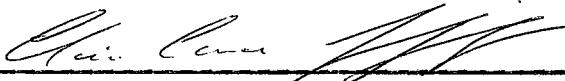


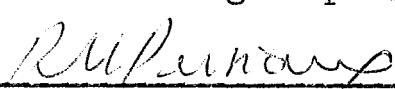
IMPLEMENTAÇÃO DE UM ALGORITMO DUAL PARA
OTIMIZAÇÃO DE SISTEMAS DECOMPONÍVEIS

Dilermando Ferreira Lopes Filho

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO
DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO
PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO
DO GRAU DE MESTRE EM CIÊNCIA (M. Sc.)

Aprovada por:


Clóvis Caesar Gonzaga - presidente


Ronaldo Cesar Marinho Persiano


Rodrigo A. Restrepo

RIO DE JANEIRO
ESTADO DA GUANABARA - BRASIL
AGOSTO DE 1972

A meus pais e a Sueli.

RESUMO

Este trabalho trata da otimização de sistemas decomponíveis. Um problema decomponível de programação matemática é proposto e sua solução obtida por coordenação dual. Dois conceitos básicos que aparecem em programação matemática de sistemas de grande porte são usados: linearização externa e relaxação.

Um algoritmo tipo plano cortante que permite o abandono de vínculos plenamente satisfeitos é obtido com base em resultados de teoria de dualidade. Sua aplicação ao problemas linearizado externamente resulta em um método convergente para solução do problema original. Um programa de computador foi desenvolvido com este propósito.

ABSTRACT

This work deals with the optimization of decomposable systems. A decomposable mathematical programming problem is proposed and its solution is achieved by means of dual coordination. Two basic concepts arising in large scale mathematical programming are used: outer linearization and relaxation.

A cutting plane algorithm without nested constraint sets is synthesized based on results from duality theory. Its application to the outer linearized problem results in a convergent method for the solution of the original problem. A computer program has been developed.

SUMÁRIO

<u>Capítulo I</u>	Introdução	
<u>Secção 1</u>	- Objetivo primário	1
<u>Secção 2</u>	- Forma organizacional associada a S	5
<u>Secção 3</u>	- Resolução do problema S	7
<u>Secção 4</u>	- Notação	10
 <u>Capítulo II</u>	O problema S e sua resolução por coordenação dual	
<u>Secção 1</u>	- O problema de tomada de decisão . .	13
<u>Secção 2</u>	- Resolução do problema S	16
<u>Secção 3</u>		
 <u>Capítulo III</u>	Estabilidade do problema S	
	Equivalência entre S e D	
	Algoritmo mestre de resolução de D	
<u>Secção 1</u>	- Introdução	23
<u>Secção 2</u>	- Estabilidade do problema S	24
<u>Secção 3</u>	- Equivalência entre S e D	27
<u>Secção 4</u>	- Linearização externa do problema D	28
<u>Secção 5</u>	- Algoritmo mestre para resolução do problema D	32
 <u>Capítulo IV</u>	Implementação do algoritmo mestre	
<u>Secção 1</u>	- Separador e mapeamentos cortantes .	37

<u>Secção 2</u>	- Algoritmo mestre plano cortante	43
<u>Secção 3</u>	- Versão implementada de AMPO	45
<u>Secção 4</u>	- Mecanismo de geração de cortantes	49

Capítulo V Implementação do algoritmo dual

<u>Secção 1</u>	- Introdução	55
<u>Secção 2</u>	- Resolução dos subproblemas D^i	56
<u>Secção 3</u>	- Adição-e-abandono de vínculos	60
<u>Secção 4</u>	- Fluxograma final	62

Apêndice A

<u>Secção 1</u>	- Teoria de dualidade	67
<u>Secção 2</u>	- Mapeamentos ponto-conjunto	71

Apêndice B

<u>Secção 1</u>	- Entradas e saídas do programa	73
<u>Secção 2</u>	- Mensagens de erro	75
<u>Secção 3</u>	- Dimensionamento	75
<u>Secção 4</u>	- Subrotinas	78
<u>Secção 5</u>	- Fluxogramas	81

<u>Bibliografia</u>	139
---------------------	-----------	-----

I INTRODUÇÃO

1. Objetivo primário

Consiste nosso objetivo primário em definir um problema de tomada de decisão, dar-lhe uma interpretação e então propor um algoritmo de resolução implementável, discutindo-lhe os aspectos computacionais.

Definir um problema de tomada de decisão, segundo Varaiya [43], significa identificar em um dado contexto os seguintes elementos:

- a - um conjunto de agentes, aos quais cabem as tomadas de decisão;
- b - para cada um destes agentes, um critério de preferência (por exemplo, uma função-objetivo);
- c - um conjunto de decisões viáveis associado a cada um destes agentes;
- d - uma estrutura de informação, que permita a cada agente realizar observações sobre o seu "meio ambiente";
- e - uma descrição ou modelo do sistema, por intermédio do qual é possível examinar as interações entre o meio ambiente e as decisões tomadas por um dado agente.

Ainda segundo Varaiya [43], o comportamento

de cada agente é definido por uma regra de decisão, que corresponde à sua resposta a uma dada observação (estímulo) efetuada. Podemos ainda, segundo Marschak e Radner [34], definir como forma organizacional uma particular escolha de uma estrutura de informação e um conjunto de regras de decisão, uma para cada agente.

Passamos agora a definir o nosso problema de tomada de decisão, identificando os seus diversos elementos constitutivos.

1 Problema S - Geoffrion [15]

$$\text{maximize} \sum_{i=1}^k f_i(x_i)$$

sujeito a:

$$x_i \in X_{ii} \quad (1 \leq i \leq k)$$

$$\sum_{i=1}^k g_{ii}(x_i) \leq b$$

onde ($1 \leq i \leq k$):

x_i - é o vetor de decisão, de dimensão n_i , associado ao agente i ;

$X_i \in R^{n_i}$ - é o conjunto de decisões viáveis associado ao agente i ;

$f_i: X_{ii} \rightarrow R$ - é uma função-critério com valores em R associada ao agente i ;

$g_i: X_{ii} \rightarrow R^m$ - é uma função-demanda com valores em R^m , associada ao agente i , expressando o uso de m recursos produtivos dados correspondente à decisão x_i ;

$b \in R^m$ - é um vetor de R^m representando as disponibilidades de m recursos produtivos dados.

S é a formulação geral de uma grande classe de problemas, cujo protótipo seria um modelo econômico competitivo. Uma primeira interpretação de S seria a seguinte: um agente central (gerente) deseja repartir uma certa quantidade de m recursos produtivos em disponibilidade (vetor b) entre k subsistemas, onde cada subsistema toma suas próprias decisões viáveis ($x_i \in X_i$, $1 \leq i \leq k$) referentes à utilização de sua quota de recursos produtivos, de modo a maximizar o "lucro" do sistema como um todo.

Apresentaremos agora dois exemplos de problemas de interesse prático, que podem ser colocados sob a forma do problema S .

2 Exemplo: um problema de exploração de recursos florestais [36].

Deseja-se estabelecer um plano plurianual, abrangendo um período de m anos, de derrubada de árvores plantadas em n áreas diferentes, maximizando-se a produção total. Sejam:

c_{ij} - cubagem total (m^3 de madeira) correspondente à área j ($1 \leq j \leq n$) quando a derrubada é efetuada no ano i ($1 \leq i \leq m$);

r_j - número de acres da área j ($1 \leq j \leq n$);

x_{ij} - fração da área j ($1 \leq j \leq n$) a ser derrubada no ano i ($1 \leq i \leq m$);

a_i - número total de acres que podem ser derrubados no ano i ($1 \leq i \leq m$).

O problema de exploração de recursos florestais poderá assim ser formulado:

$$\text{maximize} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

sujeito a:

$$\sum_{j=1}^n r_{ij} x_{ij} \leq a_i \quad (1 \leq i \leq m)$$

$$\sum_{i=1}^m x_{ij} \leq 1 \quad (1 \leq i \leq m)$$

$$x_{ij} \geq 0 \quad (1 \leq i \leq m) \quad (1 \leq j \leq n)$$

3 Exemplo: um problema de planejamento de produção de energia elétrica [35].

Deseja-se escalonar a produção de energia elétrica destinada ao consumo de n áreas conectadas entre si, minimizando-se o custo total de produção. Sejam:

c_{ji} - consumo total de energia elétrica correspondente à área j ($1 \leq j \leq n$);

x_{1j} - produção total de energia elétrica correspondente à área j ($1 \leq j \leq n$);

x_{2ji} - quantidade total de energia elétrica intercambiada entre a área j ($1 \leq j \leq n$) e outras áreas, pelas linhas de transmissão ($x_{2ji} \geq 0$ significa que a área j está recebendo energia elétrica proveniente de outras áreas);

$P_j(x_{1j}, x_{2ji})$ - dissipação de energia elétrica por perdas variadas na área j ($1 \leq j \leq n$);

$F_{ji}(x_{1j})$ - custo de produção de energia elétrica na área j ($1 \leq j \leq n$) por unidade de medida de energia elétrica.

O problema de escalonamento da produção de energia elétrica poderia assim ser formulado:

$$\text{minimize } \sum_{j=1}^n F_{ji}(x_{1j})$$

sujeito a:

$$\sum_{j=1}^n x_{2j} \leq 0$$

$$x_{1j} - x_{2j} - p_j(x_{1j}, x_{2j}) \geq c_j \quad (1 \leq j \leq n)$$

$$x_{1j} \geq 0 \quad (1 \leq j \leq n)$$

nenhuma restrição de sinal a x_{2j}
 $(1 \leq j \leq n)$

Uma outra ilustração interessante poderia ser vista, por exemplo, em [3].

2. Forma organizacional associada a S

Vemos pois que o modelo proposto para o problema S é bastante geral, o que permite associá-lo a diversas formas organizacionais. Dentre estas, adotaremos a seguinte:

S é o modelo de planejamento de uma companhia (sistema) consistindo de um gerente e k divisões (subsistemas), cada uma delas produzindo e vendendo determinados bens.

Dentro deste particular enfoque, o programa de produção da i-ésima divisão é representado pelo vetor de atividades x_i ($1 \leq i \leq k$), cada função $f_i(x_i)$ ($1 \leq i \leq k$) representando a parcela de contribuição da i-ésima divisão ao "lucro" total da companhia.

