f}(\mathbf{v}) \text{ if } \|\mathbf{v}\| > M,$$

which was proved in [86].

Example. The simple two-region model of an nMOS-transistor is

$$\begin{aligned} i_{\rm ds} = 2k'r[(v_{\rm gs} - v_{\rm t})v_{\rm ds} - \frac{1}{2}v_{\rm ds}^2] \quad &\text{if} \quad v_{\rm ds} \leq v_{\rm gs} - v_{\rm t} \\ i_{\rm ds} = k'r(v_{\rm gs} - v_{\rm t})^2 \quad &\text{if} \quad v_{\rm ds} > v_{\rm gs} - v_{\rm t}, \end{aligned}$$

and $i_{\rm gs} = 0$. The transistor can therefore be represented as a passive voltage-controlled resistor with one branch from drain to source and a second from gate to source. As in the ideal transformer case this model does not satisfy Definition 5 because of condition 2 : $\boldsymbol{\rho}^t\mathbf{f}(\boldsymbol{\rho})/\|\mathbf{f}(\boldsymbol{\rho})\|_\infty = |v_{\rm ds}|$, and this can remain bounded even if $\|\mathbf{f}(\boldsymbol{\rho})\|_\infty = |i_{\rm ds}|$ goes to infinity. The problem disappears if the model is replaced by a more realistic one that takes the breakdown of the gate oxide into consideration. If in addition one takes a finite source-to-drain resistance in saturation, then both branches can be treated as branches of type U.

Theorem 6 *Suppose the circuit is composed of weakly active resistors (each branch is either voltage-controlled with all voltages admissible or current-controlled with all currents admissible) and that the branches can be partitioned in branches of type U, H and B. Suppose the following two topological conditions are satisfied :*

1. *There is no cutset exclusively composed of voltage-controlled branches of type B and similarly directed voltage-controlled branches of type H.*

2. *There is no loop exclusively composed of current-controlled branches of type B and similarly directed current-controlled branches of type H.*

Then all branch voltages and branch currents in the circuit modified as in figure 2.14 remain bounded for $0 \leq t < \infty$. *Hence the circuit has at least one solution and the octahedral algorithm converges provided the regularity assumptions 1–4 hold.*

Proof. Similar to Theorem 4. □

Example. It is well known that existence of solutions in transistor networks can usually be taken for granted [86, 17]. This also follows from the previous theorem : a circuit with bipolar transistors, positive linear resistors, exponential diodes, and independent sources has at least one solution if it contains

1. no cutsets exclusively composed of independent current sources and similarly oriented *pn*-junctions (diode or transistor branches),

2. no loops exclusively composed of independent voltage sources.

Before concluding this section, it is interesting to cite a related result from [97].

Theorem 7 *Suppose a circuit is composed by passive elements and independent voltage sources, that each voltage source has a nonzero series resistance, and that each voltage source is connected to a common ground. Then the mapping* $\mathbf{f}(\cdot)$ *describing the nodal equations* $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ *satisfies* $\mathbf{x}^t\mathbf{f}(\mathbf{x}) > 0$ *if* $\|\mathbf{x}\| \to \infty$.

As a consequence of Theorem 2, this is sufficient for convergence of the octahedral algorithm. Theorems 6 and 7 overlap but are not equivalent. In Theorem 6 no *a priori* assumptions are made about the independent sources. Theorem 7 on the other hand allows all passive elements (including for instance ideal transformers).

2.6 Circuits with ideal operational amplifiers

The previous sections established convergence of the algorithm by appealing to the fundamental requirement of *energy conservation*. It should therefore come as no surprise that convergence is much less certain in the presence of active elements such as ideal operational amplifiers or controlled sources. In this section, circuits with ideal operational amplifiers, modelled by nullator-norator pairs [4] will be studied (see fig. 2.18). Beside ideal opamps, weakly active resistors of type U, independent voltage sources and independent current

[4] A nullator is a two-terminal element characterized by $v = i = 0$. A norator is a two-terminal element with both v and i unrestricted.

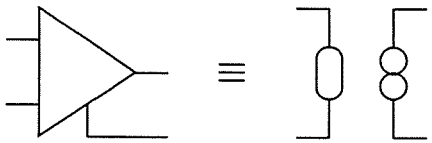


Figure 2.18: Equivalence of an ideal opamp and a nullator-norator pair.

sources will be included in the analysis. This scope is sufficiently broad, since all controlled sources can be realized with ideal opamps and positive resistors. The discussion below can be extended to circuits with resistors of type B (other than independent sources) or H, but this will not be done here for the sake of clarity. So, differently to the previous sections, the independent sources and the other resistors will be treated separately.

Before applying the octahedral algorithm, we need to formulate network equations as in (2.10)–(2.13), and this requires the availability of a pair of conjugate trees.

Definition 6 [46, 45] *Consider the graph of a network composed of resistors, independent sources and an equal number of nullators and norators. Two trees of the graph are called conjugate if one tree includes all nullators, and no norator, the other tree includes all norators and no nullator, both contain all voltage sources and no current sources, and they both include the same resistors.*

Conjugate trees exist in most circuits, except for ill-defined cases such as an opamp without feedback (see [46, 45]). In figure 2.19 for example, branches 1 and 2 and the voltage source form a pair of conjugate trees with the nullator, resp. the norator.

Assume a pair of conjugate trees are given and partition the resistor branches in four classes :

1. the voltage-controlled tree resistors, with characteristics $\mathbf{i}_1 = \hat{\mathbf{i}}_1(\mathbf{v}_1)$,
2. the current-controlled tree resistors, with characteristics $\mathbf{v}_2 = \hat{\mathbf{v}}_2(\mathbf{i}_2)$,
3. the voltage-controlled cotree resistors, with characteristics $\mathbf{i}_3 = \hat{\mathbf{i}}_3(\mathbf{v}_3)$,
4. the current-controlled cotree resistors, with characteristics $\mathbf{v}_4 = \hat{\mathbf{v}}_4(\mathbf{i}_4)$.

Let the fundamental cutset matrix for the tree with the norators be given by :

$$\begin{array}{c} \begin{array}{cccc} \text{volt} & \text{nor} & 1 & 2 \end{array} \\ \left(\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right. \end{array} \begin{array}{cccc} \text{curr} & \text{null} & 3 & 4 \\ \mathbf{Q}_{\text{volt,curr}} & \mathbf{Q}_{\text{volt,null}} & \mathbf{Q}_{\text{volt},3} & \mathbf{Q}_{\text{volt},4} \\ \mathbf{Q}_{\text{nor,curr}} & \mathbf{Q}_{\text{nor,null}} & \mathbf{Q}_{\text{nor},3} & \mathbf{Q}_{\text{nor},4} \\ \mathbf{Q}_{1,\text{curr}} & \mathbf{Q}_{1,\text{null}} & \mathbf{Q}_{13} & \mathbf{Q}_{14} \\ \mathbf{Q}_{2,\text{curr}} & \mathbf{Q}_{2,\text{null}} & \mathbf{Q}_{23} & \mathbf{Q}_{24} \end{array} \left.\vphantom{\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}}\right),$$

and for the tree with the nullators by :

$$\begin{array}{c} \begin{array}{cccc} \text{volt} & \text{nor} & 1 & 2 \end{array} \\ \left(\begin{array}{cccc} 1 & \tilde{\mathbf{Q}}_{\text{volt,nor}} & 0 & 0 \\ 0 & \tilde{\mathbf{Q}}_{\text{null,nor}} & 0 & 0 \\ 0 & \tilde{\mathbf{Q}}_{1,\text{nor}} & 1 & 0 \\ 0 & \tilde{\mathbf{Q}}_{2,\text{nor}} & 0 & 1 \end{array}\right. \end{array} \begin{array}{cccc} \text{curr} & \text{null} & 3 & 4 \\ \tilde{\mathbf{Q}}_{\text{volt,curr}} & 0 & \tilde{\mathbf{Q}}_{\text{volt},3} & \tilde{\mathbf{Q}}_{\text{volt},4} \\ \tilde{\mathbf{Q}}_{\text{null,curr}} & 1 & \tilde{\mathbf{Q}}_{\text{null},3} & \tilde{\mathbf{Q}}_{\text{null},4} \\ \tilde{\mathbf{Q}}_{1,\text{curr}} & 0 & \tilde{\mathbf{Q}}_{13} & \tilde{\mathbf{Q}}_{14} \\ \tilde{\mathbf{Q}}_{2,\text{curr}} & 0 & \tilde{\mathbf{Q}}_{23} & \tilde{\mathbf{Q}}_{24} \end{array} \left.\vphantom{\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}}\right).$$

Then, as in Section 2.4, we take $\mathbf{v}_1, \mathbf{v}_2, \mathbf{i}_2$ and $\mathbf{i}_4$ as basic variables for the equations

$$\mathbf{0} = \hat{\mathbf{i}}_1(\mathbf{v}_1) + \mathbf{Q}_{13}\hat{\mathbf{i}}_3(\tilde{\mathbf{Q}}_{13}^t\mathbf{v}_1 + \tilde{\mathbf{Q}}_{23}^t\mathbf{v}_2 + \tilde{\mathbf{Q}}_{\text{volt},3}^t\mathbf{v}_{\text{volt}}) + \mathbf{Q}_{14}\mathbf{i}_4 + \mathbf{Q}_{1,\text{curr}}\mathbf{i}_{\text{curr}} \quad (2.17)$$

$$\mathbf{0} = \mathbf{i}_2 + \mathbf{Q}_{23}\hat{\mathbf{i}}_3(\tilde{\mathbf{Q}}_{13}^t\mathbf{v}_1 + \tilde{\mathbf{Q}}_{23}^t\mathbf{v}_2 + \tilde{\mathbf{Q}}_{\text{volt},3}^t\mathbf{v}_{\text{volt}}) + \mathbf{Q}_{24}\mathbf{i}_4 + \mathbf{Q}_{2,\text{curr}}\mathbf{i}_{\text{curr}} \quad (2.18)$$

$$\mathbf{0} = \hat{\mathbf{v}}_2(\mathbf{i}_2) - \mathbf{v}_2 \quad (2.19)$$

$$\mathbf{0} = \hat{\mathbf{v}}_4(\mathbf{i}_4) - \tilde{\mathbf{Q}}_{14}^t\mathbf{v}_1 - \tilde{\mathbf{Q}}_{24}^t\mathbf{v}_2 - \tilde{\mathbf{Q}}_{\text{volt},4}\mathbf{v}_{\text{volt}}, \quad (2.20)$$

where $\mathbf{v}_{\text{volt}}$ and $\mathbf{i}_{\text{curr}}$ are the constant voltages and currents of the independent sources. The interpretation of the octahedral algorithm when applied to this set of equations is the same as in figure 2.14. The branches with nullators, norators and independent sources are not modified.

Example. Assume all resistors in the circuit of fig. 2.19 are voltage-controlled of type U. Then equations (2.17)–(2.20) become

$$\begin{aligned} \hat{\imath}_1(v_1) - I &= 0 \\ \hat{\imath}_2(v_2) - \hat{\imath}_3(v_1 + E) &= 0. \end{aligned}$$

Application of the octahedral algorithm amounts to modifying resistors 1 and 2 as the voltage-controlled tree resistor of fig. 2.15. It is easy to check that the resistor currents and voltages are bounded. For instance, if $|v_1| \to \infty$, then $|i_1| \to \infty$ and this leads to a contradiction of Kirchhoff's current law in the cutset formed by branches 1, 4 and the nullator.

The situation changes if the current source is replaced by a fourth resistor (again voltage-controlled and of type U, see fig. 2.20). Now the algorithm may

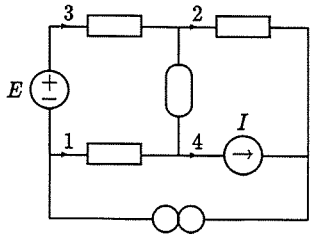


Figure 2.19: Circuit with three voltage-controlled resistors that satisfies the convergence criteria.

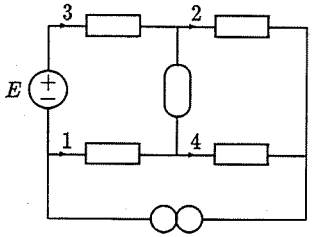


Figure 2.20: Example with four voltage-controlled resistors. Here the octahedral algorithm may diverge for certain choices of the parameters.

diverge because there is no contradiction in $v_1 \to \infty$, $v_2 \to \infty$, $v_3 \to \infty$ and $v_4 \to \infty$.

In general, the topological structure that must be excluded in order to be sure of convergence, is the *uniform partial orientation* defined in [46, 45].

Definition 7 *Assume that a circuit is given with two-terminal resistors, independent voltage and current sources, and an equal number of nullators and norators, then a partial orientation of the resistors*[5] *is uniform if conditions 1 and 2 below are satisfied :*

[5] A partial orientation is obtained by either orienting each resistor in one of both possible ways, or leaving it unoriented.

1. *each oriented resistor is part of a uniform cutset exclusively composed of oriented resistors, norators and independent current sources,*

2. *each oriented resistor is part of a uniform loop exclusively composed of oriented resistors, norators and independent voltage sources.*

The trivial partial orientation where none of the resistors is oriented is always uniform. The following theorem gives a sufficient condition for convergence of the octahedral algorithm applied to eqs. (2.17)–(2.20). Roughly speaking, it states that either the algorithm converges to a solution or it ends up with a uniform partial orientation of the resistors.

Theorem 8 *Assume that a circuit is given with two-terminal weakly active resistors of type U, independent sources, and an equal number of nullators and norators. Suppose a pair of conjugate trees can be found and that there exists no uniform partial orientation of the resistors (except the trivial orientation). Then all branch voltages and branch currents in the circuit modified as in figure 2.14 remain bounded for $0 \leq t < \infty$. Hence the circuit has at least one solution and provided the regularity assumptions 1–4 hold, the octahedral algorithm converges.*

Proof. As in the proof of Theorem 4, each composite branch in figure 2.14 will be treated as one single branch. Let b be the total number of branches, and define S as in the proof of Theorem 4. Assume S is not bounded, then there is at least one orthant $C \subset R^{2b}$ s.t. $C \cap S$ is unbounded. Define two constants M and N such that the branches can be partitioned in four sets as follows :

1. $\alpha = \{k \mid |i_k| \leq N, |v_k| \leq M, \forall(\mathbf{v},\mathbf{i}) \in S \cap C\}$,

2. $\beta = \{k \mid |i_k| \leq N, \forall(\mathbf{v},\mathbf{i}) \in S \cap C$, and $\forall R, \exists(\mathbf{v},\mathbf{i}) \in S \cap C : |v_k| > R\}$. This set can only contain independent current sources and norators,

3. $\gamma = \{k \mid |v_k| \leq M, \forall(\mathbf{v},\mathbf{i}) \in S \cap C$, and $\forall R, \exists(\mathbf{v},\mathbf{i}) \in S \cap C : |i_k| > R\}$. The set γ can only contain independent voltage sources and norators,

4. $\delta = \{k \mid \forall R_1, R_2 : \exists(\mathbf{v},\mathbf{i}) \in S \cap C$ with $|i_k| > R_1$, and $\exists(\mathbf{v},\mathbf{i}) \in S \cap C$ with $|v_k| > R_2\}$. The set δ can only contain resistor branches and norators. If branch $k \in \delta$ is a resistor branch, one has that $v_k.i_k \geq 0\ \forall(\mathbf{v},\mathbf{i}) \in C \cap S$, i.e. either i_k and v_k both go to $+\infty$ or they both go to $-\infty$.

From this we can show that of these four sets only α is nonempty .

- The set δ contains no resistors. Orient all resistors in δ as their voltage and current and do not orient the other branches. Then this partial orientation is uniform as can be seen as follows. Choose any resistor branch $k \in \delta$ and take a point $(\mathbf{v}, \mathbf{i}) \in S \cap C$ with $|v_k| > bR$ and $|i_k| > bR$ and $R > \max(M, N)$. Then we can make the following observations :

  1. Through k there is a cutset of branches with voltages at least R, and with all voltages directed similarly in the cutset (cf. the proof of Theorem 4). Because $R > M$ this cutset can only contain branches of classes δ and β, i.e. oriented resistors, independent current sources and norators. Moreover the orientation of the cutset is such that the first condition of Definition 7 is satisfied.
  2. Similarly, there is a loop through k with branches with currents at least R and with all currents directed similarly in the cutset. This loop satisfies the second condition in Definition 7.

  Since there can be no uniform partial orientation except the trivial one, we conclude that δ contains no resistors.

- The set β is empty. Pick a branch $k \in \beta$ and an operating point $(\mathbf{v}, \mathbf{i}) \in S \cap C$ with $|v_k| > b.M$. Through k there must be a cutset of branches with voltage greater than M, and with all voltages directed similarly in the cutset. This cutset must be composed exclusively of branches of classes δ and β, i.e. norators and independent current sources. The existence of such a cutset is impossible, because the norators and the current sources are in the cotree of one of the conjugate trees. Hence β is empty.

- The proof that γ is empty is dual to the previous case. In a similar way it can be shown that $\delta = \emptyset$ because there can be no cutset and no loop exclusively composed of norators. □

The corresponding uniqueness result is given by the next theorem which is a special case of a result in [45]. The proof is similar to Theorem 5.

Theorem 9 *Suppose a circuit is given containing independent voltage sources, independent current sources, resistors with strictly increasing characteristics, and an equal number of norators and nullators. Suppose a pair of conjugate trees can be found and that there exists no uniform partial orientation of the resistor branches (except the trivial orientation). Then the network has a unique solution.*

2.7 Conclusion

In this chapter a new and circuit-theoretical interpretation of variable dimension algorithms for solving the sets of nonlinear equations that arise from DC-analysis of electrical circuits is given. These methods possess important advantages over the continuous deformation methods that are better known in circuit theory. The principle forms a recursive extension of the idea in [11], and unlike this and other methods, it does not suffer from restart problems. Moreover, as will be shown in the next chapter, variable dimension methods are also faster, because most of the work is carried out in a lower-dimensional space.

Its broad convergence makes the octahedral algorithm a powerful constructive tool for proving existence and uniqueness in resistive circuits. In a circuit with non-coupled, weakly active resistors, convergence is guaranteed in all cases where existence of solutions can be shown on purely topological grounds (Section 2.4). We have also discussed the criteria for convergence in circuits with coupling among the resistors (Section 2.5). These criteria include common electronic circuits. The chapter closes with a condition for convergence in circuits with ideal operational amplifiers (Section 2.6). This condition can be weakened by extending the basic algorithm as will be done in the fourth chapter.

The interpretation of variable dimension algorithms as path-following methods in the $2n$-dimensional space of $(\mathbf{x}, \mathbf{y})$-pairs (the sets $X(\mathbf{s})$, $Y(\mathbf{s})$, ...in Sections 2.2.3–2.3.3) is taken from [53, 54, 55]. The main contribution of the first half of this chapter is to elicit the circuit-theoretical meaning of these algorithms. The proof of the index theorem (Theorem 3) is also original, although the result is hardly surprising since most homotopy methods possess the same property.

The circuit-theoretical equivalent of the octahedral algorithm is of crucial importance for the second half of the chapter where it is indispensable for proving convergence in a purely topological way, i.e. relying on graph theory only. The proof of Theorem 4 is adapted from a similar result of [46], where existence of solutions is proved by considering the homotopy between a linear mapping and the original resistor characteristics. The results of Sections 2.5 and 2.6 are new. In Section 2.5 a topological criterion is given for the existence of solutions in circuits with coupled resistors, by generalizing the definitions of *type U*, *type B*, and *type H* branches. In Section 2.6 the approach of [45, 46] concerning the qualitative study of uniqueness of solutions in circuits with ideal operational amplifiers is extended in several aspects : (1) the resistor characteristics need not be monotonic, (2) we deal with existence of solutions,

and from this deduce uniqueness if the characteristics are monotonic, (3) the proof is constructive. It should also be stressed that discussions on comparable homotopy methods in circuit theory usually treat the convergence conditions in much less detail, by neglecting to formulate the criteria in topological terms, or by excluding active elements such as ideal operational amplifiers or controlled sources from the analysis.

Chapter 3

Numerical implementation

3.1 Introduction

In the previous chapter the octahedral algorithm was shown to provide a unified way to prove existence and uniqueness theorems for resistive circuits. The importance of this approach is that it is constructive and that the algorithm can be carried out in a practically feasible way. This will be the subject for the present chapter.

In order to implement the basic scheme of Section 2.2.6, one needs a numerical technique to trace a one-dimensional curve defined by a set of nonlinear equations. Two classes of methods are available for this : homotopy continuation methods [125, 16] and methods based on simplicial pivoting. This thesis will only deal with the latter category. The general principles are explained by Allgower and Georg in [4, 5] and by Todd in [92]. These methods will be reviewed briefly in Section 3.2. The subsequent sections focus on the application to the octahedral algorithm, as well as on the exploitation of structure typical for electrical circuits. Section 3.5 gives a list of test problems.

3.2 Tracing a one-dimensional path through a triangulated region.

3.2.1 The reflected Freudenthal triangulation

To illustrate the basic concepts, we take one of the simplest triangulations known : the *reflected Freudenthal triangulation*, denoted K'. This will also serve as a building block for the more complicated constructions to follow. Figure 1.6 shows K' for R^2. Each n-simplex in K' can be represented in a

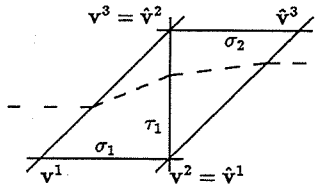


Figure 3.1: Pivoting through a simplicial subdvision.

compact form by a triplet $(\mathbf{v}^1, \pi, \mathbf{t})$ where

- $\mathbf{v}^1$ is a positive integer n-vector,
- π is a permutation of the numbers $1, 2, \ldots, n$ (its members are denoted as $\pi_1, \ldots, \pi_n$),
- $\mathbf{t}$ is an n-vector with elements taken from $\{-1, +1\}$ and indicates the orthant to which the simplex belongs.

Then the $n+1$ vertices of the simplex $\sigma(\mathbf{v}^1, \pi, \mathbf{t})$ are defined by $\mathbf{v}^{k+1} = \mathbf{v}^k + t_{\pi_k} \cdot \mathbf{e}^{\pi_k}$ for $k = 1, \ldots, n$, with $\mathbf{e}^i$ the i-th unit vector. This rule for generating the vertices from the representation $(\mathbf{v}^1, \pi, \mathbf{t})$ is called the *representation rule* for the triangulation K'.

To trace a path through a region of R^n triangulated by K', one has to generate the sequence of adjacent n-simplices through which the path passes. Suppose the path traverses the n-simplex σ_1 represented by $(\mathbf{v}^1, \pi, \mathbf{t})$ and leaves it through the exit facet τ_1 (see fig. 3.1, where $\pi = (1, 2)$). The facet τ_1 is uniquely determined by specifying the vertex $\mathbf{v}^j$ of σ_1 not belonging to τ_1 (in this case $\mathbf{v}^1$). The way j is found will be discussed in the next section. The new n-simplex σ_2 that will be entered next is the unique simplex that shares the facet τ_1 with σ_1 and its representation $(\hat{\mathbf{v}}^1, \hat{\pi}, \hat{\mathbf{t}})$ can be derived easily from $(\mathbf{v}^1, \pi, \mathbf{t})$ and j by the so-called *replacement rules* :

case 1. If $j = 1$: then $\hat{\mathbf{v}}^1 = \mathbf{v}^2$, $\hat{\pi} = (\pi_2, \ldots, \pi_n, \pi_1)$, and $\hat{\mathbf{t}} = \mathbf{t}$,

case 2. If $1 < j < n+1$: then $\hat{\mathbf{v}}^1 = \mathbf{v}^1$, $\hat{\pi} = (\pi_1, \ldots, \pi_{j-1}, \pi_{j+1}, \pi_j, \pi_{j+2}, \ldots, \pi_n)$, and $\hat{\mathbf{t}} = \mathbf{t}$,

case 3. If $j = n+1$ and $v^1_{\pi_n} \neq 0$: then $\hat{\mathbf{v}}^1 = \mathbf{v}^1 - t_{\pi_n} \mathbf{e}^{\pi_n}$, $\hat{\pi} = (\pi_n, \pi_1, \ldots, \pi_{n-1})$ and $\hat{\mathbf{t}} = \mathbf{t}$,

case 4. If $j = n+1$ and $v^1_{\pi_n} = 0$: then $\hat{\mathbf{v}}^1 = \mathbf{v}^1$, $\hat{\pi} = \pi$, and $\hat{t}_{\pi_n} = -t_{\pi_n}$, $\hat{t}_{\pi_k} = t_{\pi_k}$ for $k \neq n$.

Simplices σ_1 and σ_2 will have the n vertices of τ_1 in common but the numbering of the vertices will be different in σ_1 and σ_2. The relationship is represented by a permutation ψ of the numbers $1, 2, \ldots, n+1$, the so-called *vertex correspondence* : $\mathbf{v}^i = \hat{\mathbf{v}}^{\psi_i}$, for $i \neq j$, and $\hat{\mathbf{v}}^{\psi_j}$ is the unique vertex of σ_2 not contained in τ_1. The vertex correspondence for each of the four cases above is as follows :

case 1. If $j = 1$, then $\psi = (n+1, 1, \ldots, n)$,

case 2. If $1 < j < n+1$, then $\psi = (1, 2, \ldots, n+1)$,

case 3. If $j = n+1$ and $v^1_{\pi_n} \neq 0$, then $\psi = (2, \ldots, n+1, 1)$,

case 4. If $j = n+1$ and $v^1_{\pi_n} = 0$, then $\psi = (1, 2, \ldots, n+1)$.

Although the triangulations actually used may be much more complicated than K', it is important to keep in mind that the same basic concepts appear in the description of each triangulation : the representation rule, the replacement rule, and the vertex correspondence.

3.2.2 Pivoting through a triangulated region

The process of generating adjacent simplices as described above, is called *pivoting*. This paragraph describes how the choice of the vertex j to be dropped is made if the path in R^n is defined implicitly as $\mathbf{g}^{-1}(\mathbf{0})$ with $\mathbf{g}(\cdot)$ a mapping from R^n to R^{n-1}. Let $\mathbf{G}(\cdot)$ be the piecewise-linear mapping obtained from $\mathbf{g}(\cdot)$ by identifying $\mathbf{G}(\cdot)$ and $\mathbf{g}(\cdot)$ at the vertices of the triangulation and extending $\mathbf{G}(\cdot)$ affinely on each simplex.

Definition 8 *If σ is a k-simplex ($k = n$ or $k = n-1$) with vertices $\mathbf{v}^1, \ldots, \mathbf{v}^{k+1}$, define the label matrix $\mathbf{A}_\sigma$ of σ as*

$$\mathbf{A}_\sigma = \begin{pmatrix} \mathbf{g}(\mathbf{v}^1) & \mathbf{g}(\mathbf{v}^2) & \ldots & \mathbf{g}(\mathbf{v}^{k+1}) \\ 1 & 1 & \ldots & 1 \end{pmatrix}.$$

The i-th column of $\mathbf{A}_\sigma$ is called the label of vertex $\mathbf{v}^i$ (denoted as $\mathbf{l}(\mathbf{v}^i)$).

Definition 9 *An n- or $(n-1)$- simplex σ is called complete if there exists a nonnegative solution $\boldsymbol{\lambda}$ to $\mathbf{A}_\sigma \boldsymbol{\lambda} = \mathbf{e}^n$.*

Using this vector $\boldsymbol{\lambda}$ one can construct a vector $\mathbf{v} = \mathbf{v}^1\lambda_1 + \cdots + \mathbf{v}^{k+1}\lambda_{k+1}$. The vector $\mathbf{v}$ belongs to σ, and from the definition of $\mathbf{G}(\cdot)$ it follows that $\mathbf{G}(\mathbf{v}) = \mathbf{0}$. Hence we have the following theorem [4, 92].

Theorem 10 *The set $\mathbf{G}^{-1}(\mathbf{0})$ intersects σ if and only if σ is complete.*

For the time being, assume that each complete n-simplex has exactly two complete facets. This is equivalent to saying that $\mathbf{G}^{-1}(\mathbf{0})$ enters and leaves each n-simplex through an $(n-1)$-dimensional face. The following theorem describes how the second complete facet (the *exit facet*) of a complete simplex can be generated given the first (the *entrance facet*).

Theorem 11 [94] *Suppose σ is a complete n-simplex with exactly two complete facets. Let the label matrix of σ be $\mathbf{A}_\sigma$ and suppose two $(n+1)$-vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\beta} \neq \mathbf{0}$ are given with $\mathbf{A}_\sigma \boldsymbol{\lambda} = \mathbf{e}^n$ and $\mathbf{A}_\sigma \boldsymbol{\beta} = \mathbf{0}$. Assume that $\sigma \setminus \mathbf{v}^i$ is a complete facet of σ and that $\beta_i < 0$ (otherwise reverse the sign of $\boldsymbol{\beta}$). Then the other complete facet of σ is $\sigma \setminus \mathbf{v}^j$, determined by*

$$\lambda_j/\beta_j = \min\{\frac{\lambda_k}{\beta_k} \mid \beta_k > 0\}.^1$$

The basic steps in one pivot then become :[2]

1. From the previous pivot, we are given a complete n-simplex σ with entrance facet $\sigma \setminus \mathbf{v}^i$, and a vector $\boldsymbol{\lambda}$ such that $\mathbf{A}_\sigma \boldsymbol{\lambda} = \mathbf{e}^n$. Compute a vector $\boldsymbol{\beta}$ such that $\mathbf{A}_\sigma . \boldsymbol{\beta} = \mathbf{0}$ and $\beta_i < 0$.

2. Apply the *minimum ratio test* : find the index j for which

   $$\lambda_j/\beta_j = \min\{(\lambda_k/\beta_k) \mid \beta_k > 0\}.$$

   Assume this test yields a unique j, otherwise apply lexicographical pivoting as explained below. This determines the exit face $\sigma \setminus \mathbf{v}^j$.

3. From the replacement rules for the triangulation used, find the representation of the simplex $\hat{\sigma}$ that shares the facet $\sigma \setminus \mathbf{v}^j$ with σ. Let ψ be the vertex correspondence for this replacement.

4. Compute the label of vertex $\hat{\mathbf{v}}^{\psi_j}$ of $\hat{\sigma}$. Form the label matrix $\mathbf{A}_{\hat{\sigma}}$ from $\mathbf{A}_\sigma$ by replacing column j by the new label, and by permuting the columns according to ψ. The vector $\hat{\boldsymbol{\lambda}}$ is obtained as $\hat{\lambda}_{\psi_k} = \lambda_k - (\lambda_j/\beta_j).\beta_k$, $k = 1, \ldots, n+1$. Now everything has been prepared for the next pivot : one has a complete n-simplex $\hat{\sigma}$ with entrance facet $\hat{\sigma} \setminus \mathbf{v}^{\psi_j}$, and a vector $\hat{\boldsymbol{\lambda}}$ that satisfies $\mathbf{A}_{\hat{\sigma}} \hat{\boldsymbol{\lambda}} = \mathbf{e}^n$.

[1] One has $\beta_k > 0$ for at least one k, as can be seen from the last equation of $\mathbf{A}_\sigma \boldsymbol{\beta} = \mathbf{0}$.

[2] The question how the process starts and ends, will be dealt with later. For the time being we only consider the steps needed for going from one simplex to the next.

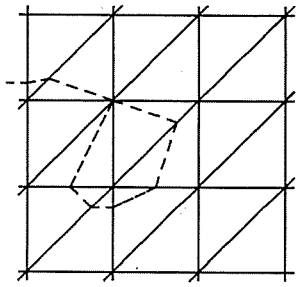


Figure 3.2: Cycling due to degeneracy.

3.2.3 Lexicographical pivoting

The standing assumption on which the preceding algorithm relies is the unique exit-entry property of each complete n-simplex. This property is guaranteed by the construction of the triangulation as long as the minimum ratio test yields unique results. If a tie occurs, we are in the degenerate situation that $\mathbf{G}^{-1}(\mathbf{0})$ intersects a face of dimension lower than $n-1$ and the path is no longer well defined. Phenomena like bifurcations and cycling that were excluded by the unique exit-entry principle become possible (cf. fig. 3.2). This situation is the piecewise-linear counterpart of the problems with regularity assumptions 2–4 in Section 2.2.4. The classical technique for solving this degeneracy is to perturb $\mathbf{G}^{-1}(\mathbf{0})$ to $\mathbf{G}^{-1}([\epsilon, \epsilon^2, \ldots, \epsilon^{n-1}]^t)$ with ϵ an arbitrary small positive number. This is called *lexicographical pivoting* [4, 92].

We say that a matrix $\mathbf{W}$ is *lexicononnegative*, $\mathbf{W} \gg \mathbf{0}$, if each row of $\mathbf{W}$ is zero or has its first nonzero element positive. For a set of row vectors $\mathbf{w}^i$, $i = 1, \ldots, k$ we write $\mathbf{w}^j = \text{lexicomin}\{\mathbf{w}^i \mid i = 1, \ldots, k\}$ if $\mathbf{w}^i - \mathbf{w}^j \gg \mathbf{0}$, $i = 1, \ldots, k$.

Definition 10 *A k-simplex σ ($k = n-1$ or $k = n$) is called very complete if there is a lexicononnegative matrix Λ with columns $\lambda^1, \ldots, \lambda^n$ such that $\mathbf{A}_\sigma \lambda^1 = \mathbf{e}^n$ and $\mathbf{A}_\sigma \lambda^i = \mathbf{e}^{i-1}$ for $i = 2, \ldots, n$.*

Theorem 12 *The set $\mathbf{G}^{-1}([\epsilon, \epsilon^2, \ldots, \epsilon^{n-1}]^t)$ intersects σ for all sufficiently small positive ϵ iff σ is very complete. Each very complete n-simplex has exactly two very complete facets.*

Theorem 11 can now be completed as :

Theorem 13 [94] *Suppose σ is a very complete n-simplex. Let the label matrix of σ be $\mathbf{A}_\sigma$ and suppose an $(n+1) \times n$-matrix Λ and an $(n+1)$-vector $\beta \neq \mathbf{0}$ are given with $\mathbf{A}_\sigma \Lambda = [\mathbf{e}^n, \mathbf{e}^1, \ldots, \mathbf{e}^{n-1}]$ and $\mathbf{A}_\sigma \beta = \mathbf{0}$. Assume that $\sigma \setminus \mathbf{v}^i$ is a very complete facet of σ and that $\beta_i < 0$ (otherwise reverse the sign of β). Then the other very complete facet of σ is $\sigma \setminus \mathbf{v}^j$, determined by*

$$\lambda_j / \beta_j = \text{lexicomin}\{\frac{\lambda_k}{\beta_k} \mid \beta_k > 0\}$$

where the λ_k are the rows of Λ.

Lexicographical pivoting is a very robust but quite expensive solution to degeneracy. This scheme, however, only adds to the computational burden if degeneracy occurs, which may be expected to be unlikely. Additional columns of the matrix Λ should only be generated if a tie in the minimum ratio test occurs and only until the tie is resolved.

3.3 Application to the octahedral algorithm

We now return to the implementation of the octahedral algorithm. The previous discussion leaves two questions unanswered if one wants to apply the triangulation K' to this algorithm.

1. How can the pivoting process be started and when does it end ? The discussion above only explains how to continue a path from one simplex into an adjacent one.

2. The main feature of a variable dimension algorithm is that the path in R^n is confined to cells of varying dimension (see figure 1.5). The algorithm must also generate a sequence of simplices of varying dimension. Figure 1.7 shows the example of figure 1.5 for a triangulation with grid size 0.8.

One elegant way to interpret the pivoting process as in figure 1.7 is the following construction [9, 128], which will also clarify the origin of the notion *octahedral algorithm*. Define the n-octahedron O as $O = \text{cvx}\{\mathbf{e}^i, -\mathbf{e}^i : i = 1, \ldots, n\}$ (see figure 3.3 for $n = 2$, O is the unit ball in the 1-norm).

Now we add an $(n+1)$-th coordinate lying between 0 and 1 to all vectors (see figure 3.4), we place the n-octahedron O at level 0 and R^n, triangulated by K', at level 1. Our goal is to construct a subdivision $\mathcal{O}$ of the set $S = \text{cvx}\,((R^n \times \{1\}) \cup (O \times \{0\}))$ in such a way that the restriction of $\mathcal{O}$ to $R^n \times \{1\}$ can be identified with K'. One can then interpret figure 1.7 as if one

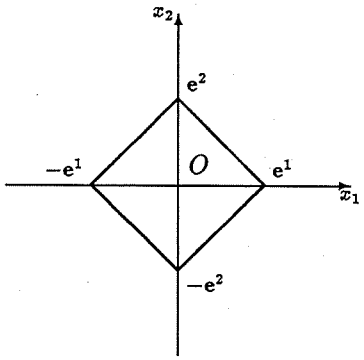


Figure 3.3: The n-octahedron O for $n = 2$.

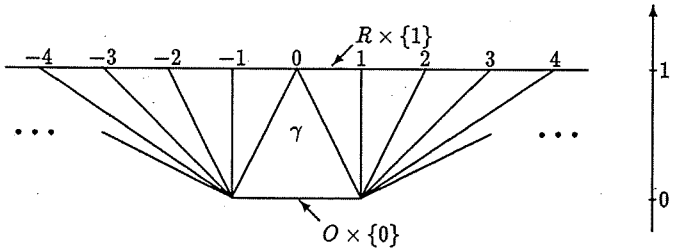


Figure 3.4: The subdivision $\mathcal{O}$ for $n = 1$. One adds an $(n+1)$-th dimension (vertical axis), places the n-octahedron O at level 0 and R^n at level 1 and then subdivides the set $S = \text{ cvx } ((R^n \times \{1\}) \cup (O \times \{0\}))$. For $n > 1$ the cell γ is non-simplicial.

were pivoting through a sequence of $(n+1)$-simplices in $\mathcal{O}$, but only displaying the part of these simplices that lies on $R^n \times \{1\}$.

The subdivision $\mathcal{O}$ contains one non-simplicial cell $\gamma = \text{ cvx } ((O \times \{0\}) \cup (\{\mathbf{0}\} \times \{1\}))$. The other $(n+1)$-cells are simplices and are obtained as follows. Take a sign vector $\mathbf{s} \neq \mathbf{0}$ and let m be the dimension of $X(\mathbf{s})$ (as defined in Section 2.2.3, $m-1$ is the number of zero elements in $\mathbf{s}$). Let σ be any m-simplex of K' that lies in $X(\mathbf{s})$. Let τ be the $(n-m)$-face of O defined as $\tau = \text{ cvx } \{s_k \mathbf{e}^k, \text{ for all } k : s_k \neq 0\}$. Then the simplex cvx $((\sigma \times \{1\}) \cup (\tau \times \{0\}))$ is an $(n+1)$-simplex from $\mathcal{O}$. In this way the complete subdivision $\mathcal{O}$ can be constructed. Its replacement rule is described in Appendix A.

A simplicial algorithm that uses this triangulation, starts at the cell γ and generates a sequence of $(n+1)$-simplices of $\mathcal{O}$ until a simplex with exit facet on $R^n \times \{1\}$ is found. Assume that we want to trace the path of the algorithm in Section 2.2.6 by approximating $\mathbf{f}(\cdot)$ by a piecewise-linear $\mathbf{F}(\cdot)$ on the triangulation K'. The *labelling rule* that will govern the pivoting through $\mathcal{O}$ is derived as follows. Let σ be an m-simplex in $X(\mathbf{s})$ where m is the dimension of $X(\mathbf{s})$ and let $\mathbf{v}^1, \ldots, \mathbf{v}^{m+1}$ be the vertices of σ. Then the path traced by the algorithm of Section 2.2.6 can intersect σ only if convex combinations $\mathbf{x}$ of $\mathbf{v}^1, \ldots, \mathbf{v}^{m+1}$ can be found for which $F_k(\mathbf{x}) = 0$ if $s_k = 0$ and $s_k F_k(\mathbf{x}) \leq 0$ if $s_k \neq 0$. This is equivalent to requiring that the $(n+1) \times (n+2)$ system

$$\sum_{k=1}^{m+1} \begin{pmatrix} \mathbf{f}(\mathbf{v}^i) \\ 1 \end{pmatrix} \mu_k + \sum_{k:s_k \neq 0} \begin{pmatrix} s_k \mathbf{e}^k \\ 0 \end{pmatrix} \nu_k = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} \tag{3.1}$$

has a solution with $\mu_k \geq 0, \nu_k \geq 0$.

This observation leads us to the following labelling rule. Each vertex $\mathbf{v}$ of $\mathcal{O}$ on level $R^n \times \{1\}$ receives a label $\mathbf{l}(\mathbf{v}) = [\mathbf{f}(\overline{\mathbf{v}})^t, 1]^t$, where $\overline{\mathbf{v}}$ is the projection of the $(n+1)$-vector $\mathbf{v}$ on R^n. The label of a vertex $\mathbf{v}$ of $O \times \{0\}$ is $\mathbf{l}(\mathbf{v}) = [\overline{\mathbf{v}}^t \; 0]^t$. The algorithm of Section 2.2.6 then becomes :

1. Take $s_k := 1$ if $f_k(\mathbf{0}) \leq 0$ and $s_k := -1$ if $f_k(\mathbf{0}) > 0$. The first $(n+1)$-simplex σ of $\mathcal{O}$ in the sequence that will be generated has vertices

$$\mathbf{v}^1 = [0 \ldots 0 \; 1]^t, \quad \mathbf{v}^2 = [s_1 \; \ldots \; s_n \; 1]^t,$$
$$\mathbf{v}^3 = [s_1 (\mathbf{e}^1)^t \; 0]^t, \; \ldots \; , \mathbf{v}^{n+2} = [s_n (\mathbf{e}^n)^t \; 0]^t.$$

The label matrix of σ is

$$\mathbf{A}_\sigma := \begin{pmatrix} \mathbf{f}(\mathbf{0}) & \mathbf{f}(\mathbf{s}) & s_1 \mathbf{e}^1 & \ldots & s_n \mathbf{e}^n \\ 1 & 1 & 0 & \ldots & 0 \end{pmatrix}.$$

The system $\mathbf{A}_\sigma \boldsymbol{\lambda} = \mathbf{e}^{n+1}$ has the nonnegative solution $\lambda_1 := 1$, $\lambda_2 := 0$, $\lambda_3 := |f_1(\mathbf{0})|, \ldots, \lambda_{n+2} := |f_n(\mathbf{0})|$. So both σ and its facet $\tau = \sigma \setminus \mathbf{v}^2$ are complete. Set $i := 2$.

2. Calculate a vector β for which $\mathbf{A}_\sigma\beta = \mathbf{0}$ and $\beta_i < 0$. From the minimum ratio test

$$\lambda_j/\beta_j = \min\{(\lambda_k/\beta_k) \mid \beta_k > 0\}.$$

decide which vertex $\mathbf{v}^j$ of σ will be dropped (if necessary, apply the lexicographical minimum ratio test).

3. If the facet $\sigma \setminus \mathbf{v}^j$ lies in $R^n \times \{1\}$, stop : the facet $\sigma \setminus \mathbf{v}^j$ contains a zero of $\mathbf{F}(\cdot)$. Otherwise, use the replacement rules for $\mathcal{O}$ (appendix A) to find the representation of the $(n+1)$-simplex $\hat{\sigma}$ that shares the facet $\sigma \setminus \mathbf{v}^j$ with σ. Let the vertex correspondence for this replacement be ψ.

4. Update $\boldsymbol{\lambda}$ as $\lambda_{\psi_k} := \lambda_k - (\lambda_j/\beta_j)\beta_k$ for $k = 1,\ldots,n$. Update $\mathbf{A}_\sigma$ by evaluating the label of vertex ψ_j of $\hat{\sigma}$ and by reordering the columns according to ψ. Set $\sigma := \hat{\sigma}$ and $i := \psi_j$. Go to step 2.

Remarks.

- The algorithm above starts at $\mathbf{x}^0 = \mathbf{0}$ and makes unit steps in each direction. To start from another point or to set arbitrary steps δ_k, $k = 1,\ldots n$ in each direction, simply modify the labelling of points $\mathbf{v}$ on $R^n \times \{1\}$ as $\mathbf{l}(\mathbf{v}) = [\mathbf{f}(\mathbf{x})^t\ 1]^t$ with $x_k = x_k^0 + \delta_k\overline{v}_k$, $k = 1,\ldots,n$.

- After convergence one can restart the algorithm with the approximate zero as starting point and with smaller grid sizes δ_k for getting a more accurate solution. Note that a similar procedure in Katzenelson's algorithm could suffer from convergence problems because of the restrictions on the starting point mentioned in Section 1.3. We come back to the choice of the refinement rate in Section 3.4.2.

- An alternative to restarting the algorithm is to use a refining triangulation [9], i.e. to put additional layers $R^n \times \{k\}$, $k = 2,3,\ldots$ on top of the subdivision $\mathcal{O}$, and to triangulate these levels with smaller and smaller simplices.

