

Computergestützte Schaltungsanalyse : verfügbare Sprachen und Methoden

Autor(en): **Vogel, J.S.**

Objektyp: **Article**

Zeitschrift: **Bulletin des Schweizerischen Elektrotechnischen Vereins, des
Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de
l'Association Suisse des Electriciens, de l'Association des
Entreprises électriques suisses**

Band (Jahr): **69 (1978)**

Heft 18

PDF erstellt am: **13.09.2024**

Persistenter Link: <https://doi.org/10.5169/seals-914935>

Nutzungsbedingungen

Die ETH-Bibliothek ist Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Inhalten der Zeitschriften. Die Rechte liegen in der Regel bei den Herausgebern.

Die auf der Plattform e-periodica veröffentlichten Dokumente stehen für nicht-kommerzielle Zwecke in Lehre und Forschung sowie für die private Nutzung frei zur Verfügung. Einzelne Dateien oder Ausdrucke aus diesem Angebot können zusammen mit diesen Nutzungsbedingungen und den korrekten Herkunftsbezeichnungen weitergegeben werden.

Das Veröffentlichen von Bildern in Print- und Online-Publikationen ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Die systematische Speicherung von Teilen des elektronischen Angebots auf anderen Servern bedarf ebenfalls des schriftlichen Einverständnisses der Rechteinhaber.

Haftungsausschluss

Alle Angaben erfolgen ohne Gewähr für Vollständigkeit oder Richtigkeit. Es wird keine Haftung übernommen für Schäden durch die Verwendung von Informationen aus diesem Online-Angebot oder durch das Fehlen von Informationen. Dies gilt auch für Inhalte Dritter, die über dieses Angebot zugänglich sind.

Computergestützte Schaltungsanalyse: Verfügbare Sprachen und Methoden ¹⁾

Von J.S. Vogel

621.3.049.7: 681.3 01:: 62.001.13;

Die starke Verbesserung des Preis-Leistungs-Verhältnisses von Computern und die Entwicklung neuer numerischer Verfahren für die Lösung von Gleichungs- und Differentialgleichungssystemen haben den Einsatz des Rechners für die Schaltungsentwicklung wirtschaftlich gemacht. Die Ausbildung der Ingenieure trägt dieser Entwicklung Rechnung, und eine grössere Anzahl von Analysenprogrammen gestatten es, Schaltungen in kurzer Zeit zu testen, zu selektionieren und optimal auszulegen.

Die Anforderungen, welche der Benutzer an Netzwerkprogramme stellt, werden beschrieben und die Erfüllung dieser Ansprüche bei verschiedenen Programmen einander gegenübergestellt. Wesentliche Fortschritte wurden in den letzten Jahren bei der konzisen Formulierung der Netzwerkgleichungen, bei der Lösung von linearen und nichtlinearen Gleichungssystemen und bei der Integration steifer Differentialgleichungen erzielt. Als Beispiele für diese Entwicklungen wird für jeden Bereich ein Verfahren kurz erläutert.

Le rapport prix/performance des calculateurs électroniques étant devenu bien meilleur et de nouveaux procédés numériques ayant été conçus pour la solution de systèmes d'équations différentielles et autres, l'emploi d'un calculateur pour l'élaboration de circuits se justifie économiquement. La formation des ingénieurs tient compte de cette évolution, et un grand nombre de programmes d'analyses permettent de contrôler, sélectionner ou élaborer d'une façon optimale des circuits.

Les exigences que pose l'utilisateur à des programmes de réseaux sont décrites et la satisfaction de ces exigences par divers programmes est comparée. Ces dernières années, d'importants progrès ont été réalisés pour une formulation concise des équations de réseaux, pour la solution de systèmes d'équations linéaires ou nonlinéaires, ainsi que pour l'intégration d'équations différentielles «stiff». Un procédé est décrit pour chaque domaine, à titre d'exemple de ces développements.

1. Einführung

Der Einsatz des Computers in der Elektrotechnik hat in den letzten 10 Jahren einen derartigen Umfang angenommen, dass gewisse Arbeiten, wie beispielsweise die optimale Auslegung von gedruckten Schaltungen, ohne den Rechner fast nicht denkbar sind. Zwei Hauptgründe können für die Entwicklung dieser computergestützten Arbeitsweise angeführt werden: Einerseits trägt die Ausbildung der Ingenieure an den höheren technischen Lehranstalten dem Bedürfnis nach Kenntnissen über den Computer und Informationen über dessen sinnvolle Benützung Rechnung. Immer mehr Ingenieure suchen und finden dementsprechend bei rechenintensiven Problemen den Zugang zum Elektronenrechner und fordern zweckdienliche Software an. Andererseits ist das Preis-Leistungsverhältnis in der elektronischen Datenverarbeitung dank der modernen Technologie derart günstig geworden, dass heute in kürzester Zeit, selbst auf Kleinrechnern, Arbeiten durchgeführt werden können, an welche man sich anfangs der sechziger Jahre überhaupt nicht herangewagt hat. Als Beispiel sei hier nur die Analyse einer nichtlinearen Schaltung, in der sich transiente Vorgänge abspielen, angeführt, wobei infolge von toleranzbehafteten Parametern die Monte-Carlo-Technik zur Anwendung kommen muss.

Die Software-Palette, die dem Elektro-Ingenieur heute zur Verfügung steht, ist reichhaltig. Die erwähnten Programme zur Schaltungsauslegung werden ergänzt durch solche zur optimalen Verdrahtung. Für die Analyse von digitalen und, in grösserem Umfang, von analogen Schaltungen werden laufend neue Systeme entwickelt, und schliesslich sind schon seit Jahren die Simulatoren kontinuierlicher Systeme für Aufgaben aus der Elektrotechnik im Gebrauch. Im nachfolgenden wird besonders auf die Netzwerkanalysenprogramme eingegangen, welche von Computer-Dienstleistungsunternehmen oder an Hochschulrechenzentren angeboten werden.

¹⁾ Einleitendes Referat der SEV-Informationstagung über «der Computer als Werkzeug der Elektronik» vom 23. Mai 1978. Der Sammelband aller Referate kann beim SEV, Abt. VVW noch bezogen werden (Fr. 40.- + Versandspesen). Der Tagungsbericht ist in Bull. SEV/VSE 69(1978)16, S. 893...894 enthalten.