Podemos distinguir na formulação do problema S dois tipos de restrição. O primeiro, representado pela inequação vetorial

$$\sum_{i=1}^k g_i(x_i) \leq b$$

é uma restrição de recursos. Realmente, para realizar um certo programa de produção x_i , cada divisão necessitará receber do gerente uma certa quota $g_i(x_i)$ dos recursos produtivos comuns que são oferecidos a todas as divisões, cujas disponibilidades expressam-se pelo vetor b . O segundo tipo de restrição, representado pelos conjuntos viáveis X_i , expressam restrições impostas pela tecnologia de cada divisão, bem como levam em conta limitações de recursos produtivos próprios, particulares à divisão.

Dentro deste contexto, a função-objetivo do problema S expressa o imperativo de otimizar o funcionamento da companhia, no sentido de determinar-se um ponto de operação que maximize seu "lucro" total.

Tal interpretação adotada para S foi pela primeira vez dada por Baumol e Fabian [1], e também por Zschau [49]. Outras interpretações ligeiramente diferentes foram dadas por Gale [11], Kornai e Liptak [28], Malinvaud [31] e Weitzman [46].

Duas observações interessantes podem ainda ser feitas a respeito do problema S . Primeiro, é possível modificar o formato de S para abranger os casos em que há movimentação de produtos semi-acabados de uma para outra ou outras divisões para posterior acabamento, como o que ocorre, por exemplo, em linhas de produção. O exemplo (3) é uma ilustração desta possibilidade, representando-se a troca ou intercâmbio de energia elétrica entre as diversas áreas pela restrição

$$\sum_{j=1}^n x_{2j} \leq 0$$

Segundo, é possível também modificar o formato do problema S para abranger situações não-determinísticas. Uma discussão a respeito desta possibilidade pode ser encontrada, por exemplo, em Jennergren [24].

3. Resolução do problema S

Uma vez proposta e interpretada a forma organizacional correspondente ao problema S, resta sintetizar um algoritmo implementável, de convergência garantida, que se preste à resolução de S, uma vez explicitados todos os seus elementos constitutivos.

A síntese deste algoritmo é o resultado último de um estudo sistemático, que começa pela análise do tamanho e da estrutura do problema S, à luz dos conceitos de programação matemática de sistemas de grande porte [16], [30].

Não há o que se possa chamar de definição de sistemas (ou problemas) de grande porte, que aparecem frequentemente em planejamento econômico, em gestão empresarial e em engenharia. Sendo assim, contentar-nos-emos em afirmar que problemas de grande porte caracterizam-se não só pelo tamanho, porém, melhor, pelo tamanho em conjunção com sua estrutura não sendo possível resolver-se o problema de grande porte pelas técnicas usuais de programação matemática (por exemplo, programação linear e algoritmos implementáveis de programação não-linear).

A idéia básica da programação matemática de sistemas de grande porte reside em explorar convenientemente a estrutura do problema que se deseja resolver. Em uma primeira fase, chamada manipulação, busca-se reescrever o problema dado sob uma forma equivalente, mais adequada para a resolução, e que recebe o nome de programa mestre. Este programa mestre, em uma segunda fase, é reduzido a uma sequência de problemas implementáveis, de menor porte, pelo uso de diversas estratégias de solução, cuja aplicação fornece enfim um algoritmo implementável de resolução do problema originalmente proposto.

Estes conceitos - manipulações e estratégias de solução - unificados foram por Geoffrion [16], e resultam das contribuições individuais de dezenas de nomes importantes.

A característica mais importante da estrutura do problema S é que a função-objetivo e a inequação voto - rial que leva em conta a restrição de recursos produtivos co - muns são aditivamente separáveis por divisão, ou seja, estamos tratando de uma estrutura tipicamente multidivisional. O proble - ma S , mercê desta particularidade notável, classifica-se en - tão como decomponível, sendo possível resolvê-lo por decentraliza - ção.

Decentralização, com vistas à nossa forma or - ganizacional, significa atribuir a cada divisão da companhia uma larga autonomia, delegando autoridade, responsabilidades e poder de decisão aos níveis inferiores da hierarquia (gerentes das divisões). Aos níveis superiores desta mesma hierarquia (no caso, o gerente da companhia), é deixada apenas a tarefa de de - finir uma política global e a tomada de decisões importantes, que afetem a companhia como um todo [25]. Além do mais, a polí - tica de decentralização permite tratar o caso, mais geral, em que nenhum agente possui informações ou conhecimentos comple - tos. Essa observação será melhor explicada posteriormente.

Sob um ponto de vista mais imediatista, de - centralizar significa resolver o problema S desmembrando-o em k subproblemas de menor porte (um subproblema correspondendo a cada divisão), assumindo o gerente da companhia a tarefa de coordenar a formulação de tais subproblemas de maneira que qual - quer combinação de soluções ótimas dos mesmos seja uma solução ótima para S . Essa tarefa de coordenação é executada via um processo iterativo de troca de informações entre o gerente e as divisões.

A aplicação do conceito de decentralização

permite explorar em nosso proveito a estrutura e característica - cas dos subproblemas assim obtidos, de resolução comprovadamen - te mais imediata que S . Outra vantagem, sob o ponto de vista computacional, é a viabilidade de processamento em paralelo.

Mais especificamente, o programa mestre a ser obtido, isto é, a versão decentralizada de S , será obtida pela manipulação chamada dualização. O programa mestre, após a dualização, é novamente manipulado por linearização externa, linearizando-se totalmente sua função-objetivo pelo uso de hiperplanos-suporte, que, como veremos, aparecem naturalmente como resultado da dualização. Nesta fase o algoritmo de resolução do programa mestre é ainda conceitual, pois esta linearização total é obtida com o emprego de um número infinito de hiperplanos-suporte. Estas manipulações são apresentadas, discutidas e justificadas nos capítulos II e III.

Essa dificuldade é removida pela aplicação da estratégia de relaxação [7], o que constitui assunto para o capítulo IV. Relaxar o programa mestre linearizado externamente significará trabalhar, a cada iteração do algoritmo de resolução, com um número finito de hiperplanos-suporte. A cada iteração são gerados novos hiperplanos-suporte e, sob certas condições estabelecidas por Eaves e Zangwill [10], abandonados hiperplanos-suporte mais antigos, mantendo-se pois o tamanho do programa a ser otimizado dentro de limites computacionalmente aceitáveis. Essa combinação linearização externa/relaxação foi inicialmente sugerida por Kelley [26] e Dantzig e Madansky [8].

A coordenação desse processo iterativo de geração e abandono de aproximações lineares inspira-se em algoritmos tipo plano-cortante, primeiramente sugeridos por Kelley [26] e Cheney e Goldstein [5]. Importantíssimos são os resultados recentemente obtidos por Eaves e Zangwill [10], por nós aproveitados, e que garantem a convergência do algoritmo de resolução com o abandono de aproximações lineares. Esses aspectos são discutidos minuciosamente no capítulo IV.

Comentaremos ainda uma dificuldade adicional até à presente data ainda não superada, ou seja, que o processo de geração de hiperplanos-suporte - apresentado no capítulo V - é um processo essencialmente infinito. Com vistas a aplicações práticas esse processo de geração é então truncado, restando-nos tão-somente apontar a necessidade de compatibilizar convergência e truncamento.

As hipóteses assumidas no decorrer deste trabalho, explicitadas no capítulo II, são, basicamente, convexidade, compacidade, e diferenciabilidade.

No apêndice A são apresentados alguns resultados teóricos sobre teoria de dualidade transcritos de [17] e sobre mapeamentos ponto-conjunto [22], [2]. No apêndice B são apresentados e comentados os diagramas de bloco do algoritmo implementado de resolução do problema S, mais as listagens correspondentes a um problema de tomada de decisão por nós proposto e resolvido.

4. Notação

A notação empregada no decorrer deste trabalho é, sempre que possível, a de uso corrente. Outras notações aqui não comentadas são de compreensão imediata pelo simples exame do contexto onde situadas.

Os conjuntos são denotados por letras maiúsculas gregas ou latinas, as matrizes por letras maiúsculas latinas. Vetores são assimilados a matrizes-coluna, e representados por letras minúsculas latinas. O produto escalar de dois vetores é denotado por um ponto (.). Componentes ou famílias indexadas de vetores são indicadas por superíndices minúsculos latinos.

Os capítulos são enumerados por algarismos

romanos, e suas diversas secções por algarismos arábicos. Exemplos, definições importantes, teoremas e lemas são numerados se quencialmente por meio de algarismos arábicos colocados à esquerda do corpo do texto.

Referências ao próprio texto são dadas entre parênteses, citando-se o número correspondente e, se necessário também o capítulo. As referências bibliográficas, arroladas após o apêndice B, são referenciadas por seus respectivos números colocados entre colchetes.

II O PROBLEMA S E SUA RESOLUÇÃO VIA DUAL

1. O problema de tomada de decisão

Seja o seguinte problema de tomada de decisão S , determinístico, de estrutura multidivisional, constante de k divisões (subsistemas) sob a coordenação de um agente central [15] :

$$S \quad \text{maximize} \quad \sum_{i=1}^k f_i(x_i)$$

sujeito a:

$$x_i \in X_i \quad (1 \leq i \leq k)$$

$$\sum_{i=1}^k g_i(x_i) \leq b$$

onde ($1 \leq i \leq k$):

x_i - é o vetor de decisão, de dimensão n_i , associado ao agente i ;

$X_i \subseteq R^{n_i}$ - é o conjunto de decisões viáveis associado ao agente i ;

$f_i: X_i \rightarrow R$ - é uma função-critério com valores em R associada ao agente i ;

$g_i: X_i \rightarrow R^m$ - é uma função-demanda com valores em R^m , associada ao agente i , expressando o uso de m recursos produtivos dados correspondente à de-

cisão x_i ;

$b \in R^m$ - é um vetor de R^m representando as disponibilidades de m recursos produtivos dados.

Já vimos no capítulo I que S pode ser considerado como um problema de alocação de recursos, já que o objetivo do agente central (gerente) é distribuir quotas de m recursos produtivos comuns (cujas disponibilidades representam-se pelas m componentes do vetor b) entre os k subsistemas da companhia, visando-se maximizar o lucro da companhia como um todo. Foram também apresentadas várias interpretações e exemplos para o problema S , e, em linhas gerais, foi delineada a maneira de obtenção de um algoritmo implementado de resolução para S , através dos conceitos e técnicas de programação matemática de sistemas de grande porte.

Assumiremos doravante como válidas as seguintes hipóteses básicas associadas ao problema S , hipóteses estas que hão de acompanhar-nos no decorrer de todo este trabalho.

Hipóteses associadas a S :

- (a) os conjuntos X_i ($1 \leq i \leq k$) são convexos, compactos e não-vazios;
- (b) as funções f_i e cada componente das funções $-g_i$ ($1 \leq i \leq k$) são côncavas, diferenciáveis e explicitamente conhecidas em X_i ($1 \leq i \leq k$);
- (c) o problema S é consistente, isto é, seu conjunto de soluções viáveis é não-vazio;
- (d) não há indivisibilidades.