3.4 Pivoting operations on a modified label matrix.

3.4.1 The modified label matrix.

Section 3.2 contains the theoretical steps for pivoting on the label matrix $\mathbf{A}_\sigma$, without numerical details. Straightforward implementation (by updating the

inverse of $\mathbf{A}_\sigma$) could lack numerical stability or be inefficient in the light of the abundant structure present in realistic circuits. The numerical implementation will therefore be commented in this section. The scheme presented here is inspired by the papers of Todd [93, 94]. All computations are carried out on the QR-factorization of a matrix $\mathbf{B}_\sigma$ derived from the label matrix $\mathbf{A}_\sigma$ as follows.

Let σ be an $(n+1)$-simplex of $\mathcal{O}$ with $m+1$ vertices $\mathbf{v}^1,\ldots,\mathbf{v}^{m+1}$ lying in $R^n \times \{1\}$ and $n-m+1$ vertices $\mathbf{v}^{m+2},\ldots,\mathbf{v}^{n+2}$ in $O \times \{0\}$. Then define $\mathbf{B}_\sigma =$

$$\left[\ \mathbf{l}(\mathbf{v}^1)\ \ \mathbf{l}(\mathbf{v}^2)-\mathbf{l}(\mathbf{v}^1)\ \ \ldots\ \ \mathbf{l}(\mathbf{v}^{m+1})-\mathbf{l}(\mathbf{v}^m)\ \ \mathbf{l}(\mathbf{v}^{m+2})\ \ \mathbf{l}(\mathbf{v}^{m+3})\ \ \ldots\ \ \mathbf{l}(\mathbf{v}^{n+2})\ \right].$$

In other words, $\mathbf{B}_\sigma = \mathbf{A}_\sigma.\mathbf{P}$ with

$$\mathbf{P} = \begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & & & m+1 & m+2 & m+3 & & & n+2 \end{array} \\ \begin{pmatrix} 1 & -1 & & & & & & & & \\ & 1 & \ddots & & & & & & & \\ & & \ddots & & -1 & & & & & \\ & & & & 1 & & & & & \\ & & & & & 1 & & & & \\ & & & & & & 1 & & & \\ & & & & & & & \ddots & & \\ & & & & & & & & & 1 \end{pmatrix} \end{array}.$$

The fundamental reason for preferring to work with $\mathbf{B}_\sigma$ rather than with $\mathbf{A}_\sigma$ is that $\mathbf{B}_\sigma$ inherits much of the structure of the Jacobian matrix of the mapping $\mathbf{f}(\cdot)$ owing to the "coordinate"-structure of the triangulations used.

Note that it is as easy to solve linear systems $\mathbf{A}_\sigma\boldsymbol{\beta} = \mathbf{0}$, necessary in the pivoting process, if one has $\mathbf{B}_\sigma$ available instead of $\mathbf{A}_\sigma$: one solves $\mathbf{B}_\sigma\hat{\boldsymbol{\beta}} = \mathbf{0}$ and hence derives $\boldsymbol{\beta} = \mathbf{P}\hat{\boldsymbol{\beta}}$.

Instead of working directly on the matrix $\mathbf{B}_\sigma$ itself, we will store it in a factored form using the QR-factorization : $\mathbf{B}_\sigma\mathbf{S} = \mathbf{QR}$, where $\mathbf{Q}$ is an $(n+1)\times(n+1)$-orthonormal matrix and $\mathbf{R}$ is an $(n+1)\times(n+2)$ upper triangular matrix (by this we mean a square upper triangular matrix with a full column added at the end). The matrix $\mathbf{S}$ is an $(n+2)\times(n+2)$-permutation matrix that reorders the columns of $\mathbf{B}_\sigma$ in order to optimize the efficiency of the updating. One of the functions of this permutation is to keep the columns corresponding to $\mathbf{l}(\mathbf{v}^1)$ and $\mathbf{l}(\mathbf{v}^{m+2})$ in the final two positions of $\mathbf{R}$.

The details of the updating of the QR-decomposition of $\mathbf{B}_\sigma$ when the algorithm moves from one simplex σ to an adjacent $\hat{\sigma}$ will not be included here. Similarly as in [93] the matrix $\mathbf{R}$ is first modified by appending two full columns at the end and deleting two other columns. This matrix is then reduced to

upper-triangular form by a series of Givens-rotations [38]. The cost of this updating is of order $O(n^2)$.

3.4.2 Acceleration by quasi-Newton steps.

One of the benefits of working with $\mathbf{B}_\sigma$ is that each time the algorithm arrives at a complete n-simplex on level $R^n \times \{1\}$ (and thus has found an approximate solution to $\mathbf{f}(\mathbf{x}) = \mathbf{0}$), we have an approximate Jacobian available without additional calculations. If $\mathbf{v}^1, \ldots, \mathbf{v}^{n+1}$ are the vertices of this complete facet then $\mathbf{B}_\sigma$ has the form

$$\left[\mathbf{l}(\mathbf{v}^1) \;\; \mathbf{l}(\mathbf{v}^2) - \mathbf{l}(\mathbf{v}^1) \;\; \mathbf{l}(\mathbf{v}^3) - \mathbf{l}(\mathbf{v}^2) \;\; \ldots \;\; \mathbf{l}(\mathbf{v}^{n+1}) - \mathbf{l}(\mathbf{v}^n) \;\; \mathbf{l}(\mathbf{v}^{n+2}) \right].$$

Each vertex $\mathbf{v}^{j+1}$, $j = 1, \ldots, n$ differs from its predecessor in only one component, say in component π_j and therefore the j-th column of $\mathbf{B}_\sigma$ is a finite difference approximation to $\partial \mathbf{f}/\partial x_{\pi_j}$. Moreover, we have taken care to keep the columns corresponding to $\mathbf{l}(\mathbf{v}^1)$ and $\mathbf{l}(\mathbf{v}^{m+2})$ in the last two positions in $\mathbf{R}$. The QR-factors from the approximate Jacobian matrix can then be retrieved without extra computations from the first n rows and n columns of $\mathbf{Q}$ and $\mathbf{R}$.

This approximate Jacobian matrix can now be used for accelerating the algorithm by a combination with a quasi-Newton method [28, 29]. Quasi-Newton methods have been proposed as an alternative to the classical Newton-Raphson iteration, and avoid the use of derivatives. The rate of convergence is therefore somewhat slower than Newton's method (superlinear instead of quadratic), but the amount of work per iteration step is reduced by one order. For instance, a series of Broyden-updates can be made using the QR-decomposition as in [39]. Each step requires one function-evaluation and $O(n^2)$ operations. If after a number of iterations these steps deteriorate, one proceeds with the simplicial algorithm and uses the information from the quasi-Newton steps to choose the grid size and the starting point of the next cycle. The grid size will depend on the total reduction of the error during the series of quasi-Newton steps, and on the size of the steps made [120]. It is also possible to refine each coordinate at a different rate.

3.4.3 Exploitation of linear structure in the equations.

One final topic still to be discussed is the exploitation of the structure present in electrical circuits. The worst case complexity of traversing a simplicial subdivision can grow as $n!$ as the dimension of the problem increases. It takes $n!$ simplices of unit size to triangulate a unit cube and, in a worst case, one would have to pivot through all of these before the cube is crossed. This

problem was also cited in [132] in a reaction to [12]. The reason for subdividing the domain of the mapping with simplices rather than with larger cells is that it is generally impossible to approximate a nonlinear mapping by a piecewise-linear one that coincides with it on the vertices of the subdivision if the cells of the subdivision are not simplices. However, if the equations have special structure, a piecewise-linear approximation on a simplicial subdivision can have larger pieces of linearity (for instance rectangles if the equations have the form $\mathbf{f}(\mathbf{x}) = \mathbf{f}^1(x_1) + \cdots + \mathbf{f}^n(x_n)$). Two ways for exploiting this phenomenon are

1. A *global* approach [95, 132, 21] is to identify the pieces of linearity (e.g. rectangles) and to traverse each of these by one pivot. This approach is very efficient but it works for a restricted class of problems only.

2. A *local* approach [94] that exploits structure only to the extent that it is encountered by the algorithm. As it moves from one simplex into an adjacent one, the algorithm checks whether the path is actually linear on the union of these two simplices. If it is, the pivot can be carried out at a negligible cost. This approach places no restrictions on the equations and will be adopted here.

The general structure that will be exploited here is as follows. We call two coordinates i and k *independent* if

$$\mathbf{f}(\mathbf{x} + \Delta_i \mathbf{e}^i + \Delta_k \mathbf{e}^k) - \mathbf{f}(\mathbf{x} + \Delta_i \mathbf{e}^i) = \mathbf{f}(\mathbf{x} + \Delta_k \mathbf{e}^k) - \mathbf{f}(\mathbf{x})$$

holds for all Δ_i and Δ_k (loosely speaking these means that there are no cross-terms involving x_i and x_k). In nodal analysis this occurs if there is no coupling between nodes i and k or if they are only coupled by linear elements. The extreme case with $\mathbf{f}(\mathbf{x}) = \mathbf{f}^1(x_1) + \cdots + \mathbf{f}^n(x_n)$ is called *complete separability*.

Now, assume the simplicial algorithm replaces a vertex $\mathbf{v}^j$ of σ and that $\mathbf{v}^j$ differs from its predecessor $\mathbf{v}^{j-1}$ in only one component, π_{j-1}, and from its successor in only one component, π_j. This situation is very common owing to the structure in the triangulation. Then the replacement rule will yield a new simplex $\hat{\sigma}$ with $\mathbf{v}^j$ replaced by $\hat{\mathbf{v}}^j = \mathbf{v}^{j-1} + \mathbf{v}^{j+1} - \mathbf{v}^j$ [9]. If π_j and π_{j-1} are independent, then $\mathbf{f}(\hat{\mathbf{v}}^j) = \mathbf{f}(\mathbf{v}^{j-1}) + \mathbf{f}(\mathbf{v}^{j+1}) - \mathbf{f}(\mathbf{v}^j)$ and the linear approximation on σ will also be valid on $\hat{\sigma}$. In this case, the label matrices $\mathbf{B}_\sigma$ and $\mathbf{B}_{\hat{\sigma}}$ are related very simply because

$$\mathbf{l}(\hat{\mathbf{v}}^j) - \mathbf{l}(\hat{\mathbf{v}}^{j-1}) = \mathbf{l}(\mathbf{v}^{j+1}) - \mathbf{l}(\mathbf{v}^j) \qquad \text{and} \qquad \mathbf{l}(\hat{\mathbf{v}}^{j+1}) - \mathbf{l}(\hat{\mathbf{v}}^j) = \mathbf{l}(\mathbf{v}^j) - \mathbf{l}(\mathbf{v}^{j-1})$$

and no function evaluation or update of the QR-decomposition is needed. The vectors $\hat{\boldsymbol{\lambda}}$ and $\hat{\boldsymbol{\beta}}$ that solve $\mathbf{A}_{\hat{\sigma}}\hat{\boldsymbol{\lambda}} = \mathbf{e}^n$ and $\mathbf{A}_{\hat{\sigma}}\hat{\boldsymbol{\beta}} = \mathbf{0}$ differ from $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$ in

only 3 components :

$$\begin{array}{rclcrcl} \hat{\lambda}_{j-1} & = & \lambda_{j-1}+\lambda_j & & \hat{\beta}_{j-1} & = & \beta_{j-1}+\beta_j \\ \hat{\lambda}_j & = & -\lambda_j & \text{and} & \hat{\beta}_j & = & -\beta_j \\ \hat{\lambda}_{j+1} & = & \lambda_{j+1}+\lambda_j & & \hat{\beta}_{j+1} & = & \beta_{j+1}+\beta_j \end{array} \tag{3.2}$$

and only two new ratios have to be computed for the next minimum ratio test. This makes the amount of work needed in a cheap pivot negligible in comparison with a full pivot. Nevertheless it is important to have an estimate of the cost of a cheap pivot, since, depending on the structure in the equations, the number of cheap pivots may amount to more than 90% of the total number of pivots.

One has the following situation. Denote the i-th ratio as r_i :

$$\begin{aligned} r_i = \lambda_i/\beta_i \quad &\text{if} \quad \beta_i > 0, \\ r_i = \infty \quad &\text{if} \quad \beta_i \leq 0, \end{aligned}$$

for $i = 1, \ldots, n+2$. In a cheap pivot these ratios are processed as follows.

- Find the index j for which r_j is minimal.
- Replace r_j by ∞.
- Replace r_{j+1} and r_{j-1} by their new values (3.2).

The search for the minimum ratio requires $O(n)$ comparisons but can be accelerated for large n by keeping the ratios *partially sorted* in a so-called *priority queue*, see [3]. This amounts to the following. A binary tree is formed, in which each leaf corresponds to an index j, $j = 1, \ldots, n+2$. The tree is as balanced as possible, i.e. each row is complete except possibly the last one. The tree is also *partially ordered*: if leaf j is the parent of leaves k and l, then $r_j \leq r_k$ and $r_j \leq r_l$. This implies that the minimum ratio can always be found at the root of the tree. In order to update the tree if one ratio r_k is changed one needs the following steps. First compare r_k with the value of its parent r_j.

- If $r_k < r_j$, then interchange leaves k and j. Now compare r_k with its new parent, and interchange if necessary. In this way leaf k can be shifted upwards until r_k is greater than or equal to the value of its new parent.
- If $r_k \geq r_j$, then compare r_k with the smallest value of its two children (say r_m and r_n, with $r_m \leq r_n$). If $r_m < r_k$ then interchange leaves k and m. In this way leaf k can be shifted downwards until r_k is less than the values of its two children.

The complexity of this operation is proportional to the number of layers in the tree, i.e. $O(\log_2 n)$. In this way the amount of comparisons needed in a cheap pivot can be reduced from $O(n)$ if the ratios are unsorted to $O(\log_2 n)$.

It is also instructive to compare this strategy for exploiting separability to the one advocated in [132]. The global approach in [132] exploits the fact that for completely separable mappings, the pieces of linearity are *rectangles*. Each of these is crossed in one single step. This method

- avoids the use of cheap pivots alltogether,
- has a slightly more involved minimum-ratio test (involving $2n$ ratios instead of $n+1$),
- has a limited applicability, because it is valid for completely separable mappings only.

It is therefore faster than the local technique described above. In the case of complete separablity the local technique amounts to one full pivot for crossing the first simplex in each rectangle, and then a series of cheap pivots to cross all other simplices encountered in the rectangle. The extra amount of work therefore depends on the average number of simplices intersected by a straight line. Calculations have proved that this average grows as $n^{3/2}$ [79]. The total amount of work to cross the rectangle can therefore be estimated as one full pivot plus $O(n^{3/2}.\log_2 n)$ comparisons. This should be compared with $O(n)$ extra comparisons in the global approach. In spite of this difference in speed, we preferred to use the local technique because of its general validity.

3.4.4 Premultiplying with the inverse Jacobian matrix

It has sometimes been suggested to accelerate restart simplicial methods by premultiplying the mapping $\mathbf{f}(\cdot)$ by the inverse of an approximate Jacobian matrix at the beginning of each major cycle. As shown in Section 3.4.2 a finite-difference approximation of the Jacobian matrix is available under factored form and could be used for this purpose. It can also be shown that the inversion of $\mathbf{J}$ need not be carried out explicitly : instead of using the system of (3.1) with $\mathbf{f}(\cdot)$ replaced by $\mathbf{J}^{-1}\mathbf{f}(\cdot)$, one can as well replace the labelling $s_k\mathbf{e}^k$ of the vertices on level 0 by $s_k\mathbf{J}_{\cdot,k}$.

Nevertheless, premultiplying was not applied in the examples listed below, due to a number of disadvantages. First of all it has been experienced to decrease the efficiency of the algorithm to some extent [120]. Besides, convergence might even be lost, because there is no guarantee that $\mathbf{J}^{-1}\mathbf{f}(\cdot)$ will satisfy the necessary convergence criteria, even if $\mathbf{f}(\cdot)$ does.

An important case where premultiplying is advantageous and maybe inevitable, is described in the next chapter.

3.5 Computational results.

The computational load of the algorithm can now be summarized as follows.

- When no structure is exploited, a pivot requires one function evaluation, and $O(n^2)$ floating point operations.
- The cost of a cheap pivot of the type discussed in the previous paragraph, is $O(\log_2 n)$ comparisons.
- Each quasi-Newton step requires one function evaluation and $O(n^2)$ floating point operations.

Therefore, the number of function evaluations will be used as measure of the complexity. In comparison, each iteration in the Newton-Raphson method requires $n+1$ function evaluations and $O(n^3)$ operations. The amount of work per function evaluation should therefore be roughly the same in both algorithms. Improvement of the efficiency by sparse matrix techniques is not taken into account here.

3.5.1 The test problems in [132]

A homotopy method that has received a lot of interest during the past two decades, is Katzenelson's algorithm. It was first formulated for monotonic equations[3] in [52], and later generalized in [72]. It is usually applied to piecewise-linear mappings on a subdivision by hyperplanes. In [12] it was proposed to use a simplicial subdivision instead, and, in a reaction to this, Yamamura and Horiuchi showed in [132] that considerable savings can be made if the mappings are separable and a rectangular subdivision is used. A comparison with the computational results in their paper will allow us to judge the advantages of the octahedral algorithm over Katzenelson's method.

As a classical measure of the complexity, the number of function evaluations needed for convergence will be counted. By one function evaluation will be meant evaluating $\mathbf{f}(\mathbf{x})$ in a given point $\mathbf{x}$, as well as the evaluation of the difference in function values $\mathbf{f}(\mathbf{x}^1)-\mathbf{f}(\mathbf{x}^2)$ between two adjacent vertices of one simplex. The reason for not counting the latter as two function evaluations, is

[3] in this sense that the determinant of the Jacobian matrix has the same sign on all cells of the subdivision

that adjacent vertices usually differ in one coordinate only, and this makes the evaluation of the difference actually much cheaper than one single evaluation.

As detailed above the amount of work in the octahedral algorithm can be estimated from the number of full pivots and the number of quasi-Newton steps as follows : each full pivot and each quasi-Newton step requires one function evaluation and $O(n^2)$ floating point operations.

Yamamura and Horiuchi also apply a refining subdivision, and list the number of *steps* in each cycle. Per step one function evaluation and $O(n^2)$ floating point operations are needed, as in the octahedral algorithm. To this the amount of work to build the initial factorization should be added, namely n function evaluations, and $O(n^3)$ floating point operations. In the tables below the computational effort per cycle will therefore be estimated as the number of steps $+\ n$.

Example 1

A set of two equations (see fig. 1.3)

$$\begin{aligned} g(x_1)+x_1+x_2 &= 69, \\ g(x_2)+x_1+x_2 &= 75, \end{aligned}$$

with $g(x) = 0.081x^3 - 2.04x^2 + 13.6x$. Table 3.1 compares the results of Katzenelson's algorithm with the octahedral algorithm.[4] Both methods were started from $(0,0)$ with initial grid size 5. The octahedral algorithm is slightly faster. In addition, it was mentioned in [132] that Katzenelson's algorithm fails to converge if $(10,10)$ is taken as starting point. This problem was also mentioned in Chapter 1. The octahedral algorithm is immune to such problems and converges from $(10,10)$ in 13 function evaluations.

Example 2

A system of 50 nonlinear equations

$$g(x_k) + \sum_{i=1}^{50} x_i = k, \quad k = 1,\ldots,50,$$

where $g(x) = 2.5x^3 - 10.5x^2 + 11.8x$. Table 3.2 gives the results for Katzenelson's algorithm, when started at the origin with unit grid size.

[4] As in [132, 120], we will use $\|\mathbf{f}(\mathbf{x})\|_\infty$ as an estimate of the accuracy. Better criteria are given in [29].

cycle	No. steps	No. function eval.	error $\|f(x)\|$
1	7	9	1.0E+01
2	1	3	4.1E+00
3	1	3	8.5E−01
4	1	3	3.2E−02
5	1	3	4.2E−05
6	1	3	7.1E−11
		24	

cycle	Total no. pivots.	No. quasi-Newton steps	No. function eval.	error $\|f(x)\|_\infty$
1	4	7	15	3.2E−12

Table 3.1: Convergence of Katzenelson's algorithm (left) and the octahedral algorithm (right) for Example 1 [132].

cycle	No. steps.	No. function eval.	Error $\|f(x)\|$
1	95	145	1.9E+01
2	1	51	5.5E+00
3	1	51	1.5E+00
4	1	51	4.0E−01
5	1	51	1.0E−02
6	1	51	8.0E−03
7	1	51	5.1E−05
8	1	51	2.1E−09
		502	

cycle	Total no. pivots.	No. quasi-Newton steps	No. function eval.	Error $\|f(x)\|_\infty$
1	1846	1	166	2.0E+00
2	107	8	64	9.7E−09
			230	

Table 3.2: Convergence of Katzenelson's algorithm (left) and the octahedral algorithm (right) for Example 2 [132].

Besides, the authors mention that if a simplicial subdivision is used instead of their rectangular subdivision, the number of steps in the first cycle was 17081.

With the same starting point and the same initial grid size, the octahedral algorithm performs as summarized in table 3.2. The number of function evaluations in the octahedral algorithm is less than one half the number of function evaluations in the rectangular algorithm. Moreover one can make the following important observation : Katzenelson's algorithm on a simplicial subdivision with unit grid size requires 17081 pivots to reach the end simplex. For the octahedral algorithm, this figure is only 1846. Probably, this difference should be explained by the variable dimension feature of the octahedral algorithm. Katzenelson's algorithm generates a sequence of 50-dimensional simplices, whereas the octahedral algorithm exploits the fact that much faster progress can be made along lower-dimensional cells of the subdivision (the dimension of the simplices generated by the octahedral algorithm varies between 1 and 50).

Example 3

This is the four-transistor circuit from [16], shown in fig. 3.5. As mentioned above the algorithm in [132] is restricted to completely separable mappings and this for instance excludes nodal analysis of transistor circuits. Instead of nodal analysis, a set of eight cut-set equations is taken, using the base-emitter and base-collector voltages as unknowns. The rectangular algorithm converges as in table 3.3.

Since the pivot-saving technique used in our implementation is not restricted to separable mappings, the algorithm was run on the nodal equations for this circuit (10 equations). Comparing these results is difficult since the algorithm might have converged to a different solution (it is known from [16] that the circuit has nine solutions).

Example 4

Example 4 is a TTL-inverter circuit with eight transistors [132, fig.4] . Again, the octahedral algorithm was applied to the nodal equations ($n = 14$). The starting point was $x_k = 2.5$, $k = 1, \ldots, n$, and the initial grid size 0.25. The results are summarized in table 3.4.

It is difficult to compare these results with those from [132] because, as in the previous example, another system of equations was used. Moreover only the results for one cycle are listed in [132].

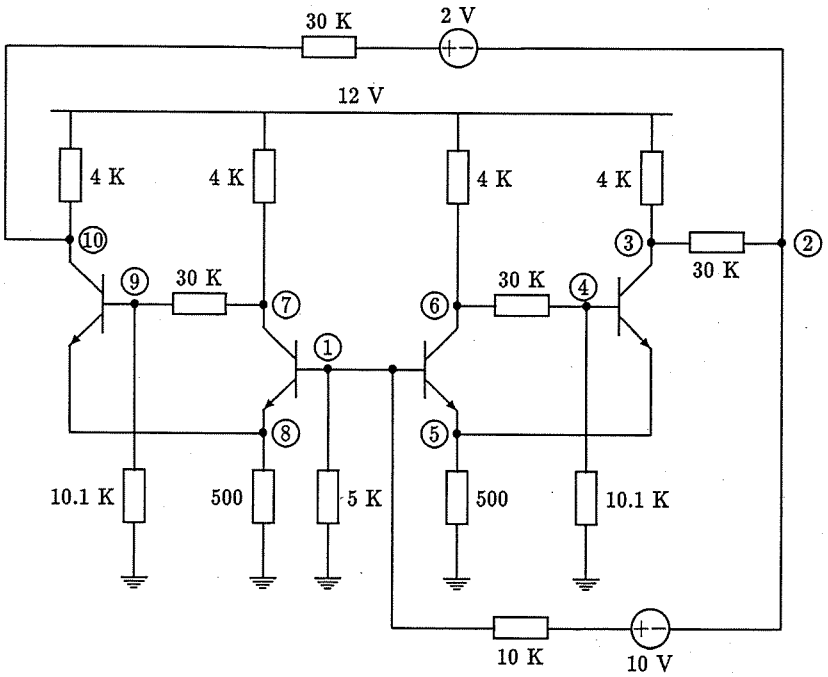


Figure 3.5: Four-transistor circuit of Example 3.

Cycle	No. steps.	No. funct. eval.	Error $\|f(x)\|$
1	23	31	5.5E−03
2	29	37	5.4E−03
3	1	9	5.0E−03
4	5	13	1.7E−03
5	3	11	2.7E−04
6	3	11	2.1E−04
7	1	9	3.6E−05
8	1	9	2.0E−06
9	1	9	5.6E−09
10	1	9	4.3E−14
		146	

Cycle	Total no. pivots.	No. quasi-Newton steps	No. function eval.	Error $\|f(x)\|_\infty$
1	27	1	18	3.2E−01
2	55	0	30	3.1E−01
3	21	0	14	2.7E−02
4	24	0	20	2.1E−01
5	24	2	16	8.3E−02
6	27	5	20	1.7E−09
			118	

Table 3.3: Convergence of Katzenelson's algorithm (left) and the octahedral algorithm for Example 3 [132].

cycle	Total no. pivots.	No. quasi-Newton steps	No. function eval.	Error $\|f(x)\|_\infty$
1	335	1	103	1.1E+00
2	109	1	44	1.0E+00
3	143	1	40	1.0E+00
4	68	1	32	6.5E−01
5	68	1	33	6.8E−01
6	167	1	47	1.1E−01
7	84	1	32	1.5E−03
8	79	1	29	2.4E−09
			362	

Table 3.4: Convergence of the octahedral algorithm for Example 4.

cycle	Total no. pivots.	No. quasi-Newton steps	No. function eval.	Error $\|f(x)\|_\infty$
1	889	0	120	1.8E+00
2	260	0	75	5.3E−01
3	45	0	31	3.6E−01
4	45	5	30	1.1E−09
			246	

Table 3.5: Convergence of the octahedral algorithm for Example 5.

Example 5

A system of 11 separable equations

$$f_i(\mathbf{x}) = x_i - \exp\cos(i.x_{11}), \quad i = 1,\ldots,10, \quad \text{and} \quad x_{11} = \sum_{i=1}^{10} x_i.$$

An accurate result was obtained in [132] in 74 steps (the number of cycles is not mentioned). For the octahedral algorithm one has the results in table 3.5.

3.5.2 Test problems from [4]

Here three test mappings will be studied. They were used in [4] to illustrate the performance of a homotopy continuation method, and in [120] to compare several simplicial variable dimension algorithms (including the octahedral algorithm).

Problem 1 : $f_i(\mathbf{x}) = x_i - \frac{1}{2n}(\sum_{j=1}^n x_j^3 + i)$ for $i = 1,\ldots,n$.

Problem 2 : $f_i(\mathbf{x}) = x_i - \exp\cos(i\sum_{j=1}^n x_j)$ for $i = 1,\ldots,n$.

Problem 3 : $f_1(\mathbf{x}) = \prod_{j=1}^n x_j - 1$, $f_i(\mathbf{x}) = \sum_{j=1}^n x_j + x_i - n - 1$ for $i = 2,\ldots,n$.

The algorithm was run with initial grid size $\delta = 0.5$ in problems 1 and 3, and $\delta = 1$ in problem 2. The starting point is the origin and the program was terminated when $\|\mathbf{f}(\mathbf{x})\|_\infty < 10^{-8}$.

The results are given in table 3.6. Bearing in mind that separability was not exploited in [120], one can conclude that these results are comparable with theirs, and better than those in [4].

		Total no. pivots.	No. quasi-Newton steps	No. function eval.	No. function eval. in [120]	No. Jacobian eval. in [4]
Problem 1	$n = 10$	10	3	17	16	50
	$n = 20$	20	3	27	27	40
	$n = 30$	30	3	37	-	34
	$n = 40$	41	3	48	-	46
	$n = 50$	50	3	57	-	36
	$n = 60$	60	3	67	-	38
	$n = 70$	70	3	77	-	50
	$n = 80$	81	3	88	-	54
	$n = 90$	90	3	97	-	56
	$n = 100$	100	3	107	-	34
Problem 2	$n = 2$	3	4	15	-	44
	$n = 3$	15	7	34	34	221
	$n = 4$	52	5	71	64	223
	$n = 5$	161	6	177	111	513
	$n = 6$	100	5	119	91	1212
	$n = 7$	485	9	486	843	1980
	$n = 8$	653	6	632	-	2974
	$n = 9$	2204	9	2191	-	3690
	$n = 10$	1795	9	1711	-	4644
Problem 3	$n = 5$	10	0	14	23	52
	$n = 10$	20	0	24	50	74
	$n = 15$	30	0	34	71	97
	$n = 20$	40	0	44	93	73
	$n = 25$	50	0	54	-	81
	$n = 30$	60	0	64	-	108
	$n = 35$	70	0	74	-	121
	$n = 40$	98	0	96	-	121
	$n = 45$	117	0	112	-	123
	$n = 50$	108	0	110	-	129

Table 3.6: Convergence of the octahedral algorithm for the three test problems. Comparison with the implementation of the octahedral algorithm in [120] and a homotopy continuation method [4].

	Total no. pivots.	No. quasi-Newton steps	No. function eval.
$n = 10$	1239	5	246
$n = 15$	5086	5	564
$n = 20$	10622	2	1037
$n = 25$	25736	3	1861
$n = 30$	42325	4	2435
$n = 35$	57232	4	2640
$n = 40$	88580	2	3618
$n = 45$	137125	3	5353
$n = 50$	139509	4	4584

Table 3.7: Convergence of the octahedral algorithm for Example 6.

3.5.3 Additional examples

Example 6

Example 5 above was a reformulation of problem 2 in the previous section, in order to obtain a separable mapping. This results in important savings, as can be observed from comparing tables 3.5 and 3.6. We now consider the problem

$$f_i(\mathbf{x}) = x_i - \exp\cos(i.x_{n+1}), \quad i = 1, \ldots, n, \quad \text{and} \quad x_{n+1} = \sum_{i=1}^{n} x_i,$$

for different values of n (see table 3.7). The starting point is $x_k = 0, \quad k = 1, \ldots, n+1$. The initial grid size is one.

Example 7

This 9-node circuit is the example of figure 1a in [49] and contains 8 bipolar transistors. The origin is taken as the starting point for nodal analysis. Obtaining a solution with an accuracy of 8.10^{-12} takes 134 function evaluations and 189 pivots (see table 3.8). To get an idea of the efficiency of the implementation, we ran SPICE on the same example. DC-analysis converged after 10 iterations, and after 26 iterations when the starting point was fixed to the origin with the .NODESET option. These results are comparable with the amount of work in the octahedral algorithm.

cycle	Total no. pivots.	No. quasi-Newton steps	No. function eval.	Error $\|f(x)\|_\infty$
1	49	1	25	3.5E−01
2	25	1	22	3.5E−02
3	24	1	17	3.0E−02
4	32	1	26	2.8E−02
5	38	1	18	1.6E−02
6	21	10	26	8.9E−11
			134	

Table 3.8: Convergence of the octahedral algorithm for Example 7.

cycle	Total no. pivots.	No. quasi-Newton steps	No. function eval.	Error $\|f(x)\|_\infty$
1	15	1	17	1.0E−01
2	12	3	15	5.9E−09
			32	

Table 3.9: Convergence of the octahedral algorithm for Example 8.

Example 8

It is well known that conventional algorithms for solving sets of equations may have problems with MOSFET-circuits, due to ill-conditioning or singularity of the Jacobian matrix. One of the interesting properties of a derivative-free method like the present one is its robustness against such difficulties. As an illustration, consider the two stage Miller opamp of fig. 3.6. Table 3.9 shows the results of a simulation with $V_{\text{IN}} = 0$.

3.6 Conclusions

The implementation described in this chapter combines the robustness of a homotopy algorithm with the asymptotic speed of a locally convergent quasi-

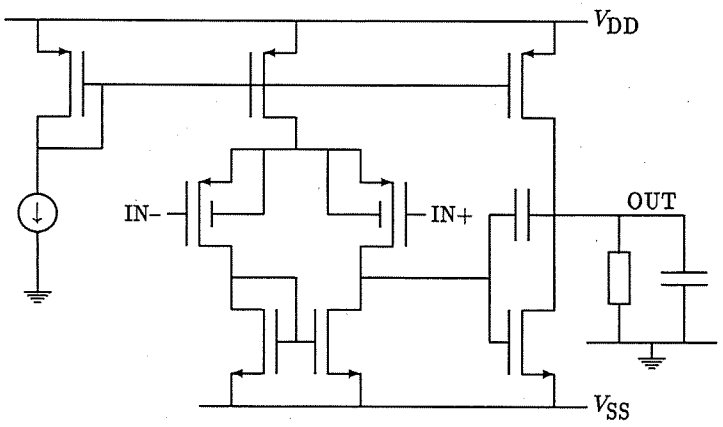


Figure 3.6: Two-stage Miller opamp.

Newton method, and also avoids the use of derivatives. The global rate of convergence is difficult to predict but the method can be seen as a heuristic attempt to improve the efficiency of the path-following process by using a very coarse approximation as long as one is far away from the solution : by approximating the problem by a lower-dimensional one, and by refining the triangulation as one approaches the solution.

From the numerical examples one can draw the following conclusions.

- The octahedral algorithm on a simplicial subdivision is faster than Katzenelson's algorithm on an identical subdivision. This is due to the fact that Katzenelson's algorithm generates a sequence of full-dimensional simplices, while variable dimension algorithms confine the path to spaces of varying dimension.

- Although in [132] Yamamura and Horiuchi use Katzenelson's algorithm as a restart algorithm, convergence of their algorithm is not guaranteed from the second cycle onwards, due to the possibility of cycling. Variable dimension algorithms are true restart algorithms and therefore easy to combine with grid refinement and iterative methods.

- The octahedral algorithm also compares favorably with the homotopy continuation method for which the results were listed in [4]. In view of

the results reported by Trajković et al. in [97, 98] it would be interesting to make a more extensive comparison between homotopy continuation and simplical methods. One could for instance implement the octahedral algorithm using the HOMPACK software package for homotopy continuation [125], and compare the results with the ones given here.

- The structure-exploiting technique advocated here is more general than the one in [132], which is restricted to completely separable mappings (it is therefore not applicable to nodal analysis of transistor circuits). In case of complete separability, the technique in [132] is faster, and should also be recommended for the octahedral algorithm.

- Since the algorithm is derivative-free, it can be applied in cases where the Jacobian matrix is singular at certain points (a situation occurring frequently in MOSFET circuits, unless precautions are taken).

The idea of using the matrix decomposition (QR or LU) of a modified label matrix in the implementation of simplicial algorithms is due to Todd [93]. We adapted his scheme for the octahedral algorithm. One of the modifications is the use of a *rectangular* matrix $\mathbf{B}_\sigma$ instead of a square one. This simplifies the implementation considerably. Acceleration of piecewise-linear methods by using quasi-Newton steps, is a classical technique [4]. The structure-exploiting scheme of Section 3.4.3 is based on [94], except for the arrangement of the ratios in a priority queue.

Chapter 4

DC-Analysis of active circuits

4.1 Introduction

Continuous deformation algorithms for D.C.-analysis exploit eventual passivity of the components and their convergence is therefore not affected by local behaviour. As was shown in Chapter 2 for the octahedral algorithm, the convergence conditions are satisfied in most circuits containing transistors, positive resistors and independent sources. Moreover, in the typical cases where the algorithm fails, a solution does not exist or its existence depends on a very specific choice of the parameters (for example a circuit containing a loop of independent voltage sources).

At the same time however one has to exclude many circuits with active elements such as operational amplifiers or controlled sources. The sufficient condition derived in Section 2.6 for instance excludes the circuit of figure 2.20 although it has a well-defined solution for almost all choices of the parameters. In the case of linear resistors R_1, R_2, R_3 and R_4 one has the equations

$$\begin{aligned} v_1 - \frac{R_3}{R_2} v_2 &= -E, \\ \frac{R_4}{R_1} v_1 - v_2 &= 0, \end{aligned}$$

and the solution is well-defined as long as $R_4.R_3 \neq R_1.R_2$.

The purpose of this chapter is to show how the octahedral algorithm can be extended to process a broad class of active circuits. The basic ideas are exposed in Section 4.2, whereas Section 4.4 elaborates on the significance of these results to electrical circuits. Section 4.5 deals with the possibility of finding multiple solutions. Section 4.3 contains a summary of the method and a number of technical details.

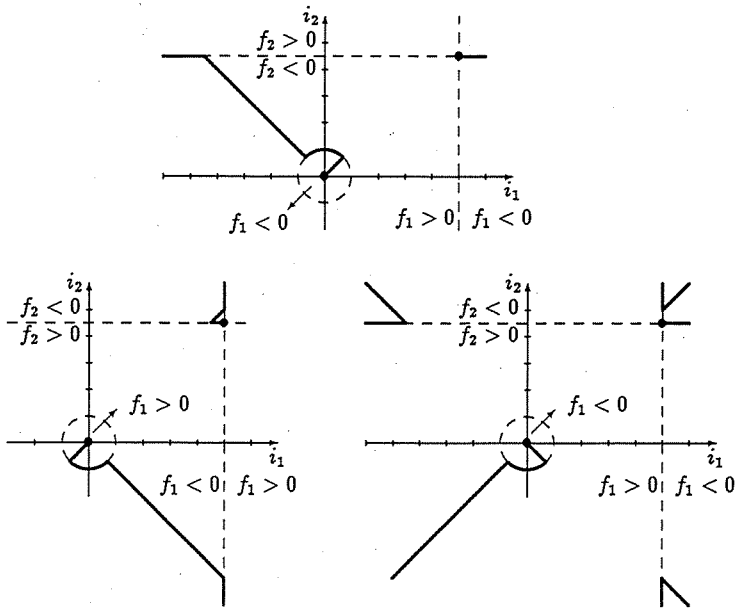


Figure 4.1: Solution set of the modified circuit in figure 2.1 if the signs of $f_1(\cdot)$ and $f_2(\cdot)$ are changed.

4.2 The nonparallel condition as a sufficient condition for existence of solutions

It is instructive to see what happens to the solution set of the modified circuit in figure 2.1 if one changes the signs of the functions $f_1(\cdot)$ and $f_2(\cdot)$. Figure 4.1 shows the effect for

$$\begin{aligned} f_1(i_1, i_2) &= \alpha_1.(i_1^2 + i_2^2 - 4)(i_1 - 10), \\ f_2(i_1, i_2) &= \alpha_2.(i_2 - 9), \end{aligned} \tag{4.1}$$

and $(\alpha_1, \alpha_2) = (-1, +1)$, $(+1, -1)$ and $(-1, -1)$, respectively. In each of these cases the algorithm diverges because the origin and the solution lie on separate components of the solution set. However, one can make the following important

observation. In the first two cases, the solution set consists of two unbounded branches pointing in opposite directions. If one could jump from one branch to the other, the algorithm might be continued and a solution still be found. In the third case one would need four of these 'reflections'. In the next two sections it will be explained how and when these reflections can be carried out. The reasoning is based on ideas from [96, 119] and will lead to the intended generalization of the octahedral algorithm to circuits with active elements.

4.2.1 The antipodal property

Let C_r^n and ∂C_r^n be defined as

$$C_r^n = \{\mathbf{x} \in R^n \mid \|\mathbf{x}\|_\infty \leq r\}, \quad \partial C_r^n = \{\mathbf{x} \in R^n \mid \|\mathbf{x}\|_\infty = r\}.$$

Then the following is one possible formulation of the Borsuk-Ulam theorem [44].

Theorem 14 *Let* $\mathbf{f}(\cdot) : C_r^n \to R^n$ *be a continuous mapping satisfying* the antipodal property, *i.e.* $\mathbf{f}(\mathbf{x}) = -\mathbf{f}(-\mathbf{x})$ *for all* $\mathbf{x} \in \partial C_r^n$. *Then there exists an* $\mathbf{x}^* \in C_r^n$ *with* $\mathbf{f}(\mathbf{x}^*) = \mathbf{0}$.

The octahedral algorithm can be modified for mappings satisfying the antipodal property by making use of the type of reflections suggested in [96]. The idea is based on the following simple observation. If the homotopy path reaches the boundary of C_r^n in some point $\mathbf{x}$, then point $-\mathbf{x}$ must also belong to the solution set of the modified circuit. This is clear from fig.2.2 : if $\mathbf{x}$ and $\mathbf{y} = \mathbf{f}(\mathbf{x})$ satisfy the constitutive relation of fig.2.2, then so will $-\mathbf{x}$ and $\mathbf{f}(-\mathbf{x}) = -\mathbf{f}(\mathbf{x})$. The algorithm can therefore be switched from $\mathbf{x}$ to $-\mathbf{x}$, and the path through point $-\mathbf{x}$ can be followed into C_r^n. With this modification the path is confined to stay inside C_r^n. Using standard arguments from the theory of homotopy algorithms, it can be shown under appropriate regularity assumptions that cycling is excluded and that the endpoint must be a zero of $\mathbf{f}(\cdot)$.

4.2.2 The nonparallel condition

The second step towards a generalization of Theorem 2 is to replace the *antipodal property* by the more general *nonparallel property* introduced below.

Theorem 15 *Let* $\mathbf{f}(\cdot) : C_r^n \to R^n$ *be a continuous mapping satisfying* the nonparallel property, *i.e. if* $\mathbf{x} \in \partial C_r^n$, *then* $\mathbf{f}(\mathbf{x}) \neq \alpha \mathbf{f}(-\mathbf{x})$ *for all* $\alpha > 0$. *Then there exists an* $\mathbf{x}^* \in C_r^n$ *with* $\mathbf{f}(\mathbf{x}^*) = \mathbf{0}$.

In [119] the proof of this theorem was reduced to Theorem 14 as follows. Choose a number $r' > r$ and define a mapping $\mathbf{g}(\cdot)$ on $C_{r'}^n$ as follows :

1. if $\mathbf{x} \in C_r^n$, then $\mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathbf{x})$,

2. if $\mathbf{x} \in C_{r'}^n \setminus C_r^n$, then $\mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathbf{y}) - \frac{\|\mathbf{x}\|_\infty - r}{r' - r}\mathbf{f}(-\mathbf{y})$ with $\mathbf{y} = \frac{r}{\|\mathbf{x}\|_\infty}\mathbf{x}$.

The mapping $\mathbf{g}(\cdot)$ is odd on $\partial C_{r'}^n$, and hence the algorithm of the previous section can be applied. On the other hand, $\mathbf{g}(\cdot)$ will have no zeros in $C_{r'}^n \setminus C_r^n$ if $\mathbf{f}(\cdot)$ satisfies the nonparallel condition. The algorithm must therefore converge to a zero of $\mathbf{f}(\cdot)$ in C_r^n.

Example. Figure 4.2 applies this reasoning to the equations (4.1) with $\alpha_1 = -1$ and $\alpha_2 = -1$. The mapping is nonparallel on C_{18}^2 and is extended to an odd mapping

$$\begin{aligned} f_1(i_1, i_2) &= -2.(i_1^2 + i_2^2 - 4).i_1 \\ f_2(i_1, i_2) &= -2.i_2 \end{aligned}$$

on C_{20}^2.

Remarks.

- For affine mappings $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$, the nonparallel condition is equivalent to nonsingularity of $\mathbf{A}$ [119]. This should be compared with the situation without reflections, where the algorithm only converges if $\mathbf{A}$ is a P-matrix.

- As mentioned in Section 2.2.7, the index of a solution found by the octahedral algorithm is always $+1$. By a similar proof, it can be shown that the solution obtained by the modified algorithm with reflections has index $+1$ if found after an even number of reflections and -1 if found after an odd number of reflections.

- The mappings for which convergence of the octahedral algorithm was proved in Chapter 2 have degree $+1$. The degree of a mapping satisfying the nonparallel condition can be any odd integer.

A more formal description of the octahedral algorithm when this extension is added can be found in Section 4.3, together with a number of practical details.