2. Anforderungen an Analysenprogramme

Die Anforderungen, die an die Software gestellt werden, sind oft stark problembezogen und dementsprechend derart vielfältiger Natur, dass selbst die modernsten Mehrzweckprogramme nicht allen Ansprüchen genügen können. Der versierte Ingenieur wird sich deshalb auch im allgemeinen nicht nur auf die Benützung eines einzigen Programms beschränken, sondern von Fall zu Fall versuchen, das für die gestellte Aufgabe geeignetste System zu finden. Die Mehrzahl der nachfolgenden, häufigsten Anforderungen werden heute von mehreren Analysenprogrammen erfüllt.

a) *Benutzerfreundliche Sprache:* Die Codiersprache zur Beschreibung der Schaltungstopologie und der Elemente sollte auch von EDV-Laien innert weniger Stunden erlernt werden können. Die Instruktionen müssen selbsterklärend und auch für diejenigen verständlich sein, die das Programm nicht geschrieben haben. Befehle wie

```
RLAST, KN5 - GROUND = 5.6 KOHM  
START FREQ = 100  
POINTS PER DECADE = 20  
PRINT VC1, IR3
```

erfüllen diese Forderung. Ersatzschemas für die gebräuchlichsten Halbleiter (BJT, FET, MOSFET, Dioden) sollten wie die üblichen Schaltungselemente bereits vorprogrammiert vorhanden sein oder zumindest nach einmaliger Definition permanent in der Systembibliothek abgespeichert und nach Bedarf wieder aufgerufen werden können. Das letztere gilt auch für häufig gebrauchte Teilschaltungen, z.B. ein logisches Gatter oder einen Operationsverstärker, und führt auf eine erhebliche Reduktion des Programmieraufwands.

Die Benutzerfreundlichkeit eines Systems steht und fällt auch mit der zugehörigen Dokumentation, welche nicht nur die Syntax beschreibt, sondern auch Aufschluss über die verwendeten numerischen Verfahren mit deren Fallstricken und Limiten geben muss.

b) *Beliebig umfangreiche oder selektive Resultatausgabe:* Ein Analysenprogramm muss dem Benutzer nicht nur alle jene

Informationen liefern, die er durch Messungen im Labor erhalten kann, sondern ihm Resultate auch numerisch oder graphisch in wunschgemäss skalierter, selektiver Form auf Papier oder am Bildschirm ausgeben. Neben den üblichen Darstellungen von Zeitverläufen und Bodediagrammen wünscht man neuerdings im Zusammenhang mit Monte-Carlo-Analysen Streudiagramme, Histogramme, Envelopendarstellungen u.a.m. (Fig. 1).

c) *Einfache Änderung von Schaltungsparametern:* Soll eine Schaltung optimal ausgelegt werden, so wird man mehrfache Analysen mit wechselnden Parameterwerten durchführen wollen, ohne das Programm jeweils neu aufsetzen zu müssen. Wünschenswert sind auch Möglichkeiten zur Topologieänderung, u.a. zur Entfernung von Elementen oder Einfügung neuer Knoten und Zweige. Befehle wie

```
CHANGE, CIRCUIT, TOPOLOGY, REMOVE,
RLAST, CSTREU
CHANGE, CIRCUIT, TOPOLOGY, ADD LZUL,
A-BASIS = 20 MILLIHENRY
```

gehören bei einigen Systemen zum Standardwerkzeug.

d) *Automatische Genauigkeitskontrolle:* In Rechenprozessen, wie der Lösung von Gleichungssystemen, der Integration oder der Bestimmung von Polen und Nullstellen, können für schlecht gearbete Probleme Schwierigkeiten numerischer Art auftreten. Die dabei entstehenden Rechenfehler schaukeln sich oft auf, und die Resultate verlieren jeglichen Aussagewert. Der Programmbenützer muss in solchen Fällen gewarnt und, sofern dies überhaupt möglich ist, auf die Ursache der Fehler hingewiesen werden.

e) *Behandlung aktiver und nichtlinearer Schaltungen:* Obwohl in vielen Bereichen der Analogtechnik mit linearen Netzwerken gearbeitet wird, muss ein modernes Analysenprogramm, insbesondere im Zusammenhang mit Halbleiterschaltungen, in der Lage sein, Nichtlinearitäten zu behandeln. Oft werden letztere in Form eines FORTRAN-ähnlichen Ausdrucks beschrieben:

```
CSPERR, BASIS-COLL = CO*EXP(-VJDI/K),
```

sind durch Interpolation aus einer Tabelle zu entnehmen:

```
CTABEL, (0,3 PF), (1, 2.71 PF), (2, 2.46 PF), (4, 2.01 PF)
```

oder durch eine Polynomfunktion definiert:

```
G 15, 7 - 4, POLY (3), 6 - 2, 6E - 3, 0.025, 0.008, 0.002
```

Polynomkoeffizienten

In diesem Beispiel wird mittels eines Polynoms dritten Grades eine Stromquelle zwischen den Knoten 7 und 4 definiert, welche durch eine Spannung zwischen den Knoten 6 und 2 gesteuert ist.

f) *Verschiedene Analysenarten:* Zur Bestimmung von Arbeitspunkten, Transfercharakteristiken, Belastungen und Sensitivitäten benötigt man die *Gleichstromanalyse*. Die *Wechselstromanalyse* dient der Bestimmung von linearen Übertragungsfunktionen, Verstärkungseigenschaften, Frequenzgängen, Polen und Nullstellen, Verzerrungsprodukten, Vierpol- und N-Port-Parametern sowie von Rauschfaktoren. Eine *transiente Analyse* ist erwünscht, wenn man periodische oder aperiodische Prozesse in nichtlinearen Netzwerken beobachten oder Schaltvorgänge untersuchen will. Vielfach möchte man Netzwerkelemente mit Toleranzen versehen und prüfen, ob

eine Schaltung bei Abweichungen von den Parameter-Nominalwerten immer noch ihre Funktion erfüllt. Für eine Toleranzanalyse wird jeder Netzwerkparameter dementsprechend mit einer Wahrscheinlichkeitsverteilungsfunktion versehen und unter Einsatz von *Monte-Carlo-Verfahren* die Anzahl der Ausfälle gezählt bzw. die Ausbeute bestimmt. Aufgrund der Sensitivitätsberechnung oder der Monte-Carlo-Analyse kann oft auch eine Berechnung des schlimmsten Falls, d.i. eine sog. Worst-Case-Analyse, angeschlossen werden.