Cabe aqui um comentário a respeito das hipóteses supracitadas. Exigências de concavidade de funções-critério a serem otimizadas, e de convexidade e compactade de con-

juntos viáveis são típicas em economia [36]. A validade destas hipóteses em exemplos clássicos de otimização aplicados à economia, como por exemplo a teoria da escolha do consumidor e teoria da produção, é que permite eliminar dificuldades acarretadas pela existência de máximos locais. Hipóteses sobre convexidade e fechamento de conjuntos viáveis de produção aparecem, por exemplo, em modelos de economia competitiva, v.g. modelo de Arrow-Debreu-Mackenzie [29].

Exigimos diferenciabilidade por simultaneamente garantir-nos continuidade e aplicabilidade de métodos de otimização tipo direções viáveis, que aplicaremos à resolução dos subproblemas, conforme discutido no capítulo V. A combinação continuidade/compacidade permitir-nos-á ainda garantir que estes subproblemas possuem solução ótima, e que poderemos confundir supremos (infimos) e máximos (mínimos).

Talvez não seja muito realista exigir que se conheçam explicitamente as expressões matemáticas para as funções f_i e g_i ($1 \leq i \leq k$). Em um problema de alocação de recursos seria mais natural supor que estas entidades são conhecidas apenas implicitamente, isto é, cada divisão seria capaz de responder apenas a perguntas específicas concernentes a estas entidades, quando formuladas de maneira a terem um sentido no contexto da economia ("Quais seriam seus planos de produção sob condições tais e tais?"), sendo ainda necessário proceder-se a estimativas dos valores locais destas mesmas funções, de modo a ainda permitir a aplicabilidade de métodos tipo direções viáveis.

Admitimos conhecimento explícito apenas como hipótese simplificativa. De fato, nossa abordagem do problema de alocação de recursos proposto poderia ser modificada de modo a tratar do caso, mais geral, de conhecimento implícito, ou mesmo ainda problemas de alocação de recursos formulados em termos de funções-utilidade. Citamos, nesta linha de desenvolvimento,

[18] e [19].

Finalmente, cumpre lembrar que há problemas que, embora não possam ser encarados como problemas de alocação de recursos, mesmo assim são estruturalmente idênticos a S, e portanto passíveis de receberem tratamento idêntico.

2. Resolução do problema S

Já identificamos na Introdução a estrutura do problema S como sendo do tipo multidivisional, sendo a função-critério e as restrições do sistema, como um todo, linearmente separáveis nas variáveis de decisão de cada divisão. Vimos também que esta característica estrutural permite-nos buscar resolver S via decentralização, cujas vantagens já foram apontadas em (I.3).

É possível, pois, resolver S a nível de subsistemas (divisões), coordenando adequadamente a otimização de cada um destes subsistemas. Passamos agora a apresentar e interpretar as técnicas de coordenação aplicáveis à resolução do problema S.

Coordenação via primal:

A grande dificuldade na resolução do problema S, conforme formulado no item anterior, reside no fato de que a companhia (sistema) opera sob uma política de limitação dos diversos recursos produtivos que são oferecidos às divisões (subsistemas). De fato, caso existisse essa restrição, ou seja, se a companhia dispusesse de quantidades ilimitadas de recursos produtivos, o problema S seria resolúvel imediatamente por uma série de k subproblemas absolutamente desvinculados, do tipo:

$$\begin{aligned} & \text{maximize } f_i(x_i) \\ & x_i \in X_i \end{aligned}$$

para $1 \leq i \leq k$.

Admitindo que a companhia opera sob um regime de limitação de recursos produtivos, vamos examinar a coordenação primal proposta para resolução de S, após o que examinaremos a coordenação dual.

- (a) o gerente destina a cada divisão uma certa quota de recursos produtivos $y_i \in R^m$ ($1 \leq i \leq k$);
- (b) cada divisão, fazendo uso da quota a ela destinada, maximiza seu lucro, isto é, resolve o subproblema:

$$\text{maximize } f_i(x_i)$$

sujeito a:

$$x_i \in X_i$$

$$g_i(x_i) \leq y_i$$

para $1 = i = k$, informando ao gerente como alterar-se-ia este valor ótimo assim computado, para um aumento ou diminuição (no sentido marginal) da quota y_i ($1 \leq i \leq k$) de recursos produtivos a ela destinada;

- (d) com base nas informações supridas pelas divisões, o gerente reavalia iterativamente o valor das quotas de recursos produtivos distribuídos às divisões, até que as alterações (no sentido marginal) no valor das quotas não sejam mais efetivas no sentido de que não mais contribuam para aumentar o lucro da companhia.

É preciso impor uma restrição adicional que diz respeito exclusivamente ao gerente, isto é, exigir-se-á que

sejam permanentemente respeitadas as limitações de recursos produtivos da companhia e que todos os subproblemas sejam consistentes:

$$(y_i) \in Y \Leftrightarrow \left\{ \begin{array}{l} \{y_i | y_i \in R\} \\ \sum_{i=1}^k y_{ii} \leq b, g_i(x_i) \leq y_{ii}, \\ \text{para algum } x_i \in X_i, 1 \leq i \leq k \end{array} \right.$$

Essa coordenação via primal corresponde a uma manipulação de S via projecção, isto é, parametrizamos S por projeção do mesmo no espaço Y das variáveis y_i ($1 \leq i \leq k$).

O programa mestre assim obtido, equivalente ao problema S [15], será escrito como:

$$P \quad \text{maximize} \sum_{i=1}^k v_i(y_i)$$

sujeito a:

$$y_i \in Y \quad (1 \leq i \leq k)$$

onde definimos ($1 \leq i \leq k$):

$$P_y^i \quad v_i(y_i) \triangleq \underset{x_i \in X_i}{\text{máximo}} \{ f_i(x_i) \mid g_i(x_i) \leq y_i \}$$

A versão correspondente a P quando o problema S é completamente linear encontra-se em [49].

O programa mestre P pode ser resolvido por aplicação do princípio de decomposição de Dantzig e Wolfe [6], ou como sugerido por Geoffrion em [15]. Em particular citamos Campos [4], que resolve P pelo método de direções viáveis.

A coordenação via primal permite que os algoritmos implementados de resolução de P façam uso de uma solução viável inicial conhecida para S. Soluções viáveis para o

problema S são obtidas a cada iteração do algoritmo, o que é vantajoso se desejarmos truncar, em um dado momento, o processo iterativo.

Coordenação via dual:

A inserção da teoria de dualidade em programação não-linear teve seu início com o teorema clássico de minimax de von Neumann [45] e foi pela primeira vez explicitamente enunciada por Gale, Kuhn e Tucker [13]. Somente em data mais recente a teoria de dualidade começou a ser empregada em programação não-linear, iniciando-se por programação quadrática [9].

A idéia básica da coordenação via dual surge como uma bem sucedida extensão da teoria de preços em uma economia de mercado. Sugeriu-se então que a decentralização poderia ser levada a cabo criando-se uma economia em miniatura, com seu próprio sistema de preços, dentro da companhia [33], [21]. A descoberta de que os multiplicadores de Lagrange podem ser interpretados como preços levou a considerar a possibilidade de utilizar o novo valor informacional neles descoberto como um instrumento a serviço da decentralização. Citamos [11], [12], [27].

Seguindo esta linha de raciocínio, vamos examinar o seguinte tipo de coordenação via dual proposto para resolução do problema S :

- (a) o gerente estipula preços λ_j ($1 \leq j \leq m$) para os diversos recursos produtivos que serão colocados à disposição das k divisões, e formula a cada divisão a seguinte pergunta: "Qual a quota de recursos produtivos que desejam adquirir, a tais e tais preços, de modo a maximizar o lucro líquido da divisão?"
- (b) para responder a esta pergunta cada divisão resolve o seguinte problema ($1 \leq i \leq k$):

$$\underset{x_i \in X_i}{\text{maximize}} \quad f_i(x_i) - \lambda \cdot g_i(x_i)$$

onde $\lambda \triangleq (\lambda_1, \lambda_2, \dots, \lambda_m)$ é o vetor de preços anunciado pelo gerente. Cada divisão informa então ao gerente qual a quota de recursos produtivos que deseja adquirir aos preços vigentes;

- (c) com base nas informações supridas pelas divisões, o gerente reavalia iterativamente o vetor de preços até que as demandas totais das divisões:

$$\sum_{i=1}^k g_i(x_i)$$

não ultrapassem as limitações de recursos produtivos cujas disponibilidades são representadas pelo vetor b . Os recursos produtivos porventura em excesso são oferecidos gratuitamente (isto é, a preço zero) às divisões.

Essa coordenação via dual corresponde a uma manipulação do problema S via dualização, obtendo-se o seguinte programa mestre [15] :

$$6 \quad D \quad \underset{\lambda}{\text{minimize}} \quad \sum_{i=1}^k w_i(\lambda) + \lambda \cdot b$$

onde definimos ($1 \leq i \leq k$):

$$7 \quad D_\lambda^i \quad w_i(\lambda) \triangleq \underset{x_i \in X_i}{\text{supremo}} \quad f_i(x_i) - \lambda \cdot g_i(x_i)$$

A equivalência entre S e D será estabelecida no capítulo III, fazendo-se uso dos resultados fornecidos pela teoria de dualidade. O programa mestre D assim definido é que será objeto de estudo desta tese.

Uma desvantagem acarretada pelo emprêgo de coordenação via dual é que não nos podemos beneficiar do eventual conhecimento de uma solução viável inicial de S , e que, tipicamente, a resolução de D só fornecerá uma solução viável de S ao cabo do processo iterativo. Neste caso, a única solução viável de S que temos certeza seja obtida é a própria solução ótima procurada. Ou seja, não há garantias de que sejam preservadas as propriedades de viabilidade no decorrer do processo iterativo. Todavia, ao truncarmos o processo iterativo, após um número razoável de iterações, é de se supor que a solução quase-ótima assim obtida não apresente desvios marcantes no que tange à viabilidade, desta maneira sendo um resultado aproveitável em termos práticos.

Tanto na coordenação via primal, quanto na coordenação via dual, o processo de coordenação pode ser encarado como um diálogo estruturado entre os dois níveis hierárquicos da nossa forma organizacional: o gerente (nível superior) e as divisões (nível inferior), com tomadas de decisão e trocas de informação de parte a parte. Ambos os tipos de coordenação, conforme formulados (P , D) permitem-nos ainda tirar uma conclusão interessante: em qualquer caso o gerente prescinde totalmente do conhecimento das funções f_i , g_i e dos conjuntos viáveis X_i , isto é, o gerente não precisa saber do "modus operandi" de cada divisão: os subproblemas P_y^i e D_λ^i são resolvidos totalmente a nível de divisão, bastando-lhe tão-somente conhecer as disponibilidades de recursos produtivos da companhia (vetor b) e ser informado dos valores locais (supridos pelas divisões) no decorrer do processo iterativo.