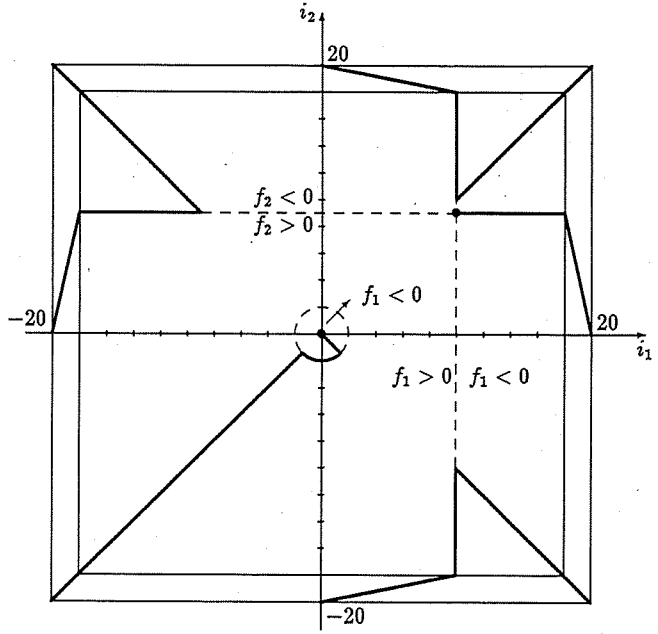


Figure 4.2: Solution set in figure 4.1 if the mapping is modified to an odd mapping on C^2_{20}.

4.3 Summary of the algorithm

4.3.1 The octahedral algorithm with reflections

Below the modified form of the algorithm of Section 2.2.6 will be given. Let $\mathbf{f}(\cdot)$ be an n-dimensional mapping which satisfies the antipodal property on ∂C^n_τ. The algorithm traces part of the one-dimensional manifold $\mathcal{F} \cap \mathcal{Z}$ through the $2n$-dimensional space of $(\mathbf{x}, \mathbf{y})$-pairs. Each time the path reaches the boundary of $C^n_\tau \times R^n$, a reflection is applied.

Start

Let $\mathbf{x}^* := \mathbf{0}$, $\mathbf{y}^* := \mathbf{f}(\mathbf{0})$, and for $k = 1, \ldots, n$, $s_k := 1$ if $y^*_k \leq 0$ and $s_k := -1$ if $y^*_k > 0$. Then $(\mathbf{x}^*, \mathbf{y}^*)$ lies on an n-face of $Z(\mathrm{s}) \cap (C^n_\tau \times R^n)$ and belongs to $\mathcal{Z} \cap \mathcal{F}$.

Repeat

Trace $\mathcal{F} \cap \mathcal{Z}$ through $Z(\mathrm{s}) \cap (C^n_\tau \times R^n)$ from $(\mathbf{x}^*, \mathbf{y}^*)$ onwards until another point $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ on the boundary of $Z(\mathrm{s}) \cap (C^n_\tau \times R^n)$ is reached. Inside $Z(\mathrm{s}) \cap (C^n_\tau \times R^n)$, $\mathcal{F} \cap \mathcal{Z}$ is determined by

(i) A set of n equations $\mathbf{f}(\mathbf{x}) - \mathbf{y} = \mathbf{0}$. Let $\alpha = \{k \mid s_k = 0\}$ and $\beta = \{k \mid s_k \neq 0\}$. Because $\mathbf{y}_\alpha = \mathbf{0}$ and $\mathbf{x}_\beta = \|\mathbf{x}\|_\infty . s_\beta$, the number of unknowns is $n + 1$: $(\mathbf{x}_\alpha, \mathbf{y}_\beta, t)$ with $t = \|\mathbf{x}\|_\infty$.

(ii) A set of inequalities :

- $x_k \leq \|\mathbf{x}\|_\infty$, if $s_k = 0$,
- $x_k \geq -\|\mathbf{x}\|_\infty$, if $s_k = 0$,
- $s_k y_k \leq 0$, if $s_k \neq 0$,
- $\|\mathbf{x}\|_\infty \leq r$.

At the point $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ where $\mathcal{F} \cap \mathcal{Z}$ hits an n-face of $Z(\mathrm{s}) \cap (C^n_\tau \times R^n)$, exactly one of the constraints in (ii) holds with equality. This determines the next piece of $\mathcal{Z}$ to be entered :

- if $\tilde{x}_k = \|\tilde{\mathbf{x}}\|_\infty$, and $s_k = 0$, then set $s_k := 1$, and take $\mathbf{x}^* := \tilde{\mathbf{x}}$ and $\mathbf{y}^* := \tilde{\mathbf{y}}$.
- if $\tilde{x}_k = -\|\tilde{\mathbf{x}}\|_\infty$, and $s_k = 0$, then set $s_k := -1$, and take $\mathbf{x}^* := \tilde{\mathbf{x}}$ and $\mathbf{y}^* := \tilde{\mathbf{y}}$.
- if $\tilde{y}_k = 0$, and $s_k \neq 0$, set $s_k := 0$, $\mathbf{x}^* := \tilde{\mathbf{x}}$, $\mathbf{y}^* := \tilde{\mathbf{y}}$.
- if $\|\tilde{\mathbf{x}}\|_\infty = r$ then set $\mathrm{s} := -\mathrm{s}$, $\mathbf{x}^* := -\tilde{\mathbf{x}}$ and $\mathbf{y}^* := -\tilde{\mathbf{y}}$.

until $\mathrm{s} = \mathbf{0}$
Stop : $\mathbf{y}^* = \mathbf{f}(\mathbf{x}^*) = \mathbf{0}$.

4.3.2 Practical considerations

In order to turn the above ideas into a practical algorithm, a number of technical details must be taken care of.

1. The mapping $\mathbf{f}(\cdot)$ will usually not be defined on a symmetrical hypercube C_r^n centered around the origin, but on some hyperrectangle instead. Let $B_{[\mathbf{l},\mathbf{u}]}^n$ be defined as

$$B_{[\mathbf{l},\mathbf{u}]}^n = \{\mathbf{x} \in R^n \mid \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}, \tag{4.2}$$

   where the inequalities are to be taken componentwise, and where $\mathbf{l}$ and $\mathbf{u}$ are two vectors with $\mathbf{l} < \mathbf{u}$. Assume a mapping $\mathbf{f}(\cdot)$ on $B_{[\mathbf{l},\mathbf{u}]}^n$ is given which is nonparallel on $\partial B_{[\mathbf{l},\mathbf{u}]}^n$, i.e. if $\mathbf{x} \in \partial B_{[\mathbf{l},\mathbf{u}]}^n$, then $\mathbf{f}(\mathbf{x}) \neq \alpha\mathbf{f}(\mathbf{l}+\mathbf{u}-\mathbf{x})$, for all positive α. The problem of finding a zero of $\mathbf{f}(\cdot)$ can be reduced to finding a zero of a nonparallel mapping $\mathbf{g}(\cdot)$ on C_1^n by defining $\mathbf{g}(\mathbf{x})$ as $\mathbf{g}(\mathbf{x}) = \mathbf{f}(\mathbf{y})$ where $y_k = (l_k+u_k)/2+(u_k-l_k).x_k/2$, $k = 1,\ldots,n$.

2. A more difficult point is the following. In order to increase the efficiency of simplicial algorithms, it is common practice to run the algorithm several times with finer and finer subdivisions, and with the result of one run as the starting point for the next. This is impossible with the algorithm as stated above, because it must always be started in the origin. The following problem reformulation allows to restart the algorithm at any arbitrary point in C_r^n.

   Assume we want to find a zero of a mapping $\mathbf{g}(\cdot)$, nonparallel on ∂C_1^n, using $\mathbf{x}^0 \in C_1^n$ as an initial guess. This can be achieved by applying the octahedral algorithm with reflections on a mapping $\mathbf{h}(\cdot) : C_2^n \rightarrow R^n$ defined as follows.

   (a) If $\mathbf{x} \in C_1^n$ then $\mathbf{h}(\mathbf{x}) = \mathbf{g}(\mathbf{y})$ with $\mathbf{y}$ defined as

$$y_k = x_k^0 + (1-x_k^0)x_k \quad \text{if} \quad x_k \geq 0$$
$$y_k = x_k^0 + (1+x_k^0)x_k \quad \text{if} \quad x_k \leq 0.$$

   (b) If $\mathbf{x} \in C_2^n \setminus C_1^n$ then $\mathbf{h}(\mathbf{x}) = \mathbf{g}(\mathbf{y}) - (\|\mathbf{x}\|_\infty - 1)\mathbf{g}(-\mathbf{y})$ with $\mathbf{y}$ defined as follows :

   for $k = 1,\ldots,n$:

   - if $x_k^0 > 0$, then let $\lambda = 2.x_k^0/(1+x_k^0)$, and

$$y_k = x_k^0 + \frac{x_k-\lambda(\|\mathbf{x}\|_\infty-1)}{\|\mathbf{x}\|_\infty-\lambda(\|\mathbf{x}\|_\infty-1)}(1-x_k^0)$$
$$\text{if } \lambda(\|\mathbf{x}\|_\infty - 1) \leq x_k \leq \|\mathbf{x}\|_\infty,$$
$$y_k = x_k^0.\frac{(2-\|\mathbf{x}\|_\infty)\lambda+2x_k}{\|\mathbf{x}\|_\infty\lambda}$$
$$\text{if } -\lambda \leq x_k \leq \lambda(\|\mathbf{x}\|_\infty - 1)$$
$$y_k = -x_k^0 + \frac{x_k+\lambda}{\|\mathbf{x}\|_\infty-\lambda}(1-x_k^0)$$
$$\text{if } -\|\mathbf{x}\|_\infty \leq x_k \leq -\lambda,$$

   - if $x_k^0 < 0$, then let $\lambda = -2.x_k^0/(1-x_k^0)$, and

$$y_k = -x_k^0 + \frac{x_k-\lambda}{\|\mathbf{x}\|_\infty-\lambda}(1+x_k^0)$$
$$\text{if } \lambda \leq x_k \leq \|\mathbf{x}\|_\infty$$
$$y_k = x_k^0\frac{(2-\|\mathbf{x}\|_\infty)\lambda-2x_k}{\|\mathbf{x}\|_\infty\lambda}$$
$$\text{if } -\lambda(\|\mathbf{x}\|_\infty - 1) \leq x_k \leq \lambda$$
$$y_k = x_k^0 + \frac{x_k+\lambda(\|\mathbf{x}\|_\infty-1)}{\|\mathbf{x}\|_\infty-\lambda(\|\mathbf{x}\|_\infty-1)}(1+x_k^0)$$
$$\text{if } -\|\mathbf{x}\|_\infty \leq x_k \leq -\lambda(\|\mathbf{x}\|_\infty - 1).$$

   - if $x_k^0 = 0$, then $y_k = x_k/\|\mathbf{x}\|_\infty$.

   It can be verified that $\mathbf{h}(\cdot)$ is odd on ∂C_2^n and that zeros of $\mathbf{h}(\cdot)$ are in one-to-one correspondence with zeros of $\mathbf{g}(\cdot)$.

3. A second problem with restarting the algorithm is related to the index of the solution. As mentioned above the index of the solution found by the algorithm can be either $+1$ or -1, depending on the number of reflections. Now assume the algorithm ends at a solution with negative index, then restarting will not lead to an efficient refinement of the solution, because at least one reflection is needed to arrive at the same point again. This is a waste of time and the benefit of restarting would be lost. The problem can be solved by replacing $\mathbf{f}(\cdot)$ by $\mathbf{J}^{-1}\mathbf{f}(\cdot)$, where $\mathbf{J}$ is the Jacobian matrix of $\mathbf{f}(\cdot)$ or an approximation of it. As shown in Section 3.4.4, this premultiplying can be carried out easily.

4. A final topic is the treatment of degeneracy. The algorithm as sketched above will work for *generic* mappings $\mathbf{f}(\cdot)$, but might fail in degenerate cases. The classical cure of degeneracy in simplicial algorithms is lexicographical pivoting, which is based on the idea of perturbing $\mathbf{f}(\cdot)$ as $\mathbf{f}(\cdot) - \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} = [\epsilon, \epsilon^2, \ldots, \epsilon^n]^t$ and ϵ an infinitesimal positive number. This scheme breaks down in the present context, because an odd perturbation is needed here. The answer proposed in [96] provides an odd perturbation, but cannot be applied to the octahedral algorithm. An

α_1	α_2	α_3	α_4	α_5	Number of funct. eval.
1	1	1	1	1	152
1	1	1	1	−1	186
1	1	1	−1	−1	285
1	1	−1	−1	−1	489
1	−1	−1	−1	−1	500
−1	−1	−1	−1	−1	929

Table 4.1: Convergence of the algorithm on (4.3) for different choices of α_i.

ad hoc solution is to use lexicographical pivoting but to reverse the sign of ϵ with each reflection. This was successful on all test examples, although theoretically it cannot be excluded that the path comes back to the origin after an odd number of reflections.

As an illustration we modify one of the test problems of Chapter 3 (Section 3.5.2, problem 2).
Example. Take

$$f_i(\mathbf{x}) = \alpha_i(x_i - \exp(\cos(i\sum_{j=1}^{5} x_j))), \quad i = 1,\ldots 5, \tag{4.3}$$

with each $\alpha_i = \pm 1$. If $\alpha_i = 1$, $i = 1,\ldots,5$, this mapping satisfies the coercivity condition of Theorem 2, and in general it is always nonparallel on the boundary of C_3^5. Table 4.1 shows the number of function evaluations needed for different choices of α_i. The more the asymptotic behaviour deviates from the coercivity condition of Section 2.2.5, the more reflections are needed. This clearly slows down the algorithm.

4.4 Application of the nonparallel condition to nonlinear circuits

4.4.1 Mappings of the form $\mathbf{f(x) = Ag(x) + Bx + c}$

Equations $\mathbf{f(x) = 0}$ with

$$\mathbf{f(x) = Ag(x) + Bx + c} \tag{4.4}$$

have been studied in connection with nonlinear circuits at several places [87, 17]. These efforts aimed at proving uniqueness or existence of solutions under some qualitative and global properties of the mapping $\mathbf{g}(\cdot)$ (for instance eventual passivity, eventual increasingness, ...). As shown in [17] all of these results can be proved by application of Theorem 15. Here we will deal with a slightly different question. Since our interest is in solving the above equation for as broad a class of problems as possible, we will attempt to incorporate quantitative information on $\mathbf{g}(\cdot)$ in order to relax the restrictions on matrices **A** and **B**.

The main theorem in this section is Theorem 18. It is an extension to similar theorems in [87, 17].

Definition 11 *A continuous mapping $g(\cdot) : R \to R$ is said to be*

- *of type U if $g(x) \to +\infty$ as $x \to +\infty$, and $g(x) \to -\infty$ as $x \to -\infty$,*
- *of type H^+ if $g(x) \to +\infty$ as $x \to +\infty$, and $|g(x)|$ remains bounded as $x \to -\infty$,*
- *of type H^- if $g(x) \to -\infty$ as $x \to -\infty$, and $|g(x)|$ remains bounded as $x \to \infty$,*
- *of type B if $|g(x)|$ remains bounded as $|x| \to \infty$.*

Definition 12 *Let $\underline{\gamma}$ and $\overline{\gamma}$ be two real numbers with $0 \leq \underline{\gamma} < \overline{\gamma} \leq +\infty$. Then a nonlinear mapping $g(\cdot) : R \to R$ of type U, H^+ or H^- lies asymptotically in sector $(\underline{\gamma}, \overline{\gamma})$ if :*

$$\exists N > 0 \ \ s.t. \ \ |x|_g > N \Rightarrow \underline{\gamma} < g(x)/x < \overline{\gamma},$$

and

$$\forall M_1 \ \ \exists M_2 \geq 0 : |x|_g > M_2 \Rightarrow |g(x) - \underline{\gamma}x| > M_1 \text{ and } |\overline{\gamma}x - g(x)| > M_1.$$

Here $|x|_g$ is defined as $|x|_g = |x|$ if $g(\cdot)$ is of type U, as $|x|_g = \max(0, x)$ if $g(\cdot)$ is of type H^+, and as $|x|_g = \max(0, -x)$ if $g(\cdot)$ is of type H^-.

Theorem 16 [33, 84] *If **A** is an $n \times n$ matrix then the following properties are equivalent :*

1. *$\det(\mathbf{A} + \mathbf{D}) \neq 0$ for all diagonal matrices **D** with $d_k > 0$, $k = 1,\ldots,n$.*
2. *All principal minors of **A** are nonnegative.*

3. *If* $\mathbf{D} \neq \mathbf{0}$ *is a diagonal matrix with* $d_k \in \{-1, 0, +1\}$ *for* $k = 1, \ldots, n$, *then there exists a nonzero vector* $\mathbf{p}$ *with* $\mathbf{Dp} \geq \mathbf{0}$, $\mathbf{D}^2\mathbf{p} = \mathbf{p}$ *and* $\mathbf{DA}^t\mathbf{p} \geq \mathbf{0}$.

Such a matrix is called a P_0-matrix. The following is Willson's generalization of the P_0-matrix concept to pairs of matrices [126].

Definition 13 *Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be two* $n \times n$ *matrices, then* $\mathcal{C}(\mathbf{A}, \mathbf{B})$ *is the set of matrices constructed as follows :* $\mathbf{M} \in \mathcal{C}(\mathbf{A}, \mathbf{B})$ *iff for* $k = 1, \ldots, n$ *either* $\mathbf{M}_{.,k} = \mathbf{A}_{.,k}$ *or* $\mathbf{M}_{.,k} = \mathbf{B}_{.,k}$.

Theorem 17 [126] *If* $\mathbf{A}$ *and* $\mathbf{B}$ *are two* $n \times n$ *matrices, then the following properties are equivalent :*

1. $\det(\mathbf{AD} + \mathbf{B}) \neq 0$ *for all diagonal matrices* $\mathbf{D}$ *with* $d_k > 0$, $k = 1, \ldots, n$,
2. *There exists a matrix* $\mathbf{M} \in \mathcal{C}(\mathbf{A}, \mathbf{B})$ *such that* $\det \mathbf{M} \neq 0$ *and such that* $\det \mathbf{M}. \det \mathbf{N} \geq 0$ *for all* $\mathbf{N} \in \mathcal{C}(\mathbf{A}, \mathbf{B})$,
3. *If* $\mathbf{D} \neq \mathbf{0}$ *is a diagonal matrix with* $d_k \in \{-1, 0, +1\}$ *for* $k = 1, \ldots, n$, *then there exists a vector* $\mathbf{p}$ *for which* $\mathbf{DA}^t\mathbf{p} \geq \mathbf{0}$, $\mathbf{DB}^t\mathbf{p} \geq \mathbf{0}$ *and* $\mathbf{D}(\mathbf{A}^t + \mathbf{B}^t)\mathbf{p} \neq \mathbf{0}$.

A pair of matrices satisfying these properties is called a $\mathcal{W}_0$-pair. It is also clear how this generalizes the P_0-matrix concept : a matrix $\mathbf{A}$ is a P_0 matrix iff $(\mathbf{I}, \mathbf{A}) \in \mathcal{W}_0$.

Theorem 18 below gives a condition for mapping (4.4) to possess the nonparallel property.

Theorem 18 *Let* $\mathbf{g}(\cdot)$ *in (4.4) be a diagonal mapping with the following properties*

1. *If* $g_k(\cdot)$ *is of type* U, H^+ *or* H^-, *then there exist numbers* $0 \leq \underline{\gamma}_k < \overline{\gamma}_k \leq \infty$ *s.t.* $g_k(\cdot)$ *lies asymptotically in sector* $(\underline{\gamma}_k, \overline{\gamma}_k)$.
2. *For all nonzero diagonal matrices* $\mathbf{D}$ *with diagonal elements* $d_k \in \{-1, 0, +1\}$, $k = 1, \ldots, n$ *there exists a vector* $\mathbf{p}$ *s.t.*
   for $k = 1, \ldots, n$:

   - *if* $g_k(\cdot)$ *is of type* U *and* $d_k \neq 0$, *then*

$$d_k[(\mathbf{p}^t\mathbf{A})_k\underline{\gamma}_k + (\mathbf{p}^t\mathbf{B})_k] \geq 0, \tag{4.5}$$
$$d_k[(\mathbf{p}^t\mathbf{A})_k\overline{\gamma}_k + (\mathbf{p}^t\mathbf{B})_k] \geq 0,^1 \tag{4.6}$$

[1] If $\overline{\gamma}_k = \infty$, this inequality should be read as $d_k(\mathbf{p}^t\mathbf{A})_k \geq 0$.

   - *if* $g_k(\cdot)$ *is of type* H^+ *and* $d_k = 1$, *or* $g_k(\cdot)$ *is of type* H^- *and* $d_k = -1$, *then*

$$d_k[(\mathbf{p}^t\mathbf{A})_k\underline{\gamma}_k + (\mathbf{p}^t\mathbf{B})_k] \geq 0, \tag{4.7}$$
$$d_k[(\mathbf{p}^t\mathbf{A})_k\overline{\gamma}_k + (\mathbf{p}^t\mathbf{B})_k] \geq 0, \tag{4.8}$$

   - *if* $g_k(\cdot)$ *is of type* H^+ *and* $d_k = -1$, *or* $g_k(\cdot)$ *is of type* H^- *and* $d_k = 1$, *then*

$$d_k(\mathbf{p}^t\mathbf{B})_k \geq 0 \tag{4.9}$$

   - *if* $g_k(\cdot)$ *is of type* B *and* $d_k \neq 0$, *then*

$$d_k(\mathbf{p}^t\mathbf{B})_k \geq 0, \tag{4.10}$$

   - *at least one of the inequalities (4.5), (4.6), (4.7), (4.8), (4.9), (4.10) holds with strict inequality.*

3. *In 2. the vector* $\mathbf{p}$ *can be chosen in such a way that* $-\mathbf{p}$ *satisfies the requirements under 2. for matrix* $-\mathbf{D}$.

Then there is some $r > 0$ *such that* $\mathbf{f}(\cdot)$ *satisfies the nonparallel condition on* ∂C_r.

Proof. Let S be defined as the set of all vectors $\mathbf{x}$ for which $\mathbf{f}(\mathbf{x}) = \alpha\mathbf{f}(-\mathbf{x})$ for some $\alpha > 0$, and assume the theorem is invalid and that S is unbounded. Then it must be possible to choose n numbers $d_k \in \{-1, 0, +1\}$, not all equal to zero, and a positive constant N such that

$$\forall r \; \exists \mathbf{x} \in S \text{ with } d_k x_k > r \text{ if } d_k \neq 0, \text{ and } |x_k| < N \text{ if } d_k = 0. \tag{4.11}$$

Given this specific set of d_k's, a vector $\mathbf{p}$ can be found as stated in the theorem. For this $\mathbf{p}$ it will be shown that there exists an r_1 s.t.

$$\begin{array}{c} d_k x_k > r_1 \text{ if } d_k \neq 0 \quad \text{and} \quad |x_k| < N \text{ if } d_k = 0 \\ \Downarrow \\ \mathbf{p}^t\mathbf{f}(\mathbf{x}) > 0 \quad \text{and} \quad \mathbf{p}^t\mathbf{f}(-\mathbf{x}) < 0. \end{array} \tag{4.12}$$

From this it follows that (4.11) is false and hence S must be bounded.

First, observe that

$$\mathbf{p}^t\mathbf{f}(\mathbf{x}) = \mathbf{p}^t\mathbf{A}\mathbf{g}(\mathbf{x}) + \mathbf{p}^t\mathbf{B}\mathbf{x} + \mathbf{p}^t\mathbf{c} \tag{4.13}$$

is bounded below for all $\mathbf{x}$ with $d_k x_k \geq 0$ if $d_k \neq 0$ and $|x_k| < N$ if $d_k = 0$. This can be seen by examining each term in the inner product (4.13) separately. If $d_k = 0$, then $|x_k| < N$ and $[\mathbf{p}^t\mathbf{A}]_k g_k(\mathbf{x}) + [\mathbf{p}^t\mathbf{B}]_k x_k$ is bounded. Also $\mathbf{p}^t\mathbf{c}$ is a constant. Owing to (4.5)–(4.10), one can distinguish six possibilities for each $d_k \neq 0$:

- Assume that $d_k \neq 0$ and $g_k(\cdot)$ is of type U, or that $d_k = 1$ and $g_k(\cdot)$ is of type H^+, or that $d_k = -1$ and $g_k(\cdot)$ is of type H^-.

  1. If $[\mathbf{p}^t\mathbf{A}]_k = 0$ and $[\mathbf{p}^t\mathbf{B}]_k = 0$, then $[\mathbf{p}^t\mathbf{A}]_k g_k(\mathbf{x}) + [\mathbf{p}^t\mathbf{B}]_k x_k = 0$.
  2. If $[\mathbf{p}^t\mathbf{A}]_k = 0$ and $d_k[\mathbf{p}^t\mathbf{B}]_k > 0$, then $[\mathbf{p}^t\mathbf{A}]_k g_k(\mathbf{x}) + [\mathbf{p}^t\mathbf{B}]_k x_k \geq 0$ if $d_k x_k \geq 0$.
  3. If $d_k[\mathbf{p}^t\mathbf{A}]_k > 0$, then

     $$[\mathbf{p}^t\mathbf{A}]_k g_k(\mathbf{x}) + [\mathbf{p}^t\mathbf{B}]_k x_k = ([\mathbf{p}^t\mathbf{A}]_k \underline{\gamma}_k + [\mathbf{p}^t\mathbf{B}]_k) x_k + [\mathbf{p}^t\mathbf{A}]_k (g_k(\mathbf{x}) - \underline{\gamma}_k x_k). \tag{4.14}$$

     The first term in (4.14) is nonnegative, the second is bounded below and becomes positive if $d_k x_k \to \infty$.
  4. If $d_k[\mathbf{p}^t\mathbf{A}]_k < 0$, then

     $$[\mathbf{p}^t\mathbf{A}]_k g_k(\mathbf{x}) + [\mathbf{p}^t\mathbf{B}]_k x_k = ([\mathbf{p}^t\mathbf{A}]_k \overline{\gamma}_k + [\mathbf{p}^t\mathbf{B}]_k) x_k - [\mathbf{p}^t\mathbf{A}]_k (\overline{\gamma}_k x_k - g_k(\mathbf{x})). \tag{4.15}$$

     As in the previous case, this is bounded below, and positive for $d_k x_k \to \infty$.

- Assume that $d_k \neq 0$ and $g_k(\cdot)$ is of type B, or that $d_k = -1$ and $g_k(\cdot)$ is of type H^+, or that $d_k = 1$ and $g_k(\cdot)$ is of type H^-.

  5. If $d_k(\mathbf{p}^t\mathbf{B})_k = 0$, then $[\mathbf{p}^t\mathbf{A}]_k g_k(\mathbf{x}) + [\mathbf{p}^t\mathbf{B}]_k x_k$ is bounded for all $d_k x_k \geq 0$.
  6. If $d_k(\mathbf{p}^t\mathbf{B})_k > 0$, then $[\mathbf{p}^t\mathbf{A}]_k g_k(\mathbf{x}) + [\mathbf{p}^t\mathbf{B}]_k x_k$ is bounded below for all $d_k x_k \geq 0$, and becomes positive if $d_k x_k \to \infty$.

In addition it follows from the requirements made in the theorem that there must be an index j for which one of the cases 2, 3, 4 or 6 holds. The term $[\mathbf{p}^t\mathbf{A}]_j g_j(\mathbf{x}) + [\mathbf{p}^t\mathbf{B}]_j x_j$ in the expression for $\mathbf{p}^t\mathbf{f}(\mathbf{x})$ grows unboundedly if $d_j x_j \to \infty$, and therefore there exists an r_2 such that

$$d_k x_k > r_2 \text{ if } d_k \neq 0 \text{ and } |x_k| < N \text{ if } d_k = 0 \implies \mathbf{p}^t\mathbf{f}(\mathbf{x}) > 0.$$

In the same way one can show the existence of an r_3 for which

$$d_k x_k > r_3 \text{ if } d_k \neq 0 \text{ and } |x_k| < N \text{ if } d_k = 0 \implies \mathbf{p}^t\mathbf{f}(-\mathbf{x}) < 0.$$

It now suffices to take $r_1 = \max\{r_2, r_3\}$ to see that (4.12) holds and assumption (4.11) is false. □

Theorem 19 *Let $\underline{\Gamma}$ and $\overline{\Gamma}$ be two diagonal matrices, with diagonal elements $\underline{\gamma}_k$ and $\overline{\gamma}_k$ respectively, and $0 \leq \underline{\gamma}_k < \overline{\gamma}_k \leq \infty$ for $k = 1, \ldots, n$. Assume that $\mathbf{g}(\cdot)$ in (4.4) is a diagonal mapping with each $g_k(\cdot)$ of type U and asymptotically lying in sector $(\underline{\gamma}_k, \overline{\gamma}_k)$, and that*

$$(\mathbf{A}\overline{\Gamma} + \mathbf{B}, \mathbf{A}\underline{\Gamma} + \mathbf{B}) \in \mathcal{W}_0.^2 \tag{4.16}$$

Then there is some $r > 0$ such that $\mathbf{f}(\cdot)$ satisfies the nonparallel condition on ∂C_r.

Proof. Owing to (4.16) one can find, for all nonzero diagonal matrices $\mathbf{D}$, with $d_k \in \{-1, 0, +1\}$, a vector $\mathbf{p}$ such that

$$\mathbf{p}^t(\mathbf{A}\underline{\Gamma} + \mathbf{B})\mathbf{D} \geq \mathbf{0}, \quad \mathbf{p}^t(\mathbf{A}\overline{\Gamma} + \mathbf{B})\mathbf{D} \geq \mathbf{0} \tag{4.17}$$

and at least one of the two vectors in (4.17) is nonzero. This fact combined with Theorem 18 proves the theorem. □

Examples.

- If one takes each $\underline{\gamma}_k = 0$ and each $\overline{\gamma}_k = \infty$, then Theorem 19 requires $(\mathbf{A}, \mathbf{B})$ to be a $\mathcal{W}_0$-pair. If in addition $\mathbf{A} = \mathbf{I}$, one finds that $\mathbf{B}$ must belong to P_0, which confirms a well-known fact (cf. Section 2.2.5).

- If each $g_k(\cdot)$ is of type B, then Theorem 18 is equivalent with requiring $\mathbf{B}$ to be nonsingular. This follows from the fact that for each diagonal matrix $\mathbf{D}$ with -1, 0 or 1 on its main diagonal, exactly one of the following alternatives holds (see Appendix B) :

  1. There exists a vector $\mathbf{x} > \mathbf{0}$ such that $\mathbf{BDx} = \mathbf{0}$.
  2. There exists a vector $\mathbf{p}$ such that $\mathbf{p}^t\mathbf{BD} \geq \mathbf{0}$ and $\mathbf{p}^t\mathbf{BD} \neq \mathbf{0}$.

  Theorem 18 requires the second alternative to hold. Hence the first alternative is false for all nonzero $\mathbf{D}$, and $\mathbf{B}$ must be nonsingular.

4.4.2 Circuits containing controlled sources

Example. As an illustration of the above results, consider the circuit of fig.4.3. Assume all three resistors as well as the controlled sources have nonlinear characteristics of type U. The resistors in branches 3 and 4 are voltage-controlled whereas the resistor in branch 7 is current-controlled. The circuit can be de-

[2] If $\overline{\gamma}_k = \infty$, then in this exprssion the k-th column of $\mathbf{A}\overline{\Gamma} + \mathbf{B}$ can be replaced by the k-th column of $\mathbf{A}$.

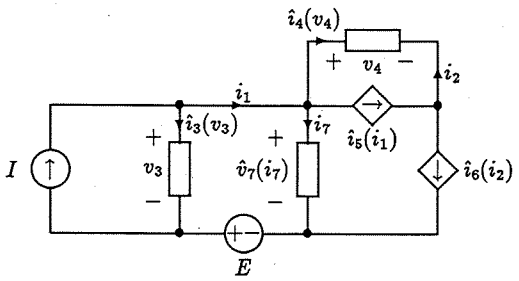


Figure 4.3: Nonlinear circuit with two controlled sources and three resistors.

scribed by equations of the form (4.4) :

$$\begin{pmatrix} 0 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{\imath}_5(i_1) \\ \hat{\imath}_6(i_2) \\ \hat{\imath}_3(v_3) \\ \hat{\imath}_4(v_4) \\ \hat{v}_7(i_7) \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \\ v_3 \\ v_4 \\ i_7 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ I \\ 0 \\ E \end{pmatrix}. \tag{4.18}$$

Referring to Theorem 19, one can consider several possibilities depending on the sectors where the nonlinear characteristics are bound to lie. Let the sectors for $\hat{\imath}_5(i_1)$, $\hat{\imath}_6(i_2)$, $\hat{\imath}_3(v_3)$, $\hat{\imath}_4(v_4)$ and $\hat{v}_7(i_7)$ be denoted by $(\underline{\gamma}_1, \overline{\gamma}_1)$, $(\underline{\gamma}_2, \overline{\gamma}_2)$, $(\underline{\gamma}_3, \overline{\gamma}_3)$, $(\underline{\gamma}_4, \overline{\gamma}_4)$ and $(\underline{\gamma}_7, \overline{\gamma}_7)$, respectively. Then one has in equation (4.16) :

$$\mathbf{A}\overline{\mathbf{\Gamma}} + \mathbf{B} = \begin{pmatrix} 1 & -\overline{\gamma}_2 & 0 & 0 & -1 \\ -\overline{\gamma}_1 & 1+\overline{\gamma}_2 & 0 & 0 & 0 \\ 0 & \overline{\gamma}_2 & \overline{\gamma}_3 & 0 & 1 \\ \overline{\gamma}_1 & -\overline{\gamma}_2 & 0 & \overline{\gamma}_4 & 0 \\ 0 & 0 & -1 & 0 & \overline{\gamma}_7 \end{pmatrix},$$

$$\mathbf{A}\underline{\mathbf{\Gamma}} + \mathbf{B} = \begin{pmatrix} 1 & -\underline{\gamma}_2 & 0 & 0 & -1 \\ -\underline{\gamma}_1 & 1+\underline{\gamma}_2 & 0 & 0 & 0 \\ 0 & \underline{\gamma}_2 & \underline{\gamma}_3 & 0 & 1 \\ \underline{\gamma}_1 & -\underline{\gamma}_2 & 0 & \underline{\gamma}_4 & 0 \\ 0 & 0 & -1 & 0 & \underline{\gamma}_7 \end{pmatrix}.$$

Hence one has to examine 2^5 determinants of the form

$$\det \begin{pmatrix} 1 & -\gamma_2 & 0 & 0 & -1 \\ -\gamma_1 & 1+\gamma_2 & 0 & 0 & 0 \\ 0 & \gamma_2 & \gamma_3 & 0 & 1 \\ \gamma_1 & -\gamma_2 & 0 & \gamma_4 & 0 \\ 0 & 0 & -1 & 0 & \gamma_7 \end{pmatrix} = \gamma_4(\gamma_3\gamma_7(1+\gamma_2-\gamma_1\gamma_2)+1+\gamma_2), \tag{4.19}$$

by taking either $\gamma_1 = \overline{\gamma}_1$ or $\gamma_1 = \underline{\gamma}_1$, either $\gamma_2 = \overline{\gamma}_2$ or $\gamma_2 = \underline{\gamma}_2$ etc.

If the nonlinear functions in (4.18) can be any functions of type U, one must take $\underline{\gamma}_k = 0$ and $\overline{\gamma}_k = \infty$ and condition (4.16) reduces to $(\mathbf{A}, \mathbf{B}) \in \mathcal{W}_0$. This is clearly false since there are matrices in $\mathcal{C}(\mathbf{A}, \mathbf{B})$ with positive and with negative determinant. Hence there will exist specific choices for the functions $\hat{\imath}_5(i_1)$, $\hat{\imath}_6(i_2)$, $\hat{\imath}_3(v_3)$, $\hat{\imath}_4(v_4)$ and $\hat{v}_7(i_7)$ for which equation (4.18) does not possess the nonparallel property.

The nonparallel property can be satisfied by strengthening the requirements on the nonlinearities. If for instance one takes

$$\begin{aligned} (\underline{\gamma}_1, \overline{\gamma}_1) = (\underline{\gamma}_2, \overline{\gamma}_2) &= (0, (\sqrt{5}+1)/2) \\ (\underline{\gamma}_3, \overline{\gamma}_3) = (\underline{\gamma}_4, \overline{\gamma}_4) = (\underline{\gamma}_7, \overline{\gamma}_7) &= (0, \infty) \end{aligned}$$

then all determinants in (4.19) are nonnegative and the nonparallel condition is satisfied.

On the other hand, taking

$$\begin{aligned} (\underline{\gamma}_1, \overline{\gamma}_1) = (\underline{\gamma}_2, \overline{\gamma}_2) &= (\sqrt{3}+1, \infty) \\ (\underline{\gamma}_3, \overline{\gamma}_3) = (\underline{\gamma}_7, \overline{\gamma}_7) &= (1, \infty) \\ (\underline{\gamma}_4, \overline{\gamma}_4) &= (0, \infty) \end{aligned}$$

ensures that all determinants (4.19) are nonpositive, and again the nonparallel property is satisfied.

Let us now consider an arbitrary circuit containing independent sources, nonlinear resistors (voltage-controlled or current-controlled) and nonlinear controlled sources of all four types. A directed graph can be defined starting from the circuit, by replacing each independent source and each resistor by a branch with arbitrary direction, and each controlled source by two branches as shown in fig 4.4. To this graph the following operations will be applied [69] :

- $\mathcal{O}(k)$ means removing branch k, while leaving the nodes intact.
- $\mathcal{S}(k)$ means removing branch k, while coalescing the two nodes (shrinking).

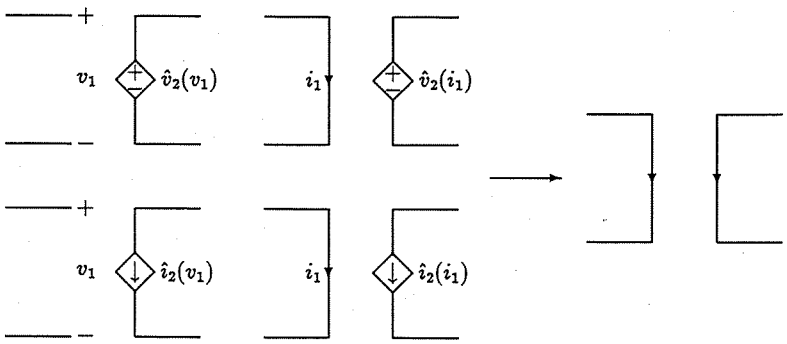


Figure 4.4: Each controlled source is replaced by two directed branches.

- $\mathcal{Z}(\cdot)$ applied to a controlled source with input branch j and output branch k means
  - $\mathcal{O}(j)$ and $\mathcal{S}(k)$ if the source is a VCVS,
  - $\mathcal{O}(j)$ and $\mathcal{O}(k)$ if the source is a VCCS,
  - $\mathcal{S}(j)$ and $\mathcal{S}(k)$ if the source is a CCVS,
  - $\mathcal{S}(j)$ and $\mathcal{O}(k)$ if the source is a CCCS.

Assume the graph satisfies the following *interconnection assumption* :

- There are no loops composed exclusively by independent voltage sources, input branches of CCVS's or CCCS's, and output branches of CCVS's or VCVS's.
- There are no cutsets composed exclusively by independent current sources, input branches of VCCS's or VCVS's, and output branches of CCCS's or VCCS's.

It is then possible to find a tree containing all independent voltage sources, all input branches of CCCS's and CCVS's, all output branches of CCVS's and VCVS's, and no independent current sources, no input branches of VCVS's and VCCS's, and no output branches of VCCS's and CCCS's. Partition the elements as follows

(1) the input branches of the CCVS's (currents $\mathbf{i}_1$, voltages $\mathbf{v}_1 = \mathbf{0}$),

(2) the input branches of the CCCS's (currents $\mathbf{i}_2$, voltages $\mathbf{v}_2 = \mathbf{0}$),

(3) the output branches of the CCVS's (currents $\mathbf{i}_3$, voltages $\mathbf{v}_3 = \hat{\mathbf{v}}_3(\mathbf{i}_1)$),

(4) the output branches of the VCVS's (currents $\mathbf{i}_4$, voltages $\mathbf{v}_4 = \hat{\mathbf{v}}_4(\mathbf{v}_7)$),

(5) the voltage-controlled tree resistors (currents $\mathbf{i}_5 = \hat{\mathbf{i}}_5(\mathbf{v}_5)$, voltages $\mathbf{v}_5$),

(5') the current-controlled tree resistors (currents $\mathbf{i}_{5'}$, voltages $\hat{\mathbf{v}}_{5'}(\mathbf{i}_{5'})$),

(6) the input branches of the VCCS's (currents $\mathbf{i}_6 = \mathbf{0}$, voltages $\mathbf{v}_6$),

(7) the input branches of the VCVS's (currents $\mathbf{i}_7 = \mathbf{0}$, voltages $\mathbf{v}_7$),

(8) the output branches of the VCCS's (currents $\hat{\mathbf{i}}_8(\mathbf{v}_6)$, voltages $\mathbf{v}_8$),

(9) the output branches of the CCCS's (currents $\hat{\mathbf{i}}_9(\mathbf{i}_2)$, voltages $\mathbf{v}_9$),

(10) the current-controlled cotree resistors (currents $\mathbf{i}_{10}$, voltages $\hat{\mathbf{v}}_{10}(\mathbf{i}_{10})$),

(10') the voltage-controlled cotree resistors (currents $\mathbf{i}_{10'} = \hat{\mathbf{i}}_{10'}(\mathbf{v}_{10'})$, voltages $\mathbf{v}_{10'}$).

It is assumed that all nonlinearities ($\hat{\mathbf{v}}_3(\cdot)$, $\hat{\mathbf{v}}_4(\cdot)$, ...) are diagonal mappings. Let

$$\mathbf{Q} = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 5' & 6 & 7 & 8 & 9 & 10 & 10' \end{matrix} \\ \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{1,6} & \mathbf{Q}_{1,7} & \mathbf{Q}_{1,8} & \mathbf{Q}_{1,9} & \mathbf{Q}_{1,10} & \mathbf{Q}_{1,10'} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{2,6} & \mathbf{Q}_{2,7} & \mathbf{Q}_{2,8} & \mathbf{Q}_{2,9} & \mathbf{Q}_{2,10} & \mathbf{Q}_{2,10'} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{3,6} & \mathbf{Q}_{3,7} & \mathbf{Q}_{3,8} & \mathbf{Q}_{3,9} & \mathbf{Q}_{3,10} & \mathbf{Q}_{3,10'} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{4,6} & \mathbf{Q}_{4,7} & \mathbf{Q}_{4,8} & \mathbf{Q}_{4,9} & \mathbf{Q}_{4,10} & \mathbf{Q}_{4,10'} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{Q}_{5,6} & \mathbf{Q}_{5,7} & \mathbf{Q}_{5,8} & \mathbf{Q}_{5,9} & \mathbf{Q}_{5,10} & \mathbf{Q}_{5,10'} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{Q}_{5',6} & \mathbf{Q}_{5',7} & \mathbf{Q}_{5',8} & \mathbf{Q}_{5',9} & \mathbf{Q}_{5',10} & \mathbf{Q}_{5',10'} \end{pmatrix} \end{array}$$

be the fundamental cutset matrix for this circuit if $\mathcal{S}(\cdot)$ is applied to the voltage sources and $\mathcal{O}(\cdot)$ to the current sources. One can then write down circuit equations in the form $\mathbf{A}\mathbf{g}(\mathbf{x}) + \mathbf{B}\mathbf{x} + \mathbf{c} = \mathbf{0}$ with :

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{Q}_{1,9} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{1,8} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{1,10'} \\ \mathbf{0} & \mathbf{Q}_{2,9} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{2,8} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{2,10'} \\ \mathbf{0} & \mathbf{Q}_{5,9} & \mathbf{1} & \mathbf{0} & \mathbf{Q}_{5,8} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{5,10'} \\ \mathbf{0} & \mathbf{Q}_{5',9} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{5',8} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{5',10'} \\ -\mathbf{Q}_{3,6}^t & \mathbf{0} & \mathbf{0} & -\mathbf{Q}_{5',6}^t & \mathbf{0} & -\mathbf{Q}_{4,6}^t & \mathbf{0} & \mathbf{0} \\ -\mathbf{Q}_{3,7}^t & \mathbf{0} & \mathbf{0} & -\mathbf{Q}_{5',7}^t & \mathbf{0} & -\mathbf{Q}_{4,7}^t & \mathbf{0} & \mathbf{0} \\ -\mathbf{Q}_{3,10}^t & \mathbf{0} & \mathbf{0} & -\mathbf{Q}_{5',10}^t & \mathbf{0} & -\mathbf{Q}_{4,10}^t & \mathbf{1} & \mathbf{0} \\ -\mathbf{Q}_{3,10'}^t & \mathbf{0} & \mathbf{0} & -\mathbf{Q}_{5',10'}^t & \mathbf{0} & -\mathbf{Q}_{4,10'}^t & \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad (4.20)$$

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} \hat{\mathbf{v}}_3(\mathbf{i}_1) \\ \hat{\mathbf{i}}_9(\mathbf{i}_2) \\ \hat{\mathbf{i}}_5(\mathbf{v}_5) \\ \hat{\mathbf{v}}_{5'}(\mathbf{i}_{5'}) \\ \hat{\mathbf{i}}_8(\mathbf{v}_6) \\ \hat{\mathbf{v}}_4(\mathbf{v}_7) \\ \hat{\mathbf{v}}_{10}(\mathbf{i}_{10}) \\ \hat{\mathbf{i}}_{10'}(\mathbf{v}_{10'}) \end{pmatrix}, \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \mathbf{Q}_{1,10} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \mathbf{Q}_{2,10} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{Q}_{5,10} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \mathbf{Q}_{5',10} & 0 \\ 0 & 0 & -\mathbf{Q}^t_{5,6} & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -\mathbf{Q}^t_{5,7} & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\mathbf{Q}^t_{5,10} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\mathbf{Q}^t_{5,10'} & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4.21}$$

Theorem 19 can now be used to examine whether these equations satisfy the nonparallel property.