3. Verfügbare Mehrzweckprogramme

An vielen Hochschulen werden laufend neue Analysenprogramme entwickelt, welche aber oft auf spezifische, gerade aktuelle Probleme zugeschnitten sind. In Tab. I, die keinen Anspruch auf Vollständigkeit erhebt, sind die wichtigsten, momentan in der Schweiz allgemein zugänglichen Mehrzweck-Analysenprogramme aufgeführt. Dabei wurde das Programm ECAP-360 mit seinen Varianten PAN und GOCAP weggelassen, da es vielen der aufgeführten Forderungen nicht mehr entspricht.

Die Programme ASTAP [1]²⁾, LISA [2] und SIMELEC [3] stammen von einem Computer-Hersteller, SPICE-2 [4] wurde an der Stanford University, CORNAP [5] an der Cornell University entwickelt, SCEPTRE [6] entspringt einer gemeinsamen Entwicklung der U.S. Air Force und eines Computer-Herstellers und wurde an einer Universität weiterentwickelt [7]. NAP-3 [8] und ANP-3 [9] werden seit einiger Zeit in Dänemark erfolgreich eingesetzt. Weitere Programme, die in kürzlich erschienenen Gegenüberstellungen [10; 11] gute Qualifikationen erhalten haben, in der Schweiz bisher aber nicht installiert wurden, sind u.a. NET II, CIRCUS II, ISPICE und BELAC. Die Bewertung der verschiedenen Systeme erfolgte in diversen Publikationen einerseits aufgrund der Erfüllung der in Abschnitt 2 beschriebenen Anforderungen, darüber hinaus in bezug auf die topologischen Limiten, d.h. die Grösse der erlaubten Netzwerke, die Rechengeschwindigkeit, Genauigkeit und schliesslich den Speicherbedarf.

In Tab. I sind einige der wichtigsten Eigenschaften der ausgewählten Programme zusammengestellt, wobei zu vermerken ist, dass bei all diesen Systemen mit verschiedenartigen Erweiterungen versehene Versionen im Umlauf sind. Die Erweiterungen von SUPERSCEPTRE gegenüber SCEPTRE sind durch Einklammerung der entsprechenden Tabelleneinträge gekennzeichnet.

4. Numerische Methoden

Eine Anzahl von neueren numerischen Verfahren hat den Zeitbedarf von Netzwerkanalysen stark verringert und dadurch die computergestützte Schaltungsentwicklung wirtschaftlich gemacht. Insbesondere erzielte man grössere Fortschritte im Bereich der konzisen Formulierung von Netzwerkbeziehungen mittels topologischer Matrizen, bzw. mittels Tableaux, der Lösung von linearen und nichtlinearen Gleichungssystemen, bei der numerischen Integration von steifen Differentialgleichungen und bei der zeitsparenden Durchführung von Monte-Carlo-Analysen.

Nachfolgend wird eine kleine Auswahl dieser Methoden, die bereits in vielen Analysenprogrammen implementiert sind, kurz beschrieben.

²⁾ Siehe Literatur am Schluss des Aufsatzes.

4.1 Die Formulierung der Netzwerkgleichungen unter Benützung von Graphen

Mit Hilfe von topologischen Matrizen, welche sich auf gerichtete Graphen beziehen, lassen sich die Netzwerkgleichungen in eleganter Form beschreiben [12; 13]. Die Formulierung kann aufgrund von Zweig-Knoten-, Zweig-Maschen- oder Trennbündelmatrix erfolgen; auch gemischte Verfahren [13] sind in Gebrauch. Stellt Y die Zweig-Admittanzmatrix eines Netzwerkes dar, so lautet das Ohmsche Gesetz, d. h. die

Beziehung zwischen Elementströmen j und -spannungen v , in Matrixform

$$\underline{j} = \underline{Y} \underline{v} \quad (1)$$

Bezeichnet man die Knoteninzidenzmatrix mit A , so lässt sich das Kirchhoffsche Knotengesetz mittels