Encerrando estes comentários, apontamos ainda outras políticas de decentralização aplicáveis à resolução do problema S , a saber, [39], [21], [46].

III ESTABILIDADE DO PROBLEMA S
EQUIVALÊNCIA ENTRE S E D
ALGORITMO MESTRE DE RESOLUÇÃO DE D

1. Introdução

Para estabelecer a equivalência entre o problema S (1) e o problema D (6), faremos uso dos resultados obtidos por Geoffrion [17], baseando-se apenas em resultados de análise convexa.

Em linhas gerais, apresentaremos preliminarmente alguns conceitos básicos, em particular a definição de uma certa função-perturbação associada ao problema S, cujas importantes propriedades serão examinadas. Com o auxílio desta função-perturbação definiremos o que seja estabilidade do problema S. O requisito de estabilidade figurará como hipótese básica do teorema de dualidade forte, o qual estabelecerá formalmente a equivalência entre S e D.

Uma vez estabelecida esta equivalência, mostraremos que o problema D pode ser manipulado por linearização externa. Esta linearização externa, obtida pelo emprêgo de hiperplanos-suporte, permitir-nosá aproximar a função-objetivo do problema D por uma outra função-objetivo, seccionalmente linear, em térmos da qual será formulado o algoritmo mestre de resolução do problema D.

Será útil reapresentar os problemas S e D, conforme anteriormente definidos:

$$S \quad \text{maximize} \quad \sum_{i=1}^k f_i(x_i)$$

sujeito a:

$$x_i \in X_i$$

$$\sum_{i=1}^k g_i(x_i) \leq b$$

$$D \quad \text{minimize} \quad \sum_{i=1}^k w_i(\lambda) + \lambda \cdot b$$

sujeito a:

$$\lambda \in \Lambda \triangleq \{ \lambda \in \mathbb{R}^m \mid \lambda \geq 0 \}$$

onde a função $w_i : \Lambda \rightarrow \mathbb{R} \cup \{+\infty\}$ é calculada como ($1 \leq i \leq k$):

$$D_\lambda^i \quad w_i(\lambda) \triangleq \supremo_{x_i \in X_i} [f_i(x_i) - \lambda \cdot g_i(x_i)]$$

Admitindo válidas as hipótese apresentadas em (II.1), e fazendo uso de resultados a serem apresentados oportunamente, podemos adiantar que garantiremos a finitude das funções w_i em seus respectivos domínios de definição, e que poderemos assimilar o supremo ao máximo. Da mesma forma, podemos definir o valor ótimo do problema S como o valor de sua função-objetivo calculado para uma solução ótima de S. O valor ótimo do problema D é definido de maneira idêntica.

2. Estabilidade do problema S

O conceito de estabilidade, apresentado pela primeira vez por Gale [12], surge naturalmente quando se estuda

a variação do valor ótimo do problema S em função de alterações introduzidas em seu "segundo membro". As definições seguintes explicarão melhor o que temos em mente.

Definição: o problema perturbado SP associado ao problema S é:

$$8 \quad \text{SP} \quad \text{maximize} \quad \sum_{i=1}^k f_i(x_i)$$

sujeito a:

$$x_i \in X_i \quad (1 \leq i \leq k)$$

$$\sum_{i=1}^k g_i(x_i) - b \leq p$$

onde o vetor $p \in R^m$ é chamado de vetor perturbação.

Definição: o conjunto de perturbações viáveis é o conjunto de valores assumidos pelo vetor perturbação tais que o problema perturbado SP é consistente, isto é:

$$9 \quad \Pi \triangleq \left\{ p \in R^m \mid (\exists x_i \in X_i) (1 \leq i \leq k) \left(\sum_{i=1}^k g_i(x_i) - b \leq p \right) \right\}$$

O conjunto Π é necessariamente não-vazio, pois $0 \in \Pi$, uma vez admitida a consistência do problema S, hipótese esta apresentada em (II.1).

Definição: a função-perturbação $v: \Pi \rightarrow R$ associada ao problema perturbado SP é calculada como:

$$10 \quad v(p) \triangleq \supremo \quad \sum_{i=1}^k f_i(x_i)$$

sujeito a:

$$x_i \in X_i$$

$$\sum_{i=1}^k g_i(x_i) - b \leq p$$

$$p \in \mathbb{P}$$

Observamos que a função-perturbação acima está bem definida, uma vez que, para qualquer perturbação viável p o valor correspondente $v(p)$ é finito, devido às hipóteses feitas em (II.1) de compacidade dos conjuntos X_i ($1 \leq i \leq k$) e continuidade das funções f_i ($1 \leq i \leq k$).

A função-perturbação associa a cada perturbação viável no "lado direito" do problema S um valor ótimo correspondente ao problema perturbado SP. Para $p=0$, $v(0)$ é o valor ótimo do problema S.

A função-perturbação assim definida é convexa, e não decrescente [17], sendo uma construção de fundamental importância na demonstração da equivalência entre S e D.

Definição: o problema S é dito estável se $v(0)$ é finito e se:

11

$$\lim_{\theta \rightarrow +0} \frac{v(0) - v(\theta y)}{\theta \|y\|} < \infty \quad (\forall y \neq 0)$$

A definição acima [17] independe da particular norma usada. Todas as condições de qualificação de vínculos para o problema S implicam em estabilidade do mesmo, por exemplo a condição de qualificação de Slater [32].

O limite assim definido é o negativo da derivada direcional de v na direção da perturbação y . Uma vez satisfeita a condição de estabilidade, isso significa que o valor ótimo do problema S é finito e que nenhuma perturbação no "lado direito" do mesmo fará com que seu valor ótimo varie com in-

clinação infinita.

3. Equivalência entre S e D

Poderemos agora estabelecer a equivalência formal entre os problemas S e D, enunciando o teorema da dualidade forte, cuja demonstração consta do apêndice A. Este teorema foi demonstrado por Geoffrion [17] fazendo uso da definição e propriedades da função-perturbação v associada ao problema S, conforme definida no item anterior.

12 Teorema: (dualidade forte)

se o problema S (1) é estável (11), então:

- (a) o problema D possui uma solução ótima;
- (b) os valores ótimos dos problemas S e D são iguais;
- (c) λ^* é uma solução ótima do problema D se e somente se $-\lambda^*$ é subgradiente de v em $y=0$;
- (d) qualquer solução ótima λ^* do problema D caracteriza o conjunto de todas as soluções ótimas (caso existam) do problema P como soluções dos subproblemas:

Nota: $-\lambda^*$ é subgradiente de v em $y=0$ se $v(0)$ é finito e se:

$$v(y) \geq v(0) - \lambda^* \cdot y$$

para todo $y \in \mathbb{R}^m$ (ver apêndice A).

$$\text{maximize} \sum_{i=1}^k [f_i(x_i) - \lambda^* g_i(x_i)]$$

sujeito a:

$$x_i \in X_i \quad (1 \leq i \leq k)$$

que satisfazem:

$$\sum_{i=1}^k g_i(x_i) \leq b$$

$$\lambda^* \cdot \left(\sum_{i=1}^k g_i(x_i) - b \right) = 0$$

As conclusões (a), (b) e (d) do teorema da dualidade forte validam pois a resolução do problema S por ordenação via dual. Em especial, a conclusão (b) elimina a possibilidade da existência de uma "brecha de dualidade" entre os valores ótimos dos problemas S e D. É interessante ainda notar que é possível, pela conclusão (d), determinar o conjunto de todas as soluções ótimas (caso existam) do problema S a partir do conhecimento de qualquer uma das soluções ótimas do problema D. A conclusão (c) revela a conexão existente entre o conjunto de soluções ótimas do problema D e a função-perturbação v (10) associada ao problema S. Ainda essa mesma conclusão permite interpretar o conjunto de soluções ótimas do problema D como o conjunto de multiplicadores ótimos de Kuhn e Tucker para o problema S.

4. Linearização externa do problema D

O problema D (6) não é de resolução imediata, já que sua função-objetivo não é explicitamente conhecida como função de seu argumento λ . Isso porque as funções w_i (7) também não são explicitamente conhecidas como função de seu argumento λ , podendo apenas ser calculadas ponto a ponto.

Veremos em seguida que cada uma das funções w_i ($1 \leq i \leq k$), que se pode facilmente demonstrar serem convexas, pode ser reformulada como a envoltória de todos os seus suportes lineares, de onde surge a idéia de aproxima-las por linearização externa, fazendo uso de um número finito de seus suportes lineares.

O efeito global dessa manipulação será, pois introduzir uma função seccionalmente linear, explicitamente conhecida, definida em Λ e com valores em R , que aproxima a função-objetivo do problema D. Essa aproximação assim introduzida é satisfatória sob o ponto de vista computacional e, sob condições a serem estabelecidas no capítulo IV, não compromete as propriedades de convergência do algoritmo implementado de resolução do problema S que nos propomos construir.

O teorema que se segue mostra que hiperplanos-suporte a cada uma das funções w_i ($1 \leq i \leq k$) aparecem naturalmente como subprodutos da resolução dos correspondentes subproblemas D_{λ}^i (7).

13 Teorema: seja $\bar{\lambda} \in \Lambda$. Então $D_{\bar{\lambda}}^i$ admite pelo menos uma solução $\bar{x}_i \in X_i$ com $w_i(\bar{\lambda})$ finito, para $1 \leq i \leq k$. Nestas condições $-g_i(\bar{x}_i)$ é subgradiente de w_i em $\bar{\lambda}$.