If all nonlinearities are specified to lie in sector $(0, \infty)$, the condition (4.16) reduces to a purely topological criterion that was discovered by Nishi and Chua and that guarantees uniqueness of solutions when all nonlinearities are monotone increasing [69]. The connection between criterion (4.16) and the result in [69] is stated in the following theorem.

Theorem 20 *The matrix pair* $(\mathbf{A}, \mathbf{B})$ *defined in (4.20) and (4.21) is a* $\mathcal{W}_0$*-pair if it is impossible,*

- *by applying* $\mathcal{S}(\cdot)$ *to each independent voltage source and* $\mathcal{O}(\cdot)$ *to each independent current source,*
- *by applying either* $\mathcal{O}(\cdot)$ *or* $\mathcal{S}(\cdot)$ *to each of the resistors,*
- *by applying* $\mathcal{Z}(\cdot)$ *to some controlled sources,*

to arrive at a simpler circuit with the following properties.

1. *The input branches of the controlled sources form a tree.*
2. *The output branches of the controlled sources form a tree.*
3. *Let the fundamental cutset matrix belonging to the tree formed by the input branches be* $\tilde{\mathbf{Q}} = (\mathbf{1} \;\; \mathbf{\Delta})$*. Assume that input and output branches are numbered in the same order, i.e., row* k *and column* k *of* $\mathbf{\Delta}$ *correspond to the input and the output branch of the same controlled source. Then* $(-1)^l \det \mathbf{\Delta} < 0$*, where* l *is the number of VCVS's plus the number of CCVS's in the reduced circuit.*

Proof. Since for the discussion here the distinction between v.c. and c.c. resistors is immaterial, the resistors $5'$ and $10'$ will be absorbed in 5 and 10

respectively. The cut-set matrix then becomes

$$\mathbf{Q} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \mathbf{Q}_{1,6} & \mathbf{Q}_{1,7} & \mathbf{Q}_{1,8} & \mathbf{Q}_{1,9} & \mathbf{Q}_{1,10} \\ 0 & 1 & 0 & 0 & 0 & \mathbf{Q}_{2,6} & \mathbf{Q}_{2,7} & \mathbf{Q}_{2,8} & \mathbf{Q}_{2,9} & \mathbf{Q}_{2,10} \\ 0 & 0 & 1 & 0 & 0 & \mathbf{Q}_{3,6} & \mathbf{Q}_{3,7} & \mathbf{Q}_{3,8} & \mathbf{Q}_{3,9} & \mathbf{Q}_{3,10} \\ 0 & 0 & 0 & 1 & 0 & \mathbf{Q}_{4,6} & \mathbf{Q}_{4,7} & \mathbf{Q}_{4,8} & \mathbf{Q}_{4,9} & \mathbf{Q}_{4,10} \\ 0 & 0 & 0 & 0 & 1 & \mathbf{Q}_{5,6} & \mathbf{Q}_{5,7} & \mathbf{Q}_{5,8} & \mathbf{Q}_{5,9} & \mathbf{Q}_{5,10} \end{pmatrix} \end{array}$$

The matrices $\mathbf{A}$ and $\mathbf{B}$ in (4.20) and (4.21) form a $\mathcal{W}_0$-pair iff the matrix

$$\begin{pmatrix} 0 & \mathbf{Q}_{1,9} & 0 & \mathbf{Q}_{1,8} & 0 & \mathbf{Q}_{1,10} \\ 0 & \mathbf{Q}_{2,9} & 0 & \mathbf{Q}_{2,8} & 0 & \mathbf{Q}_{2,10} \\ 0 & \mathbf{Q}_{5,9} & 0 & \mathbf{Q}_{5,8} & 0 & \mathbf{Q}_{5,10} \\ -\mathbf{Q}^t_{3,6} & 0 & -\mathbf{Q}^t_{5,6} & 0 & -\mathbf{Q}^t_{4,6} & 0 \\ -\mathbf{Q}^t_{3,7} & 0 & -\mathbf{Q}^t_{5,7} & 0 & -\mathbf{Q}^t_{4,7} & 0 \\ -\mathbf{Q}^t_{3,10} & 0 & -\mathbf{Q}^t_{5,10} & 0 & -\mathbf{Q}^t_{4,10} & 0 \end{pmatrix} \tag{4.22}$$

is a P_0-matrix. If it is not, it has a negative principal minor, which can be found by symmetrically deleting rows and columns in (4.22). This corresponds to applying $\mathcal{Z}(\cdot)$ to certain controlled sources, to short-circuiting certain tree resistor branches and to open-circuiting certain cotree resistor branches. Without loss of generality it may therefore be assumed that the circuit has already been reduced in this way and that the determinant of matrix (4.22) is negative. We will now prove that if this circuit is further simplified by open-circuiting the remaining tree resistors, and short-circuiting the remaining cotree resistors, then the conditions in the theorem are satisfied.

If we let ν_1 be the number of CCCS's, ν_2 be the number of CCVS's, ν_3 be the number of VCVS's, ν_4 be the number of VCCS's, ν_5 be the number of tree resistors, and ν_6 be the number of cotree resistors, then we know from the negativity of the determinant of (4.22) that

$$(-1)^\alpha \det \begin{array}{c} \\ \nu_2 \\ \nu_1 \\ \nu_5 \\ \nu_4 \\ \nu_3 \\ \nu_6 \end{array} \begin{array}{c} \begin{array}{cccccc} \nu_1 & \nu_4 & \nu_6 & \nu_2 & \nu_5 & \nu_3 \end{array} \\ \begin{pmatrix} \mathbf{Q}_{1,9} & \mathbf{Q}_{1,8} & \mathbf{Q}_{1,10} & 0 & 0 & 0 \\ \mathbf{Q}_{2,9} & \mathbf{Q}_{2,8} & \mathbf{Q}_{2,10} & 0 & 0 & 0 \\ \mathbf{Q}_{5,9} & \mathbf{Q}_{5,8} & \mathbf{Q}_{5,10} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\mathbf{Q}^t_{3,6} & -\mathbf{Q}^t_{5,6} & -\mathbf{Q}^t_{4,6} \\ 0 & 0 & 0 & -\mathbf{Q}^t_{3,7} & -\mathbf{Q}^t_{5,7} & -\mathbf{Q}^t_{4,7} \\ 0 & 0 & 0 & -\mathbf{Q}^t_{3,10} & -\mathbf{Q}^t_{5,10} & -\mathbf{Q}^t_{4,10} \end{pmatrix} \end{array} < 0,$$

with $\alpha = \nu_1\nu_2 + \nu_4\nu_5 + \nu_4\nu_2 + \nu_6\nu_3 + \nu_6\nu_5 + \nu_6\nu_2$. The determinant can only be nonzero if $\nu_2 + \nu_5 = \nu_4 + \nu_6$. Using this expression the above equation can

be further simplified to

$$(-1)^{\nu_1\nu_2+\nu_3\nu_6+\nu_3\nu_5+\nu_3}\det\begin{pmatrix}\mathbf{Q}_{1,9}&\mathbf{Q}_{1,8}&\mathbf{Q}_{1,10}\\ \mathbf{Q}_{2,9}&\mathbf{Q}_{2,8}&\mathbf{Q}_{2,10}\\ \mathbf{Q}_{5,9}&\mathbf{Q}_{5,8}&\mathbf{Q}_{5,10}\end{pmatrix}\det\begin{pmatrix}\mathbf{Q}_{3,6}&\mathbf{Q}_{3,7}&\mathbf{Q}_{3,10}\\ \mathbf{Q}_{4,6}&\mathbf{Q}_{4,7}&\mathbf{Q}_{4,10}\\ \mathbf{Q}_{5,6}&\mathbf{Q}_{5,7}&\mathbf{Q}_{5,10}\end{pmatrix}<0. \tag{4.23}$$

In particular the second matrix in (4.23) is nonsingular, which means that branches 1,2,6,7 and 10 form another tree in the circuit. This already proves points 1 and 2 of the theorem. In order to prove part 3, let the fundamental cutset matrix for this tree be given by

$$\tilde{\mathbf{Q}}=\begin{array}{c}\begin{matrix}1&2&3&4&5&6&7&8&9&10\end{matrix}\\ \begin{pmatrix}\mathbf{1}&\mathbf{0}&\tilde{\mathbf{Q}}_{1,3}&\tilde{\mathbf{Q}}_{1,4}&\tilde{\mathbf{Q}}_{1,5}&\mathbf{0}&\mathbf{0}&\tilde{\mathbf{Q}}_{1,8}&\tilde{\mathbf{Q}}_{1,9}&\mathbf{0}\\ \mathbf{0}&\mathbf{1}&\tilde{\mathbf{Q}}_{2,3}&\tilde{\mathbf{Q}}_{2,4}&\tilde{\mathbf{Q}}_{2,5}&\mathbf{0}&\mathbf{0}&\tilde{\mathbf{Q}}_{2,8}&\tilde{\mathbf{Q}}_{2,9}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\tilde{\mathbf{Q}}_{6,3}&\tilde{\mathbf{Q}}_{6,4}&\tilde{\mathbf{Q}}_{6,5}&\mathbf{1}&\mathbf{0}&\tilde{\mathbf{Q}}_{6,8}&\tilde{\mathbf{Q}}_{6,9}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\tilde{\mathbf{Q}}_{7,3}&\tilde{\mathbf{Q}}_{7,4}&\tilde{\mathbf{Q}}_{7,5}&\mathbf{0}&\mathbf{1}&\tilde{\mathbf{Q}}_{7,8}&\tilde{\mathbf{Q}}_{7,9}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&\tilde{\mathbf{Q}}_{10,3}&\tilde{\mathbf{Q}}_{10,4}&\tilde{\mathbf{Q}}_{10,5}&\mathbf{0}&\mathbf{0}&\tilde{\mathbf{Q}}_{10,8}&\tilde{\mathbf{Q}}_{10,9}&\mathbf{1}\end{pmatrix}\end{array}.$$

Matrices $\mathbf{Q}$ and $\tilde{\mathbf{Q}}$ are related as $\tilde{\mathbf{Q}}=\mathbf{M}\mathbf{Q}$ with

$$\mathbf{M}=\begin{array}{c}\begin{matrix}\nu_1+\nu_2&\qquad&\nu_2+\nu_3+\nu_6\end{matrix}\\ \begin{matrix}\nu_1+\nu_2\\ \nu_3+\nu_4+\nu_6\end{matrix}\begin{pmatrix}\mathbf{1}&-\begin{pmatrix}\mathbf{Q}_{1,6}&\mathbf{Q}_{1,7}&\mathbf{Q}_{1,10}\\ \mathbf{Q}_{2,6}&\mathbf{Q}_{2,7}&\mathbf{Q}_{2,10}\end{pmatrix}\begin{pmatrix}\mathbf{Q}_{3,6}&\mathbf{Q}_{3,7}&\mathbf{Q}_{3,10}\\ \mathbf{Q}_{4,6}&\mathbf{Q}_{4,7}&\mathbf{Q}_{4,10}\\ \mathbf{Q}_{5,6}&\mathbf{Q}_{5,7}&\mathbf{Q}_{5,10}\end{pmatrix}^{-1}\\ \mathbf{0}&\begin{pmatrix}\mathbf{Q}_{3,6}&\mathbf{Q}_{3,7}&\mathbf{Q}_{3,10}\\ \mathbf{Q}_{4,6}&\mathbf{Q}_{4,7}&\mathbf{Q}_{4,10}\\ \mathbf{Q}_{5,6}&\mathbf{Q}_{5,7}&\mathbf{Q}_{5,10}\end{pmatrix}^{-1}\end{pmatrix}\end{array}.$$

One therefore has that

$$\frac{\begin{array}{c}\begin{matrix}\nu_2&\nu_1&\nu_4&\nu_3&\nu_6\end{matrix}\\ \det\begin{matrix}\nu_2\\ \nu_1\\ \nu_4\\ \nu_3\\ \nu_6\end{matrix}\begin{pmatrix}\mathbf{0}&\mathbf{Q}_{1,9}&\mathbf{Q}_{1,8}&\mathbf{0}&\mathbf{Q}_{1,10}\\ \mathbf{0}&\mathbf{Q}_{2,9}&\mathbf{Q}_{2,8}&\mathbf{0}&\mathbf{Q}_{2,10}\\ \mathbf{1}&\mathbf{Q}_{3,9}&\mathbf{Q}_{3,8}&\mathbf{0}&\mathbf{Q}_{3,10}\\ \mathbf{0}&\mathbf{Q}_{4,9}&\mathbf{Q}_{4,8}&\mathbf{1}&\mathbf{Q}_{4,10}\\ \mathbf{0}&\mathbf{Q}_{5,9}&\mathbf{Q}_{5,8}&\mathbf{0}&\mathbf{Q}_{5,10}\end{pmatrix}\end{array}}{\det\begin{pmatrix}\mathbf{Q}_{3,6}&\mathbf{Q}_{3,7}&\mathbf{Q}_{3,10}\\ \mathbf{Q}_{4,6}&\mathbf{Q}_{4,7}&\mathbf{Q}_{4,10}\\ \mathbf{Q}_{5,6}&\mathbf{Q}_{5,7}&\mathbf{Q}_{5,10}\end{pmatrix}}=\det\begin{array}{c}\begin{matrix}\nu_2&\nu_1&\nu_4&\nu_3&\nu_6\end{matrix}\\ \begin{matrix}\nu_2\\ \nu_1\\ \nu_4\\ \nu_3\\ \nu_6\end{matrix}\begin{pmatrix}\tilde{\mathbf{Q}}_{1,3}&\tilde{\mathbf{Q}}_{1,9}&\tilde{\mathbf{Q}}_{1,8}&\tilde{\mathbf{Q}}_{1,4}&\mathbf{0}\\ \tilde{\mathbf{Q}}_{2,3}&\tilde{\mathbf{Q}}_{2,8}&\tilde{\mathbf{Q}}_{2,9}&\tilde{\mathbf{Q}}_{2,4}&\mathbf{0}\\ \tilde{\mathbf{Q}}_{6,3}&\tilde{\mathbf{Q}}_{6,8}&\tilde{\mathbf{Q}}_{6,9}&\tilde{\mathbf{Q}}_{6,4}&\mathbf{0}\\ \tilde{\mathbf{Q}}_{7,3}&\tilde{\mathbf{Q}}_{7,8}&\tilde{\mathbf{Q}}_{7,9}&\tilde{\mathbf{Q}}_{7,4}&\mathbf{0}\\ \tilde{\mathbf{Q}}_{10,3}&\tilde{\mathbf{Q}}_{10,8}&\tilde{\mathbf{Q}}_{10,9}&\tilde{\mathbf{Q}}_{10,4}&\mathbf{1}\end{pmatrix}\end{array},$$

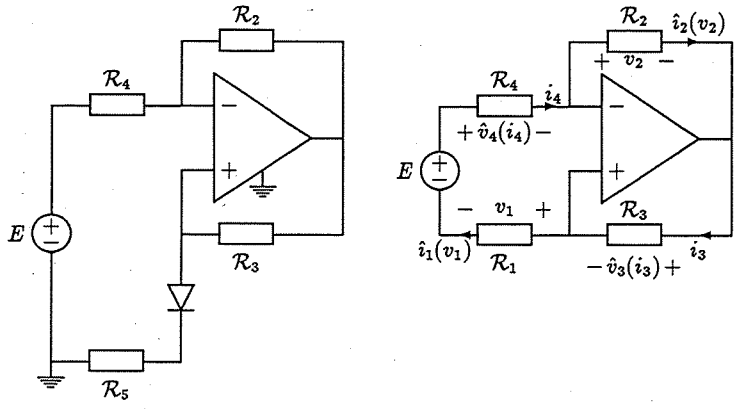


Figure 4.5: Nonlinear circuit with one ideal op-amp.

and from this and (4.23) it follows that

$$(-1)^{\nu_2+\nu_3}\det\begin{pmatrix}\tilde{\mathbf{Q}}_{1,3}&\tilde{\mathbf{Q}}_{1,9}&\tilde{\mathbf{Q}}_{1,8}&\tilde{\mathbf{Q}}_{1,4}\\ \tilde{\mathbf{Q}}_{2,3}&\tilde{\mathbf{Q}}_{2,9}&\tilde{\mathbf{Q}}_{2,8}&\tilde{\mathbf{Q}}_{2,4}\\ \tilde{\mathbf{Q}}_{6,3}&\tilde{\mathbf{Q}}_{6,9}&\tilde{\mathbf{Q}}_{6,8}&\tilde{\mathbf{Q}}_{6,4}\\ \tilde{\mathbf{Q}}_{7,3}&\tilde{\mathbf{Q}}_{7,9}&\tilde{\mathbf{Q}}_{7,8}&\tilde{\mathbf{Q}}_{7,4}\end{pmatrix}<0.$$

This proves the third part of the theorem. □

If, in addition, all nonlinear characteristics are mononotonically increasing, one can follow the reasoning of the example in Section 2.2.7 and conclude that the topological criterion of this theorem is sufficient to have uniqueness of these solution. This provides a much shorter proof of the result in [69].

4.4.3 Circuits containing ideal operational amplifiers

Example. Figure 4.5 shows a circuit with one operational amplifier, and an exponential-type diode. Resistors $\mathcal{R}_5$ and $\mathcal{R}_2$ are voltage-controlled resistors, resistors $\mathcal{R}_3$ and $\mathcal{R}_4$ are current-controlled. All four resistors are of type U, with characteristics asymptotically in sector $(0,1)$. For the sake of conciseness resistors $\mathcal{R}_5$ will be taken together with the exponential diode, as one single resistor $\mathcal{R}_1$ of type H^+, asymptotically in $(0,1)$. One can write four equations

of the form (4.4) :

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \hat{i}_1(v_1) \\ \hat{i}_2(v_2) \\ \hat{v}_3(i_3) \\ \hat{v}_4(i_4) \end{pmatrix} + \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ i_3 \\ i_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ E \end{pmatrix}. \quad (4.24)$$

In this simple case the condition stated in Theorem 18 reduces to the following : the nonparallel condition holds if for all choices $d_1, d_2, d_3, d_4 \in \{-1, 0, 1\}$, not all zero, one can find a vector $\mathbf{p}$ s.t.

$$\begin{aligned} d_1(p_1 + p_4) &\geq 0 && (4.25) \\ d_1 p_4 &\geq 0 && (4.26) \\ d_2(p_2 + p_3) &\geq 0 && (4.27) \\ d_2 p_3 &\geq 0 && (4.28) \\ d_3(p_3 - p_1) &\geq 0 && (4.29) \\ -d_3 p_1 &\geq 0 && (4.30) \\ d_4(p_4 - p_2) &\geq 0 && (4.31) \\ -d_4 p_2 &\geq 0 && (4.32) \end{aligned}$$

and at least one of the inequalities (4.26)-(4.32) holds with strict inequality. Alternatively, the existence of such a $\mathbf{p}$ is equivalent to the nonexistence of vectors $\mathbf{x}, \mathbf{y}$ s.t. (see Appendix B)

$$\begin{pmatrix} d_1 & 0 & -d_3 & 0 & 0 & 0 & -d_3 & 0 \\ 0 & d_2 & 0 & -d_4 & 0 & 0 & 0 & -d_4 \\ 0 & d_2 & d_3 & 0 & 0 & d_2 & 0 & 0 \\ d_1 & 0 & 0 & d_4 & d_1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 0, \quad (4.33)$$

with $y_1 \geq 0$, $y_2, \ldots y_4 > 0$, $x_1, \ldots x_4 > 0$ [81]. A possible interpretation of this condition is shown in fig 4.6. If we identify $i_1 = d_1 y_1$, $v_1 = d_1(y_1 + x_1)$, $i_2 = d_2 y_2$, $v_2 = d_2(y_2 + x_2)$, $i_3 = d_3(x_3 + y_3)$, $v_3 = d_3 y_3$, $i_4 = d_4(x_4 + y_4)$, and $v_4 = d_4 y_4$, then condition (4.33) will hold if there is a nonzero solution to the linear circuit of fig. 4.6 for some

$$0 \leq G_1 < 1, \quad 0 < G_2 < 1, \quad 0 < R_3 < 1, \quad 0 < R_4 < 1.$$

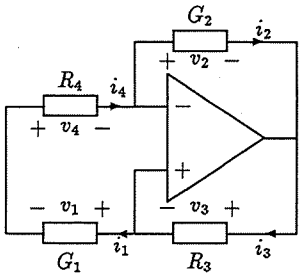


Figure 4.6: Interpretation of condition (4.33). G_1 and G_2 are conductances, R_4 and R_3 resistances.

It can be verified that no such solution exists. This confirms that equations (4.24) satisfy the nonparallel condition.

General circuit equations for ideal opamp circuits can be formulated by using the concept of *conjugate trees* (see Section 2.6). As in the previous section, an important special case occurs if all nonlinear resistor characteristics are of type U and asymptotically in sector $(0, \infty)$. The conditions in Theorem 19 then again reduce to a purely topological condition, the nonexistence of a so-called *uniform partial orientation*.

4.5 Computing multiple solutions

4.5.1 Method

The reflection algorithm also leads to a modification of the octahedral algorithm to find multiple solutions. The idea is as follows. When a solution $\mathbf{x}^*$ is reached, one knows that the pair $(\mathbf{x}^*, \mathbf{f}(\mathbf{x}^*))$ lies in the horizontal segment of each characteristic of fig. 2.2. Moreover, $t = \|\mathbf{x}^*\|_\infty$, and there must be an index k for which $t = x_k^*$ or $t = -x_k^*$. One can then replace the k-th constitutive relation by the one in fig. 4.7 and continue the algorithm. Since fig. 4.7 defines an active resistor, the homotopy path may now diverge, but as long as $\mathbf{f}(\cdot)$ satisfies the nonparallel condition, this can be overcome by reflection

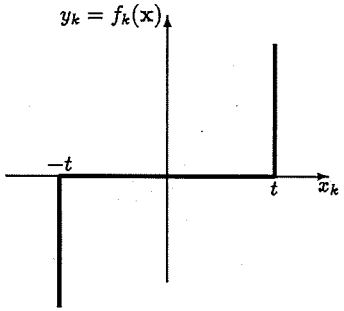


Figure 4.7: Modified constitutive relation.

	v_1	v_2	v_3	v_4	Number of funct. eval.
1	2.81	2.89	6.85	10	71
2	2.83	3.29	7.95	8.83	183
3	3.35	3.84	9.49	6.68	249

Table 4.2: Solutions for Example 1 and required number of function evaluations.

steps as above. In this way, a number of zeros can be enumerated, until the path arrives back at the origin.

4.5.2 Examples

Example 1. Consider the Schmitt-trigger circuit in figure 4.8. The algorithm can be applied to the nodal equations. We take $v_k^0 = 5 \quad k = 1,\ldots,4$ as starting point, and initial grid size 0.2. The algorithm finds the three solutions of table 4.2 after 71, 183 and 249 function evaluations, respectively.

Example 2. Fig.3.5 is taken from [16]. Nodal analysis leads to a system of 10 equations. The solutions found by the algorithm are listed in table 4.3. Here the algorithm was started in $v_k^0 = 5, \quad k = 1,\ldots,10$ with grid size 0.3.

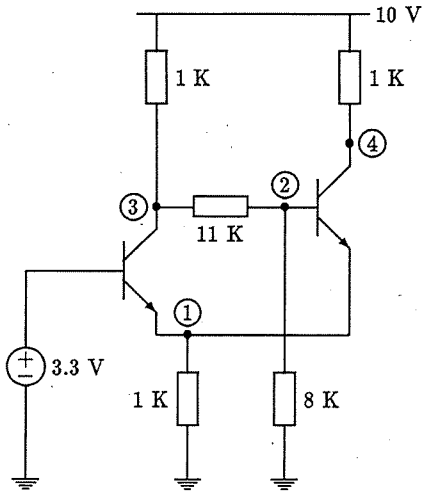


Figure 4.8: Schmitt-trigger.

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	Number of funct. eval.
1	1.87	−1.04	10.47	0.36	1.39	1.44	1.44	1.39	0.36	10.70	268
2	1.87	−2.57	2.99	1.90	1.42	8.17	1.45	1.39	0.36	10.52	1122
3	1.86	−2.88	1.52	1.92	1.43	9.70	1.44	1.38	0.36	10.48	1482
4	1.85	−4.33	1.50	1.90	1.42	9.62	8.05	1.40	1.88	3.30	2024
5	1.89	−4.18	2.60	1.92	1.44	8.26	8.26	1.44	1.92	2.83	2294
6	1.87	−2.57	10.29	0.36	1.39	1.45	8.17	1.42	1.90	3.23	3366
7	1.86	−2.93	10.24	0.36	1.38	1.44	9.92	1.44	1.92	1.52	3742
8	1.84	−4.36	3.17	1.87	1.39	8.01	9.90	1.42	1.90	1.50	4284
9	1.73	−4.80	1.39	1.81	1.33	10.47	10.61	1.34	1.82	1.41	4503

Table 4.3: Solutions for Example 2 and required number of function evaluations.

4.6 Conclusion

In this chapter an important extension of the octahedral algorithm is given. It makes the algorithm applicable to circuits with ideal operational amplifiers and controlled sources, a case traditionally excluded from discussions of homotopy methods.

The ideas in this chapter should be used in combination with existing techniques. One can distinguish the following hierarchy. In general it is clearly a wise choice to try a conventional iterative method in the first place. These methods offer fast convergence, but may fail due to non-monotonicity of the equations if the initial guess is too poor. If such problems occur, homotopy methods offer a way out owing to their insensitivity to local behaviour. Instead they rely on assumptions about the global properties of the equations (Chapter 2 contains a number of typical examples). This is justified in most circuits with independent sources and eventually passive components and, in particular, in common electronic circuits. In case of active elements, however, this assumption 'by default' may turn out to be wrong, and the homotopy path may diverge, unless a means is provided to revise the initial assumption. This is precisely what the reflection steps described above amount to.

The criteria for convergence of this generalized method have been described in Section 4.4 for circuits with controlled sources and operational amplifiers, taking into account the topology of the circuit, and the asymptotic behaviour of the nonlinearities. An important open problem is to characterize the circuits where it is possible to estimate the degree of the circuit equations, or the index of solutions on topological grounds. This information would greatly increase the efficiency of the algorithm.

One of the important advantages of homotopy methods, beside global convergence, lies in their capability of finding multiple solutions in a systematic way. An extension of the octahedral algorithm for this purpose has been illustrated in Section 4.5.

The idea of reflecting the path in order to achieve broader convergence or to find multiple solutions, was suggested in [96, 119] for the $2n$-ray algorithm. Here we have applied this idea to the octahedral algorithm and have also discussed some practical details (Section 4.3.2) not touched upon in the references.

The main contribution in this chapter is an investigation of the meaning of the nonparallel criterion in circuits with active elements. This nonparallel condition has also been studied in connection with mappings $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{g}(\mathbf{x}) + \mathbf{B}\mathbf{x} + \mathbf{c}$ by Chua and Wang [17]. Theorem 18 generalizes their result and the one in [87] in two important ways : (1) it includes as a special case all extreme

situations that are given as examples at the end of Section 4.4.1; (2) it takes into account the sectors in which the unbounded components of $\mathbf{g}(\cdot)$ lie.

Seen as theorems on existence of solutions in circuits with controlled sources, the results of Sections 4.4.2 generalize the work of Nishi and Chua [69, 70]. We consider *non-monotonic* resistor characteristics, *nonlinear* controlled sources, and *upper and lower bounds* on all characteristics. Again we first examine existence of solutions and hence deduce uniqueness. This approach also leads to a much shorter proof of the topological condition given in [69].

Similarly, the inclusion of *upper and lower bounds*, and the use of *non-monotonic characteristics*, form an extension of Hasler's work on circuits with ideal operational amplifiers [45, 46].

Chapter 5

Nonlinear circuits and the complementarity problem

5.1 Introduction

The linear complementarity problem (LCP) [23] is defined as follows : given an $n \times n$-matrix $\mathbf{M}$ and an n-vector $\mathbf{q}$, find vectors $\mathbf{x}$ and $\mathbf{y}$ such that

$$\mathbf{y} = \mathbf{M}\mathbf{x} + \mathbf{q}, \quad \mathbf{y} \geq \mathbf{0}, \, \mathbf{x} \geq \mathbf{0}, \quad \mathbf{y}^t.\mathbf{x} = 0. \tag{5.1}$$

If the affine mapping $\mathbf{y} = \mathbf{M}\mathbf{x} + \mathbf{q}$ in (5.1) is replaced by an arbitrary nonlinear mapping $\mathbf{y} = \mathbf{f}(\mathbf{x})$, one has the nonlinear complementarity problem [51] :

$$\mathbf{y} = \mathbf{f}(\mathbf{x}), \quad \mathbf{y} \geq \mathbf{0}, \, \mathbf{x} \geq \mathbf{0}, \quad \mathbf{y}^t.\mathbf{x} = 0. \tag{5.2}$$

These problems have been the object of intensive study as a unified description of numerous problems in mathematical programming and game theory. Problem (5.1) can also be interpreted as the description of a linear n-port $\mathbf{y} = \mathbf{M}\mathbf{x} + \mathbf{q}$, terminated with n ideal diodes (see fig. 5.1). As an extension of this, the LCP can be used as a general model for piecewise-linear circuits, a fact that has been exploited in [90, 99, 100].

The LCP has also an important historical significance. It was Lemke's pivoting algorithm for the LCP that led to the development of simplicial algorithms for solving sets of nonlinear equations [88, 32]. In the first half (Section 5.2) of this chapter we illustrate this by showing the connection between Lemke's algorithm and the methods discussed in the previous chapters.

Section 5.3 applies the LCP in an entirely different setting. It will be shown how a generalized version of the LCP can be used to analyze a broad class of piecewise-linear circuits [100, 26]. The method is very general, finds all solutions but is limited to small problems.

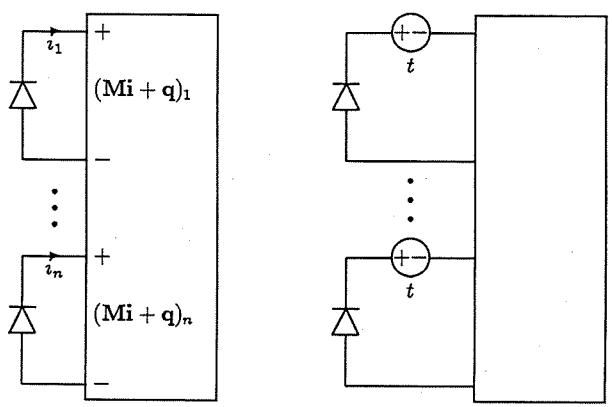


Figure 5.1: Interpretation of the LCP (5.1) as the description of a network with linear resistors and ideal diodes (left). Principle of Lemke's algorithm (right).

5.2 Relation between variable dimension algorithms and Lemke's algorithm

The purpose of this section is to show that the methods described in the previous chapters can be viewed as applications of Lemke's algorithm to an underlying nonlinear complementarity problem. The discussion is based on [54].

5.2.1 The linear complementarity problem and Lemke's algorithm

To fix the ideas, we assume that problem (5.1) is the description of a current-controlled n-port. In other words, $\mathbf{x}$ is a vector of port currents and $\mathbf{y}$ is a vector of port voltages. We also assume that $\mathbf{q} \not\geq \mathbf{0}$, otherwise the solution is trivial.

Lemke's algorithm is based on the insertion of voltage sources as in fig. 5.1. The circuit equations then become :

$$\mathbf{v} = \mathbf{M}\mathbf{i} + \mathbf{q} + \mathbf{e}t, \quad \mathbf{v} \geq \mathbf{0}, \, \mathbf{i} \geq \mathbf{0}, t \geq 0, \quad \mathbf{v}^t\mathbf{i} = 0, \tag{5.3}$$

where $\mathbf{e}$ is the n-vector consisting of all ones. Instead of $\mathbf{e}$, any other positive vector can be used as well. Lemke's algorithm generates a path S of points $(t, \mathbf{i}, \mathbf{v})$ satisfying (5.3). An initial point is easily found as

$$t^0 = \min\{t \geq 0 : \mathbf{q} + \mathbf{e}t \geq \mathbf{0}\}, \quad \mathbf{i}^0 = \mathbf{0}, \quad \mathbf{v}^0 = \mathbf{q} + \mathbf{e}t^0.$$

Barring degeneracy, it can be shown that S must be a piecewise-linear path. Moreover it cannot be a loop because it contains the half-line $\{(t, \mathbf{0}, \mathbf{q} + \mathbf{e}t) : t \geq t^0\}$. It must therefore go to infinity or reach an endpoint with $t = 0$, i.e. a solution to (5.1). A classical result states that the algorithm always converges if $\mathbf{M}$ is a P-matrix [23].

5.2.2 The nonlinear complementarity problem

In case the n-port in fig. 5.1 is nonlinear and described by $\mathbf{v} = \mathbf{f}(\mathbf{i})$ we obtain the nonlinear complementarity problem (5.2). Lemke's algorithm can be extended to this problem. After inserting voltage sources the equations become :

$$\mathbf{v} = \mathbf{f}(\mathbf{i}) + \mathbf{e}t, \quad \mathbf{v} \geq \mathbf{0},\ \mathbf{i} \geq \mathbf{0}, t \geq 0, \quad \mathbf{v}^t\mathbf{i} = 0. \tag{5.4}$$

Again, one starts with

$$t^0 = \min\{t \geq 0 : \mathbf{f}(\mathbf{0}) + \mathbf{e}t \geq \mathbf{0}\}, \quad \mathbf{i}^0 = \mathbf{0}, \quad \mathbf{v}^0 = \mathbf{f}(\mathbf{0}) + \mathbf{e}t^0,$$

and then attempts to decrease t until $t = 0$. Of course the path will no longer be piecewise-linear, but it can be approximated by replacing $\mathbf{f}(\cdot)$ by a piecewise-linear approximation on a simplicial subdivision.

5.2.3 Equivalence between a set of nonlinear equations and the nonlinear complementarity problem

As an example, it will now be shown how the $2n$-ray algorithm of section 2.3.1 (see fig. 5.2) can be interpreted as Lemke's algorithm applied to an underlying nonlinear complementarity problem. In fig. 5.2 one can distinguish $2n$ loops (fig. 5.3). With each loop a voltage E_k (the sum of the voltages across the diodes in the loop, all measured in the same direction, and such that $E_k \geq 0$) and a current I_k (oriented such that $I_k \geq 0$) can be associated. Then it follows

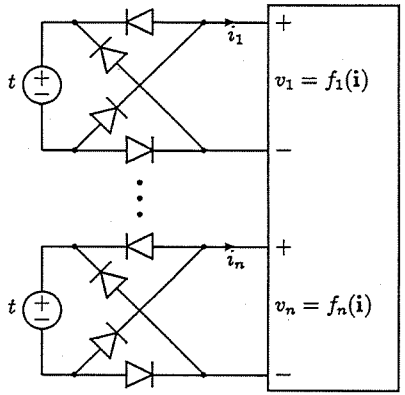


Figure 5.2: The $2n$-ray algorithm applied to a current-controlled n-port. Initially the parameter $t = \infty$. The algorithm attempts to decrease t until $t = 0$.

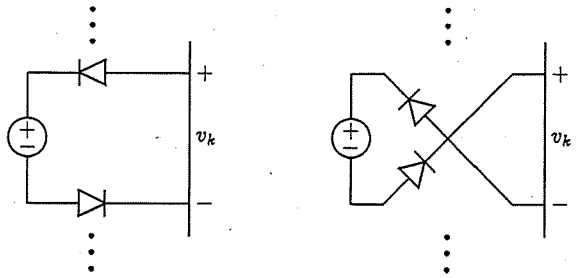


Figure 5.3: Loops k and $n + k$ in the circuit of fig. 5.2.

that

$$\begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \\ E_{n+1} \\ E_{n+2} \\ \vdots \\ E_{2n} \end{bmatrix} = \begin{bmatrix} -1 & 0 & & 0 \\ 0 & -1 & & 0 \\ & & \ddots & \\ 0 & 0 & & -1 \\ 1 & 0 & & 0 \\ 0 & 1 & & 0 \\ & & \ddots & \\ 0 & 0 & & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} + \mathbf{e}t,$$

or

$$\mathbf{E} = \mathbf{B}^t.\mathbf{v} + \mathbf{e}t,$$

and

$$\begin{bmatrix} i_1 \\ \vdots \\ i_n \end{bmatrix} = \begin{bmatrix} -1 & 0 & & 0 & 1 & 0 & & 0 \\ 0 & -1 & & 0 & 0 & 1 & & 0 \\ & & \ddots & & & & \ddots & \\ 0 & 0 & & -1 & 0 & 0 & & 1 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_n \\ I_{n+1} \\ I_{n+2} \\ \vdots \\ I_{2n} \end{bmatrix},$$

or $\mathbf{i} = \mathbf{B}.\mathbf{I}$. When $t = 0$ this leads to the nonlinear complementarity problem

$$\mathbf{E} = \mathbf{B}^t.\mathbf{f}(\mathbf{BI}), \quad \mathbf{I} \geq \mathbf{0}, \ \mathbf{E} \geq \mathbf{0}, \quad \mathbf{I}^t.\mathbf{E} = 0. \tag{5.5}$$

The path traced by the $2n$-ray algorithm is the solution to the augmented circuit of fig. 5.2 :

$$\mathbf{E} = \mathbf{B}^t.\mathbf{f}(\mathbf{BI}) + \mathbf{e}.t, \quad \mathbf{I} \geq \mathbf{0}, \ \mathbf{E} \geq \mathbf{0}, t \geq 0, \quad \mathbf{I}^t.\mathbf{E} = 0,$$

and can therefore be identified with the path found by Lemke's algorithm applied to (5.5). A similar equivalence can be established for the octahedral algorithm and the $(3^n - 1)$-ray algorithm by using different matrices $\mathbf{B}$.

5.3 The complete solution of a piecewise-linear circuit

5.3.1 Introduction

In this section we present an entirely different application of the linear complementarity problem [100, 25, 26]. In these references a method is described

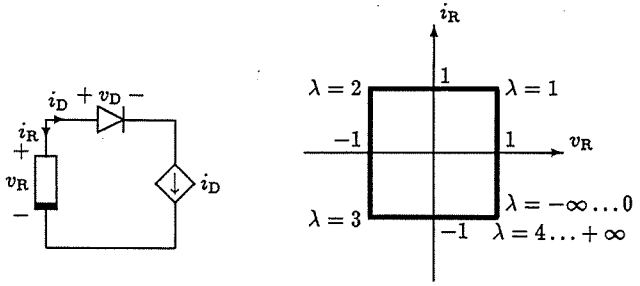


Figure 5.4: Piecewise-linear circuit containing a nonlinear resistor, an ideal diode and a current-controlled current source.

to find all solutions in a very broad class of piecewise-linear circuits. The algorithm is based on a generalization of the LCP to model piecewise-linear circuits. Figures 5.4 and 5.5 illustrate the power of the method. In both cases, the solution set (shown in fig 5.8 and fig 5.9) is unbounded. We now give a brief account of the LCP-model the method is based on.

5.3.2 A parametrization of piecewise-linear resistors

A piecewise-linear curve in R^m (fig 5.6) characterized by a set of $n+1$ breakpoints $\mathbf{x}^0, \mathbf{x}^1, \ldots, \mathbf{x}^n$, and two directions $\mathbf{x}^{-\infty}$ and $\mathbf{x}^{+\infty}$, can be parametrized as follows :

$$\mathbf{x} = \mathbf{x}^0 + \mathbf{x}^{-\infty}.\lambda^- + (\mathbf{x}^1 - \mathbf{x}^0).\lambda^+ + \sum_{k=2}^{n} (\mathbf{x}^k - 2\mathbf{x}^{k-1} + \mathbf{x}^{k-2}).(\lambda - k + 1)^+ + (\mathbf{x}^{+\infty} - \mathbf{x}^n + \mathbf{x}^{n-1}).(\lambda - n)^+, \tag{5.6}$$

where $\lambda^+ = \max(\lambda, 0)$ and $\lambda^- = \max(-\lambda, 0)$. Substituting the auxiliary variables $\lambda_i = \lambda - i$, or, equivalently,

$$\lambda_i^+ - \lambda_i^- = \lambda^+ - \lambda^- - i, \tag{5.7}$$

in (5.6) and adding the conditions (5.7) gives the complete piecewise-linear description :

$$\begin{aligned} \mathbf{x} &= \mathbf{x}^0 + \mathbf{X}^-.\boldsymbol{\Lambda}^- + \mathbf{X}^+.\boldsymbol{\Lambda}^+, \\ \mathbf{E}.(\boldsymbol{\Lambda}^+ - \boldsymbol{\Lambda}^-) &= \mathbf{u}, \\ \boldsymbol{\Lambda}^- \geq \mathbf{0}, \ \boldsymbol{\Lambda}^+ &\geq \mathbf{0}, \\ (\boldsymbol{\Lambda}^-)^t.\boldsymbol{\Lambda}^+ &= 0, \end{aligned} \tag{5.8}$$

where

$$\boldsymbol{\Lambda}^- = [\lambda^- \ \lambda_1^- \ \ldots \lambda_n^-]^t,$$

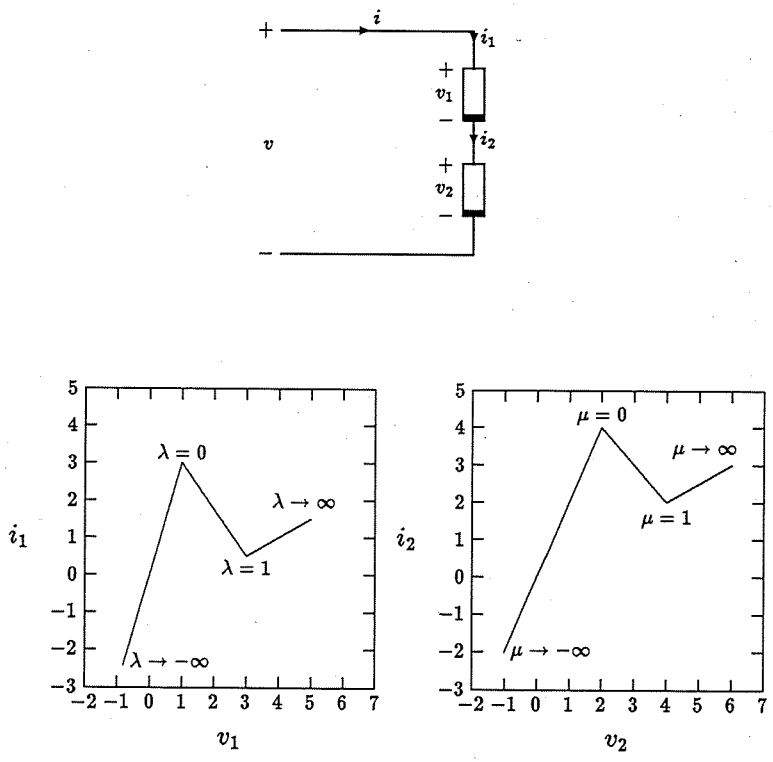


Figure 5.5: Series connection of two piecewise-linear resistors.