$$A^T \underline{i} = 0 \quad (2)$$

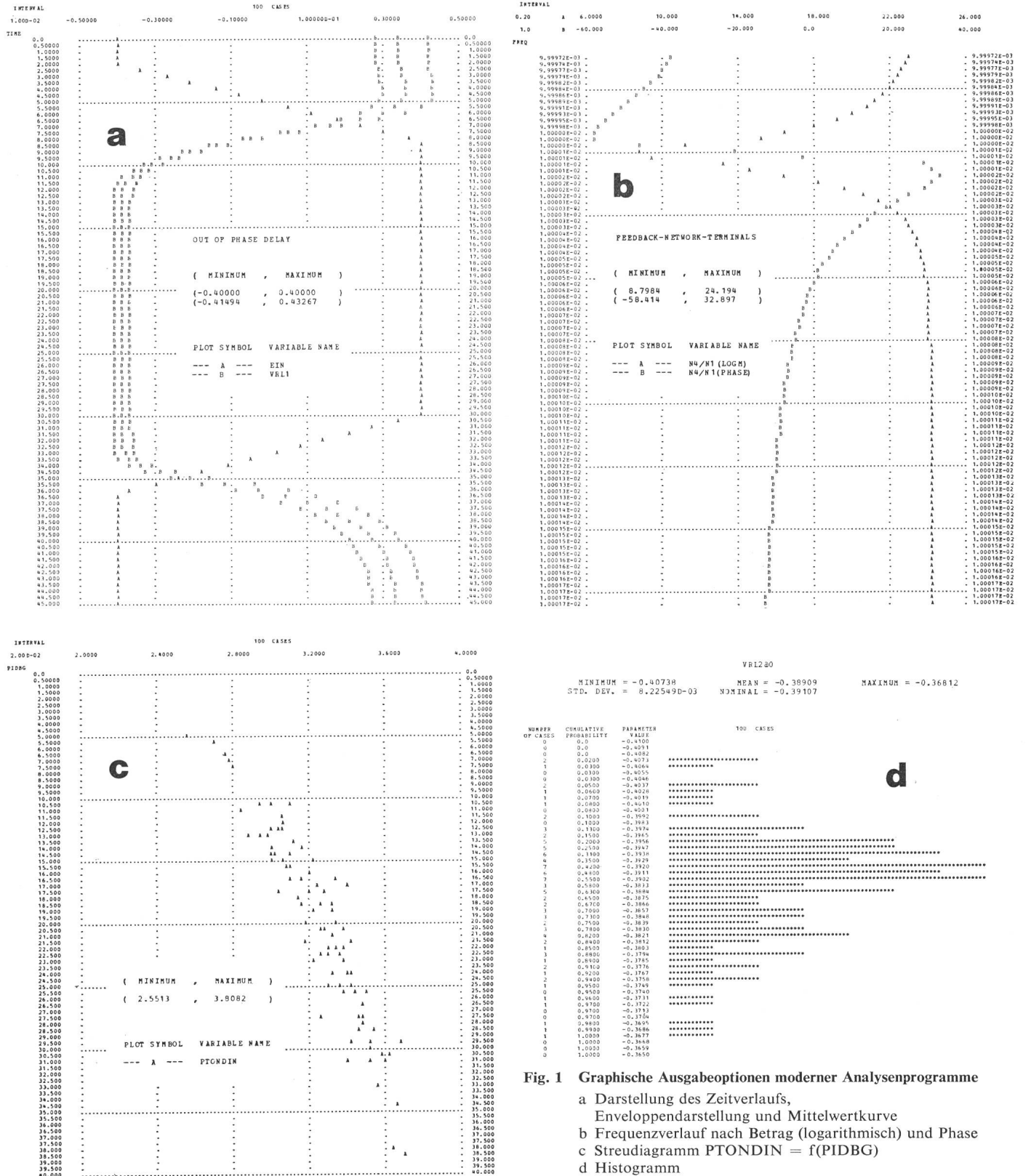


Fig. 1 Graphische Ausgabeoptionen moderner Analysenprogramme
a Darstellung des Zeitverlaufs, Enveloppendarstellung und Mittelwertkurve
b Frequenzverlauf nach Betrag (logarithmisch) und Phase
c Streudiagramm PTONDIN = f(PIDBG)
d Mittelwertkurve

Merkmale	ANP3	ASTAP	CORNAP	LISA	NAP 2	SIMELEC	(SUPER) SCEPTRE	SPICE 2
Topologische Limiten								
Knoten	40	n		50	50	1050	300	n
Zweige	100	n	70	125	200	1000	300	n
Analysenarten								
DC	x	x		x	x	x	x	x
AC	x	x	x	x	x	x	(x)	x
Transient	x	x	x	x	x	x	x	x
Sensitivität	x			x	x	x	(x)	x
Monte Carlo		x				x	(x)	
Temperatur					x			x
Ausfall		x						
Worst Case	x				x		(x)	
Nichtlinearitäten		x			x	x	x	x
Halbleitermodelle		x			x	x	x	x
Teilschaltungen		x				x	x	x
Aufruf von Unterprogrammen		x			x		x	
Parametermodifikation	x	x		x	x	x	x	
Topologische Änderungen	x			x				
Vierpolparameter		x		x				
N-Port-Parameter				x				
Übertragungsfunktionen			x	x			(x)	x
Pole, Nullstellen	x		x	x				
Rauschen				x	x			x
Tabellenbearbeitung		x			x	x	x	
Fourieranalyse					x	x	(x)	x

n: nur vom verfügbaren Speicherplatz abhängig

aufstellen, wobei \underline{i} für die Zweigströme (Elementströme minus Quellen) steht und der Index T die Transposition anzeigt (Fig. 2). Auch der Zusammenhang zwischen den Knotenspannungen \underline{u}' und den Zweigspannungen \underline{u} lässt sich mit der A -Matrix einfach darstellen

$$\underline{u} = A \underline{u}' \quad (3)$$

und schliesslich ergeben sich die Kirchhoffschen Maschengleichungen zu

$$C^T u = 0 \quad (4)$$

wenn C die Zweig-Maschenmatrix ist.

Bezeichnet man die unabhängigen Stromquellen mit \underline{I} und die Spannungsquellen mit \underline{U} und berücksichtigt die Zusammenhänge

$$\underline{v} = \underline{u} + \underline{U} \quad \text{und} \quad \underline{i} = \underline{j} - \underline{I} \quad (5, 6)$$

so ergibt sich aus den Gl. (1) bis (6)

$$(A^T YA) \cdot \underline{u}' = A^T \cdot (\underline{I} - YU) \quad (7)$$

mithin ein Gleichungssystem zur Bestimmung des Knotenspannungsvektors \underline{u}' . Das Produkt

$$Y^K = A^T YA \quad (8)$$

wird als Knoten-Leitwertmatrix bezeichnet, der Ausdruck

$$\underline{I}^K = A^T (\underline{I} - YU) \quad (9)$$

als äquivalenter Stromvektor. Damit ergibt sich aus Gl. (7)

$$Y^K \underline{u}' = \underline{I}^K \quad (10)$$

Dies ist eine der möglichen Darstellungen der Netzwerkgleichungen; andere Formen, welche die Zweig-Maschen- oder die Trennbündelmatrix verwenden, findet man oft in der Literatur, und sie sind bei der Implementierung von Netzwerk-Analysenprogrammen zum Einsatz gekommen.

In manchen Programmen (z.B. LISA) wird die Knotenadmittanzmatrix Y^K direkt aus der Schaltungsbeschreibung erstellt, dies ohne Umweg über topologische Matrizen. Ein neueres Programm [14] erweitert das beschriebene Konzept, das ursprünglich für lineare Schaltungen entwickelt wurde, auf nichtlineare Netzwerkbeziehungen unter Inkaufnahme eines iterativen Lösungsprozesses bei der Gleichstromanalyse und der wiederholten iterativen Lösung nach jedem Zeitschritt bei der transienten Analyse.