A demonstração deste teorema baseia-se no seguinte resultado, provado em [32] :

14 Lema: uma função Θ semicontínua superiormente, definida em um conjunto compacto $\Gamma \subset R^n$, é limitada superiormente e atinge em Γ o valor:

$$\bar{\Theta} = \sup_{x \in \Gamma} \Theta(x)$$

Demonstração: a existência de $\bar{x}_i \in X_i$ é garantida pelas hipóteses

apresentadas em (II.1), com a aplicação do lema anterior. Portanto, $D_{\bar{\lambda}}^i$ ($1 \leq i \leq k$) admite solução ótima para qualquer $\bar{\lambda} \in \mathbb{R}^m$, em particular para todo $\bar{\lambda} \in \Lambda$. A conclusão final é obtida com as seguintes desigualdades:

$$\begin{aligned} w_i(\lambda) &= \sup_{x_i \in X_i} [f_i(x_i) - \lambda \cdot g_i(x_i)] \geq \\ &\geq f_i(\bar{x}_i) - \lambda \cdot g_i(\bar{x}_i) = \\ &= f_i(\bar{x}_i) - \bar{\lambda} \cdot g_i(\bar{x}_i) - g_i(x_i) \cdot (\lambda - \bar{\lambda}) = \\ &= w_i(\bar{\lambda}) - g_i(x_i) \cdot (\lambda - \bar{\lambda}) \end{aligned}$$

pela própria definição de w_i . Logo:

$$w_i(\lambda) \geq w_i(\bar{\lambda}) - g_i(x_i) \cdot (\lambda - \bar{\lambda}) \quad (\forall \lambda \in \Lambda)$$

A demonstração está completa.

15 Teorema: a função w_i ($1 \leq i \leq k$) coincide em Λ com a envoltória de todos os seus suportes lineares calculados em Λ .

Demonstração: aplicando-se os resultados obtidos do teorema anterior, poderemos escrever:

$$w_i(\lambda) \geq w_i(\bar{\lambda}) - g_i(\bar{x}_i) \cdot (\lambda - \bar{\lambda})$$

para todo $\lambda, \bar{\lambda} \in \Lambda$, onde $\bar{x}_i \in X_i$ é uma solução ótima de $D_{\bar{\lambda}}^i$ (7). Tomando-se o supremo desta desigualdade sobre todo $\bar{\lambda} \in \Lambda$, temos:

$$w_i(\lambda) \geq \sup_{\bar{\lambda} \in \Lambda} [w_i(\bar{\lambda}) - g_i(\bar{x}_i) \cdot (\lambda - \bar{\lambda})]$$

para todo $\lambda \in \Lambda$. Uma vez que podemos tomar $\lambda = \bar{\lambda}$, a igualdade de

ve ocorrer, e o supremo deverá ser atingido. Logo:

$$w_i(\lambda) = \max_{\bar{\lambda} \in \Lambda} [w_i(\bar{\lambda}) - g_i(\bar{x}_i) \cdot (\lambda - \bar{\lambda})]$$

para todo $\lambda \in \Lambda$, desta maneira completando a prova do teorema.

Como resultado de tudo o que foi exposto neste item, poderemos então aproximar cada função w_i ($1 \leq i \leq k$) por uma função seccionalmente linear w_i^j , definida em Λ e com valores em \mathbb{R} , e calculada como:

16 $w_i^j(\lambda) \triangleq \max_{j=0,1,\dots,\hat{j}} [w_i(\bar{\lambda}^j) - g_i(\bar{x}_i^j) \cdot (\lambda - \bar{\lambda}^j)]$

Os vetores $\bar{\lambda}^j$, $j=0,1,\dots,\hat{j}$, constituem uma base para a linearização externa da função w_i . Cada vetor $\bar{x}_i^j \in X_i$ é uma solução ótima dos correspondentes subproblemas $D_{\bar{\lambda}^j}^i$, para $j=0,1,\dots,\hat{j}$.

É interessante notar desde já que toda liberdade é concedida na escolha desta base, tanto quanto ao número total \hat{j} de vetores participantes, quanto aos particulares valores escolhidos. Sob o ponto de vista computacional, é vantajoso que a base seja constituída por um número mínimo de elementos - o que minimizará o tamanho do problema a ser otimizado - , e que estes elementos tendam a concentrar-se na região de Λ onde interessa-nos seja mais fina a aproximação, ou seja, nas vizinhanças do ponto de ótimo procurado.

A linearização externa assim introduzida é a construção básica para a síntese do algoritmo implementado de resolução do problema S por coordenação via dual. Aprenderemos no capítulo seguinte como proceder a alterações de base no decorrer do processo iterativo, à medida que nos aproximamos de uma solução ótima para o problema D.

5. Algoritmo mestre para resolução de D

Passamos agora a apresentar o algoritmo mestre para resolução do problema D, obtido por linearização externa de sua função-objetivo e posterior aplicação da estratégia de relaxação, conforme visto no item anterior.

A estratégia de relaxação é particularmente eficaz quando aplicada a problemas de otimização que envolvem um número muito grande de restrições. A idéia básica consiste em resolver uma versão dita relaxada do problema originalmente proposto, isto é, levando-se em conta um subconjunto das restrições, desta maneira diminuindo-se a complexidade do problema a ser resolvido. Se a solução ótima obtida para esta versão relaxada satisfaz as restrições não levadas em conta, temos em mãos uma solução ótima do problema originalmente proposto. Caso contrário, constrói-se uma nova versão relaxada, passando-se a considerar restrições que se verifiquem tenham sido violadas.

Um refinamento importante consiste em abandonar restrições plenamente satisfeitas, o que será explorado no capítulo seguinte. Vemos, pois, que a estratégia de relaxação substitui um dado problema de otimização com restrições por uma sequência de problemas menores, alterando-se iterativamente o subconjunto de restrições em pauta, até que se obtenha uma solução ótima que satisfaça a todas as restrições, ou seja, uma solução ótima para o problema originalmente proposto. Um dos exemplos mais marcantes de aplicação da estratégia de relaxação é o método simplex para programação linear.

O algoritmo mestre apresentado em seguida será posteriormente modificado, quando então estabeleceremos a convergência da versão implementável a ser obtida no próximo capítulo.

17

Algoritmo mestre

passo 1 Escolha $\bar{\lambda}^0 \in \Lambda$. Resolva os subproblemas $D_{\bar{\lambda}^0}^i$:

$$w_i(\bar{\lambda}^0) = \max_{x_i \in X_i} [f_i(x_i) - \bar{\lambda}^0 \cdot g_i(x_i)]$$

para $1 \leq i \leq k$, usando um algoritmo de otimização adequado (por exemplo, método de direções viáveis). Seja $(\bar{x}_1^0, \bar{x}_2^0, \dots, \bar{x}_k^0)$ uma solução ótima assim obtida, e $(-g_1(\bar{x}_1^0), -g_2(\bar{x}_2^0), \dots, -g_k(\bar{x}_k^0))$ o correspondente conjunto de subgradientes. Faça $\hat{v}=0$.

passo 2 Resolva a aproximação linear corrente ao problema D , isto é:

$$\text{minimize } \sum_{i=1}^k w_i^0(\lambda) + \lambda \cdot b$$

onde definimos ($1 \leq i \leq k$):

$$w_i^0(\lambda) = \max_{j=0,1,\dots,\hat{v}} [w_i(\bar{\lambda}^j) - g_i(\bar{x}_i^j) \cdot (\lambda - \bar{\lambda}^j)]$$

usando um algoritmo de otimização adequado. Seja $\bar{\lambda}^{v+1}$ uma solução ótima assim determinada.

passo 3 Determine uma solução ótima $(\bar{x}_1^{v+1}, \bar{x}_2^{v+1}, \dots, \bar{x}_k^{v+1})$ para os subproblemas $D_{\bar{\lambda}^{v+1}}^i$ ($1 \leq i \leq k$). Seja $(-g_1(\bar{x}_1^{v+1}), -g_2(\bar{x}_2^{v+1}), \dots, -g_k(\bar{x}_k^{v+1}))$ o correspondente conjunto de subgradientes.

passo 4 Se $\bar{\lambda}^{v+1}$ é uma boa aproximação da solução exata do problema D , pare. Caso contrário, faça $\hat{v} = \hat{v} + 1$ e vá para 2.

O algoritmo mestre (17) assim proposto gera iterativamente uma sequência de hiperplanos-suporte a cada uma das funções w_i ($1 \leq i \leq k$). Assim sendo, a cada iteração a aproximação corrente ao minimando do problema D é sucessivamente refinada pela adição de novos hiperplanos-suporte.

Desta maneira, o tamanho do problema a ser resolvido no passo 2 do algoritmo mestre proposto crescerá de iteração a iteração, segundo uma progressão aritmética. Sob o ponto de vista computacional o algoritmo mestre (17) é, pois, conceptual. No capítulo IV veremos como contornar esta dificuldade, pelo abandono, sob condições controladas, de hiperplanos-suporte mais antigos.

No passo 4 do algoritmo mestre foi sugerida, de maneira vaga, uma certa regra de parada. Uma possível regra de parada derivaria da observação [15] de que a sequência de valores ótimos correspondentes à resolução do passo 2 do algoritmo mestre é uma sequência monotônica crescente de limites inferiores ao valor ótimo do problema D . Analogamente, a cada solução ótima $\bar{\lambda}^{j+1}$ assim computada no passo 2 corresponderá um valor ótimo calculado quando da resolução dos problemas $D_{\bar{\lambda}^{j+1}}^i$ ($1 \leq i \leq k$) no passo 3 do algoritmo mestre, valores ótimos estes que se constituem em limites superiores ao valor ótimo do problema D .

Desta maneira, a regra de parada sugerida por esta observação seria: pare se:

$$LS^{j+1} - LI^{j+1} \leq \varepsilon$$

onde ε é um escalar não negativo arbitrariamente escolhido e:

$$LI^{j+1} \triangleq \sum_{i=1}^k w_i(\bar{\lambda}^{j+1}) + \bar{\lambda}^{j+1} \cdot b$$

$$LS^{j+1} \triangleq \min_{j=0,1,\dots,k} \left[\sum_{i=1}^k w_i(\bar{\lambda}^{j+1}) + \bar{\lambda}^{j+1} \cdot b \right]$$

Todavia, implementaremos uma regra de parada mais simples, isto é, pare se:

$$\sum_{i=1}^k (w_i^j(\bar{\lambda}^{j+1}) - w_i^{j-1}(\bar{\lambda}^j)) \leq \epsilon_1$$

$$\text{e } \parallel \bar{\lambda}^{j+1} - \bar{\lambda}^j \parallel \leq \epsilon_2$$

onde ϵ_1 e ϵ_2 são escalares não negativos arbitrariamente escolhidos.

Como comentário final, enfatizamos, mais uma vez, que o principal obstáculo à implementação do algoritmo mestre (17) é, como já foi apontado, o crescimento do tamanho do programa de iteração para iteração. No capítulo IV estudaremos condições sob as quais é possível aplicar a estratégia de relaxação com abandono de vínculos ao algoritmo mestre. Esta técnica, em linhas gerais, constitui-se no abandono de hiperplanos-supor-te mais "antigos", desta maneira fazendo com que o tamanho do programa esteja sempre dentro de limites computacionalmente admissíveis. Naturalmente, as condições que nos interessam determinar deverão ser compatíveis com a convergência do algoritmo mestre.