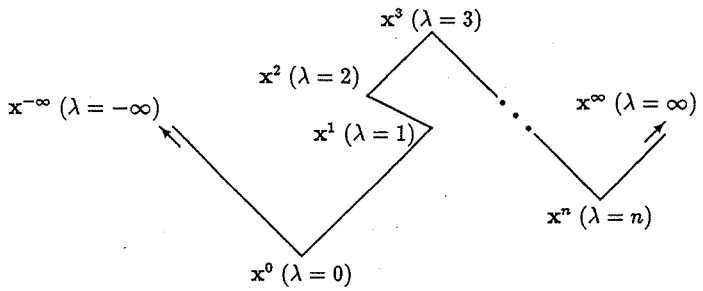


Figure 5.6: Piecewise-linear curve.

$$\boldsymbol{\Lambda}^+ = [\lambda^+ \ \lambda_1^+ \ \ldots \lambda_n^+]^t,$$
$$\mathbf{X}^- = \left[\mathbf{x}^{-\infty} \ \ \mathbf{0} \ \ \ldots \ \ \mathbf{0} \right] \in R^{m\times(n+1)},$$
$$\mathbf{X}^+ = \left[\mathbf{0} \ \ \mathbf{x}^1 - \mathbf{x}^0 \ \ \mathbf{x}^2 - 2\mathbf{x}^1 + \mathbf{x}^0 \ \ \ldots \ \ \mathbf{x}^{+\infty} - \mathbf{x}^n + \mathbf{x}^{n-1} \right] \in R^{m\times(n+1)},$$
$$\mathbf{E} = \begin{bmatrix} 1 & -1 & 0 & \ldots & 0 \\ 1 & 0 & -1 & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 0 & 0 & \ldots & -1 \end{bmatrix} \in R^{n\times(n+1)},$$
$$\mathbf{u} = [1 \ \ 2 \ \ \ldots \ \ n]^t.$$

Example 1 : The edges of the square (fig. 5.4)
This is an example of a curve which is not controlled by one of the variables. It can be parametrized by assigning a coordinate λ as in fig. 5.4 :

$$\begin{bmatrix} v_R \\ i_R \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix}.\lambda^+ + \begin{bmatrix} -2 \\ -2 \end{bmatrix}.\lambda_1^+ + \begin{bmatrix} 2 \\ -2 \end{bmatrix}.\lambda_2^+ + \begin{bmatrix} 2 \\ 2 \end{bmatrix}.\lambda_3^+ - \begin{bmatrix} 2 \\ 0 \end{bmatrix}.\lambda_4^+,$$

$$\lambda_1^+ - \lambda_1^- = \lambda^+ - \lambda^- - 1,$$
$$\lambda_2^+ - \lambda_2^- = \lambda^+ - \lambda^- - 2,$$
$$\lambda_3^+ - \lambda_3^- = \lambda^+ - \lambda^- - 3,$$
$$\lambda_4^+ - \lambda_4^- = \lambda^+ - \lambda^- - 4$$
$$\lambda^+.\lambda^- = \lambda_1^+.\lambda_1^- = \lambda_2^+.\lambda_2^- = \lambda_3^+.\lambda_3^- = \lambda_4^+.\lambda_4^- = 0,$$
$$\lambda^+, \lambda^- \geq 0, \ \ \lambda_i^+, \lambda_i^- \geq 0, \ \ i = 1 \ldots 4.$$

Example 2 : the resistors of fig 5.5 are described by

$$\begin{bmatrix} i_1 \\ v_1 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} + \begin{bmatrix} -3 \\ -1 \end{bmatrix}.\lambda^- + \begin{bmatrix} -2.5 \\ 2 \end{bmatrix}.\lambda^+ + \begin{bmatrix} 3.5 \\ 0 \end{bmatrix}.\lambda_1^+,$$

$$\begin{aligned} &\lambda_1^+ - \lambda_1^- = \lambda^+ - \lambda^- - 1, \\ &\lambda^+, \lambda^-, \lambda_1^+, \lambda_1^- \geq 0, \\ &\lambda^+.\lambda^- + \lambda_1^+.\lambda_1^- = 0, \end{aligned}$$

$$\begin{bmatrix} i_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix} + \begin{bmatrix} -4 \\ -2 \end{bmatrix}.\mu^- + \begin{bmatrix} -2 \\ 2 \end{bmatrix}.\mu^+ + \begin{bmatrix} 3 \\ 0 \end{bmatrix}.\mu_1^+,$$

$$\begin{aligned} &\mu_1^+ - \mu_1^- = \mu^+ - \mu^- - 1, \\ &\mu^+, \mu^-, \mu_1^+, \mu_1^- \geq 0, \\ &\mu^+.\mu^- + \mu_1^+.\mu_1^- = 0. \end{aligned}$$

5.3.3 Formulation of circuit equations

Determination of operating points

The parametrization just derived can be used to cast the circuit equations of a piecewise-linear resistive circuit in the standard form of a generalized linear complementarity problem, as will now be shown. In the next section we will demonstrate that the extension to driving-point analysis and transfer characteristics only requires minor modifications.

As a first step all nonlinear resistors are extracted from the circuit as shown in fig. 5.7. The resulting N-port contains only linear resistive elements and independent sources, and can always be described by its constraint matrix description :

$$\mathbf{C}.[i_1\ v_1\ i_2\ v_2\ \ldots\ i_N\ v_N]^t = \mathbf{b} \tag{5.9}$$

where $\mathbf{C}$ is a matrix with $2N$ columns and an arbitrary number of rows. The linear complementarity problem to be solved can easily be derived by taking the piecewise-linear description of each resistor ($k = 1 \ldots N$) :

$$\begin{bmatrix} i_k \\ v_k \end{bmatrix} = \begin{bmatrix} i_k^0 \\ v_k^0 \end{bmatrix} + \mathbf{X}_k^-.\boldsymbol{\Lambda}_k^- + \mathbf{X}_k^+.\boldsymbol{\Lambda}_k^+, \tag{5.10}$$

$$\begin{aligned} &\mathbf{E}_k.\boldsymbol{\Lambda}_k^+ - \mathbf{E}_k.\boldsymbol{\Lambda}_k^- = \mathbf{u}_k, \\ &\boldsymbol{\Lambda}_k^-, \boldsymbol{\Lambda}_k^+ \geq \mathbf{0}, \\ &(\boldsymbol{\Lambda}_k^+)^t.\boldsymbol{\Lambda}_k^- = 0, \end{aligned}$$

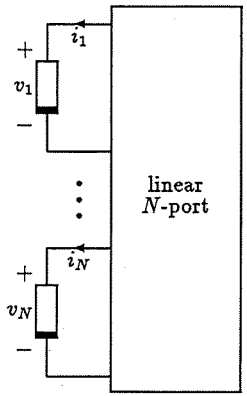


Figure 5.7: All nonlinear resistors are extracted from the circuit. The remaining N-port is linear.

and substituting in (5.9) :

$$\mathbf{C}\begin{bmatrix} \mathbf{X}_1^- & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_2^- & \ldots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{X}_N^- \end{bmatrix}\begin{bmatrix} \boldsymbol{\Lambda}_1^- \\ \vdots \\ \boldsymbol{\Lambda}_N^- \end{bmatrix}$$

$$+\mathbf{C}\begin{bmatrix} \mathbf{X}_1^+ & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_2^+ & \ldots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{X}_N^+ \end{bmatrix}\begin{bmatrix} \boldsymbol{\Lambda}_1^+ \\ \vdots \\ \boldsymbol{\Lambda}_N^+ \end{bmatrix} = \mathbf{b} - \mathbf{C}\begin{bmatrix} i_0^0 \\ v_0^0 \\ i_1^0 \\ v_1^0 \\ \vdots \\ i_N^0 \\ v_N^0 \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{E}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{E}_N \end{bmatrix}\begin{bmatrix} \boldsymbol{\Lambda}_1^+ \\ \vdots \\ \boldsymbol{\Lambda}_N^+ \end{bmatrix} - \begin{bmatrix} \mathbf{E}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{E}_N \end{bmatrix}\begin{bmatrix} \boldsymbol{\Lambda}_1^- \\ \vdots \\ \boldsymbol{\Lambda}_N^- \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_N \end{bmatrix},$$

$$\begin{aligned} &\boldsymbol{\Lambda}_i^- \geq \mathbf{0},\ \boldsymbol{\Lambda}_i^+ \geq \mathbf{0} \text{ for } i = 1 \ldots N, \\ &(\boldsymbol{\Lambda}_i^+)^t.\boldsymbol{\Lambda}_i^- = 0 \text{ for } i = 1 \ldots N. \end{aligned}$$

Solving these equations will yield the complete solution set.

Example 1 (continued). The description of the resistor in fig 5.4 has already been given in Section 5.3.2. The description of the diode is trivial :

$$\begin{bmatrix} i_D \\ v_D \end{bmatrix} = \begin{bmatrix} \lambda_D^+ \\ -\lambda_D^- \end{bmatrix}, \quad \lambda_D^+, \lambda_D^- \geq 0, \quad \lambda_D^+ . \lambda_D^- = 0.$$

Substituting these descriptions in the network equation :

$$[1\ 0\ 1\ 0] \begin{bmatrix} i_R \\ v_R \\ i_D \\ v_D \end{bmatrix} = 0,$$

leads to the equations :

$$\begin{bmatrix} 2 & -2 & -2 & 2 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \lambda^+ \\ \lambda_1^+ \\ \lambda_2^+ \\ \lambda_3^+ \\ \lambda_4^+ \\ \lambda_D^+ \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda^- \\ \lambda_1^- \\ \lambda_2^- \\ \lambda_3^- \\ \lambda_4^- \\ \lambda_D^- \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \\ 4 \end{bmatrix},$$

$$\begin{aligned} &\lambda_i^+, \ \lambda_i^- \geq 0, \ i = 1 \ldots 4, \\ &\lambda^+, \lambda^-, \lambda_D^+, \lambda_D^- \geq 0, \\ &\lambda^+ . \lambda^- = \lambda_1^+ . \lambda_1^- = \lambda_2^+ . \lambda_2^- = \lambda_3^+ . \lambda_3^- = \lambda_4^+ . \lambda_4^- = \lambda_D^- . \lambda_D^+ = 0. \end{aligned}$$

Driving-point and transfer characteristics.

The analysis given in the previous section can easily be extended to the determination of the driving-point plot (i_{in} vs. v_{in}) of a one-port. Together with all nonlinear resistors, we extract the input branch. The constraint matrix description of the remaining $(N+1)$-port is

$$\mathbf{C}.[i_{in}\ v_{in}\ i_1\ v_1\ i_2\ v_2\ \ldots\ i_N\ v_N]^t = \mathbf{b}. \tag{5.11}$$

In order to formulate this as a generalized linear complementarity problem we write $i_{in} = i_{in}^+ - i_{in}^-$, and $v_{in} = v_{in}^+ - v_{in}^-$, with $i_{in}^- \geq 0$, $v_{in}^- \geq 0$, $i_{in}^+ \geq 0$, $v_{in}^+ \geq 0$, and $i_{in}^- . i_{in}^+ + v_{in}^- . v_{in}^+ = 0$. Together with the description of the nonlinear resistors

(5.10) this yields

$$\mathbf{C} \begin{bmatrix} -1 & 0 & 0 & \ldots & 0 \\ 0 & -1 & 0 & \ldots & 0 \\ 0 & 0 & \mathbf{X}_1^- & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \ldots & \mathbf{X}_N^- \end{bmatrix} \begin{bmatrix} i_{IN}^- \\ v_{IN}^- \\ \boldsymbol{\Lambda}_1^- \\ \vdots \\ \boldsymbol{\Lambda}_N^- \end{bmatrix}$$

$$+ \mathbf{C} \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & \mathbf{X}_1^+ & \ldots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \ldots & \mathbf{X}_N^+ \end{bmatrix} \begin{bmatrix} i_{IN}^+ \\ v_{IN}^+ \\ \boldsymbol{\Lambda}_1^+ \\ \vdots \\ \boldsymbol{\Lambda}_n^+ \end{bmatrix} = \mathbf{b} - \mathbf{C} \begin{bmatrix} 0 \\ 0 \\ i_0^0 \\ v_0^0 \\ i_1^0 \\ v_1^0 \\ \vdots \\ i_N^0 \\ v_N^0 \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{E}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{E}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_1^+ \\ \vdots \\ \boldsymbol{\Lambda}_N^+ \end{bmatrix} - \begin{bmatrix} \mathbf{E}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_2 & \ldots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{E}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_1^- \\ \vdots \\ \boldsymbol{\Lambda}_N^- \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_N \end{bmatrix},$$

$$\begin{aligned} &\boldsymbol{\Lambda}_i^- \geq \mathbf{0}, \ \boldsymbol{\Lambda}_i^+ \geq \mathbf{0} \text{ for } i = 1 \ldots N, \quad v_{in}^-, i_{in}^-, v_{in}^+, i_{in}^+ \geq 0, \\ &(\boldsymbol{\Lambda}_i^+)^t . \boldsymbol{\Lambda}_i^- = 0 \text{ for } i = 1 \ldots N, \quad v_{in}^- . v_{in}^+ + i_{in}^- . i_{in}^+ = 0. \end{aligned}$$

It is important to note that the matrices in this problem are rectangular.

The extension to the computation of transfer characteristics is straightforward. For instance, in order to trace the v_{out}-v_{in}-characteristic of a circuit the input and the output branch are extracted together with all nonlinear resistors and one applies the same reasoning to the resulting $(N+2)$-port.

Example 2 (continued).
Substituting the resistor descriptions of Section 5.3.2 in the constraint matrix description

$$\begin{bmatrix} 0 & 1 & 0 & -1 & 0 & -1 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} . \begin{bmatrix} i \\ v \\ i_1 \\ v_1 \\ i_2 \\ v_2 \end{bmatrix} = 0,$$

yields the equations

$$\begin{bmatrix} -1 & 0 & 3 & 0 & 4 & 0 \\ 0 & -1 & 1 & 0 & 2 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} i^- \\ v^- \\ \lambda^- \\ \lambda_1^- \\ \mu^- \\ \mu_1^- \end{bmatrix}$$

$$+ \begin{bmatrix} 1 & 0 & 2.5 & -3.5 & 0 & 0 \\ 0 & 1 & -2 & 0 & -2 & 0 \\ 1 & 0 & 0 & 0 & 2 & -3 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} i^+ \\ v^+ \\ \lambda^+ \\ \lambda_1^+ \\ \mu^+ \\ \mu_1^+ \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 4 \\ 1 \\ 1 \end{bmatrix},$$

$$v^-, i^-, \lambda^-, \lambda_1^-, \mu^-, \mu_1^- \geq 0,$$
$$v^+, i^+, \lambda^+, \lambda_1^+, \mu^+, \mu_1^+ \geq 0,$$
$$v^-.v^+ + i^-.i^+ + \lambda^-.\lambda^+ + \lambda_1^-.\lambda_1^+ + \mu^-.\mu^+ + \mu_1^-.\mu_1^+ = 0.$$

5.3.4 The generalized linear complementarity problem

The term generalized linear complementarity problem for the problems encountered above is justified by a number of differences with the classical formulation (5.1). The circuit equations derived above have the following form.

Given $\mathbf{M} \in R^{m \times n}$, $\mathbf{N} \in R^{m \times n}$, $\mathbf{q} \in R^m$, find $\mathbf{w}$, $\mathbf{z}$ and α such that

$$\begin{gathered} \mathbf{N}\mathbf{z} = \mathbf{M}\mathbf{w} + \mathbf{q}.\alpha, \\ \mathbf{w}, \mathbf{z} \geq \mathbf{0}, \quad \alpha \in \{0, 1\}, \\ \mathbf{w}^t.\mathbf{z} = 0. \end{gathered} \tag{5.12}$$

There are three distinct differences between this formulation and (5.1) :

- In (5.12) rectangular matrices are allowed ($m < n$). In the previous section it is shown how these occur in the determination of driving-point and transfer characteristics.
- Even if $m = n$, we do not exclude cases where no reordering of the variables $\mathbf{w}$ and $\mathbf{z}$ can provide an invertible matrix $\mathbf{N}$.

- The scalar α has been added to allow for solutions at infinity, i.e. directions where the solution set is unbounded. These solutions have $\alpha = 0$.

An algorithm to find the complete solution set of (5.12) is discussed in [25, 26], and will not be repeated here. The output of the algorithm is a list of extreme solutions $(\mathbf{w}^k, \mathbf{z}^k, \alpha^k)$, with $\alpha^k = 0$ for a point at infinity, and $\alpha^k = 1$ for a finite point. From this list of extreme points, the complete solution set can be generated by checking the cross-complementarity of the extreme solutions : two solutions $(\mathbf{w}^k, \mathbf{z}^k, \alpha^k)$ and $(\mathbf{w}^j, \mathbf{z}^j, \alpha^j)$ are called cross-complementary if $(\mathbf{w}^k)^t.\mathbf{z}^j = (\mathbf{w}^j)^t.\mathbf{z}^k = 0$. It is proved in [25, 26] that the complete solution set can be found by taking all convex combinations of groups of cross-complementary solutions, i.e. each solution $(\mathbf{w}, \mathbf{z}, \alpha)$ can be written as

$$\mathbf{w} = \sum_{j \in \beta} \lambda_j \mathbf{w}^j, \quad \mathbf{z} = \sum_{j \in \beta} \lambda_j \mathbf{z}^j, \quad \alpha = \sum_{j \in \beta} \lambda_j \alpha^j,$$

where $\{(\mathbf{w}^j, \mathbf{z}^j, \alpha^j) : j \in \beta\}$ is a set of pairwise cross-complementary extreme solutions and $\lambda_j > 0, \quad \forall j \in \beta$. The complete solution set can therefore be obtained by examining a graph where each node corresponds to an extreme solution of (5.12), and two nodes are connected by a branch only if they are cross-complementarty. Each clique in the graph then corresponds to a polyhedron of solutions of (5.12).

Example 1 (continued.) The solution of this problem is given in table 5.1 and drawn in fig 5.8. The significance of the finite vertices $\boxed{1}$ - $\boxed{5}$ is most easily understood. Solution $\boxed{1}$ corresponds to the point $\lambda = 0$ on resistor characteristic, solution $\boxed{2}$ to the point $\lambda = 0.5$. Moreover these two solutions are cross-complementary and all convex combinations of these are valid solutions (the segment from $\boxed{1}$ to $\boxed{2}$ on the $i - v$-characteristics). The next solution is $\boxed{3}$ ($\lambda = 2.5$), followed by $\boxed{4}$ ($\lambda = 3$) and $\boxed{5}$ ($\lambda = 4$). Checking cross-complementarity reveals that all combinations of $\boxed{3}$ and $\boxed{4}$, and of $\boxed{4}$ and $\boxed{5}$ belong to the solution set. Vertex $\boxed{6}$ lies at infinity (pointing in the direction $[i_D \ \ v_D] = [0 \ \ -1]$), and is cross-complementary with solutions $\boxed{2}$ and $\boxed{3}$. The physical meaning is that, when the diode is blocking, the voltage over it is an arbitrary negative number (a point of the form $[i_D \ \ v_D] = [0 \ \ 0] + \beta[0 \ \ -1]$, with $\beta \geq 0$). The meaning of vertices $\boxed{7}$ and $\boxed{8}$ is less essential : $\boxed{7}$, combined with the cross-complementary solution $\boxed{1}$, yields any point with $\lambda \in (-\infty, 0]$ and $\boxed{8}$, together with $\boxed{5}$, any point with $\lambda \in [0, +\infty)$. Physically, these solutions coincide with $\boxed{1}$, resp. $\boxed{5}$.

Example 2 (continued). The problem has 11 extreme solutions (see table 5.2 and fig. 5.9). Verification of the cross-complementarity conditions leads to the

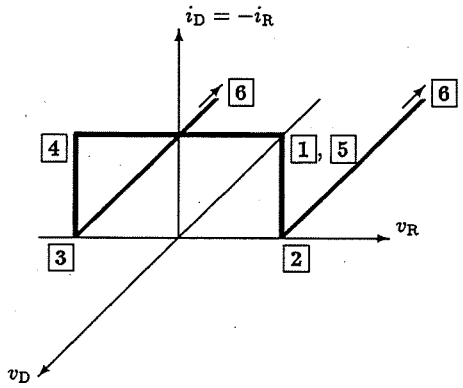


Figure 5.8: Solutions to the circuit in fig. 5.4 (Example 1).

	1	2	3	4	5	6	7	8
λ^+	0	0.5	2.5	3	4	0	0	1
λ_1^+	0	0	1.5	2	3	0	0	1
λ_2^+	0	0	0.5	1	2	0	0	1
λ_3^+	0	0	0	0	1	0	0	1
λ_4^+	0	0	0	0	0	0	0	1
λ_D^+	1	0	0	1	1	0	0	0
λ^-	0	0	0	0	0	0	1	0
λ_1^-	1	0.5	0	0	0	0	1	0
λ_2^-	2	1.5	0	0	0	0	1	0
λ_3^-	3	2.5	0.5	0	0	0	1	0
λ_4^-	4	3.5	1.5	1	0	0	1	0
λ_D^-	0	0	0	0	0	1	0	0
α	1	1	1	1	1	0	0	0

Table 5.1: Solution set of Example 1 (fig 5.4).

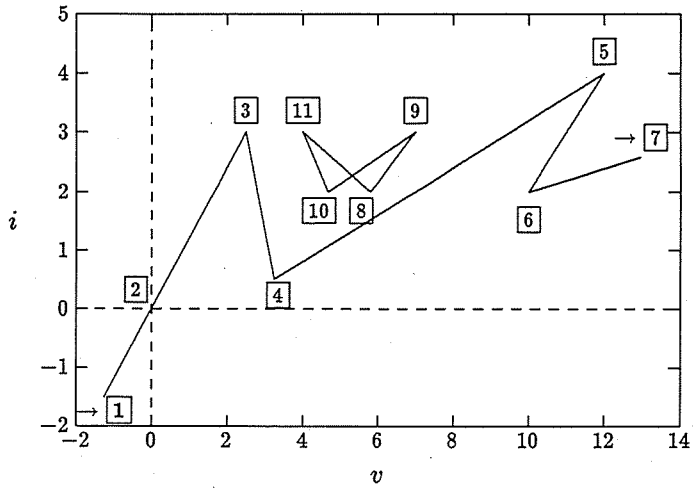


Figure 5.9: Solutions to the circuit in fig. 5.5 (Example 2).

complete driving-point characteristic which consists of two pieces : 1 – 2 – 3 – 4 – 5 – 6 – 7 and 8 – 9 – 10 – 11 – 8.

5.4 Conclusion

In this chapter we discussed two different applications of complementarity theory.

Historically, variable dimension algorithms grew out of Lemke's algorithm for the linear complementarity problem and they inherit its most typical feature, global convergence. Section 5.2 illustrates the close relationship between both methods. The interpretation of variable dimension algorithms in terms of the nonlinear complementarity problem is taken from [54]. It is included here in order to clarify the historical links between variable dimension algorithms and Lemke's algorithm. These algorithms also share a number of basic limitations. Like Lemke's algorithm, variable dimension algorithms converge only under specific conditions, and even if they can be generalized to find multiple solutions, there is no guarantee that all solutions are found.

The method published in [100, 25, 26] uses a generalized linear comple-

	1	2	3	4	5	6	7	8	9	10	11
i^-	0.77	0	0	0	0	0	0	0	0	0	0
v^-	0.64	0	0	0	0	0	0	0	0	0	0
λ^-	0.26	1	0	0	0	0	0	0	0	0.33	0
λ_1^-	0.26	2	1	0	0	0	0	0.6	1	1.33	1
μ^-	0.19	1	0.25	0.88	0	0	0	0	0	0	0
μ_1^-	0.19	2	1.25	1.88	1	0	0	0	0	0	0.5
i^+	0	0	3	0.5	4	2	0.29	2	3	2	3
v^+	0	0	2.5	3.25	12	10	1.48	5.8	7	4.67	4
λ^+	0	0	0	1	4.5	2.5	0.29	0.4	0	0	0
λ_1^+	0	0	0	0	3.5	1.5	0.29	0	0	0	0
μ^+	0	0	0	0	0	1	0.29	1	2	1	0.5
μ_1^+	0	0	0	0	0	0	0.29	0	1	0	0
α	0	1	1	1	1	1	0	1	1	1	1
i	-0.77	0	3	0.50	4	2	0.29	2	3	2	3
v	-0.64	0	2.5	3.25	12	10	1.48	5.8	7	4.67	4

Table 5.2: Solution set of Example 2 (see fig. 5.4).

mentarity problem to model a very broad class of piecewise-linear circuits, and is guaranteed to find all solutions. Due to the combinatorial explosion of complexity the method remains limited to problems of modest size. Moreover it should be remarked that, even if all solutions in a piecewise-linear circuit can be found, this merit is a relative one. Because of constraints in memory and computing capacity, the number of segments in a fixed piecewise-linear model of a nonlinear mapping $\mathbf{f}(\cdot)$ must necessarily be limited, and one might therefore easily overlook zeros of the original $\mathbf{f}(\cdot)$ during the piecewise-linear modelling step.

Chapter 6

Applications to neural networks

6.1 Introduction

An important class of neural networks is described by the differential equation

$$d\mathbf{x}/dt = \mathbf{S}(\mathbf{W}\mathbf{x} - \mathbf{c}) - \mathbf{x}, \tag{6.1}$$

where $\mathbf{x}$ is the n-dimensional state vector, $\mathbf{S}(\cdot)$ is a diagonal mapping from R^n to R^n defined by (fig. 6.1) :

$$\begin{aligned} S_i(\mathbf{x}) &= 1 \quad \text{if} \quad x_i \geq 1, \\ S_i(\mathbf{x}) &= x_i \quad \text{if} \quad -1 \leq x_i \leq 1, \\ S_i(\mathbf{x}) &= -1 \quad \text{if} \quad x_i \leq -1, \end{aligned}$$

$\mathbf{W}$ is an $n \times n$ matrix and $\mathbf{c}$ is a constant n-vector. This model has been

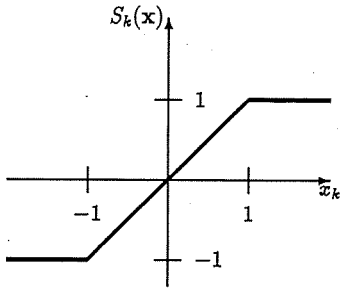


Figure 6.1: The mapping $\mathbf{S}(\cdot)$.

applied to categorical perception and pattern classification under the name of Brain-State-in-a-Box networks [6] and, more recently, in the so-called cellular neural networks for image processing [18, 19].

The more common model for these cellular neural networks is

$$d\boldsymbol{\xi}/dt = -\boldsymbol{\xi} + \mathbf{A}\mathbf{S}(\boldsymbol{\xi}) + \mathbf{B}\mathbf{u} + \boldsymbol{\chi}, \tag{6.2}$$

where $\mathbf{u}$ is an external input, $\mathbf{A}$ and $\mathbf{B}$ are the feedback and feedforward interconnection matrices respectively, and $\boldsymbol{\chi}$ is a constant vector. In addition, the matrices $\mathbf{A}$ and $\mathbf{B}$ have a very sparse and structured form due to the structure of the coefficients : neurons are connected to their nearest neigbours only, and the weights of these interconnections are space-invariant. If the matrix $\mathbf{A}$ is nonsingular, then equation (6.2) can be reduced to (6.1) by identifying $\mathbf{A} = \mathbf{W}$, $\mathbf{B}\mathbf{u} + \boldsymbol{\chi} = -\mathbf{c}$, and applying the transformation $\boldsymbol{\xi} = \mathbf{W}\mathbf{x} - \mathbf{c}$.

Cellular neural networks have been applied for a variety of tasks, see [19, 61, 64, 63]. As an example, a one-dimensional network for detecting connected components is obtained by using the template[1]

1	2	−1

for constructing the matrix $\mathbf{A}$ (see [62]). The external input is zero ($\mathbf{u} = \mathbf{0}$). The cellular neural network detects the number of connected black components on a white background (or vice versa). In order to make the network detect the number of black components, it is assumed that a 0-th and an $(n+1)$-th cell are added to the left and to the right. The states of these cells are kept fixed at -1. The operation is illustrated in fig. 6.2 : the number of black cells in the final state corresponds to the number of connected components in the initial array.

For an array with five cells this leads to an equation (6.2) with

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ 1 & 2 & -1 & 0 & 0 \\ 0 & 1 & 2 & -1 & 0 \\ 0 & 0 & 1 & 2 & -1 \\ 0 & 0 & 0 & 1 & 2 \end{bmatrix}, \quad \boldsymbol{\chi} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Applied to two-dimensional images the template can be used for counting the number of connected components along the horizontal, vertical and diagonal axes. In [62] this system was applied as part of a handwritten character recognition system.

[1] Since the coefficients between each cell and its neighbours are space-invariant, the matrix $\mathbf{A}$ can be constructed by repeating the same pattern for each cell. This pattern is called the defining *template* of the network.

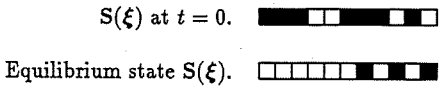


Figure 6.2: Initial and final states of an 11-dimensional connected component detector.

In this chapter it will be shown that mathematical programming and linear inequality theory can be applied to a number of important problems of analysis and synthesis of neural networks.

Section 6.3 deals with the problem of computing equilibrium points. Since proper operation of neural networks (6.1) and (6.2) depends on nonlinear dynamics and on the existence of multiple equilibrium points, it is important to have efficient analysis tools for obtaining a global picture of the net's dynamic behaviour, or at least of its equilibrium pattern. For equation (6.1) this amounts to solving the set of piecewise-linear equations

$$\mathbf{x} = \mathbf{S}(\mathbf{W}.\mathbf{x} - \mathbf{c}). \tag{6.3}$$

This task could be carried out in a trivial way by solving 3^n linear systems, but this is clearly infeasible even for moderate n. The algorithm presented in this chapter is a path-following method and is guaranteed to find at least one solution of (6.3). The possibility of finding all solutions will also be investigated. The method provides new theoretical insight into the relation between the properties of $\mathbf{W}$ and the number of equilibrium points.

In Section 6.4, a theorem will be presented on the stability of the system (6.1). Section 6.5 deals with the synthesis problem, i.e. the problem of finding a suitable matrix $\mathbf{W}$ for a specified task. Large parts of this chapter rely on the geometrical interpretation of Section 6.2.

It will sometimes be advantageous to work with the discrete-time counterpart of (6.1) and (6.2) :

$$\mathbf{x}[k+1] = \mathbf{S}(\mathbf{W}.\mathbf{x}[k] - \mathbf{c}), \tag{6.4}$$

and

$$\boldsymbol{\xi}[k+1] = \mathbf{A}\mathbf{S}(\boldsymbol{\xi}[k]) + \mathbf{B}\mathbf{u} + \boldsymbol{\chi}. \tag{6.5}$$

Most of what can be said about (6.1) and (6.2) applies to (6.4) and (6.5) as well, but the latter models are of course easier to simulate numerically.

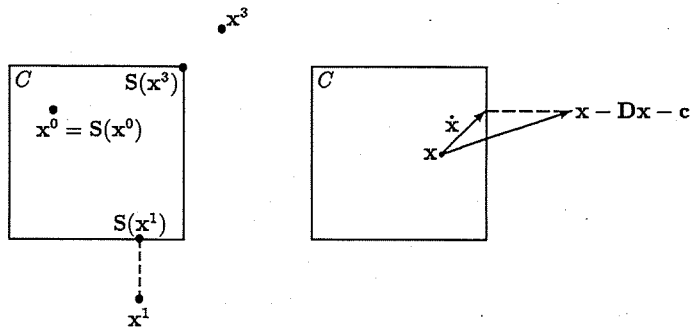


Figure 6.3: Geometrical interpretation of the mapping $\mathbf{S}(\cdot)$ as the projection on the cube C (left). Geometrical interpretation of the equation (6.6) (right).

6.2 Interpretation in the state space

The dynamical systems (6.1) and (6.4) have a simple geometrical meaning. For ease of notation, the matrix $\mathbf{W}$ is first split as $\mathbf{I} - \mathbf{D}$ where $\mathbf{I}$ is the $n \times n$ identity matrix. Then (6.1) and (6.4) become

$$d\mathbf{x}/dt = \mathbf{S}(\mathbf{x} - \mathbf{D}\mathbf{x} - \mathbf{c}) - \mathbf{x} \tag{6.6}$$

and

$$\mathbf{x}[k+1] = \mathbf{S}(\mathbf{x}[\mathbf{k}] - \mathbf{D}.\mathbf{x}[\mathbf{k}] - \mathbf{c}). \tag{6.7}$$

A geometrical interpretation of the effect of these updating rules can be formed by first noting that the mapping $\mathbf{S}(\cdot)$ performs a projection on the cube $C = \{\mathbf{x} \in R^n \mid -1 \leq x_k \leq 1, \ \ k = 1 \ldots n\}$ as shown in fig.6.3. With this in mind, equation (6.6) can be visualized as in fig.6.3 (with a similar interpretation for (6.7)). The equilibrium condition (6.3) or, equivalently,

$$\mathbf{x} = \mathbf{S}(\mathbf{x} - \mathbf{D}\mathbf{x} - \mathbf{c}), \tag{6.8}$$

is illustrated in fig.6.4. More formally, one has the following condition : *A point $\mathbf{x}^* \in C$ is an equilibrium point iff, for $k = 1 \ldots n$:*

$$\begin{aligned} x_k^* = 1 \ &\Rightarrow \ (\mathbf{D}\mathbf{x}^* + \mathbf{c})_k \leq 0 \\ -1 < x_k^* < 1 \ &\Rightarrow \ (\mathbf{D}\mathbf{x}^* + \mathbf{c})_k = 0 \\ x_k^* = -1 \ &\Rightarrow \ (\mathbf{D}\mathbf{x}^* + \mathbf{c})_k \geq 0. \end{aligned} \tag{6.9}$$

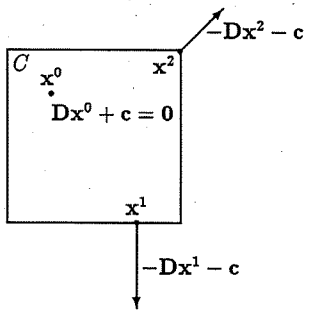


Figure 6.4: Points $\mathbf{x}^0$, $\mathbf{x}^1$ and $\mathbf{x}^2$ are equilibrium points of (6.6).

6.3 A path-following method for finding multiple equilibrium points.

6.3.1 An equivalent problem

Equations (6.9) can be seen as the description of a resistive n-port $\mathbf{y} = \mathbf{D}\mathbf{x}+\mathbf{c}$, terminated by n resistors with characteristic as depicted in fig. 6.5. Clearly, this problem has much in common with piecewise-linear circuits and the method studied here is closely related to the globally convergent methods developed for the analysis of diode-resistor circuits [90, 99]. Problem (6.9) is also known as a *linear complementarity problem with upper and lower bounds* and the method presented here is based on recent work in [122], with an extension for finding multiple solutions. As will become clear below, the algorithm is basically Wright's octahedral algorithm of Chapter 2 (complete with the reflection steps of Chapter 4), applied to the affine mapping $\mathbf{f}(\mathbf{x}) = \mathbf{D}\mathbf{x} + \mathbf{c}$. Contrary to the previous chapters however, the interest here is not in a solution to $\mathbf{D}\mathbf{x}+\mathbf{c} = \mathbf{0}$, but rather in the points where the homotopy path crosses the boundary of the cube C. Each such point corresponds to a solution of (6.9).

6.3.2 Basic principle of the algorithm.

Continuous deformation of the equations.

The idea in [122] is based on continuous deformation. Problem (6.9) is embedded in a one-parameter family of similar problems by replacing the char-

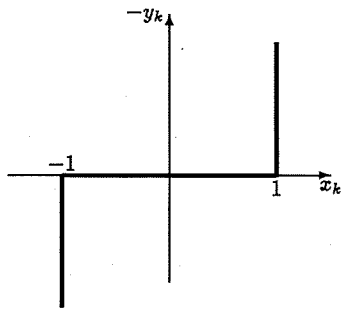


Figure 6.5: Constitutive relation of the resistors connected to the n-port $\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{c}$.

acteristics of fig. 6.5 and fig. 6.1 by the ones in fig. 6.6 in which $0 \leq t < \infty$. The algorithm starts with $t = 0$, $\mathbf{x} = \mathbf{0}$, and then follows a one-dimensional piecewise-linear curve by varying t. Each crossing of the path with the boundary of the cube C (i.e., each point on the path with $t = 1$) corresponds to a solution. Eventually the path goes to infinity or it gets stuck in a solution to $\mathbf{D}\mathbf{x} + \mathbf{c} = \mathbf{0}$. In the latter case, if the endpoint lies inside the cube C, it is a solution to (6.9) as well. In the former case, it will be shown below that the algorithm can still be continued provided the matrix $\mathbf{D}$ is nonsingular.

Recall the concept of *sign vector* which is needed for a formal description of the method : a sign vector $\mathbf{s}$ is a nonzero vector from $\{-1, 0, 1\}^n$, and indicates on which part of the three-segment characteristic of fig. 6.6 each pair (x_k, y_k) lies :

$$\begin{aligned} s_k = 1 \;&\Rightarrow\; x_k = t \text{ and } y_k \leq 0 \\ s_k = 0 \;&\Rightarrow\; -t \leq x_k \leq t \text{ and } y_k = 0 \\ s_k = -1 \;&\Rightarrow\; x_k = -t \text{ and } y_k \geq 0, \end{aligned} \tag{6.10}$$

(where $\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{c}$). The mechanism of the method will now be illustrated by two examples in R^2.

Example 1. We take

$$\mathbf{D} = \begin{bmatrix} 2 & 1 \\ -1/3 & 1/2 \end{bmatrix} \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

The path followed is shown in fig. 6.7 and consists of two segments.

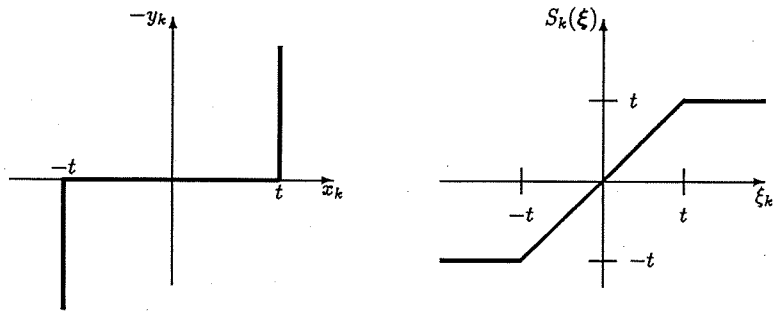


Figure 6.6: Modified constitutive relation and modified function S.

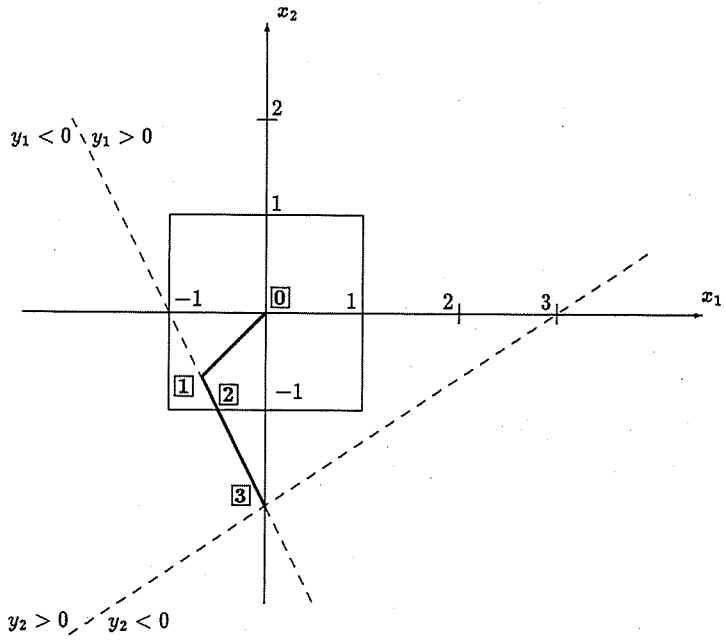


Figure 6.7: Example 1.

- On the first segment (0—1) the sign vector is $(-1,-1)$ because $y_1 > 0$ and $y_2 > 0$ in the origin. The path is described by

$$x_1 = x_2 = -t, \quad y_1 = -3t + 2, \quad y_2 = -\frac{1}{6}t + 1.$$

  This is only valid up to point 1 because y_1 and y_2 must remain positive if the sign vector is $(-1,-1)$.

- For the second segment (1—3) the sign vector becomes $(0,-1)$ and one has

$$x_2 = -t, \quad y_1 = 0, \quad y_2 = -\frac{1}{3}x_1 - \frac{1}{2}t + 1,$$

  which is valid as long as $-t \leq x_1 \leq t$ and $y_2 \geq 0$. In point 3 y_2 becomes zero, and the algorithm ends at this point since increasing t has no longer an effect on x.

- The segment 1—3 intersects the boundary of the unit cube in $(x_1, x_2) = (-0.5, -1)$ (point 2). As expected, this is a solution to (6.9) since $(y_1, y_2) = (0, 2/3)$. It can also be verified that this point is the only solution.

Example 2. A more complicated example is shown in fig.6.8 :

$$\mathbf{D} = \begin{bmatrix} -1 & 0.25 \\ 0 & -1 \end{bmatrix}, \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}.$$

The path now consists of eight segments.

- Since $y_1 > 0$ and $y_2 < 0$ in the origin, the algorithm is started with sign vector $(-1,1)$ and we have

$$x_1 = -t, \quad x_2 = t, \quad y_1 = 1.25t + 0.5, \quad y_2 = -t - 0.5.$$

  This segment is unbounded and crosses the boundary of C in the solution $(x_1, x_2) = (-1, 1)$ (point 1).

- The key observation needed to continue the algorithm is that if there is an unbounded segment with sign vector s, there can be expected to be one with sign vector $-$s as well. This needs to be proved in general, but it is clear in the two-dimensional case of fig. 6.8. Reversing s to s $= (1,-1)$, one can proceed by decreasing t along the path

$$x_1 = t, \quad x_2 = -t, \quad y_1 = -1.25t + 0.5, \quad y_2 = t - 0.5.$$

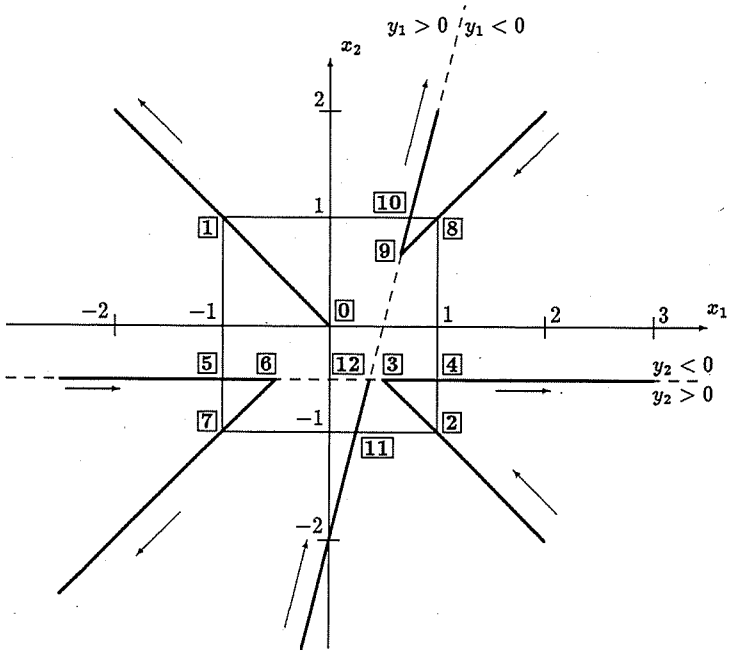


Figure 6.8: Example 2.

This satisfies the restrictions of (6.10) as long as $y_1 \leq 0$ and $y_2 \geq 0$, i.e., up to point $(x_1, x_2) = (0.5, -0.5)$ (point 3). In the meantime, a second solution has been found in $(x_1, x_2) = (1, -1)$ (point 2).

- In point 3 the sign vector switches to $\mathbf{s} = (1, 0)$, and the description of the path changes to

$$x_1 = t, \quad y_2 = 0, \quad y_1 = -t + 0.25x_2 - 0.5,$$

with the restriction that $-t \leq x_2 \leq t$. This segment is again unbounded.

- Continuing this procedure until a zero of $\mathbf{Dx}+\mathbf{c}$ (point 12) is reached, a total of nine solutions are found (points 1, 2, 4, 5, 7, 8, 10, 11 and 12). It can also be verified that the algorithm finds all solutions in this case.

Description of the path traced by the algorithm.

We now give a formal description of the algorithm. As can be seen from the examples in the preceding section, the algorithm traces a piecewise-linear path with each segment uniquely related to a particular sign vector. The defining equations for the segment corresponding to sign vector s can be expressed as follows. Define $\alpha = \{k \mid s_k = 0\}$ and $\beta = \{k \mid s_k \neq 0\}$. From fig. 6.5 and eqs. (6.10) it can be seen that $\mathbf{x}_\beta = \mathbf{s}_\beta t$, $\mathbf{y}_\alpha = \mathbf{0}$, and that the independent variables are $\mathbf{x}_\alpha$, $\mathbf{y}_\beta$ and t. These will be gathered in one single $(n+1)$-vector $\mathbf{z}$:

$$\mathbf{z}_\alpha = \mathbf{x}_\alpha, \quad \mathbf{z}_\beta = \mathbf{y}_\beta = (\mathbf{Dx}+\mathbf{c})_\beta, \quad z_{n+1} = t.$$

Also observe from fig. 6.5 that $t = z_{n+1}$ is in fact the infinity norm $\|\mathbf{x}\|_\infty$ of $\mathbf{x}$.