4.2 Lösung von linearen Gleichungssystemen

Das bekannteste Verfahren zur Lösung des linearen Gleichungssystems

$$A \underline{x} = \underline{b} \quad (11)$$

ist die Gaußsche Elimination. Besitzt die Matrix A die Dimension $N \times N$, so sind zur Bestimmung des unbekanntes Vektors \underline{x} insgesamt $z = N^3/3 - N^2/2 + N/6$ wesentliche Rechenoperationen (Multiplikationen, Divisionen) erforderlich. Die rechte Seite \underline{b} enthält die unabhängigen Quellen des Netzwerks. Will man deren Werte variieren, was z.B. bei der Bestimmung von Gleichstrom-Übertragungsfunktionen der Fall ist, so bedingt dies eine wiederholte Durchrechnung mit der gleichen Anzahl

	N = 20			N = 50			N = 200		
	K = 1	K = 5	K = 20	K = 1	K = 5	K = 20	K = 1	K = 5	K = 20
Gauss $K \cdot N^3/3$	<u>2 667</u>	13 333	53 333	<u>41 667</u>	208 333	833 333	$2.67 \cdot 10^6$	$133 \cdot 10^7$	$533 \cdot 10^7$
Inversion $N^3 + KN^2$	8 400	10 000	16 000	127 500	137 500	175 000	$8.04 \cdot 10^6$	$8.2 \cdot 10^6$	$8.8 \cdot 10^6$
L-U-Dec. } $N^3/3 + KN^2$ BI-Fact. }	3 067	<u>4 667</u>	<u>11 067</u>	44 167	<u>54 167</u>	<u>91 667</u>	<u>$2.71 \cdot 10^6$</u>	<u>$2.87 \cdot 10^6$</u>	<u>$3.47 \cdot 10^6$</u>

Unterstrichen ist der jeweils günstigste Wert.

von Operationen. Bei K verschiedenen rechten Seiten \underline{b} ergeben sich

$$z_k = KN^3/3 - KN^2/2 + KN/6 \tag{12}$$

wesentliche Operationen.

Für grosse Werte von K erhält man eine starke Verbesserung durch vorgängige Inversion der Matrix A und nachfolgende Multiplikation der Inversen mit den diversen Quellenvektoren \underline{b}_i

$$\underline{x}_i = A^{-1} \underline{b}_i \tag{13}$$

Die Inversion benötigt N^3 und die Matrix-Vektor-Multiplikation N^2 Operationen. Bei K rechten Seiten \underline{b} ergibt sich mit

$$z_k = N^3 + KN^2 \tag{14}$$

bereits für $k > 3$ ein Vorteil gegenüber der Gaußschen Elimination.

Mit drei neueren Verfahren [12], der L-U-Dekomposition, der Crout-Methode und der Bi-Faktorisierung kann der Rechenaufwand noch stark verringert werden. Alle drei Verfahren benötigen etwa die gleiche Anzahl von Operationen und einen fast identischen Speicheraufwand; eine kurze Erläuterung der L-U-Dekomposition soll hier als Beispiel genügen.

Entsprechend Fig. 3 wird die Matrix A in ein Produkt von zwei Dreiecksmatrizen zerlegt

$$L U = A \tag{15}$$

Die direkte Lösung von Gl. (11) durch Inversion ergäbe

$$\underline{x} = A^{-1} \underline{b} = (L U)^{-1} \underline{b} = U^{-1} L^{-1} \underline{b} \tag{16}$$

Statt dessen wird aus der Matrixgleichung

$$L \underline{y} = \underline{b} \tag{17}$$

ein Hilfslösungsvektor \underline{y}

$$\underline{y} = L^{-1} \underline{b} \tag{18}$$

ermittelt und letzterer dann als neue rechte Seite in ein weiteres zu lösendes Gleichungssystem eingeführt:

$$U \underline{x}^* = \underline{y} \tag{19}$$

Dessen Lösung ergibt sich zu

$$\underline{x}^* = U^{-1} \underline{y} = U^{-1} L^{-1} \underline{b} \tag{20}$$

Der Vergleich mit Gl. (16) zeigt, dass $\underline{x}^* \equiv \underline{x}$ ist und die Originalmatrixgleichung in zwei Schritten gelöst wurde. Die rechnerische Einsparung ergibt sich dadurch, dass zwar die Dekomposition nach Gl. (15) etwa $N^3/3$ Operationen, also gleichviel wie die Gaußsche Elimination benötigt, die nachfolgenden Lösungen der Gleichungssysteme Gl. (18) und (20) jedoch sehr einfach werden. Wegen der Dreiecksform der Matrizen L und U besteht dieser Lösungsprozess nur noch aus einem rekursiven Einsetzen bereits gefundener Komponenten der Lösungsvektoren \underline{y} bzw. \underline{x} , wobei in Gl. (17) von oben her, d.h. mit y_1 und in Gl. (19) von unten her, d.h. mit x_N begonnen wird. Die einmalige Lösung eines dreiecksförmigen Gleichungssystems erfordert etwa $N^2/2$ wesentliche Operationen, bei K verschiedenen rechten Seiten \underline{b} ergibt sich somit ein totaler Rechenaufwand von

$$z_k = N^3/3 + KN^2 \tag{21}$$

Operationen. In Tab. II ist eine Gegenüberstellung der drei erwähnten Verfahren für verschieden grosse Netzwerke mit Knotenzahlen N von 20, 50 bzw. 200 gemacht.

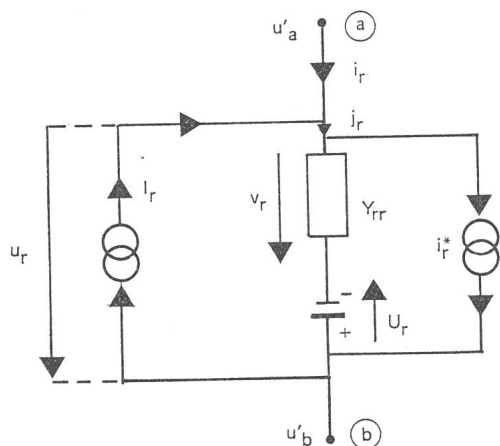


Fig. 2 Standard-Netzwerkzweig: r -ter Zweig zwischen den Knoten a und b

- u'_a, u'_b Knotenspannungen
- u_r Zweigspannung
- U_r unabhängige Spannungsquelle
- v_r Elementspannung
- i_r Zweigstrom
- j_r Elementstrom
- I_r unabhängige Stromquelle
- i_r^* abhängige (gesteuerte) Stromquelle
- Y_{rr} Selbstadmittanz, Eigenleitwert

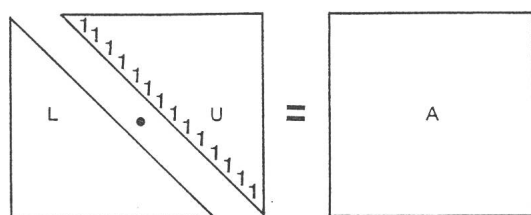


Fig. 3 L-U-Dekomposition

4.3 Lösung von nichtlinearen Gleichungssystemen

Zur Bestimmung des Arbeitspunktes einer elektronischen Schaltung, welche Halbleiter enthält, sowie zur Durchführung von transienten Analysen ist fast immer das Lösen von Systemen nichtlinearer Gleichungen erforderlich. Die in diesem Bereich bekanntesten Methoden [15] sind das Newton-Raphson-Verfahren in diversen Varianten, das Sekanten-Verfahren und die Gauss-Seidel-Methode. Letztere wird insbesondere bei der numerischen Behandlung betrieblicher und ökonomischer Modelle eingesetzt, die ersten beiden vorwiegend im Ingenieurbereich. Nachfolgend wird das Vorgehen bei einem modifizierten Newton-Raphson-Prozess kurz beschrieben.