Aspectos relativos à resolução dos subproblemas D_λ^i serão abordados no capítulo V.

IV IMPLEMENTAÇÃO DO ALGORITMO MESTRE

1. Separador e mapeamentos cortantes

Vimos no capítulo II como a dualização permitiu-nos reescrever o problema S sob uma forma equivalente, mais adequada para resolução, obtendo-se então o problema D . No capítulo III aproximamos a função-objetivo do problema D , conhecida apenas implicitamente em função de seu argumento λ , por meio de hiperplanos-suporte, obtendo uma nova função-objetivo explícitamente conhecida.

Esta nova função-objetivo foi então incorporada ao passo 2 do algoritmo mestre apresentado em (III.5), cuja implementação é obstaculada, como já vimos, pelos sucessivos aumentos em tamanho do programa a ser resolvido neste mesmo passo.

Este capítulo dedica-se a estudar condições sob as quais é possível abandonar-se hiperplanos-suporte gerados em iterações mais antigas do algoritmo mestre, desta forma impedindo-se que o tamanho do programa a ser resolvido atinja limites computacionalmente inaceitáveis. Em outras palavras, desejamos aplicar ao algoritmo mestre a estratégia de relaxação com abandono de vínculos [10], ao mesmo tempo garantindo convergência da versão implementada assim obtida.

Para isso, faremos uso da teoria generaliza-

da de convergência para algoritmos tipo plano cortante aplicada a problemas de programação matemática, conforme Eaves e Zangwill [10]. Esta teoria baseia-se em dois conceitos centrais: a função separador e o mapeamento cortante. A partir destes dois conceitos constrói-se o algoritmo mestre plano cortante [10], que serve de denominador comum aos demais algoritmos tipo plano cortante. Suas propriedades permitirão seja estabelecida a convergência da versão implementada do algoritmo mestre desenvolvido em (III.5).

Métodos de programação não-linear que fazem uso de planos cortantes foram pela primeira vez introduzidos por Kelley [26] e Cheney e Goldstein [5]. Nesta mesma linha citam-se as contribuições de Veinott [44] - método do hiperplano suporte, de Dantzig e Wolfe [6] - método plano cortante dual, e de Zangwill [47]. Contudo, todos estes métodos são estritamente cumulativos, isto é, não permitem que cortantes previamente gerados sejam abandonados ao longo do processo iterativo, embora Kelley [26] houvesse sugerido, heuristicamente, que "tal procedimento eventualmente forneceria uma solução ϵ -efetiva".

Posteriormente Topkis [41] apresentou garantias de convergência para algoritmos tipo plano cotante com abandono de vínculos para funções uniformemente côncavas, definidas em um conjunto viável convexo. Esta hipótese não é verificada no

Nota: uma função f definida em um conjunto convexo T e com valores em R é uniformemente côncava se existe uma função δ não decrescente, estritamente positiva em $(0, \infty)$ tal que, para todo x, y pertencentes a T tenhamos:

$$f((x+y)/2) \geq f(x)/2 + f(y)/2 + \delta(|x-y|) \quad [41]$$

nosso caso, já que, como veremos, transformaremos o algoritmo mestre em um programa completamente linear. Após a publicação do trabalho de Eaves e Zangwill [10], Topkis [42] reformulou alguns de seus resultados anteriormente apresentados em [41], relaxando algumas hipóteses anteriores.

Parece caber a Zangwill [48] a iniciativa de fazer uso de mapamentos ponto-conjunto no campo de programação matemática, reconhecendo que diversos métodos de programação podem ser interpretados nestes termos. As propriedades apresentadas por tais mapeamentos, sistematicamente estudadas, permitiram uma derivação mais imediata de resultados sobre convergência.

Finalmente, Eaves e Zangwill [10] expandiram estes conceitos, desenvolvendo uma teoria geral de convergência para algoritmos tipo plano cortante aplicada a problemas de programação matemática, resultados estes discutidos e aplicados a seguir.

Reescreveremos inicialmente o problema D apresentado em (II.2) sob uma forma equivalente. Considerando (Γ_i, λ) como pontos do epígrafo das funções w_i (7) ($1 \leq i \leq k$), isto é,

$$\Gamma_i \geq w_i(\lambda) = \max_{x_i \in X_i} [f_i(x_i) - \lambda \cdot g_i(x_i)]$$

podemos imediatamente reescrever D como:

$$18 \quad D \quad \begin{aligned} & \text{minimize} \sum_{i=1}^k \Gamma_i + \lambda \cdot b \\ & \Gamma \in R^k \\ & \lambda \in \Lambda \end{aligned}$$

sujeito a:

$$\begin{aligned} \Gamma_i &\geq f_i(x_i) - \lambda \cdot g_i(x_i) \\ (\forall x_i \in X_i) \quad (1 \leq i \leq k) \end{aligned}$$

Este problema com um número infinito de vínculos lineares pode ser escrito sob forma mais corrente definindo:

$$z = (\sigma_1, \sigma_2, \dots, \sigma_k, \lambda_1, \lambda_2, \dots, \lambda_m) \in \mathbb{R}^{k+m}$$

$r: \mathbb{R}^{k+m} \rightarrow \mathbb{R}$ calculada como:

$$r(z) = \sum_{i=1}^k \sigma_i + \lambda \cdot b$$

$$G \triangleq \left\{ z \in \mathbb{R}^{k+m} \mid \lambda \in \Lambda, \sigma_i + \lambda \cdot g_i(x_i) \geq f_i(x_i), \forall x_i \in X_i \quad (1 \leq i \leq k) \right\}$$

com o que obtemos a forma equivalente procurada:

19 D minimize $r(z)$
 $z \in G$

Da mesma forma, o problema correspondente ao passo 2 do algoritmo mestre definido em (III.5) será reescrito como:

20 D minimize $r(z)$
 $z \in Z^j$

onde definimos:

$$Z^j \triangleq \left\{ z \in \mathbb{R}^{k+m} \mid \lambda \in \Lambda, \sigma_i + \lambda \cdot g_i(\bar{x}_i^{j_i}) \geq f_i(\bar{x}_i^j), \quad (j=0, 1, \dots, \bar{j}) \quad (1 \leq i \leq k) \right\}$$

O conjunto Z^j corresponde a uma aproximação do conjunto viável G por linearização externa com a base $\{\bar{x}_i^j\}$, $j=0, 1, \dots, \bar{j}$, isto é, o problema (20) corresponde ao problema

(19) relaxado, considerando-se apenas \mathcal{V} entre a infinidade de vínculos lineares.

O conjunto viável G , por sua vez, pode ser considerado como o "limite" de Z^k para $k \rightarrow \infty$, isto é, G representa a região viável do problema D . É imediato que $Z^k \supseteq G \neq \emptyset$, uma vez admitida a hipótese sobre consistência do problema S .

Os algoritmos tipo plano cortante são algoritmos de relaxação que aproximam o conjunto viável de um dado problema pela intersecção de uma coleção de conjuntos, constrin- do iterativamente uma sequência de pontos de tal forma que todo ponto de acumulação pertence ao conjunto viável e é solução do problema originalmente proposto. Podemos identificar claramente G como sendo o nosso conjunto viável, e Z^k como uma de suas possíveis aproximações lineares externas, dado pela intersecção de uma coleção de k semiespaços definidos por hiperplanos-supor-te.

Observamos que a função r assim definida é linear, ou seja, D^k é um programa completamente linear. Afirma mos que a estratégia de resolução do problema D via algoritmo tipo plano cortante consistirá em resolver iterativamente os problemas linearizados D^k , gerando um conjunto de k novos vínculos a cada iteração do algoritmo, tendendo a refinaria aproximação ao conjunto viável G nas vizinhanças do ponto de ótimo procurado.

Podemos imaginar k mapeamentos cortantes que associam a cada ponto essa coleção de novos vínculos. Uma das características do mapeamento plano cortante é que os vínculos assim gerados - no nosso caso semiespaços suporte de G - excluem estritamente (ver V.4), por construção, o ponto que lhes deu origem. Observamos ainda que uma solução ótima de D^k não determina de maneira unívoca o conjunto de novos vínculos ora gerados e que deverão ser incorporados, na iteração seguinte do algoritmo,

ao novo problema D^{j+1} . Na realidade, é possível gerar-se uma coleção de conjuntos de novos vínculos.

Função separador

A sequência de pontos z^j ($j=0,1,\dots$) gerada pelo algoritmo tipo plano cortante estará contida em um conjunto fechado X , e satisfará uma propriedade muito especial, isto é, se $z^j \notin G$, ou seja, se z^j não é solução ótima do problema D^j , então z^j estará separado dos pontos z^1, z^2, \dots, z^{j-1} de pelo menos certa distância, definida por uma função separador.

Definição: dados os conjuntos fechados $\emptyset \neq G \subset X$ em um espaço métrico, a função $\delta : X \sim G \rightarrow \mathbb{R}$ é uma função separador se:

- (a) é não negativa;
- (b) se $z^j \rightarrow z^\infty$ e $\delta(z^j) \rightarrow 0$, então $z^\infty \in G$.

Um exemplo de função separador é a distância $d(z, G)$ entre um ponto $z \in X$ e o conjunto viável G . Decorre da definição acima que δ é positiva em $X \sim G$; que qualquer função p definida em $X \sim G$ e com valores em \mathbb{R}^+ , e semicontínua inferiormente, é uma função separador; que o mínimo de uma coleção finita de separadores é um separador; e que se δ é um separador e $\delta' > \delta$ então δ' é também um separador (ver [10]).

Mapeamento plano cortante

A introdução do mapeamento cortante envolve o espaço de subconjuntos \mathcal{L} dado por:

$$\mathcal{L} \triangleq \left\{ z \mid G \subset z \subset X, z \text{ fechado} \right\}$$

Supondo-se X um espaço métrico compacto, e que $G \subset X$ é um subespaço compacto, poderemos estender a \mathcal{L} uma topologia com a métrica de Hausdorff D_0 . Nestas condições \mathcal{L} se

rá compacto, em termos de topologia de conjuntos.

Com o auxílio da função separador e do espaço podemos caracterizar agora o mapeamento cortante.

Definição: um mapeamento ponto-conjunto $\varphi : X \sim G \rightarrow \mathcal{P}$ é um mapeamento cortante se existe uma função separador δ definida em $X \sim G$ e com valores em \mathbb{R}^+ tal que, para todo $z \in X \sim G$ e todo $Z \in \varphi(z)$, temos:

$$z \cap B(z, \delta(z)) = \emptyset$$

onde $B(z, \delta(z))$ é uma bola aberta centrada em z e com raio $\delta(z)$.