Define an $n \times (n+1)$-matrix $\mathbf{M}^{\mathbf{s}}$ as follows

$$\mathbf{M}^{\mathbf{s}}_{\cdot,\alpha} = \mathbf{D}_{\cdot,\alpha} \quad \mathbf{M}^{\mathbf{s}}_{\cdot,\beta} = -\mathbf{I}_{\cdot,\beta} \quad \mathbf{M}^{\mathbf{s}}_{\cdot,n+1} = \sum_{k\in\beta} \mathbf{D}_{\cdot,k} s_k = \mathbf{D}_{\cdot,\beta} \mathbf{s}_\beta. \tag{6.11}$$

With this notation the segment of the path corresponding to sign vector s is determined by all solutions to

$$\mathbf{M}^{\mathbf{s}}\mathbf{z} + \mathbf{c} = \mathbf{0}, \tag{6.12}$$

$$-z_{n+1} \leq z_k \leq z_{n+1} \quad \forall k \in \alpha, \tag{6.13}$$

$$s_k z_k \leq 0 \quad \forall k \in \beta, \tag{6.14}$$

$$z_{n+1} \geq 0. \tag{6.15}$$

If $\mathbf{M}^s$ is of rank n, the solution set to equations (6.12)–(6.15) must be empty or a one-dimensional segment (closed or semiclosed and unbounded). The case where $\mathbf{M}^s$ is not of full row rank, should be precluded by the use of a suitable perturbation scheme (see next section).

The basic step of the algorithm consists in moving from one breakpoint of the piecewise-linear path to the next. This is called pivoting. It is convenient to express each segment as a set of points $\mathbf{z} = \hat{\mathbf{z}} + \gamma\tilde{\mathbf{z}}$, $(0 \leq \gamma \leq \overline{\gamma})$ satisfying eqs. (6.12)-6.15). The vector $\hat{\mathbf{z}}$ is a breakpoint. It satisfies (6.12)-(6.15) and one of the inequalities of (6.13)-(6.15) with equalitiy (exactly one of the inequalities except in case of degeneracy). This vector $\tilde{\mathbf{z}}$ gives the direction of the segment and must be computed from $\mathbf{M}^s\tilde{\mathbf{z}} = \mathbf{0}$. In order to give $\tilde{\mathbf{z}}$ the right orientation, one considers the inequality in (6.13)-(6.15) that is satisfied with inequality by $\tilde{\mathbf{z}}$:

- if $-\hat{z}_{n+1} = \hat{z}_k$ for some $k \in \alpha$ then take $\tilde{z}_{n+1} + \tilde{z}_k > 0$,
- if $\hat{z}_{n+1} = \hat{z}_k$ for some $k \in \alpha$ then take $\tilde{z}_{n+1} - \tilde{z}_k > 0$,
- if $\hat{z}_k = 0$ for some $k \in \beta$ then take $s_k\tilde{z}_k < 0$,
- if $\hat{z}_{n+1} = 0$ then take $\tilde{z}_{n+1} > 0$.

The initial segment is given by $s_k = 1$ if $c_k < 0$, $s_k = -1$ if $c_k > 0$ (assuming $c_k \neq 0, \forall k$, see below), $\hat{\mathbf{z}} = [\hat{\mathbf{c}}^t\ 0]^t$, and $\tilde{\mathbf{z}} = [(\mathbf{D}\mathbf{s})^t\ 1]^t$.

It will now be shown how the pivoting operations can be carried out in each of the possible cases.

THE CASE OF A BOUNDED SEGMENT.

The value $\overline{\gamma}$ is determined by the first inequality in (6.13)–(6.14) to become invalid if γ increases (inequality (6.15) need not be considered since the path cannot return to the origin). This value can be obtained from the following minimum ratio test.

Let Γ be defined by

$$\begin{aligned}\Gamma &= \{k \mid k \in \alpha \text{ and } \tilde{z}_{n+1} - \tilde{z}_k < 0\} \\ &\cup \{-k \mid k \in \alpha \text{ and } \tilde{z}_{n+1} + \tilde{z}_k < 0\} \\ &\cup \{k \mid k \in \beta \text{ and } s_k\tilde{z}_k > 0\}. \end{aligned} \tag{6.16}$$

If this set Γ is empty, the segment is unbounded. This case will be treated below. For each element k of Γ, a ratio r_k can be computed as

$$r_k = -\frac{\hat{z}_{n+1} - \hat{z}_k}{\tilde{z}_{n+1} - \tilde{z}_k} \quad \text{if} \quad k \in \alpha,$$

$$r_k = -\frac{\hat{z}_{n+1} + \hat{z}_k}{\tilde{z}_{n+1} + \tilde{z}_k} \quad \text{if} \quad -k \in \alpha, \tag{6.17}$$

$$r_k = -\frac{\hat{z}_k}{\tilde{z}_k} \quad \text{if} \quad k \in \beta.$$

Assume these ratios have a unique minimum $\overline{\gamma} = r_j = \min_{k\in\Gamma} r_k$. Then the second breakpoint of the segment is given by $\hat{\mathbf{z}} + \overline{\gamma}\tilde{\mathbf{z}}$. From both breakpoints, it can easily be checked whether the segment intersects the boundary of C (and hence contains an equilibrium point).

The index j where the minimum occurs determines the next sign vector $\mathbf{s}$: only one component changes according to

$$\begin{aligned} j \in \beta &\Rightarrow s_j := 0 \\ j \in \alpha &\Rightarrow s_j := 1 \\ j \in \alpha &\Rightarrow s_j := -1. \end{aligned}$$

A treatment of ties in the minimum ratio test is relegated to the next section.

THE CASE OF AN UNBOUNDED SEGMENT.

If Γ defined in (6.16) is empty, the points $\hat{\mathbf{z}} + \gamma\tilde{\mathbf{z}}$ satisfy (6.12)–(6.15) for all $\gamma \geq 0$ and hence lie on an unbounded segment. From (6.16) one has that

$$\tilde{z}_{n+1} - \tilde{z}_k \geq 0, \quad \tilde{z}_{n+1} + \tilde{z}_k \geq 0, \ \forall k \in \alpha, \tag{6.18}$$

and

$$s_k\tilde{z}_k \leq 0 \ \ \forall k \in \beta. \tag{6.19}$$

Assume that none of these inequalities holds with equality. This is a second nondegeneracy assumption that will be relaxed in the next section. Also we know that $\tilde{z}_{n+1} > 0$, since $z_{n+1} = \|\mathbf{x}\|_\infty$ and the path is unbounded. Under these assumptions it can be verified that, provided γ is sufficiently negative, the points $\mathbf{z}$ defined by

$$z_k = \hat{z}_k + \gamma\tilde{z}_k, \quad k = 1 \ldots n,$$

and

$$z_{n+1} = -(\hat{z}_{n+1} + \gamma\tilde{z}_{n+1})$$

satisfy eqs. (6.12)-(6.15) if $\mathbf{s}$ is replaced by $-\mathbf{s}$. This is the reflection step from Example 2 above. The required value of γ is the minimum of the ratios

$$-\frac{\hat{z}_{n+1} - \hat{z}_k}{\tilde{z}_{n+1} - \tilde{z}_k} \quad \text{if} \quad k \in \alpha,$$

$$-\frac{\hat{z}_{n+1}+\hat{z}_k}{\tilde{z}_{n+1}+\tilde{z}_k} \quad \text{if} \quad -k \in \alpha,$$
$$-\frac{\hat{z}_k}{\tilde{z}_k} \quad \text{if} \quad k \in \beta.$$

For each unbounded segment one can easily check whether it contains an equilibrium point.

6.3.3 Degeneracy.

Above we have encountered two types of degeneracy. The first type, a tie in the minimum ratio test (6.17), could be resolved by subtracting an infinitesimal positive vector $[\epsilon\ \epsilon^2\ \ldots\ \epsilon^n]^t$ from $\mathbf{c}$. This would lead to standard lexicographical pivoting rules (see [92], and Chapter 3). The second type of degeneracy occurs when Γ in (6.16) is empty, but one of the inequalities (6.18) or (6.19) holds with equality. This makes it unclear whether a reflection must be applied or not, since the situation might as well be interpreted as a 'breakpoint at infinity'. Perturbing $\mathbf{c}$ does not help since the problem concerns the homogenous system $\mathbf{M}^s\tilde{\mathbf{z}} = \mathbf{0}$. An odd perturbation of the mapping $\mathbf{Dx}+\mathbf{c}$ would be required instead. The scheme presented below is a modification of the one in [96].

Two assumption will be made : *all components of the vector* $\mathbf{c}$ *are nonzero* and *the matrix* $\mathbf{D}$ *is nonsingular*. Neither of these assumptions is very restrictive. As for the first one, in most applications a small random number can be added to a zero component c_k without much harm. In the case of a singular $\mathbf{D}$, the algorithm can still be started, but it will end at a point that does not satisfy $\mathbf{Dx}+\mathbf{c}=\mathbf{0}$.

We will also use a slight modification of the algorithm, similarly to Chapter 4. Choose a number $R \geq 1$, and replace system (6.12)–(6.15) by the following two systems

$$\begin{aligned}
\mathbf{D}_{\alpha,\alpha}\mathbf{z}_\alpha + \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta z_{n+1} + \mathbf{c}_\alpha &= \mathbf{0},\\
\mathbf{D}_{\beta,\alpha}\mathbf{z}_\alpha - \mathbf{z}_\beta + \mathbf{D}_{\beta,\beta}\mathbf{s}_\beta z_{n+1} + \mathbf{c}_\beta &= \mathbf{0},\\
-z_{n+1} \leq z_k \leq z_{n+1}, \quad \forall k \in \alpha,&\\
s_k z_k \leq 0, \quad \forall k \in \beta,&\\
0 \leq z_{n+1} \leq R,&
\end{aligned} \tag{6.20}$$

and

$$\begin{aligned}
\mathbf{D}_{\alpha,\alpha}\mathbf{z}_\alpha + (\mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta - \mathbf{c}_\alpha)z_{n+1} + (R+1)\mathbf{c}_\alpha &= \mathbf{0},\\
\mathbf{D}_{\beta,\alpha}\mathbf{z}_\alpha - \mathbf{z}_\beta + (\mathbf{D}_{\beta,\beta}\mathbf{s}_\beta - \mathbf{c}_\beta)z_{n+1} + (R+1)\mathbf{c}_\beta &= \mathbf{0},
\end{aligned}$$

$$\begin{aligned}
-z_{n+1} \leq z_k \leq z_{n+1}, \quad \forall k \in \alpha,&\\
s_k z_k \leq 0, \quad \forall k \in \beta,&\\
R \leq z_{n+1} \leq R+1.&
\end{aligned} \tag{6.21}$$

The path defined by these equations is still piecewise-linear, and coincides with the original one as long as $z_{n+1} \leq R$. For points with $R \leq z_{n+1} \leq R+1$ the effect of switching to (6.21) is that of replacing $\mathbf{Dx}+\mathbf{c}$ by $\mathbf{Dx}+(R+1-t)\mathbf{c}$. This has the following consequences.

- The original algorithm terminates at $-\mathbf{D}^{-1}\mathbf{c}$. If this point has norm less than R, it will still be the endpoint of the modified path. If $\|\mathbf{D}^{-1}\mathbf{c}\|_\infty > R$, a new endpoint $\mathbf{x}^*$ will be created with $R \leq \|\mathbf{x}^*\|_\infty \leq R+1$, and satisfying $\mathbf{Dx}^* + (R+1-\|\mathbf{x}^*\|_\infty)\mathbf{c} = \mathbf{0}$ or, equivalently,

  $$\mathbf{x}^* = -(R+1-\|\mathbf{x}^*\|_\infty)\mathbf{D}^{-1}\mathbf{c}.$$

  This expression shows that $\mathbf{x}^*$ lies on a line connecting the origin and $-\mathbf{D}^{-1}\mathbf{c}$.

- From (6.21) it is clear that a reflection can be made if z_{n+1} attains the value $R+1$: one can make the changes $\mathbf{z}_\alpha \rightarrow -\mathbf{z}_\alpha$, $\mathbf{z}_\beta \rightarrow -\mathbf{z}_\beta$ and $\mathbf{s} \rightarrow -\mathbf{s}$ and continue the path.

- One is free to choose R, but the choice of a small R can reduce the number of solutions that are found. On the other hand, a small value can be effective in improving the efficiency by preventing the algorithm from being led too far astray outside C.

We now turn to the treatment of degeneracy. As described in the text, the algorithm generates a sequence of s-vectors for which the system (6.12)–(6.15) is solvable. To avoid degeneracy, one should add the requirement that the solutions to (6.12)–(6.15) be robust against small perturbations of the equations. The idea used here is to perturb the equations

$$x_k = s_k t \ \ \forall k \in \beta \quad \text{and} \quad -t \leq x_k \leq t \ \ \forall k \in \alpha,$$

which define fig. 6.5, as follows

$$x_k = s_k(t+\epsilon^k) \ \ \forall k \in \beta \quad \text{and} \quad -t-\epsilon^k \leq x_k \leq t+\epsilon^k \ \ \forall k \in \alpha,$$

for infinitesimal positive ϵ.

Systems (6.20) and (6.21) then change to

$$\begin{aligned}
&\mathbf{D}_{\alpha,\alpha}\mathbf{z}_\alpha(\epsilon) + \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta z_{n+1}(\epsilon) + \mathbf{c}_\alpha + \textstyle\sum_{k\in\beta}\mathbf{D}_{\alpha,k}s_k\epsilon^k = \mathbf{0},\\
&\mathbf{D}_{\beta,\alpha}\mathbf{z}_\alpha(\epsilon) - \mathbf{z}_\beta(\epsilon) + \mathbf{D}_{\beta,\beta}\mathbf{s}_\beta z_{n+1}(\epsilon) + \mathbf{c}_\beta + \textstyle\sum_{k\in\beta}\mathbf{D}_{\beta,k}s_k\epsilon^k = \mathbf{0},\\
&-z_{n+1}(\epsilon) - \epsilon^k \le z_k(\epsilon) \le z_{n+1}(\epsilon) + \epsilon^k, \quad \forall k \in \alpha,\\
&s_k z_k(\epsilon) \le 0, \quad \forall k \in \beta,\\
&0 \le z_{n+1}(\epsilon) \le R,
\end{aligned} \tag{6.22}$$

and

$$\begin{aligned}
&\mathbf{D}_{\alpha,\alpha}\mathbf{z}_\alpha(\epsilon) + (\mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta - \mathbf{c}_\alpha)z_{n+1}(\epsilon) + (R+1)\mathbf{c}_\alpha + \textstyle\sum_{k\in\beta}\mathbf{D}_{\alpha,k}s_k\epsilon^k = \mathbf{0},\\
&\mathbf{D}_{\beta,\alpha}\mathbf{z}_\alpha(\epsilon) - \mathbf{z}_\beta(\epsilon) + (\mathbf{D}_{\beta,\beta}\mathbf{s}_\beta - \mathbf{c}_\beta)z_{n+1}(\epsilon) + (R+1)\mathbf{c}_\beta + \textstyle\sum_{k\in\beta}\mathbf{D}_{\beta,k}s_k\epsilon^k = \mathbf{0},\\
&-z_{n+1}(\epsilon) - \epsilon^k \le z_k(\epsilon) \le z_{n+1}(\epsilon) + \epsilon^k, \quad \forall k \in \alpha,\\
&s_k z_k(\epsilon) \le 0, \quad \forall k \in \beta,\\
&R \le z_{n+1}(\epsilon) \le R+1.
\end{aligned} \tag{6.23}$$

The algorithm generates a sequence of s-vectors for which there exists an $\bar{\epsilon} > 0$ such that systems (6.22) and (6.23) are solvable for all $0 \le \epsilon \le \bar{\epsilon}$. For the sake of conciseness, the discussion will be limited to system (6.22), but everything is valid for (6.23) as well. The vector $\mathbf{z}(\epsilon)$ will be a polynomial in ϵ :

$$\mathbf{z}(\epsilon) = \mathbf{z}^0 + \sum_{j\in\beta}\mathbf{z}^{(j)}\epsilon^j,$$

where $\mathbf{z}^{(0)}$ satisfies

$$\mathbf{D}_{\alpha,\alpha}\mathbf{z}_\alpha^{(0)} + \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta z_{n+1}^{(0)} + \mathbf{c}_\alpha = \mathbf{0}, \tag{6.24}$$

$$\mathbf{D}_{\beta,\alpha}\mathbf{z}_\alpha^{(0)} - \mathbf{z}_\beta^{(0)} + \mathbf{D}_{\beta,\beta}\mathbf{s}_\beta z_{n+1}^{(0)} + \mathbf{c}_\beta = \mathbf{0}, \tag{6.24}$$

and $z^{(j)}, j \in \beta$, satisfies

$$\mathbf{D}_{\alpha,\alpha}\mathbf{z}_\alpha^{(j)} + \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta z_{n+1}^{(j)} + \mathbf{D}_{\alpha,j}s_j = \mathbf{0}, \tag{6.25}$$

$$\mathbf{D}_{\beta,\alpha}\mathbf{z}_\alpha^{(j)} - \mathbf{z}_\beta^{(j)} + \mathbf{D}_{\beta,\beta}\mathbf{s}_\beta z_{n+1}^{(j)} + \mathbf{D}_{\beta,j}s_j = \mathbf{0}. \tag{6.25}$$

A first effect of this perturbation scheme is that it ensures that the matrix $\mathbf{M}^{\mathbf{s}}$ in (6.11) will be of full row rank.

Theorem 21 *If (6.25) is solvable $\forall j \in \beta$, and $\mathbf{D}$ is nonsingular, then the $n \times (n+1)$-matrix $\mathbf{M}^{\mathbf{s}}$ defined by*

$$\mathbf{M}^{\mathbf{s}}_{\cdot,\alpha} = \mathbf{D}_{\cdot,\alpha} \quad \mathbf{M}^{\mathbf{s}}_{\cdot,\beta} = -\mathbf{I}_{\cdot,\beta} \quad \mathbf{M}^{\mathbf{s}}_{\cdot,n+1} = \mathbf{D}_{\cdot,\beta}\mathbf{s}_\beta,$$

has rank n.

Proof. If m is the number of elements in α, then it suffices to show that the matrix $[\mathbf{D}_{\alpha,\alpha} \;\; \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta]$ has rank m. Since by assumption the equations (6.25) are solvable for all $j \in \beta$, each vector $\mathbf{D}_{\alpha,j}, j \in \beta$, must lie in the column space of $[\mathbf{D}_{\alpha,\alpha} \;\; \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta]$ and hence

$$\operatorname{rank}[\mathbf{D}_{\alpha,\cdot} \;\; \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta] = \operatorname{rank}[\mathbf{D}_{\alpha,\alpha} \;\; \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta].$$

This rank cannot be less than m since this would imply that the rank of $\mathbf{D}_{\alpha,\cdot}$ is less than m and hence that $\mathbf{D}$ is singular. □

The perturbation scheme can be implemented as follows. First compute $\tilde{\mathbf{z}}$ from

$$\mathbf{D}_{\alpha,\alpha}\tilde{\mathbf{z}}_\alpha + \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta\tilde{z}_{n+1} = \mathbf{0},$$

$$\mathbf{D}_{\alpha,\alpha}\tilde{\mathbf{z}}_\alpha - \tilde{\mathbf{z}}_\beta + \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta\tilde{z}_{n+1} = \mathbf{0},$$

and give it the right orientation. Define Γ as

$$\begin{aligned}
\Gamma \;=\; &\{k \mid k \in \alpha \text{ and } \tilde{z}_{n+1} - \tilde{z}_k < 0\}\\
\cup\; &\{-k \mid k \in \alpha \text{ and } \tilde{z}_{n+1} + \tilde{z}_k < 0\}\\
\cup\; &\{k \mid k \in \beta \text{ and } s_k\tilde{z}_k > 0\}\\
\cup\; &\{n+1 \text{ if } \tilde{z}_{n+1} > 0\}.
\end{aligned}$$

Compute vectors $\mathbf{z}^{(0)}$ and $\mathbf{z}^{(j)}, j \in \beta$ that satisfy (6.24),(6.25) and, for each element k of Γ, define ratios $r_k^{(0)}$ and $r_k^{(j)}, j = 1 \ldots n$ as follows :

$$\begin{aligned}
r_k^{(0)} &= -\frac{z_{n+1}^{(0)} - z_k^{(0)}}{\tilde{z}_{n+1} - \tilde{z}_k} \text{ if } k \in \alpha,\\
r_k^{(0)} &= -\frac{z_{n+1}^{(0)} + z_k^{(0)}}{\tilde{z}_{n+1} + \tilde{z}_k} \text{ if } -k \in \alpha,\\
r_k^{(0)} &= -\frac{z_k^{(0)}}{\tilde{z}_k} \text{ if } k \in \beta,\\
r_k^{(0)} &= \frac{R - z_{n+1}^{(0)}}{\tilde{z}_{n+1}} \text{ if } k = n+1,
\end{aligned}$$

and, if $j \in \beta$,

$$\begin{aligned} r_k^{(j)} &= -\frac{z_{n+1}^{(j)} - z_k^{(j)}}{\tilde{z}_{n+1} - \tilde{z}_k} \text{ if } k \in \alpha, \\ r_k^{(j)} &= -\frac{z_{n+1}^{(j)} + z_k^{(j)}}{\tilde{z}_{n+1} + \tilde{z}_k} \text{ if } -k \in \alpha, \\ r_k^{(j)} &= -\frac{z_k^{(j)}}{\tilde{z}_k} \text{ if } k \in \beta, \\ r_k^{(j)} &= \frac{-z_{n+1}^{(j)}}{\tilde{z}_{n+1}} \text{ if } k = n+1, \end{aligned}$$

or, if $j \in \alpha$,

$$\begin{aligned} r_k^{(j)} &= 0 \text{ if } k \neq j, \\ r_k^{(j)} &= -\frac{1}{\tilde{z}_{n+1} - \tilde{z}_k} \text{ if } k = j, \\ r_k^{(j)} &= -\frac{1}{\tilde{z}_{n+1} + \tilde{z}_k} \text{ if } -k = j. \end{aligned}$$

Let $r_k(\epsilon) = r_k^{(0)} + \sum_{j=1}^{n} \epsilon^j r_k^{(j)}$. The minimum ratio test (6.17) can then be replaced by the calculation of the minimum of the polynomials $r_k(\epsilon)$ for small positive ϵ, i.e. calculation of the polynomial $r(\epsilon)$ for which

$$\exists \overline{\epsilon} > 0 \; s.t. \; \forall 0 \leq \epsilon \leq \overline{\epsilon} \; : \; r(\epsilon) = \min_{k \in \Gamma} r_k(\epsilon).$$

This minimum can be found lexicographically. It remains to be proved that the minimum is well defined i.e. that there can be no two elements $k, l \in \Gamma$ for which $r_k(\epsilon) = r_l(\epsilon) = r(\epsilon)$.

Theorem 22 *If* $\mathbf{D}$ *is nonsingular, then there can be no indices* k *and* l *for which* $r_k(\epsilon) = r_l(\epsilon) = r(\epsilon)$.

Proof. If $k \in \alpha$ or $-k \in \alpha$, this is clear from the definition of $r_k^{(j)}$. We prove the theorem for the case where $k, l \in \beta$. The remaining case $k \in \beta$, $l = n+1$, can be dealt with similarly. Define $\hat{\mathbf{z}}^{(j)} = \mathbf{z}^{(j)} + r^{(j)}\tilde{\mathbf{z}}$, then by assumption one must have $\hat{z}_k^{(j)} = \hat{z}_l^{(j)} = 0$, $\forall j$. It follows from (6.25) that $\forall j \in \beta$,

$$\begin{aligned} \mathbf{D}_{\alpha,\alpha}\hat{\mathbf{z}}_\alpha^{(j)} + \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta \hat{z}_{n+1}^{(j)} + \mathbf{D}_{\alpha,j}s_j &= 0, \\ \mathbf{D}_{k,\alpha}\hat{\mathbf{z}}_\alpha^{(j)} + \mathbf{D}_{k,\beta}\mathbf{s}_\beta \hat{z}_{n+1}^{(j)} + D_{k,j}s_j &= 0, \\ \mathbf{D}_{l,\alpha}\hat{\mathbf{z}}_\alpha^{(j)} + \mathbf{D}_{l,\beta}\mathbf{s}_\beta \hat{z}_{n+1}^{(j)} + D_{l,j}s_j &= 0. \end{aligned}$$

Hence

$$\text{rank} \begin{bmatrix} \mathbf{D}_{\alpha,\cdot} & \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta \\ \mathbf{D}_{k,\cdot} & \mathbf{D}_{k,\beta}\mathbf{s}_\beta \\ \mathbf{D}_{l,\cdot} & \mathbf{D}_{l,\beta}\mathbf{s}_\beta \end{bmatrix} = \text{rank} \begin{bmatrix} \mathbf{D}_{\alpha,\alpha} & \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta \\ \mathbf{D}_{k,\alpha} & \mathbf{D}_{k,\beta}\mathbf{s}_\beta \\ \mathbf{D}_{l,\alpha} & \mathbf{D}_{l,\beta}\mathbf{s}_\beta \end{bmatrix} \leq m+1$$

if m is the cardinality of α. This expression implies that

$$\text{rank}\mathbf{D}_{\alpha \cup \{k,l\},\cdot} \leq m+1$$

and this is impossible unless $\mathbf{D}$ is singular. □

6.3.4 Further properties.

From now on, it will be assumed that the degenerate situations described above do not occur. As shown in the previous section, this entails no loss of generality, as long as the matrix $\mathbf{D}$ is nonsingular, which will henceforth be assumed as well. A third assumption is that the path crosses the boundary of the cube C transversally, i.e., that there are no continua of equilibrium points.

To recapitulate the algorithm, it starts in the origin and generates a sequence of sign vectors $\mathbf{s}$. Each sign vector defines a linear segment by (6.10) or, equivalently, by (6.12)–(6.15). Two consecutive segments either have one endpoint in common or are unbounded and related by a reflection step. Eventually the algorithm must stop in the point $\mathbf{x} = -\mathbf{D}^{-1}\mathbf{c}$. Each time t takes the value 1, one has an equilibrium point of the neural network, and if the endpoint of the path lies in the interior of C, it is an equilibrium point as well. In general there is no guarantee that all equilibrium points can be found in this way, since besides the path generated by the algorithm, there may be cyclic sequences of segments that cannot be detected by the algorithm. Let $\mathcal{Z}(\mathbf{s})$ be the solution set to the equations (6.12-6.15) for a particular choice of $\mathbf{s}$ and let $\mathcal{Z} = \bigcup_{\mathbf{s} \neq 0} \mathcal{Z}(\mathbf{s})$. The purpose of this section is to describe the geometry of the set $\mathcal{Z}$.

As was done above, each segment $\mathcal{Z}(\mathbf{s})$ can be parametrized by its 'arclength' γ, i.e., it can be expressed as $\mathbf{z} = \hat{\mathbf{z}} + \gamma\tilde{\mathbf{z}}$, $0 \leq \gamma \leq \overline{\gamma}$, and $\tilde{\mathbf{z}}$ satisfies

$$\mathbf{D}_{\alpha,\alpha}\tilde{\mathbf{z}}_\alpha + \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta \tilde{z}_{n+1} = \mathbf{0},$$

$$\mathbf{D}_{\beta,\alpha}\tilde{\mathbf{z}}_\alpha - \tilde{\mathbf{z}}_\beta + \mathbf{D}_{\beta,\beta}\mathbf{s}_\beta \tilde{z}_{n+1} = \mathbf{0}.$$

These equations can be solved by Cramer's rule to give

$$\tilde{z}_k = -\delta \det \begin{pmatrix} \mathbf{D}_{\alpha\setminus k,\alpha\setminus k} & \mathbf{D}_{\alpha\setminus k,\beta}\mathbf{s}_\beta \\ \mathbf{D}_{k,\alpha\setminus k} & \mathbf{D}_{k,\beta}\mathbf{s}_\beta \end{pmatrix}, \quad k \in \alpha \qquad (6.26)$$

$$\tilde{z}_k = \delta \det \begin{pmatrix} \mathbf{D}_{\alpha,\alpha} & \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta \\ \mathbf{D}_{k,\alpha} & \mathbf{D}_{k,\beta}\mathbf{s}_\beta \end{pmatrix}, \quad k \in \beta \tag{6.27}$$

$$\tilde{z}_{n+1} = \delta \det \mathbf{D}_{\alpha,\alpha} \tag{6.28}$$

(where the determinant of an empty matrix is assumed to be +1). In these equations $\delta = \pm 1$ depending on the orientation in which the segment is traversed. Initially $\delta = 1$, since $\tilde{z}_{n+1}$ is increasing on the first segment while α is empty. It can be shown that δ remains the same on two connected segments if they are traversed in the same sense, and that δ is reversed with each reflection of the algorithm. This is stated by the following two theorems.

Theorem 23 *Let $\mathcal{Z}(\mathbf{s}^1)$ and $\mathcal{Z}(\mathbf{s}^2)$ be two segments that share one endpoint $\hat{\mathbf{z}}$ and let the corresponding vectors $\tilde{\mathbf{z}}^1$ and $\tilde{\mathbf{z}}^2$ be given by eqs. (6.26)-(6.28), with $\delta^1, \mathbf{s}^1, \alpha^1, \beta^1$ and $\delta^2, \mathbf{s}^2, \alpha^2, \beta^2$, resp. Then if both segments are traversed in the same sense, one has $\delta^1 = \delta^2$.*

Proof. The sign vectors $\mathbf{s}^1$ and $\mathbf{s}^2$ can differ in only one component, say component j. Assume $s_j^1 \neq 0$ and $s_j^2 = 0$. The other possibilites can be proved similarly. For ease of notation, we write $\alpha = \alpha^1$, $\beta = \beta^1$, $\mathbf{s} = \mathbf{s}^1$, $\alpha^2 = \alpha \cup j$ and $\beta^2 = \beta \setminus j$.

By (6.27) one knows that

$$s_j \tilde{z}_j^1 = s_j \delta^1 \det \begin{pmatrix} \mathbf{D}_{\alpha,\alpha} & \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta \\ \mathbf{D}_{j,\alpha} & \mathbf{D}_{j,\beta}\mathbf{s}_\beta \end{pmatrix},$$

and hence by (6.26) and (6.28)

$$\begin{aligned} s_j \tilde{z}_j^1 &= s_j \delta^1 \det \begin{pmatrix} \mathbf{D}_{\alpha,\alpha} & \mathbf{D}_{\alpha,\beta\setminus j}\mathbf{s}_{\beta\setminus j} \\ \mathbf{D}_{j,\alpha} & \mathbf{D}_{j,\beta\setminus j}\mathbf{s}_{\beta\setminus j} \end{pmatrix} + \delta^1 \det(\mathbf{D}_{\alpha\cup j,\alpha\cup j}) \\ &= (\delta^1/\delta^2)(\tilde{z}_{n+1}^2 - s_j \tilde{z}_j^2). \end{aligned}$$

Furthermore one knows that either $s_j \tilde{z}_j^1 > 0$ and $\tilde{z}_{n+1}^2 - s_j \tilde{z}_j^2 > 0$, or $s_j \tilde{z}_j^1 < 0$ and $\tilde{z}_{n+1}^2 - s_j \tilde{z}_j^2 < 0$ if both segments are traversed in the same sense. From this $\delta^1 = \delta^2$ follows. □

Theorem 24 *Let $\mathcal{Z}(\mathbf{s}^1)$ and $\mathcal{Z}(\mathbf{s}^2)$ be two unbounded segments with opposite $\mathbf{s}$ vectors $\mathbf{s}^1 = -\mathbf{s}^2$, and let the corresponding vectors $\tilde{\mathbf{z}}^1$ and $\tilde{\mathbf{z}}^2$ be given by eqs. (6.26)-(6.28), with $\delta^1, \mathbf{s}^1, \alpha^1, \beta^1$ and $\delta^2, \mathbf{s}^2, \alpha^2, \beta^2$, resp. If one segment is traversed in the direction of increasing t, and the other in the direction of decreasing t, then one has $\delta^1 = -\delta^2$.*

Proof. This follows immediately from (6.28) and from $\alpha^1 = \alpha^2$. □

In analogy with the definition of index of solutions of a set of nonlinear equations [73], one can introduce an index for each equilibrium point of the neural network.

Definition 14 *If $\mathbf{x}$ satisfies (6.9) with corresponding sign vector $\mathbf{s}$, i.e.*

$$x_k = s_k \quad \text{and} \quad s_k(\mathbf{D}\mathbf{x} + \mathbf{c})_k \leq 0 \quad \forall s_k \neq 0,$$

$$-1 \leq x_k \leq 1 \quad \text{and} \quad (\mathbf{D}\mathbf{x} + \mathbf{c})_k = 0 \quad \forall s_k = 0,$$

then the index of $\mathbf{x}$ is defined as $\operatorname{sign} \det \mathbf{D}_{\alpha,\alpha}$ where $\alpha = \{k | s_k = 0\}$. If the point $\mathbf{x} = -\mathbf{D}^{-1}\mathbf{c}$ is an equilibrium point, its index is defined as $\operatorname{sign} \det \mathbf{D}$.

The following observations follow from Theorems 23 and 24.

1. The number of solutions found by the algorithm is odd and the algorithm finds at least one solution with index +1.

2. Assume that the algorithm finds one solution $\mathbf{z}^1$, and after r reflections finds a second solution $\mathbf{z}^2$. Then the indices of $\mathbf{z}^1$ and $\mathbf{z}^2$ are opposite if r is even, and they are equal if r is odd.

From (6.28) it follows that the path behaves monotonically if $\det \mathbf{D}_{\alpha,\alpha} > 0$ for all $\alpha \subset \{1, \ldots, n\}$. In this simple case it can be proved that all solutions are found by the algorithm. It should be stressed, however, that in general the parameter t need not increase or decrease monotonically, and therefore the condition below is expected to be much too strong.

Theorem 25 *If $\det \mathbf{D}_{\alpha,\alpha} > 0, \quad \forall \alpha \subset \{1 \ldots n\}$ ($\det \mathbf{D} > 0$ is not required), then the network has either one or three solutions and all of these are found by the algorithm. If, in addition $\det \mathbf{D} > 0$, the network has a unique equilibrium.*

Proof. Since $\det \mathbf{D}_{\alpha,\alpha} > 0 \ \forall \alpha$, the sign of $\tilde{z}_{n+1}$ can only change because of a reflection (change of δ in (6.28)). Hence there can be no closed loops of connected segments and the path traced by the algorithm is the only component of $\mathcal{Z}$. In addition there can be at most one reflection. A reflection will occur if $\det \mathbf{D} < 0$. This can be seen from (6.27) : on the final segment of the path, $\mathbf{s}$ has only one nonzero component, say j, and $s_j \tilde{z}_j > 0$. Then with (6.27) one has $s_j \tilde{z}_j = \delta \det \mathbf{D} > 0$. If the algorithm ends before a reflection occurs, one has $\delta = +1$ and hence $\det \mathbf{D} > 0$. If it ends after one reflection one must have $\det \mathbf{D} < 0$. In this case of negative determinant of $\mathbf{D}$, there can be one or three equilibrium points, depending on the value of $\mathbf{c}$. In the case of positive $\det \mathbf{D}$ the number of solutions is always exactly one. □

Example. A simple 4-dimensional example of a **D**-matrix with positive principal minors is

$$\mathbf{D} = \begin{pmatrix} d_{11} & 0 & 0 & d_{14} \\ d_{21} & d_{22} & 0 & 0 \\ 0 & d_{32} & d_{33} & 0 \\ 0 & 0 & d_{43} & d_{44} \end{pmatrix},$$

with $d_{ii} > 0, \quad i = 1 \ldots 4$. Here $\det \mathbf{D} = d_{11}d_{22}d_{33}d_{44} - d_{21}d_{32}d_{43}d_{14}$.

The theorem states that a network where **D** has positive principal minors (except $\det \mathbf{D}$ itself) can have one or three equilibrium points, depending on the value of **c**. If in such a network one neuron is deleted (i.e. one row and one column of **D** are deleted), the equilibrium point becomes structurally unique. In this sense, the situation corresponds to the minimal structures known to be needed in nonlinear electrical circuits to have multiple equilibrium points [68, 34].

6.3.5 Numerical aspects.

A look at the algorithm shows that the computational load per pivot is mainly determined by the calculation of a vector $\tilde{\mathbf{z}}$ satisfying $\mathbf{M}^s\tilde{\mathbf{z}} = \mathbf{0}$ with $\mathbf{M}^s$ given by (6.11). Because of the structure of $\mathbf{M}^s$, this reduces to solving

$$\begin{bmatrix} \mathbf{D}_{\alpha,\alpha} & \mathbf{D}_{\alpha,\beta}\mathbf{s}_\beta \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{z}}_\alpha \\ \tilde{z}_{n+1} \end{bmatrix} = \mathbf{0}$$

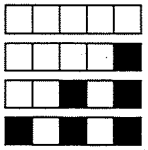
and then computing $\tilde{\mathbf{z}}_\beta$ from

$$\tilde{\mathbf{z}}_\beta = \mathbf{D}_{\beta,\alpha}\tilde{\mathbf{z}}_\alpha + \mathbf{D}_{\beta,\beta}\mathbf{s}_\beta\tilde{z}_{n+1}.$$

In a typical implementation, one would therefore store $\mathbf{D}_{\alpha,\alpha}$ under factored form (LU or QR), and update the factorization at each pivot. This updating consists in adding one row and one column to $\mathbf{D}_{\alpha,\alpha}$ (when a new element is added to α), or deleting one row and one column (when an element is deleted from α) and can be executed in $O(m^2)$ floating point operations if m is the cardinality of α [39]. Since **D** is usually very sparse, the efficiency can be further increased by using sparse-matrix techniques. This subject is well documented in the literature on large-scale linear programming.

Not much can be said about the required number of pivots, except that it is 3^n in a worst case, and probably much less on the average. The problem is reminiscent of the gap between worst-case and average-case complexity of the simplex method for linear programming and of a number of related algorithms [8, 124].

EXAMPLE 1.
We first take the connected-component detector of the introduction with $n = 5$. The algorithm applied to this problem terminates in 84 pivots with 23 solutions. Only four of these lie on the corners of the hypercube C :



This corresponds to the final states that can be generated under proper operation of the algorithm. The remaining equilibrium points have at least one component with absolute value smaller than one and are therefore unstable [18].

EXAMPLE 2.
In the hole-filling network of [61], the matrix **A** is defined by the template

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix},$$

$B = 4$ (a scalar), and $\chi_k = -1$, $\forall k$. We take a 10×10 net with input shown in fig. 6.9i. The network is designed to fill up the holes in the input image, starting from a completely black image as initial state. Applied to this problem the algorithm finds the 29 equilibrium points shown in fig. 6.9 ii)–xxx). These were all found during the first 10000 pivots. After 300000 pivots the program was interrupted. The equilibrium shown in fig. 6.9iii is the one the network is intended to reach and it can be proved that it will do so if started from a completely black initial state [61]. It is also clear that there must be more than 29 equilibrium points, because of the symmetry in the image.

6.4 Stability of neural networks.

The analysis of dynamical neural models such as (6.1) is usually limited to symmetric matrices **W** [43, 42] because in that case an energy function can easily be defined and used to study the convergence. This section contains results for the asymmetric case.

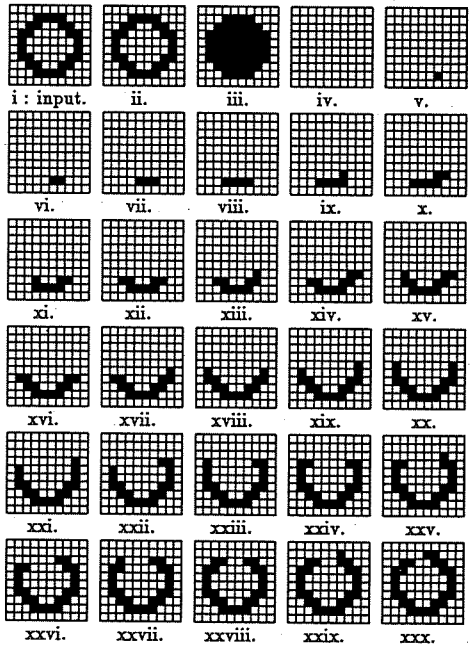


Figure 6.9: i) Input to the hole-filling neural network. ii)-xxx) Equilibrium points found by the algorithm (7784 pivots).

6.4.1 The equilibria as stationary points of a variational inequality

Equation (6.8) (the equilibrium condition for the network) has already been interpreted in various ways. Still another interpretation is as a so-called *variational inequality* : $\mathbf{x}^*$ is an equilibrium point if and only if

$$(\mathbf{y} - \mathbf{x}^*)^t.(\mathbf{D}\mathbf{x}^* + \mathbf{c}) \geq 0 \quad \forall \mathbf{y} \in C. \tag{6.29}$$

This type of problem is well known in mathematical programming as a unified description for a large class of problems [24, 75]. As a matter of fact, (6.29) holds componentwise :

$$(y_k - x_k^*).(\mathbf{D}\mathbf{x}^* + \mathbf{c})_k \geq 0 \quad \forall \mathbf{y} \in C, \text{ and } k = 1, \ldots, n. \tag{6.30}$$

In a similar way, the dynamic equation (6.1) can be written as the inequality

$$(\mathbf{y} - \mathbf{x} - \dot{\mathbf{x}})^t.(\mathbf{D}\mathbf{x} + \mathbf{c} + \dot{\mathbf{x}}) \geq 0, \quad \forall \mathbf{y} \in C. \tag{6.31}$$

It is interesting to note this link with variational inequalities for two reasons. First, as shown in the next section, one can use it in order to generalize the interpretation of equilibrium points as local minima of an energy function to the case where $\mathbf{W}$ is asymmetric. Secondly, interesting results exist on the convergence of iterative algorithms for solving this type of variational inequalities. Some of these results carry over to neural networks, because the neural dynamics can be seen as a continuous-time version of one of these iterative schemes.

6.4.2 Neural networks and equilibrium programming

The symmetric case

In the symmetric case an energy function for (6.1) can be defined as $E(\mathbf{x}) = \frac{1}{2}\mathbf{x}^t\mathbf{D}\mathbf{x} + \mathbf{x}^t\mathbf{c}$, then it follows from (6.31) with $\mathbf{y} = \mathbf{x}$ that

$$\begin{aligned} dE/dt &= \dot{\mathbf{x}}^t(\mathbf{D}\mathbf{x} + \mathbf{c}) \\ &\leq -\dot{\mathbf{x}}^t\dot{\mathbf{x}} \\ &\leq 0 \end{aligned}$$

with equality only when $\dot{\mathbf{x}} = \mathbf{0}$, so asymptotic stability is readily established if the matrix $\mathbf{W}$ is symmetric.

The asymmetric case

The concept of equilibrium programming was introduced by Zangwill and Garcia [133], as a general model of equilibrium and subsumes several traditional problems in economy, game theory, network and transportation theory. Applied to the neural model considered here, their reasoning would be as follows. In the case of symmetric coefficients one single energy function is defined and all neurons act in a coordinated way in order to meet a global objective, i.e. the minimization of the total energy. Modelled as an equilibrium programming problem, each neuron would have a *different* objective function. More specifically define the energy of the i-th neuron as

$$E_i(\mathbf{x}) = \frac{1}{2}d_{ii}x_i^2 + \sum_{j\neq i} d_{ij}x_i x_j + c_i.x_i$$

and assume that at each time instant neuron i takes the outputs $x_j, j \neq i$ of the other neurons as given and chooses to adapt its own output x_i so as to minimize its energy within the constraint $-1 \leq x_i \leq 1$. Then equilibrium $\mathbf{x}^*$ corresponds to a situation where each x_i^* solves

$$\min_{-1\leq x_i\leq 1} E_i(x_1^*, \ldots, x_{i-1}^*, x_i, x_{i+1}^*, \ldots x_n^*).$$

In this way one arrives at a system of n coupled optimization problems instead of one single. The Karush-Kuhn-Tucker (KKT) conditions for these n problems and the variational inequality (6.29) are equivalent. This shows the importance of the variational inequality formulation as a generalization of the KKT conditions to the multiple objective equilibrium programming model.

6.4.3 Stability of networks with asymmetric coefficients

Above it has already been noted, as a result of Theorem 25, that system (6.6) has a unique equilibrium state for all possible values of $\mathbf{c}$ if $\mathbf{D}$ has positive principal minors, i.e. if $\mathbf{D}$ is a P-matrix. We now proceed to describe a proper subclass of P for which the neural network is globally convergent for all values of $\mathbf{c}$. These results are adapted from Ahn [1, 2] and Pang [74, 76] who studied the convergence of iterative algorithms for variational inequalities. The following definitions will be needed [7, 77].

Definition 15 *A matrix $\mathbf{A}$ is a nonsingular M-matrix if it can be written in the form $\mathbf{A} = s\mathbf{I} - \mathbf{B}$ where s is a positive scalar, and $\mathbf{B}$ is a nonnegative matrix with spectral radius strictly smaller than s.*

Properties of nonsingular M-matrices. Let $\mathbf{A}$ be a nonsingular M-matrix. Then,

1. There exists a positive diagonal matrix $\mathbf{R}$ such that $\mathbf{RA}+\mathbf{A}^t\mathbf{R}$ is positive definite.