Soll das nichtlineare Gleichungssystem

$$u_k(x_1, x_2, \dots, x_N) = 0 \quad k = 1 \dots N \quad (22)$$

bzw. in Vektorschreibweise $\underline{u}(\underline{x}) = 0$ gelöst werden, so überträgt man die aus dem eindimensionalen Fall (Fig. 4) bekannte Taylorreihe-Approximation auf den mehrdimensionalen Fall:

$$\underline{u}(\underline{x}^{(j+1)}) = \underline{u}(\underline{x}^{(j)}) + \left(\frac{\partial \underline{u}}{\partial \underline{x}} \right)_{\underline{x}^{(j)}} \cdot \underline{\Delta x}^{(j)} + \frac{1}{2} \left(\frac{\partial^2 \underline{u}}{\partial \underline{x}^2} \right)_{\underline{x}^{(j)}} \cdot (\underline{\Delta x}^{(j)})^2 + \dots \quad (23)$$

Darin stellt die Grösse $\frac{\partial \underline{u}}{\partial \underline{x}}$ die Jacobi-Matrix dar. Der Index j zeigt den j ten iterativen Schritt an.

Im linearen Fall verschwinden die zweiten Ableitungen, d.h. $\frac{\partial^2 \underline{u}}{\partial \underline{x}^2} = 0$, wodurch das Gleichungssystem in einem Schritt gelöst werden kann:

$$\underline{u}(\underline{x}^{(j+1)}) = 0$$

Aus Gl. (23) ergibt sich der dazu notwendige Schritt $\underline{\Delta x}^{(j)}$ zu

$$\underline{\Delta x}^{(j)} = - \frac{\underline{u}(\underline{x}^{(j)})}{\left(\frac{\partial \underline{u}}{\partial \underline{x}} \right)_{\underline{x}^{(j)}}} \quad (24)$$

Da im Nenner eine Matrix vorkommt, erfordert diese Berechnung die Inversion einer $N \times N$ -Matrix. Im allgemeinen, nichtlinearen Fall stellt dieser Schritt eine erste Näherung in Richtung der Nullstelle dar, und der einfache Newton-Raphson-Algorithmus lautet dementsprechend

$$\underline{x}^{(j+1)} = \underline{x}^{(j)} + \underline{\Delta x}^{(j)} = \underline{x}^{(j)} - \frac{\underline{u}(\underline{x}^{(j)})}{\left(\frac{\partial \underline{u}}{\partial \underline{x}} \right)_{\underline{x}^{(j)}}} \quad (25)$$

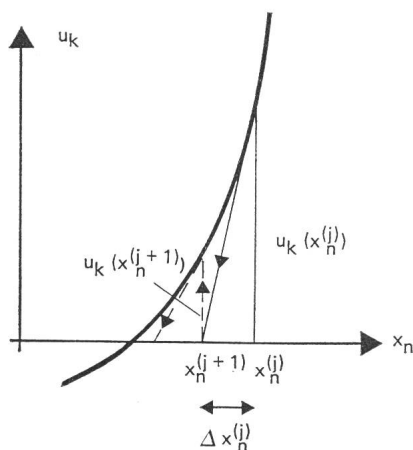


Fig. 4 Newton-Verfahren zur Lösung einer nicht-linearen Gleichung

Um ein Oszillieren des iterativen Prozesses zu verhindern bzw. die Konvergenz zu beschleunigen, wird in modifizierten Newton-Raphson-Verfahren der Schritt $\underline{\Delta x}^{(j)}$ durch einen passenden Faktor R korrigiert:

$$\underline{x}^{(j+1)} = \underline{x}^{(j)} + R \underline{\Delta x}^{(j)} \quad (26)$$

Die Wahl von R erfolgt derart, dass für jede Iteration das Skalarprodukt $y = \underline{u}^T(\underline{x}^{(j+1)}) \cdot \underline{u}(\underline{x}^{(j+1)})$ minimiert wird. In Fig. 5 sind einige mögliche Verläufe $y = f(R)$ aufgetragen. Das Minimum der Kurven erhält man durch einen eindimensionalen Suchprozess [16], was eine Anzahl von Funktionsbestimmungen für ausgewählte Werte von R bedingt. Gewöhnlich genügt es, den Kurvenverlauf durch eine quadratische oder kubische Parabel zu approximieren und hieraus analytisch den optimalen Wert R_{opt} zu bestimmen. In Ausnahmefällen sind aber aufwendigere Verfahren (z. B. Goldener Schnitt oder Fibonacci) erforderlich, um das Minimum zu lokalisieren. Im linearen Fall ergibt sich das Minimum $y = 0$ für $R = 1$. Man wird den iterativen Prozess abbrechen, sobald $\underline{u}^{(j+1)}$ einen vorgegebenen Fehlerwert unterschreitet.