Nessas condições, cada conjunto $Z \in \varphi(z)$ será chamado de cortante. Note-se que o conjunto viável G é na realidade a intersecção de todos os cortantes, isto é:

$$G = \bigcap \varphi(X \sim G)$$

Observe-se que φ associa a cada ponto de $X \sim G$ uma coleção de conjuntos de \mathcal{P} (mapeamento ponto-conjunto), ou seja, para cada ponto não pertencente ao conjunto viável G , o mapeamento cortante gera uma coleção (possivelmente infinita) de novos vínculos. Tipicamente, mapeamentos cortantes geram conjuntos de hiperplanos-suporte do conjunto viável.

2. Algoritmo mestre plano cortante

Passamos agora a apresentar o algoritmo mestre plano cortante [10], desenvolvido por Eaves e Zangwill com base nos conceitos ora apresentados. Tal algoritmo constitui-se

em um modelo bastante geral, cujas propriedades de convergência interessam-nos discutir. Nosso objetivo final será apresentar em (V.3) uma de suas versões implementadas, e então aplicá-la à solução do problema D.

Algoritmo mestre plano cortante (AMPC) - Eaves e Zangwill [10]

Seja $\delta \leq d(\cdot, G)$ uma função separadora definida em $X \sim G$ e suponha tenha sido gerada uma sequência z^0, z^1, \dots, z^j de pontos de X . Se $z^j \in G$, então pare. Caso contrário, determine um novo ponto $z^{j+1} \in X$ tal que:

$$z^{j+1} \notin B(z^i, \delta(z^i)) \quad (0 \leq i \leq j)$$

21 Teorema: dados os conjuntos fechados $\emptyset \neq G \subset X$, seja a sequência z^0, z^1, \dots de pontos de X gerados por AMPC. Então qualquer ponto de acumulação desta sequência pertencerá ao conjunto G .

Demonstração: tomado-se subsequências se necessário, consideremos que, devido à compacidade do conjunto X , a sequência z^0, z^1, \dots de pontos de X gerados por AMPC converge a um ponto de acumulação z^∞ . Então:

$$z^j \rightarrow z^\infty$$

Como por construção do AMPC temos:

$$\begin{aligned} z^j \notin G &\implies z^{j+1} \in X \\ z^{j+1} \notin B(z^j, \delta(z^j)) &\quad (0 \leq j \leq j) \end{aligned}$$

concluímos que:

$$\|z^{j+1} - z^j\| \geq \delta(z^j) \geq 0$$

Como $z^j \rightarrow z^\infty$, então:

$$\| z^{j+1} - z^j \| \rightarrow 0$$

e portanto:

$$\delta(z) \rightarrow 0$$

e por definição de função separador, conclui-se imediatamente que:

$$z^\infty \in G$$

3. Versão implementada de AMPC

Apresentaremos agora uma versão implementada do AMPC. Tipicamente, uma iteração desta versão consiste em resolver um problema de otimização em um dado conjunto viável. A partir de uma das soluções ótimas assim obtida são gerados novos vínculos a serem adicionados e caracterizados os vínculos inativos a serem abandonados. Essa adição-e-abandono corresponde a uma redefinição do conjunto viável, modificando-o para a próxima iteração do algoritmo.

Admitiremos válidas as hipóteses apresentadas em (II.1). Sejam $\varphi_i : X \sim G \rightarrow \mathcal{L}$ ($1 \leq i \leq k$) mapeamentos cortantes e $\delta : X \sim G \rightarrow R$ uma função separador associada a todos os φ_i . Vamos definir $I_i^{j_i}$ como o conjunto de índices de cortantes referentes ao subsistema i não abandonados até à j -ésima iteração inclusive, e Γ como o mapeamento ponto-conjunto definido em \mathcal{L} como:

$$\Gamma(z) = \left\{ z^* \in z \mid r(z^*) \leq r(z), (\forall z \in \mathbb{Z}) \right\}$$

O mapeamento Γ associa a cada conjunto Z , região viável do problema (19), o conjunto de pontos de ótimo correspondentes.

Versão implementada do AMPC

passo 1 Faça $z^0 = x$, $\bar{r} = 0$, $\bar{\delta} = 0$. Calcule $z^0 \in \Gamma(z^0)$.

passo 2 Suponha que até à \bar{v} -ésima iteração tenham sido calculados valores para \bar{r} e $\bar{\delta}$, e geradas a sequência de pontos z^j ($j=0,1,\dots,\bar{v}-1$) e a sequência de conjuntos Z^j ($j=0,1,\dots,\bar{v}-1$) dados por intersecção de cortantes. Se $z^{\bar{v}-1} \in G$, pare. Caso contrário, vá para 3.

passo 3 Aplique os mapeamentos cortantes e obtenha novos cortantes, escolhendo arbitrariamente:

$$c_i \in \varphi_i(z^{\bar{v}-1}) \quad (1 \leq i \leq k)$$

Defina:

$$Y = Z^{\bar{v}-1} \cap c_1 \cap \dots \cap c_k$$

e calcule $z^{\bar{v}} \in \Gamma(Y)$.

Se $r(z^{\bar{v}}) \geq \bar{r} + \bar{\delta}$, defina $Z^{\bar{v}}$ como a intersecção de todos os cortantes presentes na definição de Y e ativos em $z^{\bar{v}}$. Faça $\bar{r} = r(z^{\bar{v}})$ e $\bar{\delta} = \delta(z^{\bar{v}})$. Caso contrário, faça $Z^{\bar{v}} = Y$.

A construção de novos vínculos e o abandono (quando possível) de vínculos inativos é feita no passo 3. Vemos que este algoritmo abandona vínculos inativos em massa no instante em que o ganho em valor da função-objetivo do problema D tenha sido satisfatório, a critério do separador δ ; entretanto, todos os vínculos gerados são retidos. O abandono de vínculos inativos não altera o valor ótimo corrente do problema, já que estamos a tratar de um caso totalmente convexo.

Teorema: a versão implementada do AMPC resolve o problema:

$$\text{minimize } r(z)$$

$$z \in G$$

Demonstração: decorre da própria construção da versão implementada do AMPC que dois casos distintos podem ocorrer.

caso 1: a desigualdade não foi desobedecida por uma sequência infinita de pontos consecutivos. Neste caso, podemos extraír de $\{z^k\}$ a subsequência $\{y^k\}_{k \in \mathbb{N}}$ infinita de pontos do compacto X , tais que $r(y^k) \geq r(y^{k-1}) + \delta(y^{k-1})$. Tomando subsequências se necessário, considere que a sequência y^0, y^1, \dots converge a um ponto de acumulação y^∞ . Então:

$$y^k \xrightarrow{} y^\infty$$

Usaremos do fato de que a função $\delta: X \sim G \rightarrow \mathbb{R}$ calculada por:

$$\delta'(z) = \inf \{d(z, x) \mid r(z) \geq r(x) + \delta(x), x \in X\}$$

é também um separador [10]. Provaremos inicialmente que qualquer y^k da sequência satisfaz $y^k \notin B(y^j, \delta'(y^j))$ para todo $j < k$. Admitindo por absurdo que para algum $j < k$:

$$y^k \in B(y^j, \delta'(y^j))$$

então teremos:

$$\begin{aligned} d(y^k, y^j) &< \delta'(y^j) = \\ &= \inf \{d(y^j, x) \mid r(y^j) \geq r(x) + \delta(x), x \in X\} \end{aligned}$$

Decorre da própria construção do passo 3 que:

$$r(y^k) \geq r(y^j) + \delta(y^j) \quad (\forall j < k)$$

Obtemos então, a partir do resultado anterior, que:

$$d(y^k, y^j) < \delta'(y^j) \leq d(y^k, y^j) \quad (!)$$

Concluímos necessariamente que, para todo $j < k$, temos:

$$y^k \notin B(y^j, \delta'(y^j))$$

Por aplicação imediata do teorema (21), concluímos que:

caso 2: a desigualdade foi desobedecida por uma sequência infinita de pontos consecutivos. Tomando subsequências se necessário, considere que a sequência infinita de pontos y^0, y^1, \dots de pontos do compacto X converge a um ponto de acumulação y^∞ . Então:

$$y^k \longrightarrow y^\infty$$

Decorre da própria construção do passo 3 da versão implementada do AMPC que, nestas condições:

$$z^k \subset z^{k-1} \subset \dots$$

Como:

$$y^k \in \Psi(z^k) \subset z^k$$

então teremos:

$$y^k \subset z^{j_i}$$

para todo $j \leq k$. Ainda pelo passo 3, lembrando que:

$$z^k = z^{k-1} \cap c_1^{k-1} \cap c_2^{k-1} \cap \dots \cap c_k^{k-1}$$

decorre que:

$$y^k \in c_i^{j-1}$$

para todo $(1 \leq j \leq k)$ e todo subsistema $(1 \leq i \leq k)$. Por definição de mapeamento cortante, deverá existir uma função separador δ definida em $X \sim G$ e com valores em R^+ tal que se verifique:

$$y^k \notin B(y^{j-1}, \delta(y^{j-1}))$$

para todo $(1 \leq j \leq k)$.

Por aplicação imediata do teorema (21), concluímos que:

$$y^{\mathfrak{I}} \xrightarrow{\quad} y^{\infty} \in G$$

4. Mecanismo de geração de cortantes

O resultado fundamental do item anterior é a versão implementada do AMPC. Relembramos que o algoritmo mestre de resolução do problema D, apresentado em (III.5), foi reformulado em (IV.1) e posteriormente colocado sob a forma de algoritmo tipo plano cortante com abandono de vínculos, vínculos estes a serem gerados por mapeamentos cortantes.

Nesta linha de desenvolvimento, resta-nos agora caracterizar os mapeamentos cortantes φ_i , a função separadora δ associada e o mecanismo de geração de cortantes.

Após resolvermos o problema:

$$\begin{array}{ll} \text{D} & \text{minimize } r(z) \\ & z \in Z \end{array}$$

correspondente à \mathfrak{I} -ésima iteração do algoritmo (III.5), obtemos uma solução ótima:

$$z^{\mathfrak{I}} = (\sigma_1^{\mathfrak{I}+1}, \sigma_2^{\mathfrak{I}+1}, \dots, \sigma_k^{\mathfrak{I}+1}, \bar{\lambda}_1^{\mathfrak{I}+1}, \bar{\lambda}_2^{\mathfrak{I}+1}, \dots, \bar{\lambda}_m^{\mathfrak{I}+1}) \in X$$

e devemos então resolver os subproblemas ($1 \leq i \leq k$):

$$\begin{array}{ll} \text{maximize} & [f_{i\mathfrak{l}}(x_i) - \bar{\lambda}^{\mathfrak{I}+1} \cdot g_i(x_i)] \\ x_i \in X_i & \end{array}$$

obtendo soluções ótimas $\bar{x}_i^{\mathfrak{I}+1} \in X_i$ ($1 \leq i \leq k$) e um certo conjunto C_i

($1 \leq i \leq k$) caracterizado por:

$$C_i^{\Delta} = \left\{ (\theta, \lambda) \in \mathbb{R}^{k+1} \mid \sigma_i^{\Delta} + \lambda \cdot g_i(\bar{x}_i^{\Delta}) \geq f_i(\bar{x}_i^{\Delta}) \right\} \cap X \in \mathcal{L}$$

Como passo preliminar às demonstrações de que a geração dos conjuntos C_i^{Δ} faz-se por mapeamento plano contante, provaremos que, se $z^{\Delta} \in X \setminus G$, então z^{Δ} é separado estritamente do conjunto viável Z^{Δ} .