2. $\mathbf{A}$ belongs to the class P.

Definition 16 *A square matrix $\mathbf{A}$ is quasi-dominant if there exists a vector $\mathbf{r} > \mathbf{0}$ such that*

$$r_i A_{ii} > \sum_{j\neq i} r_j |A_{ij}|.$$

Definition 17 *The comparison matrix $\tilde{\mathbf{A}}$ of a matrix $\mathbf{A}$ is defined as $\tilde{A}_{ii} = A_{ii}$ and $\tilde{A}_{ij} = -|A_{ij}|$, for $i \neq j$.*

Lemma. [67] *A matrix $\mathbf{A}$ is quasidominant if and only if its comparison matrix $\tilde{\mathbf{A}}$ is a nonsingular M-matrix.*

Theorem 26 *The neural network converges from every initial state to the unique equilibrium if the matrix $\mathbf{D}$ is quasidominant with diagonal elements $d_{ii} \leq 1, i = 1 \ldots n$.*

Proof. It will be shown that the vector $|\dot{\mathbf{x}}| \to 0$ as $t \to \infty$. As $\mathbf{x}$ stays within a compact set, this implies that the system goes to an equilibrium. The proof proceeds in two steps. First the inequality

$$\frac{d}{dt}|\dot{\mathbf{x}}| \leq -\tilde{\mathbf{D}}|\dot{\mathbf{x}}|$$

is derived. In a second part it is proved that this inequality implies

$$|\dot{\mathbf{x}}| \to 0$$

if $\tilde{\mathbf{D}}$ is a nonsingular M-matrix.

1. Consider the dynamic equation (6.1) at time instants t and $t + dt$:

$$\begin{aligned} \mathbf{x}(t) + \dot{\mathbf{x}}(t) &= \mathbf{S}(\mathbf{W}\mathbf{x}(t) - \mathbf{c}) \\ \mathbf{x}(t) + \dot{\mathbf{x}}(t).dt + \dot{\mathbf{x}}(t+dt) &= \mathbf{S}(\mathbf{W}\mathbf{x}(t) + \mathbf{W}\dot{\mathbf{x}}dt - \mathbf{c}) \end{aligned}$$

Subtraction of these two equations yields :

$$\begin{aligned}
|\dot{\mathbf{x}}(t) - \dot{\mathbf{x}}(t)dt - \dot{\mathbf{x}}(t+dt)| &= |\mathbf{S}(\mathbf{W}\mathbf{x}(t) - \mathbf{c}) - \mathbf{S}(\mathbf{W}\mathbf{x}(t) + \mathbf{W}\dot{\mathbf{x}}dt - \mathbf{c})| \\
&\leq |\mathbf{W}\dot{\mathbf{x}}|dt \\
&\leq |\mathbf{W}||\dot{\mathbf{x}}|dt \\
|\dot{\mathbf{x}}(t+dt)| - |\dot{\mathbf{x}}(t)|(1-dt) &\leq |\mathbf{W}||\dot{\mathbf{x}}|dt \\
|\dot{\mathbf{x}}(t+dt)| - |\dot{\mathbf{x}}(t)| &\leq -(\mathbf{I} - |\mathbf{W}|)|\dot{\mathbf{x}}(t)|dt \\
\frac{d}{dt}|\dot{\mathbf{x}}| \leq -(\mathbf{I} - |\mathbf{W}|)|\dot{\mathbf{x}}| &= -\tilde{\mathbf{D}}|\dot{\mathbf{x}}|.
\end{aligned}$$

2. Now, if $\tilde{\mathbf{D}}$ is a nonsingular M-matrix, a positive diagonal matrix $\mathbf{R}$ exists such that $\tilde{\mathbf{D}}^t\mathbf{R} + \mathbf{R}\tilde{\mathbf{D}}$ is positive definite. Define $\mathbf{z} = |\dot{\mathbf{x}}|$ then we will show that

$$\frac{d}{dt}\mathbf{z} \leq -\tilde{\mathbf{D}}.\mathbf{z}$$

with $\tilde{\mathbf{D}}$ a nonsingular M-matrix and $\mathbf{z}(t) \geq \mathbf{0} \quad \forall t$ implies that $\mathbf{z} \to \mathbf{0}$. Hereto, we define the Lyapunov function

$$V(\mathbf{z}) = \frac{1}{2}\mathbf{z}^t.\mathbf{R}.\mathbf{z}$$

which is convex and has a unique minimum at $\mathbf{z} = \mathbf{0}$. Then

$$\begin{aligned}
\frac{d}{dt}V &= (\frac{d}{dt}\mathbf{z})^t.\mathbf{R}.\mathbf{z} \\
&\leq -\mathbf{z}^t.\tilde{\mathbf{D}}^t.\mathbf{R}.\mathbf{z}
\end{aligned}$$

as $\mathbf{z}(t) \geq \mathbf{0}$ for all t. From the positive definiteness of $\tilde{\mathbf{D}}^t.\mathbf{R} + \mathbf{R}.\tilde{\mathbf{D}}$ it follows that

$$\frac{d}{dt}V \leq 0.$$

with equality only when $\mathbf{z} = \mathbf{0}$.

□

6.5 An iterative method for neural network design

6.5.1 Relaxation methods for linear inequalities

One of the central problem in the use of neural networks for pattern recognition, image and signal processing is the development of training and learning

algorithms. In the cellular neural networks (6.2) for example, the templates are usually found by trial and error, or from the analogy with digital filters or cellular automata.

The purpose of this section is to present a more systematic and general method. The basic idea is to translate the design specifications into linear inequalities with the network coefficients as unknowns and then to apply an iterative method for solving these sets of linear inequalities. Let us therefore start with a brief account of the *relaxation method* for solving linear inequalities.

The method was first proposed by Agmon, Motzkin and Schoenberg (see [10]) as an iterative algorithm for solving

$$\mathbf{a}_i^t.\mathbf{x} \geq b_i, \quad i = 1, \ldots, m, \tag{6.32}$$

in n unknowns $\mathbf{x} \in R^n$. Starting from an arbitrarily chosen $\mathbf{x}^0$, an approximate solution $\mathbf{x}^k$ will be obtained after the $(k-1)$-th iteration. Assume that $\mathbf{x}^k$ violates some constraint $a_j.\mathbf{x}^k \geq b_j$ of the system (6.32). Then a correction will be made by taking

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \lambda.\frac{\mathbf{a}_j^t\mathbf{x}^k - b_j}{\mathbf{a}_j^t\mathbf{a}_j}.\mathbf{a}_j, \quad 0 < \lambda \leq 2. \tag{6.33}$$

For $\lambda = 1$, this step projects $\mathbf{x}^k$ orthogonally on the hyperplane $\mathbf{a}_j^t\mathbf{x} = b_j$. For $\lambda < 1$, one has underrelaxation, for $\lambda > 1$, overrelaxation. Several variants exist, according to which inequality is selected from the m inequalities (6.32). One possibility is to process the inequalities in a cyclical order.

The convergence of this method has been extensively studied [40, 60]. Interesting results from this analysis are :

- Finite convergence. It was shown that, if $\lambda \in (0,2)$, the relaxation method produces a sequence of iterates that is either finite or converges to a solution on the boundary of the solution polyhedron to (6.32).

- Rate of convergence. Convergence is geometric with a rate depending on a condition number defined for the system (6.32).

- Influence of overrelaxation. A condition number is introduced measuring the *obtuseness* of the solution polyhedron. If the polyhedron is full dimensional, convergence is finite for any $\lambda \in [1,2]$ if the polyhedron is obtuse, and for any $\lambda \in (\lambda^*, 2]$ if it is not obtuse, where $\lambda^* > 1$ depends on the condition number.

In the following sections, the cyclical version of this basic algorithm will be applied to the training of cellular neural networks, although the method is more generally valid (for instance for Hopfield networks and brain-state-in-a-box networks [110, 105, 108, 109]). The space-invariance of the coefficients in cellular neural networks however reduces the complexity of the problem considerably (see also [134] for a similar application of the relaxation algorithm).

6.5.2 Application to cellular neural networks

EXAMPLE 1. The connected component detector has already been presented in the introduction to this chapter. If the template used for the construction of the matrix **A** is denoted as

α_{-1}	α_0	α_1

then (6.2) and (6.5) reduce to

$$d\xi_i/dt = -\xi_i + \alpha_{-1}S(\xi_{i-1}) + \alpha_0 S(\xi_i) + \alpha_1 S(\xi_{i+1}), \tag{6.34}$$

or

$$\xi_i[k+1] = \alpha_{-1}S(\xi_{i-1}[k]) + \alpha_0 S(\xi_i[k]) + \alpha_1 S(\xi_{i+1}[k]), \tag{6.35}$$

for $i = 1, \ldots, n$ (assuming $\xi_0 = \xi_{n+1} = -1$). The discussion will be limited to the discrete-time case of (6.35). The specifications for the network are summarized in fig 6.10.[2] Each of these specificatons can be converted into a linear inequality with three unknowns α_{-1}, α_0, and α_1 :

$$\begin{bmatrix} S(\xi_{i-1}[k]) & S(\xi_i[k]) & S(\xi_{i+1}[k] \end{bmatrix} \begin{bmatrix} \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \end{bmatrix} \geq 1, \quad \text{if } S(\xi_i[k+1]) = +1,$$

$$\begin{bmatrix} S(\xi_{i-1}[k]) & S(\xi_i[k]) & S(\xi_{i+1}[k] \end{bmatrix} \begin{bmatrix} \alpha_{-1} \\ \alpha_0 \\ \alpha_1 \end{bmatrix} \leq -1, \quad \text{if } S(\xi_i[k+1]) = -1.$$

In addition, one can add the requirement that $\alpha_0 > 1$. This will ensure that all stable equilibrium points satisfy $|\xi_i| \geq 1$, for $i = 1, \ldots, n$ [18]. This requirement plus the table of fig. 6.10 leads to a system of nine inequalities. The relaxation algorithm with $\lambda = 2$ and starting from $\alpha_{-1} = \alpha_0 = \alpha_1 = 0$ converges after two cycles (i.e., 18 updates of the coefficients). The resulting template is $\alpha_{-1} = 2$, $\alpha_0 = 1.9$, $\alpha_1 = -1.3$. Fig 6.11 illustrates the operation of the network

[2] In the continuous-time case, one specifies $\dot{\xi}_i$ instead of $\xi_i[k+1]$. This also leads to a set of linear inequalities.

$S(\xi_{i-1}[k])$, $S(\xi_i[k])$, $S(\xi_{i+1}[k])$	$\xi_i[k+1]$
□□□	≤ -1
□□■	≤ -1
□■□	$\geq +1$
□■■	≤ -1
■□□	≥ 1
■□■	≤ -1
■■□	$\geq +1$
■■■	$\geq +1$

Figure 6.10: Specifications for the connected component detector.

with this template. The template functions correctly, although the numerical values differ consideralbly from those given in Section 6.1. In fact one has a polyhedron of possible templates that satisfy the above inequalities. This is an important observation from the viewpoint of analog implementation : by choosing the coefficients close to the center of the polyhedron one can maximize the insensitivity of the network against parameter deviations.

EXAMPLE 2. As a second application, one can try to reduce the number of superfluous equilibrium points of the hole-filling network of Section 6.3.5. Owing to the symmetry of the problem, one has only two unknowns in the template for **A** :

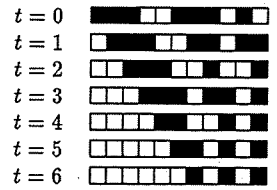


Figure 6.11: Operation of the connected-component detector.

0	β	0
β	α	β
0	β	0

The equations (6.2) and (6.5) then become ($\xi_{i,j}$ denotes the state of the cell in row i and column j of the array) :

$$d\xi_{i,j}/dt = -\xi_{i,j} + \alpha S(\xi_{i,j}) + \beta(S(\xi_{i-1,j}) + S(\xi_{i,j-1}) + S(\xi_{i+1,j}) + S(\xi_{i,j+1})) + \gamma u_{i,j} + \delta, \qquad (6.36)$$

or

$$\xi_{i,j}[k+1] = \alpha S(\xi_{i,j}[k]) + \beta(S(\xi_{i-1,j}[k]) + S(\xi_{i,j-1}[k]) + S(\xi_{i+1,j}[k]) + S(\xi_{i,j+1}[k])) + \gamma u_{i,j} + \delta, \qquad (6.37)$$

for $i = 1, \ldots, n$, $j = 1, \ldots, n$, and assuming that $\xi_{0,j} = \xi_{n+1,j} = \xi_{i,0} = \xi_{i,n+1} = -1$. We try to find values for α, β, γ and δ that satisfy the inequalities of fig. 6.12. These specifications are constructed according to the following rule :

- if $u_{i,j} = +1$, then $S(\xi_{i,j}[k+1]) = +1$,
- if $u_{i,j} = -1$ and all surrounding pixels are black ($S(\xi_{i-1,j}) = S(\xi_{i,j-1}) = S(\xi_{i+1,j}) = S(\xi_{i,j+1}) = +1$), then $S(\xi_{i,j}[k+1]) = +1$,
- if $u_{i,j} = -1$ and at least one of the surrounding pixels is white ($S(\xi_{i-1,j}) = -1$, or $S(\xi_{i,j-1}) = -1$, or $S(\xi_{i+1,j}) = -1$, or $S(\xi_{i,j+1}) = -1$), then $S(\xi_{i,j}[k+1]) = -1$.,

Together with the requirement $\alpha \geq 1$ this leads to 25 inequalities. The relaxation algorithm converges after 33 cycles to the values

$$\alpha = 1.009, \quad \beta = 2.237, \quad \gamma = 8.867, \quad \delta = 2.109.$$

Figure 6.13 shows the behaviour of this network. One can also calculate equilibrium states with the algorithm of Section 6.3. The result is shown in fig 6.14, and indicates that the number of equilibrium points is much lower than for the template given in Section 6.3.5.

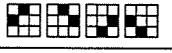
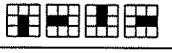
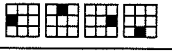
$S(\xi_{j-1,k-1}[t])$ $S(\xi_{j-1,k}[t])$ $S(\xi_{j-1,k+1}[t])$ $S(\xi_{j,k-1}[t])$ $S(\xi_{j,k}[t])$ $S(\xi_{j,k+1}[t])$ $S(\xi_{j+1,k-1}[t])$ $S(\xi_{j+1,k}[t])$ $S(\xi_{j+1,k+1}[t])$	$\xi_{j,k}[t+1]$ if $u_{j,k} = +1$	$\xi_{j,k}[t+1]$ if $u_{j,k} = -1$
	$\geq +1$	$\geq +1$
	$\geq +1$	$\geq +1$
	$\geq +1$	≤ -1
	$\geq +1$	≤ -1
	$\geq +1$	≤ -1
	$\geq +1$	≤ -1
	$\geq +1$	≤ -1
	$\geq +1$	≤ -1
	$\geq +1$	≤ -1
	$\geq +1$	≤ -1
	$\geq +1$	≤ -1
	$\geq +1$	≤ -1

Figure 6.12: Requirements for the hole-filler. The values of $S(\xi_{i-1,j-1})$, $S(\xi_{i+1,j-1})$, $S(\xi_{i-1,j+1})$ and $S(\xi_{i+1,j+1})$ are immaterial.

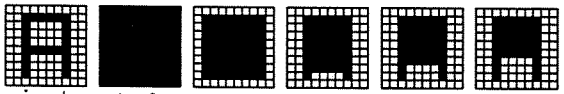


Figure 6.13: Example of the operation of the hole-filling template.

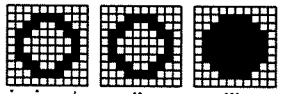


Figure 6.14: Equilibrium states of the hole-filler (found in 34540 pivots).

6.6 Conclusions

In this chapter we considered three applications of linear complementarity and linear inequalities to neural networks. Most of the results of this chapter can be traced down to the observation of the equivalence between variational inequalities (more especially the linear complementarity problem with upper bounds) and the equilibrium conditions for a neural network. From this it follows for instance that one can use the algorithm of [122] for neural networks. This is done in the first part of this chapter, but with a new extension to find multiple solutions. The main difficulty in formulating this extension was the necessity to develop a new perturbation scheme (Section 6.3.3). The algorithm for the computation of equilibrium points can be useful in the analysis of neural networks, for example during the design of templates for cellular neural networks. It can be applied to large networks, provided the extreme sparsity and the structure of the coefficients are exploited. In general there does not seem to be much hope for an efficient way to find all equilibrium points in a given neural network, unless appropriate guidelines are followed during the synthesis process. Theorem 25 might be helpful in this respect, although further work on this matter is certainly desirable. The properties pertaining to the number of solutions (Section 6.3.4) are apparently new, and demonstrate that the qualitative results from nonlinear circuit theory can often be transferred to neural networks.

In Section 6.4 neural networks are interpreted as analog systems for solv-

ing quadratic programming or more general variational inequalities. If the network has symmetric coefficients, convergence is always guaranteed, and the network can be seen as an analog machine for solving quadratic programs. In the presence of asymmetry however, oscillation and chaotic behaviour become possible, and convergence is much harder to prove. The application of convergence results for iterative methods for variational inequalities has led us to a new theorem (Theorem 26). This theorem gives a sufficient condition for convergence in networks with asymmetric coefficients.

In Section 6.5 a method was described to find templates of cellular neural networks. The method is based on the relaxation method for linear inequalities. It is very easy to implement and it is flexible because it allows to incorporate more general specifications (transitions, bounds on the coefficients, ...), than just imposing equilibrium points. This application of relaxation algorithms for finding the coefficients in neural networks was first given in [105], and has also been mentioned independently in [134].

Chapter 7

Harmonic balance and variable dimension algorithms

7.1 Introduction

Harmonic balance is a frequency-domain technique for steady-state analysis of nonlinear circuits. There exists a vast literature on the subject and the reader is referred to [57, 80, 58] for an extensive discussion and a historical overview. Here we shall restrict our attention to a brief description of the basic principles and consider the ordinary differential equation

$$\dot{\mathbf{x}} + \mathbf{g}(\mathbf{x}, \mathbf{u}) = \mathbf{0}, \tag{7.1}$$

where $\mathbf{g}(\cdot) : R^{M \times N} \rightarrow R^N$ is a continuous mapping, $\mathbf{u}(\cdot) : R \rightarrow R^M$ is a periodic waveform with period T, and $\mathbf{x}(\cdot) : R \rightarrow R^N$ is the unknown periodic steady-state solution. Equation (7.1) should be regarded as the description of a nonlinear nonautonomous circuit with $\mathbf{u}(\cdot)$ as independent sources. The unknown waveform $\mathbf{x}(\cdot)$ will be expressed as a Fourier series

$$\mathbf{x}(t) = \mathbf{X}(0) + \sum_{k=1}^{\infty} \left[\mathbf{X}^C(k) \cos k\omega t + \mathbf{X}^S(k) \sin k\omega t \right],$$

with $\omega = 2\pi/T$. In a similar way, the waveform $\mathbf{y}(\cdot) = \dot{\mathbf{x}}(\cdot) + \mathbf{g}(\mathbf{x}(\cdot), \mathbf{u}(\cdot))$ will be represented in the frequency domain as

$$\mathbf{y}(t) = \mathbf{Y}(0) + \sum_{k=1}^{\infty} \left[\mathbf{Y}^C(k) \cos k\omega t + \mathbf{Y}^S(k) \sin k\omega t \right].$$

These series will of course have to be truncated after a sufficient number of terms, say after the K-th harmonic. The variables in the harmonic balance

analysis can then be assembled in real $(2K+1)N$-vectors

$$\mathbf{X}^t = \left[\mathbf{X}(0)^t \ \mathbf{X}^C(1)^t \ \mathbf{X}^S(1)^t \ \cdots \ \mathbf{X}^C(K)^t \ \mathbf{X}^S(K)^t \right],$$

and

$$\mathbf{Y}(\mathbf{X})^t = \left[\mathbf{Y}(\mathbf{X},0)^t \ \mathbf{Y}^C(\mathbf{X},1)^t \ \mathbf{Y}^S(\mathbf{X},1)^t \ \cdots \mathbf{Y}^C(\mathbf{X},K)^t \ \mathbf{Y}^S(\mathbf{X},K)^t \right],$$

where the dependence of $\mathbf{Y}$ on $\mathbf{X}$ has been denoted explicitly. As a result, the problem of finding the steady-state solution of (7.1) is converted into one of solving the set of $(2K+1).n$ algebraic equations

$$\mathbf{Y}(\mathbf{X}) = \mathbf{0} \tag{7.2}$$

in the $(2K+1).N$ unknowns $\mathbf{X}$.

Since $\mathbf{g}(\cdot)$ is an arbitrary nonlinear function, $\mathbf{Y}(\mathbf{X})$ is not available under analytical form, but instead will be computed from $\mathbf{X}$ as follows.

1. Transform $\mathbf{X}$ back to the time domain, to obtain a sampled waveform $x(k.T/S), k = 0, \ldots, S-1$. Here S is the sample rate, $S \geq 2K-1$. This operation calls for an application of the inverse discrete Fourier transform.

2. From this the samples

$$\mathbf{g}(\mathbf{x}(kT/S), \mathbf{u}(kT/S)), \quad k = 0, \ldots, S-1,$$

can be computed and transformed to the frequency domain by applying the discrete Fourier transform. The resulting Fourier coefficients will be denoted as $\mathbf{G}(\mathbf{X})$.

3. The Fourier coefficients $\mathbf{Y}(\mathbf{X})$ can now be obtained as

$$\begin{aligned} \mathbf{Y}(\mathbf{X},0) &= \mathbf{G}(\mathbf{X},0) \\ \mathbf{Y}^C(\mathbf{X},k) &= k.\omega.\mathbf{X}^S(k) + \mathbf{G}^C(\mathbf{X},k), \quad k = 1, \ldots, K \\ \mathbf{Y}^S(\mathbf{X},k) &= -k.\omega.\mathbf{X}^C(k) + \mathbf{G}^S(\mathbf{X},k), \quad k = 1, \ldots, K. \end{aligned}$$

The set of equations (7.2) can be solved by a variety of algorithms [58] : Newton's method, relaxation methods, optimization methods and continuous deformation algorithms. The latter category is the one of interest to us here. Source-stepping is perhaps the best known representative of this class. It was discussed in connection with the D.C.-operating point problem in Chapter 1, but it has also been applied to harmonic balance [47, 80, 58]. The idea is

to parametrize the independent sources $\mathbf{u}(\cdot)$ as $\gamma.\mathbf{u}(\cdot)$ with $0 \leq \gamma \leq 1$, and to trace the solution from the trivial solution with $\gamma = 0$ onwards, to the actual solution with $\gamma = 1$. The subject of this chapter is a variable dimension algorithm that could be used for the same purpose. One can cite the following reasons why variable dimension methods should be of interest for harmonic balance as well.

- Source-stepping methods overcome convergence problems as occur in Newton's method, and require no initial guess of the solution. Instead of this they have the important disadvantage that the algorithm can only start at the origin. In general the user will have at least some estimate of the solution, and the path traced by the source-stepping algorithm will be unnecessarily long. Variable dimension algorithms can start at any point and are therefore more efficient.

- One problem shared by all harmonic balance methods is the high dimension of the equations (7.2). This will be felt even more strongly with continuous deformation methods because tracing a long path through a high-dimensional space can be computationally very expensive. Variable dimension algorithms will alleviate this difficulty : they operate by recursively solving a series of lower dimensional problems, and this greatly increases the efficiency. For instance, the algorithm presented in the next section solves the harmonic balance equations of increasing orders consecutively, starting with the D.C. equations and ending with the full problem (7.2).

7.2 A variable dimension algorithm for the harmonic balance equations

The algorithm that will be used here is a combination of the 2-ray algorithm from [130, 131, 83] (see also Section 2.3) and the octahedral algorithm. These methods were explained in Chapter 2 : the only difference between both is that in the 2-ray algorithm a different parameter t_k is used for each component $k = 1, \ldots, n$ (see fig. 2.8). As a consequence, the 2-ray algorithm solves the n equations in the natural order, beginning with the first one and ending with the n-th. The order in which the equations are solved by the octahedral algorithm is not fixed and depends on the problem at hand. Owing to this greater flexibility the octahedral algorithm is known to be faster for general-purpose applications because the path can be expected to be shorter (compare figs. 1.5 and 2.10). For this reason, the octahedral algorithm will usually be

preferred, except where the structure in the equations lends itself better to the 2-ray algorithm. In harmonic balance equations (7.2) one can expect the last equations to be less important and to provide a mere refinement of the solution. It would therefore be natural to solve the equations in the given order and to use the 2-ray algorithm rather than the octahedral algorithm.

For the time being, it will be assumed that the problem is given under the form $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ with $\mathbf{f}(\cdot)$ a mapping from R^n to R^n. The origin is taken as starting point. If a better initial guess is known, one can translate the origin.

Let $m_0, m_1, \ldots m_l$ be $l+1$ integers with $0 = m_0 < m_1 < m_2 \ldots < m_l = n$. We will now look at the effect of taking

$$\begin{aligned} t_1 &= \ldots = t_{m_1}, \\ t_{m_1+1} &= \ldots = t_{m_2}, \\ &\ldots \\ t_{m_{l-1}+1} &= \ldots = t_n, \end{aligned}$$

in fig. 2.8 and of imposing that at each stage of the algorithm there must be an index $r \in \{1, \ldots, l\}$ s.t.

$$\begin{aligned} f_j(\mathbf{x}) = 0 \quad & 1 \leq j \leq m_{r-1}, \\ t_j = 0 \quad & m_r + 1 \leq j \leq n. \end{aligned}$$

A number of definitions, similar to the ones in Section 2.2, will be needed. Let $r \in \{1, \ldots, l\}$ and let $\mathbf{s}$ be a nonzero vector from $\{-1, 0, 1\}^{m_r}$ with $s_j = 0$, $j \leq m_{r-1}$. Define

- $X(r, \mathbf{s}) = \{\mathbf{x} \in R^n \mid x_j = 0 \text{ if } j > m_r;\ x_j = s_j.\max_{m_{r-1}<i\leq m_r} |x_i| \text{ if } j \leq m_r \text{ and } s_j \neq 0\}$,

- $Y(r, \mathbf{s}) = \{\mathbf{y} \in R^n \mid y_j = 0 \text{ if } s_j = 0;\ s_j y_j \leq 0 \text{ if } j \leq m_r \text{ and } s_j \neq 0\}$,

- $Z(r, \mathbf{s}) = X(r, \mathbf{s}) \times Y(r, \mathbf{s})$.

Let $\mathcal{Z}$ be the union of all sets $Z(r, \mathbf{s})$,

$$\mathcal{Z} = \bigcup Z(r, \mathbf{s}),$$

and

$$\mathcal{F} = \{(\mathbf{x}, \mathbf{y}) \mid \mathbf{y} = \mathbf{f}(\mathbf{x})\}.$$

The algorithm traces part of the one-dimensional manifold $\mathcal{F} \cap \mathcal{Z}$ through the $2n$-dimensional space of $(\mathbf{x}, \mathbf{y})$-pairs.

Start

Let $\mathbf{x}^* := \mathbf{0}$, $\mathbf{y}^* := \mathbf{f}(\mathbf{0})$, $r := 1$ and for $j = 1, \ldots, m_1$, $s_j := 1$ if $y_j^* \leq 0$ and $s_j := -1$ if $y_j^* > 0$. Then $(\mathbf{x}^*, \mathbf{y}^*)$ lies on an n-face of $Z(r, \mathbf{s})$ and belongs to $\mathcal{Z} \cap \mathcal{F}$.

Repeat

Trace $\mathcal{Z} \cap \mathcal{F}$ through $Z(r, \mathbf{s})$ from $(\mathbf{x}^*, \mathbf{y}^*)$ onwards until another point $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ on the boundary of $Z(r, \mathbf{s})$ is reached. Inside $Z(r, \mathbf{s})$, $\mathcal{F} \cap \mathcal{Z}$ is determined by

(i) A set of n equations $\mathbf{f}(\mathbf{x}) - \mathbf{y} = \mathbf{0}$. In these equations one has $2n$ variables $(\mathbf{x}, \mathbf{y})$ with the additional restrictions

- $y_j = 0$ if $s_j = 0$ and $j \leq m_r$,
- $x_j = s_j . \max_{m_{r-1} < i \leq m_r} |x_i|$ if $s_j \neq 0$ and $j \leq m_r$,
- $x_j = 0$ if $j > m_r$.

This leaves $(n+1)$ independent variables

- y_j for those $j \leq m_r$ with $s_j \neq 0$,
- y_j for $j > m_r$,
- x_j for those $j \leq m_r$ with $s_j = 0$,
- $t = \max_{m_{r-1} < i \leq m_r} |x_i|$.

The n equations define a one-dimensional path that can be followed numerically.

(ii) A set of inequalities (with $t = \max_{m_{r-1} < i \leq m_r} |x_i|$) :

- $x_j \leq t$ if $s_j = 0$ and $m_{r-1} < j \leq m_r$,
- $-t \leq x_j$ if $s_j = 0$ and $m_{r-1} < j \leq m_r$,
- $s_j y_j \leq 0$ if $s_j \neq 0$ and $m_{r-1} < j \leq m_r$,
- $t \geq 0$.

At the point $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ where $\mathcal{F} \cap \mathcal{Z}$ hits an n-face of $Z(r, \mathbf{s})$, exactly one of the constraints in (ii) holds with equality. This determines the next piece of $\mathcal{Z}$ to be entered :

- if $\tilde{x}_j = t$ and $s_j = 0$, for some $m_{r-1} < j \leq m_r$, then set $s_j := 1$.

- if $\tilde{x}_j = -t$ and $s_j = 0$, for some $m_{r-1} < j \leq m_r$, then set $s_j := -1$.
- if $\tilde{y}_j = 0$, and $s_j \neq 0$, for some $m_{r-1} < j \leq m_r$ then set $s_j := 0$.
  If now $s_i = 0$ for all i, $m_{r-1} < i \leq m_r$, and $r < l$ then take $r := r + 1$, and for $j = m_{r-1}, \ldots, m_r$, augment the vector s to an m_r-vector by taking $s_j := 1$ if $\tilde{y}_j \leq 0$, and $s_j := -1$ if $\tilde{y}_j > 0$ (the first m_{r-1} components of s remain unchanged).
- if $t = 0$ (this case is only possible when $r > 1$ and $s_j \neq 0$ for $m_{r-1} < j \leq m_r$), then take $r := r - 1$, find the index $j \in \{m_{r-1} + 1, \ldots, m_r\}$ for which $|x_j| = \max_{m_{r-1} < i \leq m_r} |x_j|$, and for this j take $s_j :=$ sign x_j, and reduce s to an m_r-vector by deleting the last components.

Take $\mathbf{x}^* := \tilde{\mathbf{x}}$ and $\mathbf{y}^* := \tilde{\mathbf{y}}$.

until $\mathbf{s} = \mathbf{0}$
Stop : $\mathbf{y}^* = \mathbf{f}(\mathbf{x}^*) = \mathbf{0}$.

In this description it is assumed that no problems with degeneracy occur. The algorithm should therefore be completed with a lexicographical perturbation scheme. The algorithm can be carried out numerically using the simplicial techniques of Chapter 3. The triangulation that should be used for this algorithm is explained in Appendix B.

The octahedral algorithm and the 2-ray algorithm can be obtained as special cases by taking

$$l = 1, \quad m_1 = n,$$

for the octahedral algorithm, and

$$l = n, \quad m_k = k, \; k = 1 \ldots n,$$

for the 2-ray algorithm.

For the harmonic balance equations (7.2) one could take $l = K$, and

$$m_1 = 3N, \quad m_2 = 5n, \quad \ldots, \; m_K = 2(K+1)N.$$

The algorithm then first searches for a solution to the first $3N$ equations (D.C. component plus fundamental frequency) and then refines the solution by taking the higher harmonics into account. With this (heuristic choice) the algorithm should be faster than the octahedral algorithm.

7.3 Convergence conditions

A general condition for convergence of the above algorithm when applied to the problem $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ is that $\mathbf{f}(\cdot)$ must satisfy

$$\exists R > 0 \text{ s.t. } \|\mathbf{x}\| > R \Rightarrow \max_k x_k f_k(\mathbf{x}) > 0.$$

A special case occurs if

$$\exists R > 0 \text{ s.t. } \|\mathbf{x}\| > R \Rightarrow \mathbf{x}^t \mathbf{f}(\mathbf{x}) > 0. \tag{7.3}$$

This guarantees that the homotopy path will be bounded and terminates at a solution (see Section 2.2.5).

The following theorem gives an application of this condition to (7.2).

Theorem 27 *Let $\mathbf{g}(\cdot) : R^N \times R^M \to R^N$ be a continuous mapping with the property that*

$$\lim_{\|\mathbf{x}\| \to \infty} \frac{\mathbf{x}^t \mathbf{g}(\mathbf{x}, \mathbf{u})}{\|\mathbf{x}\|} = \infty,$$

for all bounded $\mathbf{u}$.
Let $\mathbf{x}(\cdot) : R \to R^N$ be a periodic function

$$\mathbf{x}(t) = \mathbf{X}(0) + \sum_{k=1}^{K} \left[\mathbf{X}^C(k) \cos k\omega t + \mathbf{X}^S(k) \sin k\omega t \right],$$

and write

$$\mathbf{X}^t = \left[\mathbf{X}(0)^t \;\; \mathbf{X}^C(1)^t \;\; \mathbf{X}^S(1)^t \;\; \cdots \;\; \mathbf{X}^C(K)^t \;\; \mathbf{X}^S(K)^t \right].$$

Let $\mathbf{u}(\cdot) : R \to R^M$ be a continuous function with period T, and define $\mathbf{y}(\cdot)$ by

$$\mathbf{y} = \dot{\mathbf{x}} + \mathbf{g}(\mathbf{x}, \mathbf{u}). \tag{7.4}$$

The function $\mathbf{y}(\cdot)$ can be written as a Fourier series

$$\mathbf{y}(t) = \mathbf{Y}(0) + \sum_{k=1}^{\infty} \left[\mathbf{Y}^C(k) \cos k\omega t + \mathbf{Y}^S(k) \sin k\omega t \right].$$

Denote the first Fourier coefficients of $\mathbf{y}(\cdot)$ by

$$\mathbf{Y}(\mathbf{X})^t = \left[\mathbf{Y}(\mathbf{X}, 0)^t \;\; \mathbf{Y}^C(\mathbf{X}, 1)^t \;\; \mathbf{Y}^S(\mathbf{X}, 1)^t \;\; \cdots \;\; \mathbf{Y}^C(\mathbf{X}, K)^t \;\; \mathbf{Y}^S(\mathbf{X}, K)^t \right].$$

Then

$$\lim_{\|\mathbf{X}\| \to \infty} \mathbf{X}^t \mathbf{Y}(\mathbf{X}) > 0. \tag{7.5}$$

Proof. Equation (7.5) will follow from Parseval's theorem if one can prove that

$$\int_0^T \|\mathbf{x}(\tau)\|^2 d\tau \to \infty \Longrightarrow \int_0^T \mathbf{x}(\tau)^t (\dot{\mathbf{x}}(\tau) + \mathbf{g}(\mathbf{x}(\tau), \mathbf{u}(\tau))) d\tau > 0. \tag{7.6}$$

This can be shown in three steps

- If $\int_0^T \|\mathbf{x}(\tau)\|^2 d\tau \to \infty$, then

$$\int_0^T \|\mathbf{x}(\tau)\| d\tau \to \infty.$$

- If $\lim_{\|\mathbf{x}\| \to \infty} \mathbf{x}^t \mathbf{g}(\mathbf{x}) / \|\mathbf{x}\| = \infty$, then

$$\int_0^T \|\mathbf{x}(\tau)\| d\tau \to \infty \Longrightarrow \int_0^T \mathbf{x}(\tau)^t \mathbf{g}(\mathbf{x}(\tau), \mathbf{u}(\tau)) d\tau \to \infty.$$

- Since $\int_0^T \mathbf{x}(\tau)^t \dot{\mathbf{x}}(\tau) d\tau = 0$,

$$\int_0^T \mathbf{x}(\tau)^t (\dot{\mathbf{x}}(\tau) + \mathbf{g}(\mathbf{x}(\tau), \mathbf{u}(\tau))) d\tau = \int_0^T \mathbf{x}(\tau)^t (\mathbf{g}(\mathbf{x}(\tau), \mathbf{u}(\tau))) d\tau.$$

□

Strictly speaking, the theorem only proves existence of a solution to the harmonic balance equations (7.2) of any order. An additional reasoning would allow to conclude from this that (7.1) has a steady-state solution with period T [65]. As a theorem of existence of a periodic steady-state solution of (7.1) this result corresponds to a theorem from [14].

Example. The circuit in fig. 7.1 can be described by

$$\begin{aligned} \dot{x}_1 + g(x_1) + x_1 - x_2 - 10 \sin t &= 0 \\ \dot{x}_2 + g(x_2 + 10 \sin t) + x_2 - x_1 + 10 \sin t &= 0 \end{aligned}$$

with $g(x) = 4x^3 - x + 1$. The analysis of Section 7.1 with $K = 13$ leads to a system of 50 equations. The algorithm was started at the origin with uniform grid size 0.5. The first cycle converges in 618 pivots. The algorithm is then restarted with a smaller grid size. This second cycle converges in 516 pivots (Table 7.1). The solution is shown in fig 7.2. For comparison, the octahedral algorithm on the same set of equations required 9769 pivots to find the solution. This confirms the effectiveness of the strategy in Section 7.2.

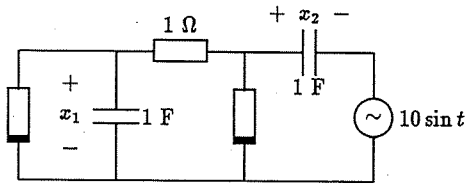


Figure 7.1: Example with two voltage-controlled resistors $i = g(v)$, where $g(x) = 4x^3 - x + 1$.

	1st cycle	2nd cycle
$k = 1$	135	625
$k = 2$	141	642
$k = 3$	159	654
$k = 4$	173	690
$k = 5$	217	749
$k = 6$	240	791
$k = 7$	288	850
$k = 8$	327	910
$k = 9$	401	999
$k = 10$	470	1061
$k = 11$	543	1098
$k = 12$	618	1134
# qN.-steps	1	13
$\|\mathbf{Y}(\mathbf{X})\|$	1.31E+00	2.88E−09

Table 7.1: Example 1. Number of pivots needed to solve the equations of order $k = 0, \ldots, 12$, number of quasi-Newton steps and accuracy obtained in each cycle of the algorithm.

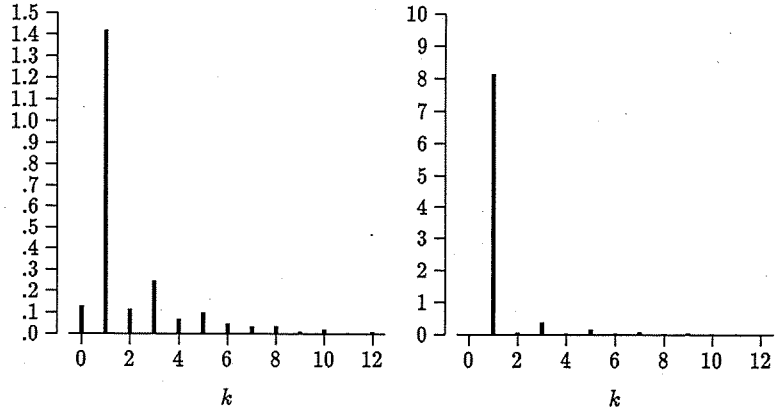


Figure 7.2: Example 1. Spectrum of x_1 (left) and x_2 (right).

7.4 Conclusion

Source-stepping algorithms for harmonic balance have been mentioned at several places in the literature. It is therefore interesting to consider variable dimension algorithms for the same purpose, because of their restart capability, and their great efficiency as compared with comparable methods. In this chapter a new variable dimension algorithm is presented that exploits the structure inherent in the harmonic balance equations.

This chapter should more or less be considered as a detailed suggestion for further research, since several aspects of the algorithm, its implementation as well as its convergence properties, are left open.

Chapter 8

Conclusions

In this thesis recent developments in nonlinear circuit theory and mathematical programming were brought together in order to find improved methods for the analysis of nonlinear circuits. Both theory and practice benefit from such a combined approach.

- Variable dimension algorithms are introduced here as globally convergent methods for DC-analysis of nonlinear circuits. We have provided a detailed study of the convergence and the implementation of the octahedral algorithm and have demonstrated its superiority over existing methods on several points.
  - Variable dimension algorithms can be started at every point, and their convergence depends only on the global, asymptotic properties of the equations. This is an important advantage when compared with Katzenelson's algorithm or source-stepping.
  - As was demonstrated from numerical examples, they are also inherently faster than Katzenelson's method, owing to the variable dimension feature.
  - Owing to the restart property, variable dimension algorithms can be run on refining simplicial subdivisions. Moreover they can be combined smoothly with conventional iterative methods. This leads to a method with the asymptotic speed of an iterative method and the immunity of a homotopy method against a bad choice of initial point or singularity of the Jacobian matrix.
  - We have also shown how to exploit separable and linear structure in the circuit equations.
  - The convergence criteria for the octahedral algorithm have been elaborated in detail, and have been formulated in topological terms.

    Moreover the presence of ideal operational amplifiers and controlled sources has been taken into account, and an extension has been given to process a broad class of active circuits.

  Sparse matrix techniques have not yet been incorporated. The algorithm basically involves standard operations on a label matrix which inherits the sparse structure of the Jacobian matrix. The sparsity of the label matrix can therefore be exploited in much the same way as the sparsity of the Jacobian matrix in the Newton-Raphson method.

- An essential step towards the derivation of convergence criteria was the simple circuit-theoretical interpretation given in Chapter 2. This made it possible to use the circuit topology in a straightforward way. At the same time, the convergence proofs lead to a unified and constructive derivation of the existence theorems known in the qualitative theory of nonlinear circuits. From these existence theorems, criteria for uniqueness of solutions can easily be established.

  The criteria in Chapter 2 essentially rely on eventual passivity of the components and on the conservation of energy. This raises the question of the applicability to active circuits. In chapter 4 the octahedral algorithm is generalized to process a large class of circuits containing operational amplifiers and controlled sources. Again the link was made with the qualitative and topological analysis of these circuits and several important new results were given.

  The necessity of exploiting the topological structure of the circuit was one the recurring themes in the derivation of convergence criteria, as well as in the description of an efficient implementation. An interesting open problem is whether it is possible to still further exploit this topological structure. The algorithm in chapter 4 for example is capable of finding multiple solutions but cannot be guaranteed to detect all of them. For general circuits, it seems a hopeless task to find efficient non-combinatorial methods for finding all solutions. It may however be feasible to derive sufficient conditions, based on the topology of the circuit, that a homotopy method finds all solutions.

Although the main emphasis lies on resistive circuits, the thesis includes two chapters on other applications.

- In chapter 6 a number of problems of analysis and synthesis of neural networks are discussed. Again the results are based on the observation of the

similarity with well-known problems in mathematical programming (variational inequalities and linear complementarity problems). The main part of the chapter is devoted to a complementary pivoting algorithm for calculating multiple equilibrium points in cellular neural networks. This is important during the design since proper operation of such networks depends on nonlinear dynamics and the presence of multiple equilibrium points. The questions that are left open here are analogous to those mentioned above : exploitation of the very sparse and structured form of the interconnection matrices, and a further investigation on the influence of the topology of the network.

The rest of chapter 6 contains a new theorem on the stability of neural networks with non-symmetric coefficients, and an iterative method for determining suitable coefficients from design specifications.

- Chapter 7 was included in order to show that variable dimension algorithms have applications beyond resistive circuits as well. We presented a modification of the octahedral algorithm that exploits the inherent structure of the harmonic balance equations.

Mathematical programming and combinatorial optimization continue to advance at a rapid pace and the implications of recent developments (e.g. interior point methods) on circuit theory remain largely unexplored. We hope that this thesis can serve as an example of how rewarding a combination of the concepts of nonlinear circuit theory with the constructive methods from mathematical programming can be.

Appendix A

Triangulations

The purpose of this appendix is to provide a formal description of the triangulations used in the text. The triangulation for the octahedral algorithm, outlined in Chapter 3 and given here in detail, is basically taken from [9, 31]. The third section of this appendix gives the modified version of this triangulation needed for the variable dimension algorithm of Chapter 7.

A.1 A few facts from nonnegative linear algebra

Points $\mathbf{x}^1, \ldots, \mathbf{x}^k$ are called *affinely independent* if $\sum_{i=1}^k \lambda_i.\mathbf{x}^i = 0$ and $\sum_{i=1}^k \lambda_i = 0$ imply $\lambda_i = 0, \quad i = 1, \ldots, k$. A *cell* is a closed, convex, polyhedral subset of R^n. The *dimension* of a cell σ, denoted $\dim \sigma$, is the maximum number of affinely independent points in σ.