4.4 Die Integration

In elektrischen Systemen ist es häufig, dass rasche und langsame Vorgänge nebeneinander ablaufen. Um eine genügend hohe Rechengenauigkeit zu erzielen war es üblich, den Integrationsschritt der kürzesten vorkommenden Zeitkonstanten anzupassen. Betrogen die Unterschiede zwischen kleinster und grösster Zeitkonstanten mehrere Zehnerpotenzen, so ergaben sich auf diese Weise sehr viele Integrations-schritte mit untragbar hohen Rechenzeiten, in extremen Fällen sogar eine Instabilität des Rechenprozesses. Neuere implizite Integrationsverfahren, mehrheitlich basierend auf Arbeiten von Gear [17], erlauben es, die Rechenschritte bedeutend grösser zu wählen und jeweils den momentanen Erfordernissen anzupassen. Insbesondere bei der Behandlung von Impulsschaltungen ist dies von Bedeutung, wenn in der Umgebung der Impulsflanken eine genügend hohe Genauigkeit beibehalten werden soll. Diese modernen Verfahren zur Behandlung von «steifen Systemen» erlauben es auch, den optimalen nächsten Integra-

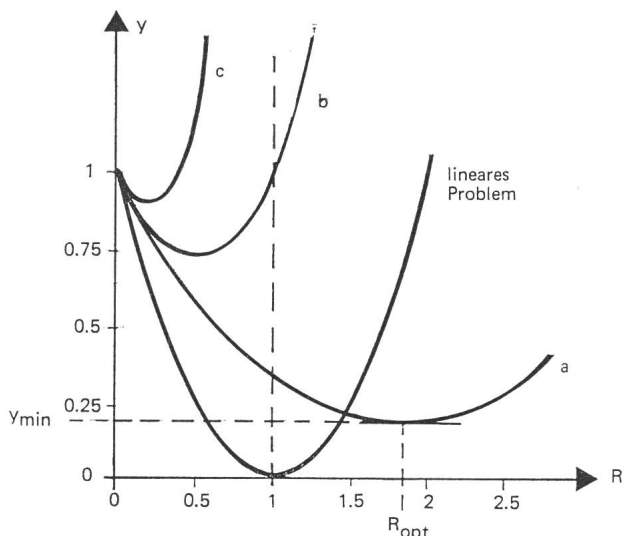


Fig. 5 Mögliche Verläufe $y = f(R)$ bei nichtlinearen und linearen Gleichungssystemen

tionsschritt jeweils auf Grund vorgegebener Fehlerkriterien zu ermitteln.

Aus Fig. 6 ist das Vorgehen beim Gear-Verfahren ersichtlich. Das Differentialgleichungssystem

$$\begin{aligned} \dot{Y}_i &= f_i(Y_1, Y_2, \dots, Y_m, t) & i &= 1 \dots s \\ 0 &= f_j(Y_1, Y_2, \dots, Y_m, t) & j &= s + 1 \dots m \end{aligned} \quad (28)$$

soll für den Zeitpunkt $t = t_n$ gelöst werden, unter der Voraussetzung Y_i und \dot{Y}_i für $i = 1 \dots m$ seien für alle vorangehenden Zeitpunkte $t_0, t_1 \dots t_{n-1}$ bekannt. Man legt zu diesem Zweck durch die letzten $k + 1$ Stützwerte ein Polynom von Grad k

$$Y_i(t) = P_i[Y_i(t_n), Y_i(t_{n-1}), \dots, Y_i(t_{n-k}), t] \quad (29)$$

wobei diese Funktionen mit Hilfe der Lagrange-Interpolationspolynome direkt angeschrieben werden können, ohne dass die Polynom-Koeffizienten (durch Lösung eines Gleichungssystems) explizit berechnet werden müssen:

$$Y_i(t) = \sum_{r=0}^k L_r \cdot Y_i(t_{n-r}) \quad (30)$$

$$\text{mit } L_r(t) = \frac{\prod_{\substack{j=0 \\ j \neq r}}^k (t - t_{n-j})}{\prod_{\substack{j=0 \\ j \neq r}}^k (t_{n-r} - t_{n-j})} \quad (31)$$

Der infolge einer endlichen Anzahl von Stützwerten entstehende Fehler wurde hier vernachlässigt.

Leitet man Gl. (29) nach der Zeit ab, so ergibt sich für $t = t_n$

$$\dot{Y}_i(t_n) = a_k \left[Y_i(t_n) + \sum_{r=1}^k b_{kr} \cdot Y_i(t_{n-r}) \right] \quad (32)$$

$$\text{mit } a_k = \sum_{j=1}^k \frac{1}{t_n - t_{n-j}} \quad (33)$$

$$b_{kr} = \frac{1}{a_k \cdot (t_n - t_{n-r})} \prod_{\substack{j=1 \\ j \neq r}}^k \frac{(t_n - t_{n-j})}{(t_{n-r} - t_{n-j})} \quad (34)$$

Setzt man Gl. (32) in die Beziehung Gl. (28) ein, so erhält man Gleichungen, in denen die Variablen $Y_1(t_n), Y_2(t_n), \dots, Y_m(t_n)$ im Zeitpunkt t_n unbekannt sind. Nimmt man alle Terme auf eine Seite, so ergibt sich die übliche Schreibweise für ein Gleichungssystem mit m Unbekannten:

$$F_1(Y_1, \dots, Y_m, t_n) = \left\{ \begin{array}{l} f_i(Y_1, \dots, Y_m, t_n) - a_k \left[Y_i(t_n) - \sum_{r=1}^k b_{kr} \cdot Y_i(t_{n-r}) \right] \\ \qquad \qquad \qquad i = 1 \dots s \\ f_j(Y_1, \dots, Y_m, t_n) \\ \qquad \qquad \qquad j = s + 1 \dots m \end{array} \right\} = 0 \quad (35)$$

Da Gl. (28) im allgemeinen Fall ein nichtlineares Differentialgleichungssystem darstellt, ist die Approximation nach Gl. (35) ein nichtlineares Gleichungssystem, das für jeden Zeitpunkt t_n neu gelöst werden muss.

Iterative Verfahren können stark beschleunigt werden, wenn man über gute Start-Näherungswerte für die Unbekannten verfügt. Im vorliegenden Fall legt man analog zu Gl. (30) ein Polynom durch die letzten $k + 1$ Stützpunkte vor dem interessierenden Zeitpunkt t_n :