The *convex hull* of a set $\sigma \subset R^n$, denoted cvx σ, is the collection of all $\sum_{i=1}^k \lambda_i \mathbf{x}^i$ with $\mathbf{x}^i \in \sigma$, $\lambda_i \geq 0$, $\sum_{i=1}^k \lambda_i = 1$ and k a positive integer. A convex subset ρ of a cell σ is called a *face* of σ if $\sum_{i=1}^k \lambda_i \mathbf{x}^i \in \rho, \sum_{i=1}^n \lambda_i = 1, \lambda_i \geq 0$, and $\mathbf{x}^i \in \sigma$ for $i = 1, \ldots, k$ imply $\mathbf{x}^i \in \rho$ for $\lambda_i > 0$. An *m-face* is a face of dimension m.

The convex hull of $m+1$ affinely independent points, cvx $\{\mathbf{v}^1, \ldots, \mathbf{v}^{m+1}\}$, is called an *m-simplex*, and the vectors $\mathbf{v}^i$ are called its *vertices*. The $(m-1)$-faces of an m-simplex are called its *facets*. An m-simplex has $m+1$ facets and each facet is the convex hull of m vertices of the simplex : we write $\tau = \sigma \setminus \mathbf{v}^i$ if $\mathbf{v}^i$ is the vertex of σ not belonging to facet τ.

A.2 The octahedral triangulation

We define here the representation and replacement rules for the triangulation of Section 3.3.

Let α be a permutation of $\{1, \ldots, m\}$. For $1 \leq i \leq j \leq n$, we denote by $\alpha[i, j]$ the rotated permutation $\hat{\alpha}$ defined by

$$\begin{aligned} \hat{\alpha}_i &= \alpha_j, \\ \hat{\alpha}_k &= \alpha_{k-1}, \text{ for } i < k \leq j \\ \hat{\alpha}_k &= \alpha_k, \text{ otherwise.} \end{aligned}$$

In other words, $\hat{\alpha}$ is α with the elements i through j rotated to the right. If $1 \leq j \leq i \leq n$, then $\hat{\alpha}[i, j]$ stands for a similar rotation to the left :

$$\begin{aligned} \hat{\alpha}_i &= \alpha_j, \\ \hat{\alpha}_k &= \alpha_{k+1}, \text{ for } j \leq k < i \\ \hat{\alpha}_k &= \alpha_k, \text{ otherwise.} \end{aligned}$$

A.2.1 Representation rule

Each simplex in the triangulation, except the non-simplicial cell γ, is represented by a 4-tuple $(\mathbf{t}, \mathbf{u}, \pi, k)$:

- $k = 1, \ldots, n$,
- π a permutation of $\{1, 2, \ldots, n+1\}$, with $\pi_{k+1} < \pi_{k+2} < \cdots < \pi_{n+1} < n+1$,
- $\mathbf{t} \in \{-1, +1\}^n$,
- $\mathbf{u} \geq \mathbf{0}$ an integer n-vector, with $u_{\pi_{k+1}} = \cdots = u_{\pi_{n+1}} = \|\mathbf{u}\|_\infty$, and if j is the index for which $\pi_j = n+1$, then $u_{\pi_i} < u_{\pi_{n+1}}$ for $i < j$.

As usual $\mathbf{e}^i$ denotes the i-th unit vector. For a given 4-tuple $(\mathbf{t}, \mathbf{u}, \pi, k)$, define $\mathbf{w}^1$ as

$$w_i^1 = t_i.u_i, \quad i = 1, \ldots, n,$$

and $\mathbf{w}^2, \ldots, \mathbf{w}^{n+2}$ as

$$\begin{aligned} \mathbf{w}^{i+1} &= \mathbf{w}^i + t_{\pi_i}\mathbf{e}^{\pi_i}, \quad i = 1, \ldots, k, \ \pi_i \neq n+1, \\ &= \mathbf{w}^i + \sum_{j=k+1}^{n+1} t_{\pi_j}\mathbf{e}^{\pi_j}, \quad \text{if } \pi_i = n+1, \\ &= t_{\pi_i}\mathbf{e}^{\pi_i}, \quad i = k+1, \ldots, n+1. \end{aligned}$$

Then the $n+2$ vertices of the simplex represented by $(\mathbf{t}, \mathbf{u}, \pi, k)$ are $\mathbf{v}^i = [(\mathbf{w}^i)^t \ \ 1]^t$, for $i = 1, \ldots, k+1$ and $\mathbf{v}^i = [(\mathbf{w}^i)^t \ \ 0]^t$, for $i = k+2, \ldots, n+2$. The vector $\mathbf{t}$ corresponds to the orthant in which the simplex $\text{cvx}\,\{\mathbf{w}^1, \ldots, \mathbf{w}^{k+1}\}$ lies. From $\mathbf{t}$ one can retrieve the sign vector $\mathbf{s}$ of the cell $X(\mathbf{s})$ to which $\mathbf{w}^1, \ldots, \mathbf{w}^{k+1}$ belong :

$$\begin{aligned} s_{\pi_i} &= 0 \quad \text{if} \quad 1 \leq i \leq k \ \text{ and } \ \pi_i \neq n+1, \\ s_{\pi_i} &= t_{\pi_i} \quad \text{if} \quad k+1 \leq i \leq n+1. \end{aligned}$$

A.2.2 Replacement rule

Assume one leaves the simplex $(\mathbf{t}, \mathbf{u}, \pi, k)$ through the facet opposite its d-th vertex ($1 \leq d \leq n+2$). The steps listed below replace $(\mathbf{t}, \mathbf{u}, \pi, k)$ by the representation of the new simplex, d by the index of the vertex opposite the entrance facet of the new simplex. It also calculates ψ, an integer $(n+2)$-vector with $\mathbf{v}^i = \hat{\mathbf{v}}^{\psi_i}$ for $i \in \{1, \ldots, n+2\} \setminus \{d\}$, where $\hat{\mathbf{v}}$ and $\mathbf{v}$ are the new and the old vertices, respectively. There are also two cases in which no replacement will be found : if the algorithm terminates (the exit facet lies in $R^n \times \{1\}$), or if algorithm returns at the origin (the exit facet belongs to the non-simplicial cell γ). This latter case will not occur under normal circumstances.

Let j be the index for which $\pi_j = n+1$ and initialize ψ as $\psi = (1, 2, \ldots, n+2)$. To simplify notation, define π_{n+2} as $\pi_{n+2} = n+2$. Also define $\mathbf{e}^{n+1}$ as $\mathbf{e}^{n+1} = \sum_{i=k+1}^{n+1} \mathbf{e}^{\pi_i}$.

One can distinguish the following cases.

1. $d = 1$. Then $\mathbf{u} := \mathbf{u} + \mathbf{e}^{\pi_1}$, $\pi := \pi[k, 1]$, $\psi := \psi[1, k+1]$, $d := k+1$.

2. $d \leq k$.

   - If $d = j+1$ and $u_{\pi_{j+1}} = u_{\pi_{n+1}}$, then compute the index $l = \min\{i = k+1, \ldots, n+2 \mid \pi_{j+1} < \pi_i\}$, and take $\pi := \pi[l-1, j+1]$, $\psi := \psi[j+1, l]$, $d := l$ and $k := k-1$.
   - Otherwise, take $\pi := \pi[d-1, d]$.

3. $d = k+1$ and $\pi_k \neq n+1$.

   - If $u_{\pi_k} \neq 0$, then $\mathbf{u} := \mathbf{u} - \mathbf{e}^{\pi_k}$, $\pi := \pi[1, k]$, $\psi := \psi[k+1, 1]$, and $d := 1$.
   - If $u_{\pi_k} = 0$, then $t_{\pi_k} := -t_{\pi_k}$.

4. $d = k+1$ and $\pi_k = n+1$.

   - If $u_{\pi_{n+1}} \neq 0$, then $\mathbf{u} := \mathbf{u} - \mathbf{e}^{\pi_k}$, $\pi := \pi[1, k]$, $\psi := \psi[k+1, 1]$, and $d := 1$.
   - If $u_{\pi_{n+1}} = 0$, then one enters the non-simplicial cell γ.

5. $d > k+1$.

   - If $d = k+2$ and $k = n$, then the exit facet lies in $R^n \times \{1\}$, and the algorithm terminates.
   - Otherwise, $\pi := \pi[j+1, d-1]$, $\psi := \psi[d, j+1]$, $d := j+1$, $k := k+1$.

A.3 The modified triangulation

This section describes the triangulation needed to implement the algorithm of Section 7.2.

A.3.1 Representation rule

Each simplex is represented by a 5-tuple $(\mathbf{t}, \mathbf{u}, \pi, k, r)$ with

- $r = 1, \ldots, l$,
- $k = m_{r-1}+1, \ldots, m_r$,
- π a permutation of $\{1, \ldots, m_r\} \cup \{n+1\}$, with $m_{r-1} < \pi_{k+1} < \pi_{k+2} < \cdots < \pi_{m_r+1} < n+1$,
- $\mathbf{t} \in \{-1, +1\}^{m_r}$,
- $\mathbf{u} \geq 0$ an integer m_r-vector, $u_{\pi_{k+1}} = \cdots = u_{\pi_{m_r+1}} = \max\{u_{m_{r-1}+1}, \ldots, u_{m_r}\}$, and, if j is the index for which $\pi_j = n+1$, then $u_{\pi_i} < u_{\pi_{m_r+1}}$ if $i < j$ and $m_{r-1} < \pi_i \leq m_r$.

The simplex represented by $(\mathbf{t}, \mathbf{u}, \pi, k, r)$ is an (m_r+1)-simplex. Define $\mathbf{w}^1$ as

$$w_i^1 = t_i . u_i, \quad i = 1, \ldots, m_r,$$

and $\mathbf{w}^2, \ldots, \mathbf{w}^{m_r+2}$ as

$$\begin{aligned} \mathbf{w}^{i+1} &= \mathbf{w}^i + t_{\pi_i} \mathbf{e}^{\pi_i}, \quad i = 1, \ldots, k, \ \pi_i \neq n+1, \\ &= \mathbf{w}^i + \sum_{j=k+1}^{m_r+1} t_{\pi_j} \mathbf{e}^{\pi_j}, \quad \text{if } \pi_i = n+1, \\ &= t_{\pi_i} \mathbf{e}^{\pi_i}, \quad i = k+1, \ldots, m_r+1. \end{aligned}$$

Then the $m_r + 2$ vertices of the simplex represented by $(\mathbf{t}, \mathbf{u}, \pi, k, r)$ are $\mathbf{v}^i = [(\mathbf{w}^i)^t \ \ 1]^t$, for $i = 1, \ldots, k+1$ and $\mathbf{v}^i = [(\mathbf{w}^i)^t \ \ 0]^t$, for $i = k+2, \ldots, m_r + 2$.

To clarify the link with the definitions of Chapter 7, we briefly describe the labelling rule needed to use this triangulation for the algorithm. The simplex cvx $\{\mathbf{w}^1, \ldots, \mathbf{w}^{k+1}\}$ lies in the cell $X(r, \mathbf{s})$ with $\mathbf{s}$:

$$\begin{aligned} s_{\pi_i} &= 0 \quad \text{if} \quad 1 \leq i \leq k \ \text{ and } \ \pi_i \neq n+1, \\ s_{\pi_i} &= t_{\pi_i} \quad \text{if} \quad k+1 \leq i \leq m_r + 1. \end{aligned}$$

One can then carry out the algorithm of Chapter 7 by simplicial pivoting if the vertices $\mathbf{v}^1, \ldots, \mathbf{v}^{m_r+2}$ are labelled as

$$\begin{aligned} \mathrm{l}(\mathbf{v}^i) &= [(\mathbf{f}_\alpha(\mathbf{w}^i))^t \ \ 1]^t, \quad i = 1, \ldots, k+1, \\ \mathrm{l}(\mathbf{v}^i) &= [(\mathbf{v}^i)^t \ \ 0]^t, \quad i = k+2, \ldots, m_r + 2, \end{aligned}$$

with $\alpha = \{1, \ldots, m_r\}$.

A.3.2 Replacement rule

Let j be the index for which $\pi_j = n+1$ and initialize ψ as $\psi = (1, 2, \ldots, m_r+2)$. Define π_{m_r+2} as $\pi_{m_r+2} = n+2$, and let $\mathbf{e}^{n+1} = \sum_{i=k+1}^{m_r+1} \mathbf{e}^{\pi_i}$.

One can distinguish the following cases.

1. $d = 1$. Then $\mathbf{u} := \mathbf{u} + \mathbf{e}^{\pi_1}$, $\pi := \pi[k, 1]$, $\psi := \psi[1, k+1]$, $d := k+1$.

2. $d \leq k$.

   - If $d = j+1$, $u_{\pi_{j+1}} = u_{\pi_{n+1}}$ and $m_{r-1} < \pi_{j+1} \leq m_r$, then compute the index $l = \min\{i = k+1, \ldots, m_r + 2 \ \mid \ \pi_{j+1} < \pi_i\}$, take $\pi := \pi[l-1, j+1]$, $\psi := \psi[j+1, l]$, $d := l$ and $k := k-1$.
   - Otherwise, take $\pi := \pi[d-1, d]$.

3. $d = k+1$ and $\pi_k \neq n+1$.

   - If $u_{\pi_k} \neq 0$, then $\mathbf{u} := \mathbf{u} - \mathbf{e}^{\pi_k}$, $\pi := \pi[1, k]$, $\psi := \psi[k+1, 1]$, and $d := 1$.
   - If $u_{\pi_k} = 0$, then $t_{\pi_k} := -t_{\pi_k}$.

4. $d = k+1$ and $\pi_k = n+1$.

   - If $u_{\pi_{m_r+1}} \neq 0$, then $\mathbf{u} := \mathbf{u} - \mathbf{e}^{\pi_k}$, $\pi := \pi[1, k]$, $\psi := \psi[k+1, 1]$, and $d := 1$.

   - If $u_{\pi_{m_r+1}} = 0$ and $r = 1$, then the path reaches the origin, and the algorithm terminates.
   - If $u_{\pi_{m_r+1}} = 0$ and $r > 1$, then
     - $r := r - 1$,
     - $\mathbf{u} := [u_1, \ldots, u_{m_r}]^t$,
     - $\mathbf{t} := [t_1, \ldots, t_{m_r}]^t$,
     - search for the index $j = \min\{1, \ldots, k-1 \mid m_{r-1} < \pi_i \leq m_r \text{ and } u_{\pi_i} = \max\{u_{m_{r-1}+1}, \ldots, u_{m_r}\}\}$, and take $\pi_{m_r+1} := \pi_j$, $\pi_j := n+1$.
     - $k := m_r$, $d := k+2$.

5. $d > k+1$.

   - If $d = k+2$, $k = m_r$ and $r = l$, then the algorithm terminates.
   - If $d = k+2$, $k = m_r$ and $r < l$, then
     - $\pi_j := \pi_{m_r+1}$, $\pi_{m_r+1} := n+1$, and $\pi_{i+1} = i, \quad i = m_r + 1, \ldots, m_{r+1}$.
     - augment the m_r vector $\mathbf{u}$ by adding $u_i := 0, i = m_r+1 \ldots m_{r+1}$,
     - augment the m_r vector $\mathbf{t}$ by adding components $t_i = \pm 1, i = m_r + 1 \ldots m_{r+1}$, (the choice of t_i is determined by $\mathbf{f}(\hat{\mathbf{x}})$, where $\hat{\mathbf{x}}$ is the piecewise-linear approximation to the solution of the first m_r equations; $t_i = 1$ if $f_i(\hat{\mathbf{x}}) \leq 0$ and $t_i = -1$ if $f_i(\hat{\mathbf{x}}) > 0$),
     - $r := r+1$, $k := k+1$, $d := k+1$,
   - Otherwise, $\pi := \pi[j+1, d-1]$, $\psi := \psi[d, j+1]$, $d = j+1$, $k := k+1$.

Appendix B

Theorems of alternative

Theorems of alternative belong to the general theory of linear inequalities [81]. Each variant deals with a specific pair of linear inequalities and asserts that exactly one of the two systems is solvable. As such they permit to prove existence or non-existence of solutions in a linear system by studying the alternative.

B.1 General form

Theorems of alternative exist in a wide variety, but they can all be derived from a basic theorem [81].

Theorem 28 *Let* $\mathbf{A}$ *be an* $m \times n$*-matrix. Let* I_i, $i = 1, \ldots, m+n$ *be real intervals. Then exactly one of the following statements is valid.*

1. *There exists an* $(n+m)$*-vector* $\boldsymbol{\xi}$ *with*

$$\begin{bmatrix} \xi_{n+1} \\ \vdots \\ \xi_{n+m} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_n \end{bmatrix},$$

   and $\boldsymbol{\xi} \in I_1 \times I_2 \times \cdots \times I_{m+n}$.

2. *There exists an* $(n+m)$*-vector* $\boldsymbol{\xi}^*$ *with*

$$\begin{bmatrix} \xi_1^* \\ \vdots \\ \xi_n^* \end{bmatrix} = -\mathbf{A}^t \begin{bmatrix} \xi_{n+1}^* \\ \vdots \\ \xi_{n+m}^* \end{bmatrix},$$

   and $\boldsymbol{\xi}^t.\boldsymbol{\xi}^* < 0, \ \ \forall \boldsymbol{\xi} \in I_1 \times I_2 \times \cdots \times I_{m+n}$.

It should be noted that the intervals I_i may be open or closed, bounded or unbounded.

B.2 Examples

Taking $I_1 = \cdots = I_n = (0, \infty)$ and $I_{n+1} = \cdots = I_{n+m} = \{0\}$ leads to the following theorem.

Theorem 29 *Let* $\mathbf{A}$ *be an* $m \times n$ *matrix. Then exactly one of the following two statements is valid.*

1. *There exists a vector* $\mathbf{x} > \mathbf{0}$ *such that* $\mathbf{A}\mathbf{x} = \mathbf{0}$.
2. *There exists a vector* $\mathbf{y}$ *with* $\mathbf{A}^t\mathbf{y} \geq \mathbf{0}$ *and* $\mathbf{A}^t\mathbf{y} \neq \mathbf{0}$.

As a second example, assume an m-vector $\mathbf{b} \neq 0$ is given and take $I_1 = \cdots = I_n = [0, \infty)$, $I_{n+k} = \{b_k\}$ for $k = 1, \ldots, m$.

Theorem 30 *Let* $\mathbf{A}$ *be an* $m \times n$ *matrix and* $\mathbf{b}$ *an* m*-vector. Then exactly one of the following two statements is valid.*

1. *There exists a vector* $\mathbf{x} \geq \mathbf{0}$ *such that* $\mathbf{A}\mathbf{x} = \mathbf{b}$.
2. *There exists a vector* $\mathbf{y}$ *with* $\mathbf{A}^t\mathbf{y} \geq \mathbf{0}$ *and* $\mathbf{b}^t\mathbf{y} < 0$.

B.3 Application to circuit theory

One of the most important examples of a theorem of alternative is the coloured branch theorem [123, 82] from graph theory. Let $\mathbf{Q}$ be the main part of the fundamental cutset matrix corresponding to a tree in a connected graph. In other words the tree branch currents and voltages, $\mathbf{i}_T$, $\mathbf{v}_T$ and the cotree branch currents and voltages $\mathbf{i}_{CT}$, $\mathbf{v}_{CT}$ are related as

$$\begin{aligned} \mathbf{i}_T &= -\mathbf{Q}\,\mathbf{i}_{CT}, \\ \mathbf{v}_{CT} &= \mathbf{Q}^t\mathbf{v}_T. \end{aligned}$$

From theorem 28 one can then derive the following by taking $\mathbf{A} = -\mathbf{Q}$.

Theorem 31 *Paint one branch of the graph dark green and the remaining branches green, red or black. Then exactly one of the following two statements holds.*

1. *There exists a current vector* $\mathbf{i}$ *such that* $i_j \geq 0$ *if branch* j *is painted green,* $i_j = 0$ *if branch* j *is painted black and* $i_j > 0$ *for the dark green branch.*

2. *There exists a voltage vector* $\mathbf{v}$ *such that* $v_j \leq 0$ *if branch* j *is painted green,* $v_j = 0$ *if branch* j *is painted red and* $v_j < 0$ *for the dark green branch.*

This can be shown to be equivalent to the more classical formulation of the coloured branch theorem [123, 82].

Theorem 32 *Paint one branch of the graph dark green and the remaining branches green, red or black, and orient the green and dark green branches. Then exactly one of the following two statements holds.*

1. *There exists a loop containing the dark green branch and formed exclusively with green branches (oriented uniformly with the dark green branch) and red branches.*

2. *There exists a cutset containing the dark green branch and formed exclusively with green branches (oriented uniformly with the dark green branch) and black branches.*

Bibliography

[1] B.H. Ahn, 'Solution of nonsymmetric linear complementarity problems by iterative methods', *Journal of Optimization Theory and Applications* 33, 175-185 (1981).

[2] B.H. Ahn, 'Iterative methods for linear complementarity problems with upperbounds on primary variables', *Mathematical Programming* 26, 295-315 (1983).

[3] A.V. Aho, J.E. Hopcroft, J.D. Ullmann, *Data structures and algorithms*, (Addison-Wesley, 1983).

[4] E. Allgower, K. Georg, 'Simplicial and continuation methods for approximating fixed points and solutions to systems of equations', *SIAM Review* 22, 28-85 (1980).

[5] E. Allgower, K. Georg, *Numerical continuation methods. An introduction*, (Springer, 1990).

[6] J.A. Anderson, 'Cognitive and psychological computation with neural models', *IEEE Transactions on Systems Man and Cybernetics* SMC-13, 799-815 (1983).

[7] A. Berman, R. Plemmons, *Nonnegative matrices in the mathematical sciences*, (Academic Press, New York, 1979).

[8] K.H. Borgwardt, *The simplex method. A probabilistic analysis*, (Springer-Verlag, New York, 1987).

[9] M.N. Broadie, B.C. Eaves, 'A variable rate refining triangulation', *Mathematical Programming* 38, 161-202, (1987).

[10] Y. Censor, 'Row-action techniques for huge and sparse systems and their applications', *SIAM Review* 23, 444-466 (1980).

[11] K.S. Chao, D.K. Liu, C.T. Pan, 'A systematic search method for obtaining multiple solutions of simultaneous nonlinear equations', *IEEE Transactions on Circuits and Systems* CAS-22, 748-753 (1975).

[12] M.J. Chien, E.S. Kuh, 'Solving nonlinear resistive networks using piecewise-linear analysis and simplicial subdivision', *IEEE Transactions on Circuits and Systems* CAS-24, 305-317 (1977).

[13] L.O. Chua, 'Nonlinear circuits', *IEEE Transactions on Circuits and Systems* CAS-31, 69-87 (1984).

[14] L.O. Chua, D.N. Green, 'A qualitative analysis of the behavior of dynamic nonlinear networks : steady-state solutions of nonautonomous networks', *IEEE Transactions on Circuits and Systems* CAS-23, 530-550 (1976).

[15] L.O. Chua, G.N. Lin, 'Nonlinear programming without computation', *IEEE Transactions on Circuits and Systems* CAS-31, 182-188 (1984).

[16] L.O. Chua, A. Ushida, 'A switching-parameter algorithm for finding multiple solutions of nonlinear resistive circuits', *International Journal of Circuit Theory and Applications* 4, 215-239 (1976).

[17] L.O. Chua, N.N. Wang, 'On the application of degree theory to the analysis of resistive nonlinear networks', *International Journal of Circuit Theory and Applications* 5, 35-68 (1977).

[18] L.O. Chua, L. Yang, 'Cellular neural networks : Theory', *IEEE Transactions on Circuits and Systems* CAS-35, 1257-1272 (1988).

[19] L.O. Chua, L. Yang, 'Cellular neural networks : Applications', *IEEE Transactions on Circuits and Systems* CAS-35, 1273-1290 (1988).

[20] L.O. Chua, R.L.P. Ying, 'Finding all solutions of piecewise-linear circuits', *International Journal of Circuit Theory and Applications* 10, 201-229 (1982).

[21] L.O. Chua, R.L.P. Ying, 'Canonical piecewise-linear analysis', *IEEE Transactions on Circuits and Systems* CAS-30, 125-140 (1983).

[22] L.O. Chua, J. Yu, Y. Yu, 'Negative resistance devices', *International Journal of Circuit Theory and Applications* 11, 161-186 (1983).

[23] R.W. Cottle, G.B. Dantzig, 'Complementary pivot theory of mathematical programming', *Linear Algebra and Its Applications* 1, 103-125 (1968).

[24] R.W. Cottle, F. Giannessi, J.-L. Lions (eds.), *Variational inequalities and complementarity problems*, (Wiley, New York, 1980).

[25] B. De Moor, L. Vandenberghe, J. Vandewalle, 'The generalized linear complementarity problem and an algorithm to find all its solutions', to be published in *Mathematical Programming*.

[26] B. De Moor, *Mathematical concepts and techniques for modelling of static and dynamic systems*, Doctoral Thesis, Department of Electrical Engineering, K.U. Leuven, June 1988.

[27] J.B. Dennis, *Mathematical programming and electrical networks*, (Chapman and Hall, London, 1959).

[28] J.E. Dennis, J.J. Moré, 'Quasi-Newton methods, motivation and theory', *SIAM Review* 19, 46-89 (1977).

[29] J.E. Dennis, R.B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, (Prentice-Hall, Englewood Cliffs, N.J., 1983).

[30] C.A. Desoer, F.F. Wu, 'Nonlinear monotone networks', *SIAM Journal of Applied Mathematics* 26, 315-333 (1974).

[31] B.C. Eaves, *A course in triangulations for solving equations with deformations*, (Springer-Verlag, New York, 1984).

[32] B.C. Eaves, H. Scarf, 'The solution of systems of piecewise linear equations', *Mathematics of Operations Research* 1, 1-27 (1976).

[33] M. Fiedler, V. Pták, 'Some generalizations of positive definiteness and monotonicity', *Numerische Mathematik* 9, 163-172 (1966).

[34] M. Fosséprez, *Topologie et comportement des circuits non linéaires non réciproques*, (Presses Polytechniques Romandes, Lausanne, 1989).

[35] C.B. Garcia, F.J. Gould, 'Relations between several path following algorithms and local and global Newton methods', *SIAM Review* 22, 263-274 (1980).

[36] C.B. Garcia, W.I. Zangwill, 'An approach to homotopy and degree theory', *Mathematics of Operations Research* 4, 390-405 (1979).

[37] C.B. Garcia, W.I. Zangwill, *Pathways to solutions, fixed points and equilibria*, (Prentice-Hall, Englewood Cliffs N.J., 1981).

[38] P.E. Gill, G.H. Golub, W. Murray, M.A. Saunders, 'Methods for modifying matrix factorizations', *Mathematics of Computation* 28, 505-535 (1974).

[39] P.E. Gill, W. Murray, 'Quasi-Newton methods for unconstrained optimization', *Journal of the Institute of Mathematics and Its Applications* 9, 91-108 (1972).

[40] J.L. Goffin, 'The relaxation method for solving systems of linear inequalities', *Mathematics of Operations Research* 5, 388-414 (1980).

[41] B. Gopinath, D. Mitra, 'When are transistors passive ?', *Bell System Technical Journal* 50, 2835-2847 (1971).

[42] H.J. Greenberg, 'Equilibria of the brain-state-in-a-box neural model', *Neural Networks* 1, 323-324 (1988).

[43] S. Grossberg, 'Nonlinear neural networks : principles, mechanisms, and architectures', *Neural Networks* 1, 17-61 (1988).

[44] V. Guillemin, A. Pollack, *Differential topology*, (Prentice-Hall, Englewood Cliffs N.J., 1974).

[45] M. Hasler, 'Non-linear non-reciprocal resistive circuits with a structurally unique solution', *International Journal on Circuit Theory and Applications* 14, 237-262 (1986).

[46] M. Hasler, J. Neirynck, *Circuits non linéaires*, (Presses Polytechniques Romandes, Lausanne, 1985).

[47] D. Hente, R.H. Jansen, 'Frequency domain continuation method for the analysis and stability investigation of nonlinear microwave circuits', *IEE Proceedings, part H* 133, 351-362 (1986).

[48] M.W. Hirsch, S. Smale, 'On algorithms for solving $f(x) = 0$', *Communications on Pure and Applied Mathematics* 32, 281-312 (1979).

[49] C.W. Ho, D.A. Zein, A.E. Ruehli, P.A. Brennan, 'An algorithm for DC solutions in an experimental general purpose interactive circuit design program', *IEEE Transactions on Circuits and Systems* CAS-24, 416-422 (1977).

[50] G. Huang, R.W. Liu, 'A simple algorithm for finding all solutions of piecewise-linear networks', *IEEE Transactions on Circuits and Systems* CAS-36, 600-609 (1989).

[51] S. Karamardian, 'The complementarity problem', *Mathematical Programming* 2, 107-129 (1972).

[52] J. Katzenelson, 'An algorithm for solving nonlinear resistor networks', *Bell System Technical Journal* 44, 1605-1620 (1965).

[53] M. Kojima, 'An introduction to variable dimension algorithms for solving systems of equations', in : *Numerical solution of nonlinear equations*, A. Dold, B. Eckman Eds. (Berlin, Springer Verlag, 1981).

[54] M. Kojima, Y. Yamamoto, 'Variable dimension algorithms : basic theory, interpretations and extensions of some existing methods', *Mathematical Programming* 24, 177-215 (1982).

[55] M. Kojima, Y. Yamamoto, 'A unified approach to the implementation of several restart fixed point algorithms and a new variable dimension algorithm', *Mathematical Programming* 28, 288-328 (1984).

[56] E.S. Kuh, I.N. Hajj, 'Nonlinear circuit theory : resistive networks', *Proceedings of the IEEE* 59, 340-355 (1971).

[57] K.S. Kundert, A. Sangiovanni-Vincentelli, 'Simulation of nonlinear circuits in the frequency domain', *IEEE Transactions on Computer-Aided Design* CAD-5, 521-535 (1986).

[58] K.S. Kundert, J.K. White, A. Sangiovanni-Vincentelli, *Steady-state methods for simulating analog and microwave circuits*, (Kluwer Academic Publishers, 1990).

[59] S.M. Lee, K.S. Chao, 'Multiple solutions of piecewise-linear resistive networks', *IEEE Transactions on Circuits and Systems* CAS-30, 84-89 (1983).

[60] J. Mandel, 'Convergence of the cyclical relaxation method for solving linear inequalities', *Mathematical Programming* 30, 1984, 218-228.

[61] T. Matsumoto, L.O. Chua, R. Furukawa, 'CNN cloning template : hole-filler', *IEEE Transactions on Circuits and Systems* CAS-37, 635-638 (1990).

[62] T. Matsumoto, L.O. Chua, H. Suzuki, 'CNN cloning template : connected component detector', *IEEE Transactions on Circuits and Systems* CAS-37, 633-635 (1990).

[63] T. Matsumoto, L.O. Chua, H. Suzuki, 'CNN cloning template : shadow detector', *IEEE Transactions on Circuits and Systems* CAS-37, 1070-1073 (1990).

[64] T. Matsumoto, L.O. Chua, T. Yokohama, 'Image thinning with a cellular neural network', *IEEE Transactions on Circuits and Systems* CAS-37, 638-640 (1990).

[65] A. Mees, *Dynamics of feedback systems*, (Wiley, 1980).

[66] J.W. Milnor, *Topology from the differentiable viewpoint*, (The University Press of Virginia, Charlottesville, 1965).

[67] P.J. Moylan, 'Matrices with positive principal minors', *Linear Algebra and Its Applications* 17, 53-58 (1977).

[68] R.O. Nielsen, A.N. Willson, 'A fundamental result concerning the topology of transistor circuits with multiple equilibria', *Proceedings IEEE* 68, 196-208 (1980).

[69] T. Nishi, L.O. Chua, 'Topological criteria for nonlinear resistive circuits containing controlled sources to have a unique solution', *IEEE Transactions on Circuits and Systems* CAS-31, 722-741 (1984).

[70] T. Nishi, L.O. Chua, 'Uniqueness of solution for nonlinear resistive circuits containing CCCS's or VCVS's whose controlling coefficients are finite', *IEEE Transactions on Circuits and Systems* CAS-33, 381-397 (1986).

[71] T. Nishi, L.O. Chua, 'Topological conditions for a resistive circuit containing negative non-linear resistors to have a unique solution', *International Journal of Circuit Theory and Applications* 15, 193-210 (1987).

[72] T. Ohtsuki, T. Fujisawa, S. Kumagai, 'Existence theorems and a solution algorithm for piecewise-linear resistor networks', *SIAM Journal on Mathematical Analysis* 8, 69-99 (1977).

[73] J.M. Ortega, W.C. Rheinboldt, *Iterative solutions of nonlinear equations in several variables*, (Academic Press, New York, 1970).

[74] J.S. Pang, 'On the convergence of a basic iterative method for the implicit complementarity problem", *Journal of Optimization Theory and Applications* 37, 149-162 (1982).

[75] J.S. Pang, 'Asymmetric variational inequality problems over product sets : applications and iterative methods', *Mathematical Programming* 31, 206-219 (1985).

[76] J.S. Pang, 'Necessary and sufficient conditions for the convergence of iterative methods for the linear complementarity problem', *Journal of Optimization Theory and Applications* 42, 1-17 (1984).

[77] R.J. Plemmons, 'M-Matrix characterizations. 1.- Nonsingular M-matrices', *Linear Algebra and Its Applications* 18, 175-188 (1977).

[78] C.P. Reames, A.N. Willson, 'Global convergence of Newton's method in the DC analysis of single-transistor circuits', *Electronics Letters* 18, 519-520 (1982).

[79] J. Renegar, 'Rudiments of an average case complexity theory for piecewise-linear path following algorithms', *Mathematical Programming* 40, 113-163 (1988).

[80] V. Rizzoli, 'State of the art and present trends in nonlinear microwave CAD techniques', *IEEE Transactions on Microwave Theory and Techniques* MTT-36, 343-363 (1988).

[81] R.T. Rockafellar, *Convex analysis*, (Princeton University Press, Princeton, NJ, 1972).

[82] R.T. Rockafellar, *Network flows and monotropic optimization*, (Wiley, New York, 1984).

[83] R. Saigal, 'A homotopy for solving large, sparse and structured fixed point problems', *Mathematics of Operations Research* 8, 557-578 (1983).

[84] I.W. Sandberg, A.N. Willson, Jr., 'Some theorems on properties of DC equations of nonlinear networks', *Bell System Technical Journal* 48, 1-34 (1969).

[85] I.W. Sandberg, 'Necessary and sufficient conditions for the global invertibility of certain nonlinear operators that arise in the analysis of networks', *IEEE Transactions on Circuit Theory* CT-18, 260-263 (1972).

[86] I.W. Sandberg, A.N. Willson, Jr., 'Existence of solutions for the equations of transistor-resistor-voltage source networks', *IEEE Transactions on Circuit Theory* CT-18, 619-625 (1971).

[87] I.W. Sandberg, A.N. Willson, Jr., 'Existence and uniqueness of solutions for the equations of nonlinear DC networks', *SIAM Journal on Applied Mathematics* 22, 173-186 (1972).

[88] H. Scarf, 'The approximation of fixed points of a continuous mapping', *SIAM Journal on Applied Mathematics* 15, 1328-1343 (1967).

[89] T.E. Stern, *Theory of nonlinear networks and systems*, (Addison-Wesley, New York, 1965).

[90] S.N. Stevens, P.M. Lin, 'Analysis of piecewise-linear resistive networks using complementary pivot theory', *IEEE Transactions on Circuits and Systems* CAS-28, 429-441 (1981).

[91] D.W. Tank, J.J. Hopfield, 'Simple 'neural' optimization networks : an A/D converter, signal decision circuit and a linear programming circuit', *IEEE Transactions on Circuits and Systems* CAS-31, 182-188 (1986).

[92] M.J. Todd, *The computation of fixed points and applications*, (Springer-Verlag, New York, 1976).

[93] M.J. Todd, 'Numerical stability and sparsity in piecewise linear algorithms', in : *Analysis and computation of fixed points*, (Academic Press, New York, 1980).

[94] M.J. Todd, 'Exploiting structure in piecewise-linear homotopy algorithms for solving equations', *Mathematical Programming* 18, 233-247 (1980).

[95] M.J. Todd, 'Traversing large pieces of linearity in algorithms that solve equations by following piecewise-linear paths', *Mathematics of Operations Research* 5, 242-257 (1980).

[96] M.J. Todd, A.H. Wright, 'A variable-dimension simplicial algorithm for antipodal fixed-point theorems', *Numer. Funct. Anal. and Optimiz.* 2, 155-186 (1980).

[97] L. Trajković, R.C. Melville, S.-C. Fang, 'Passivity and no-gain properties establish global convergence of a homotopy method for DC operating points', *Proceedings of the International Symposium om Circuits and Systems*, ISCAS-1990, (New Orleans, May 1990), pp. 914-917.

[98] L. Trajković, R.C. Melville, S.-C. Fang, 'Finding operating points of transistor circuits using homotopy methods', *Proceedings of the International Symposium on Circuits and Systems*, ISCAS-1991, (Singapore, June 1991), pp. 758-761.

[99] W.M.G. van Bokhoven, *Piecewise-linear modelling and analysis*, (Kluwer, Deventer, 1980).

[100] L. Vandenberghe, B.L. De Moor, J. Vandewalle, 'The generalized linear complementarity problem applied to the complete analysis of resistive piecewise-linear circuits', *IEEE Transactions on Circuits and Systems* CAS-36, 1382-1391 (1989).

[101] L. Vandenberghe, J. Vandewalle, 'Variable dimension algorithms for solving resistive circuits', *International Journal of Circuit Theory and Applications* 18, 443-474 (1990).

[102] L. Vandenberghe, J. Vandewalle, 'A continuous deformation algorithm for DC-analysis of active nonlinear circuits', *Journal of Circuits, Systems and Computers* 1, 327-351 (1991).

[103] L. Vandenberghe, J. Vandewalle, 'A path-following method for finding multiple equilibrium points in cellular neural networks', accepted for publication in the *International Journal of Circuit Theory and Applications.*

[104] L. Vandenberghe, B. De Moor, J. Vandewalle, 'The generalized linear complementarity problem applied to the complete analysis of resistive piecewise-linear circuits', *Proceedings of the International Symposium on Circuits and Systems*, ISCAS-1989, (Portland, May 1989), 2155-2158.

[105] L. Vandenberghe, J. Vandewalle, 'Dynamic properties of neural networks', in : *Proceedings of the Tenth Symposium on Information Theory in the Benelux*, (Houthalen, May 1989), 81-88.

[106] L. Vandenberghe, J. Vandewalle, 'Brain-state-in-a-box neural networks with asymmetric coefficients', *Proceedings of the International Joint Conference on Neural Networks*, (Washington, June 1989), I627-I630.

[107] L. Vandenberghe, J. Vandewalle, 'Variable dimension algorithms for solving nonlinear resistive circuits', *Proceedings of the European Conference on Circuit Theory and Design*, ECCTD-89, (Brighton, September 1989), 385-389.

[108] L. Vandenberghe, J. Vandewalle, 'On the training and the convergence of brain-state-in-a-box neural networks', *Proceedings of the First IEE Conference on Artificial Neural Networks*, (London, October 1989), 247-251.

[109] L. Vandenberghe, S. Tan, J. Vandewalle, 'Cellular neural networks : dynamic properties and adaptive learning algorithm', in : L.B. Almeida, C.J. Wellekens (eds.), *Neural Networks, Proceedings of the EURASIP Workshop on Neural Networks, Sesimbra, Portugal Feb. 1990*, Lecture Notes in Computer Science Vol. 412, (Springer Verlag, Berlin, 1990), pp. 141-150.

[110] L. Vandenberghe, J. Vandewalle, 'Application of relaxation methods to the adaptive training of neural networks', in : M.A. Kaashoek, J.H. van Schuppen, A.C.M. Ran, (eds.), *Signal Processing, Scattering and Operator Theory, and Numerical Methods, Proceedings of the International Symposium on the Mathematical Theory of Networks and Systems* MTNS-89, Vol. 3, (Birkhaüser, Boston, 1990), 189-196.

[111] L. Vandenberghe, J. Vandewalle, 'A globally convergent algorithm for solving a broad class of nonlinear resistive circuits', *Proceedings of the IEEE International Symposium on Circuits and Systems*, (New Orleans, May 1990), 403-406.

[112] L. Vandenberghe, J. Vandewalle, 'Finding multiple equilibrium points of cellular neural networks without enumeration', *Proceedings of the 1990 IEEE International Workshop on Cellular Neural Networks and their Applications* CNNA-90, (Budapest, December 1990), 45-54.

[113] L. Vandenberghe, J. Vandewalle, 'Variable dimension algorithms in circuit theory', *Proceedings of the IEEE Benelux/ProRISC Symposium on Circuits, Systems and Signal Processing*, (Veldhoven, April 1991), 101-104.

[114] L. Vandenberghe, J. Vandewalle, 'A continuous deformation method for resistive circuits containing active and non-reciprocal elements', *Proceedings of the IEEE International Symposium on Circuits and Systems*, ISCAS-91, (Singapore, June 1991), 762-765.

[115] L. Vandenberghe, J. Vandewalle, 'The computation of equilibrium points in cellular neural networks using complementary pivoting', *Proceedings of the European Conference on Circuit Theory and Design*, ECCTD-91, (Copenhagen, September 1991), 30-38.

[116] L. Vandenberghe, J. Vandewalle, 'DC-Analysis of active circuits by continuous deformation', *Proceedings of the European Conference on Circuit Theory and Design*, ECCTD-91, (Copenhagen, September 1991), 855-863.

[117] L. Vandenberghe, J. Vandewalle, 'A variable dimension algorithm for solving harmonic balance equations', to be presented at the *IEEE International Symposium on Circuits and Systems*, ISCAS-92, (San Diego, May 1992).

[118] L. Vandenberghe, J. Vandewalle, 'Variable dimension algorithms in the analysis of nonlinear circuits and systems', to be presented at the *International Seminar on Nonlinear Circuits and Systems*, (Moscow, June 1992).

[119] G. van der Laan, 'On the existence and approximation of zeroes', *Mathematical Programming* 28, 1-24 (1984).

[120] G. van der Laan, L.P. Seelen, 'Efficiency and implementation of simplicial zero point algorithms', *Mathematical Programming* 30, 196-217 (1984).

[121] G. van der Laan, A.J.J. Talman, 'A class of simplicial restart fixed point algorithms without an extra dimension', *Mathematical Programming* 20, 33-48 (1981).

[122] G. van der Laan, A.J.J. Talman, 'An algorithm for the linear complementarity problem with upper and lower bounds', *Journal of Optimization Theory and Applications* 62, 151-163 (1989).

[123] J. Vandewalle, L.O. Chua, 'The colored branch theorem and its applications in circuit theory', *IEEE Transactions on Circuits and Systems* CAS-27, 816-825 (1980).

[124] Y.-H. Wan, 'On the average speed of Lemke's algorithm for quadratic programming', *Mathematical Programming* 35, 236-246 (1986).

[125] L.T. Watson, 'Globally convergent homotopy methods : a tutorial', *Applied Mathematics and Computation* 31, 369-396 (1989).

[126] A.N. Willson, Jr., 'A useful generalization of the P_0-matrix concept', *Numerische Mathematik* 17, 62-70 (1971).

[127] A.N. Willson, Jr., 'Some aspects of the theory of nonlinear networks', *Proceedings of the IEEE* 61, 1092-113 (1973).

[128] A.H. Wright, 'The octahedral algorithm, a new simplicial fixed point algorithm', *Mathematical Programming* 21, 47-69 (1981).

[129] F.F. Wu, 'Existence of an operating point for a nonlinear circuit using the degree of a mapping'. *IEEE Transactions on Circuits and Systems* CAS-21, 671-677 (1974).

[130] Y. Yamamoto, 'A new variable dimension algorithm for the fixed point problem', *Mathematical Programming* 25, 329-342 (1983).

[131] Y. Yamamoto, 'A variable dimension fixed point algorithm and the orientation of simplices', *Mathematical Programming* 30, 301-312 (1984).

[132] K. Yamamura, K. Horiuchi, 'A globally and quadratically convergent algorithm for solving nonlinear resistive circuits', *IEEE Transactions on Computer-Aided Design* CAD-9, 487-499 (1990).

[133] W.I. Zangwill, C.B. Garcia, 'Equilibrium programming : the path following approach and dynamics', *Mathematical Programming* 21, 262-289 (1981).

[134] F. Zou, S. Shwarz, J.A. Nossek, 'Cellular neural network design using a learning algorithm', *Proceedings of the International Workshop on Cellular Neural Networks and Their Applications*, (Budapest 1991), 74-81.