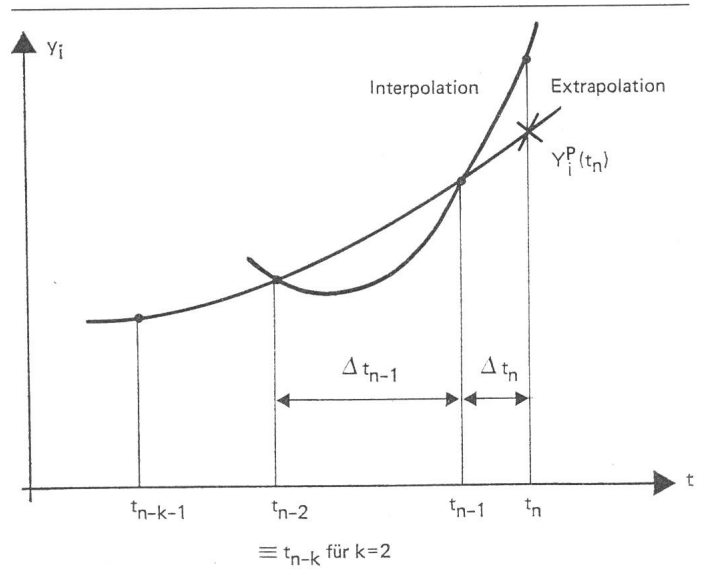


Fig. 6 Integration steifer Systeme mittels Polynom-Approximation

$$Y_i(t) = \sum_{r=0}^k L_r \cdot Y_i(t_{n-1-r}) \quad (36)$$

was unter Berücksichtigung von Gl. (31) für $t = t_n$ (Extrapolation!) auf

$$Y_i^P(t_n) = \sum_{r=0}^k d_{kr} \cdot Y_i(t_{n-1-r}) \quad (37)$$

$$d_{kr} = \prod_{\substack{j=0 \\ j \neq r-1}}^k \frac{(t_n - t_{n-1-j})}{(t_{n-r} - t_{n-1-j})} \quad (38)$$

führt. Damit besitzt man einen Satz von Voraussagewerten für den unbekanntem Vektor $Y(t_n)$, die iterativ verbessert werden.

Was die Wahl der Grösse k anbelangt, sind die Werte $k = 1 \dots 5$ gebräuchlich, in den meisten praktischen Fällen genügt jedoch $k = 1$ oder $k = 2$.

Gear-Verfahren, 1. Ordnung, $k = 1$: Die implizite Formel für die Ableitung nach Gl. (32) für den j -ten iterativen Schritt lautet

$$\dot{Y}_i^j(t_n) = \frac{1}{t_n - t_{n-1}} \left[Y_i^j(t_n) - Y_i(t_{n-1}) \right] \quad (39)$$

Der Voraussagewert für den ersten iterativen Schritt nach Gl. (36) wird

$$Y_i^P(t_n) = \frac{t_n - t_{n-2}}{t_{n-1} - t_{n-2}} Y_i(t_{n-1}) - \frac{t_n - t_{n-1}}{t_{n-1} - t_{n-2}} Y_i(t_{n-2}) \quad (40)$$

Er entspricht einer linearen Extrapolation. Es kann gezeigt werden, dass der Abschneidefehler sich sehr einfach darstellen lässt:

$$T_E = \frac{1}{t_n - t_{n-2}} \left[Y_i^*(t_n) - Y_i^P(t_n) \right] \quad (41)$$

Y_i^* ist der Konvergenzwert von Y_i nach Abbruch des iterativen Prozesses. Aufgrund der Kenntnis des Abschneidefehlers für jeden Schritt kann bei vorgegebenen absoluten und relativen Fehlerschranken für die einzelnen Integratoren jeweils der optimale nächste Integrations Schritt ermittelt werden.

5. Schlussfolgerung

Zwar entheben auch der schnellste Computer und die ausgefeiltste Software den Ingenieur nicht von seiner Aufgabe, ein Problem sauber zu formulieren und mögliche Lösungen aufgrund durchdachter Schaltungsentwürfe vor jeder Inanspruchnahme eines Rechners eingehend zu prüfen. Haben sich aber bei diesem Prozess erfolgversprechende Alternativen herausgeschält, so erlauben es die heute verfügbaren Analyseprogramme, eine Schaltung in kurzer Zeit eingehend zu testen, zu selektionieren und optimal auszulegen.

Literatur

- [1] ASTAP-advanced statistical analysis program. Program reference manual. IBM-form No. SH20-1118, Des Plaines (USA), 1973.
- [2] LISA-X linear system analysis program extended. User's manual. IBM-form No. GJ14-9602, Zürich, 1974.
- [3] SIMELEC. IBM Call-Fact 130-1, Stuttgart, 1977.
- [4] E. Cohen and D. O. Pedersen: SPICE2 user's guide. Version 2D.2. Los Angeles, University of California, Department of Electrical Engineering, 1976.
- [5] B. Tödtli: CORNAP Handbuch für das Gebiet der linearen Schaltungstechnik. Fides Rechnungszentrum Publikation No. 47. 2. Auflage. Zürich, Fides, 1972.
- [6] B. White: SCEPTRE user's manual. New Mexico, US Air Force Base, 1967.
- [7] J. C. Bowers, J. E. O'Reilly and G. A. Shaw: SUPER SCEPTRE user's guide. Tampa/Florida, University of South Florida, Electrical Engineering Department, 1974.
- [8] R. Molich-Pederson: ANP3 user's manual. Lyngby, Technical University of Denmark, Institut of Circuit Theory, 1972.
- [9] T. Rübner-Petersen: NAP2 nonlinear analysis program, users manual. Lyngby, Technical University of Denmark, Institut of Circuit Theory, 1973.
- [10] D. J. Blattner: Choosing the right programs for computer-aided design. Electronics 49(1976)9, p. 102...105.
- [11] G. W. Zobrist and J. C. Bowers: a survey of computer-aided design and analysis programs. Report No. TR AFAPL, TR 76-33 WPAFB. Tampa/Florida, University of South Florida, 1976.
- [12] A. Brameller, R. N. Allan and Y. M. Hamom: Sparsity. London, Pitman, 1976.
- [13] F. H. Branin: Computer methods of network analysis. Proc. IEEE 55(1967)11, p. 1787...1801.
- [14] C. W. Ho and D. A. Zein: ICD interactive circuit design program, user's guide. IBM Technical Report TR22, 1978.
- [15] P. Henrici: Elemente der numerischen Analysis. Hochschul-Taschenbücher 551. Mannheim, Bibliographisches Institut, 1972.
- [16] G. R. Walsh: Methods of optimization. London a.o., John Wiley, 1975.
- [17] C. W. Gear: Numerical initial value problems in ordinary differential equations. Englewood Cliffs, Prentice Hall, 1971.

Adresse des Autors

PD Dr. J. S. Vogel, Abteilung für technisch-wissenschaftliche Computeranwendungen, IBM-Schweiz, 8001 Zürich.