Teorema: O conjunto $C_i^{\Delta} \in \mathcal{L}$ ($1 \leq i \leq k$) assim obtido a partir de uma solução ótima $z^{\Delta} \notin G$ do problema D^{Δ} , separa estritamente z^{Δ} do conjunto viável associado Z^{Δ} .

Demonstração: se $z^{\Delta} \notin G$, então, por definição de G , existirá um ponto $x_i^{\Delta} \in X_i$ tal que:

$$\sigma_i^{\Delta} + \bar{\lambda}^{\Delta} \cdot g_i(x_i^{\Delta}) < f_i(x_i^{\Delta})$$

Logo, usando da compactade de X_i , obtemos:

$$\begin{aligned} \sigma_i^{\Delta+1} &\leq f_i(x_i^{\Delta}) - \bar{\lambda}^{\Delta+1} \cdot g_i(x_i^{\Delta}) = \\ &= \max_{x_i \in X_i} [f_i(x_i) - \bar{\lambda}^{\Delta+1} \cdot g_i(x_i)] = \\ &= f_i(\bar{x}_i^{\Delta+1}) - \bar{\lambda}^{\Delta+1} \cdot g_i(\bar{x}_i^{\Delta+1}) \end{aligned}$$

ou seja:

$$\sigma_i^{\Delta+1} + \bar{\lambda}^{\Delta+1} \cdot g_i(\bar{x}_i^{\Delta+1}) < f_i(\bar{x}_i^{\Delta+1})$$

para $1 \leq i \leq k$.

Uma vez estabelecida esta propriedade de se-

paração estrita, resta-nos apenas definir uma função separador $\varphi_i : X \sim G \rightarrow \mathbb{R}$ para que possamos caracterizar C_i^j ($1 \leq i \leq k$) como um cortante, após o qual identificaremos formalmente os mapeamentos cortantes φ_i ($1 \leq i \leq k$). Estes mapeamentos cortantes, mais a função separador adiante definida, integrarão a versão implementada do algoritmo dual de resolução do problema S, objetivo deste trabalho.

Neste momento será necessário lembrar que, ao apresentarmos o algoritmo mestre/plano cortante (IV.2), supusemos a existência de um separador $\delta : X \sim G \rightarrow \mathcal{L}$ tal que:

$$\delta(\cdot) \leq d(\cdot, G)$$

Definamos, com o auxílio de (7) e (16), a função $\delta : X \sim G \rightarrow \mathcal{L}$ calculada como:

$$\delta(z^j) = \frac{\alpha}{k} \sum_{i=1}^k [w_{ii}(z^{j+1}) - w_{ii}^j(z^{j+1})]$$

onde $0 < \alpha \leq 1$, e que verifica essa desigualdade. De fato, dado $z^j \in X \sim G$, e fazendo-se uso da norma do máximo, podemos escrever:

$$\begin{aligned} d(z^j, G) &\geq \min_{z \in G} \max_{1 \leq i \leq k} |z_i - z_i^j| \\ &\geq \min_{z \in G} \max_{1 \leq i \leq k} |z_i - z_i^j| \\ &= \max_{1 \leq i \leq k} \min_{z_i \in G} (z_{ii} - C_i^{j+1}) \\ &\geq \frac{\alpha}{k} \sum_{i=1}^k \max_{x_i \in X_i} [f_{ii}(x_i) - \bar{\lambda} \cdot g_{ii}(x_i)] - C_i^{j+1} \end{aligned}$$

Ou seja, usando (7) e (16):

$$\delta(z^j) \triangleq \frac{\alpha}{k} \sum_{i=1}^k [w_i(\bar{\lambda}^{j+1}) - w_i(\hat{\lambda}^{j+1})] \leq d(z^j, G)$$

Observamos que o somatório estendido às componentes surge da observação de que, se a desigualdade é verificada com respeito ao máximo valor tomado entre as componentes, também será verificada com respeito ao valor médio tomado entre todas estas componentes. O parâmetro $0 < \alpha \leq 1$ foi introduzido com vistas à implementação desta função separador, que será utilizada computacionalmente.

Teorema: A função $\delta : X \sim G \rightarrow \mathcal{L}$ assim definida é uma função separador para todo $0 < \alpha \leq 1$.

Demonstração: se $z^j \notin G$, resulta imediatamente de (7) e (10) que $\delta(z^j) > 0$, ou seja, δ é não negativa em $X \sim G$. A continuidade de δ é estabelecida lembrando-se que w_i ($1 \leq i \leq k$) é uma função secionalmente linear, e portanto contínua; e que, em vistas dos resultados de [22], transcritos no apêndice A, as funções w_i ($1 \leq i \leq k$) também são contínuas. Como toda função real definida em $X \sim G$ e semicontínua inferiormente é uma função separador [10], a demonstração está completa.

Unindo os resultados obtidos neste capítulo, surgem então de uma maneira muito natural os mapeamentos cortantes φ_i ($1 \leq i \leq k$), definidos em $X \sim G$ e com valores em \mathcal{L} , que associam a uma solução ótima z^j do problema D^j (20) k conjuntos de (novos) cortantes C_i^j ($1 \leq i \leq k$), cada cortante correspondendo a uma solução ótima do subproblema:

$$\begin{aligned} & \text{maximize } |f_i(x_i) - \bar{\lambda}^{j+1} \cdot g_i(x_i)| \\ & x_i \in X_i \end{aligned}$$

O teorema seguinte estabelece que tais mapeamentos φ_i ($1 \leq i \leq k$), associados à função separador anteriormente definida, são de fato mapeamentos cortantes.

Teorema: Os mapeamentos ponto conjunto $\varphi_i : X \sim G \rightarrow \mathcal{L}$ definidos por:

$$\varphi_i : z^{\circ} \rightarrow c_i^{\circ} \quad (1 \leq i \leq k)$$

são mapeamentos cortantes.

Demonstração: [10], com auxílio dos resultados de [2] e [22], transcritos no apêndice A.

IV.4

54

V IMPLEMENTAÇÃO DO ALGORITMO DUAL

1. Introdução

Nos capítulos I e II apresentamos o problema S, em cuja resolução estamos interessados. Mais adiante, nos capítulos III e IV, apresentamos os fundamentos teóricos que justificam a resolução de S por coordenação dual, propondo para esta finalidade o algoritmo (III.5) e sua versão implementada (IV.3).

Neste capítulo estudaremos a implementação deste algoritmo, começando pela apresentação de um método de direções viáveis escolhido para resolução dos subproblemas D_{λ}^i (7), etapa necessária à determinação de novos cortantes. Em seguida, proporemos um mecanismo computacional para implementar a adição-e-abandono de cortantes.

Outros assuntos também discutidos dizem respeito à inicialização do algoritmo, bem como às regras de parada que decidem quanto à truncagem do processo iterativo e consequente fornecimento de soluções quase-ótimas para o problema S.

Inserem-se no texto fluxogramas explicativos de diversas etapas de interesse na implementação do algoritmo (IV.3), deixando-se a apresentação do diagrama de blocos completo para o apêndice B. Ainda desse mesmo apêndice constam instruções gerais para uso do programa por nós desenvolvido.

A título de ilustração, é proposto e resolvi do um problema de otimização de um sistema decomponível, apresentando-se os resultados computacionais obtidos.

2. Resolução dos subproblemas D_{λ}^i

Para obter um cortante c_i^j ($j=0,1,\dots$) ($1 \leq i \leq k$) é necessário, como já vimos em (IV.4), resolver o sub - problema:

$$D_{\lambda}^i \quad w_i(\bar{\lambda}^j) = \underset{x_i \in X_i}{\text{máximo}} \left[f_i(x_i) - \bar{\lambda}^j \cdot g_i(x_i) \right]$$

obtendo-se uma solução ótima $x_i^{j+1} \in X_i$.

Escolhemos um método de direções viáveis [50] de primeira ordem como instrumento de resolução dos subproblemas por razões de simplicidade, facilidade de implementação e generalidade, uma vez que se pode imediatamente estender o procedimento para o caso, mais geral, em que os conjuntos viáveis X_i sejam definidos por vínculos diferenciáveis quaisquer. Além do mais, a experiência [37] reconhece serem as taxas de convergência deste método tanto mais elevadas quanto melhor a conformação da região viável, o que é de supor-se em se tratando de politopos.

Métodos de direções viáveis constroem, a partir de um dado ponto viável, uma semireta que passa por este ponto e por pontos do conjunto viável. Essa semireta tem a particularidade de ser definida pela direção viável à qual corresponda a maior taxa de ganho inicial (no sentido incremental) em valor da função-objetivo. Em seguida, o método determina um novo ponto viável sobre esta semireta, correspondente à otimização da função-objetivo a partir do ponto viável anterior, nesta

direção. Ou seja, resolve-se um dado problema de programação matemática por uma série de problemas de otimização unidirecional, processo este que não oferece maiores dificuldades sob o ponto de vista computacional.

Assumiremos por hipótese que cada conjunto viável x_i ($1 \leq i \leq k$) é um politopo dado por:

$$x_i \triangleq \left\{ x_i \in \mathbb{R}^{n_i} \mid A_i \cdot x_i + B_i \leq 0 \right\} \neq \emptyset$$

onde A_i ($m_i \times n_i$) e B_i (m_i) são, respectivamente, matrizes e vetores dados. Admitimos ainda que todos os politopos assim definidos são conjuntos limitados, desta maneira satisfazendo as hipóteses iniciais (II.1) sobre compacidade. Além do mais, uma condição de qualificação de vínculos é automaticamente garantida pela linearidade dos mesmos, por exemplo, condição de qualificação reverso-convexa [32].

Essa nova hipótese ora introduzida é interessante sob o ponto de vista computacional, tornando mais imediatas operações como testes de viabilidade e cômputo de valores locais do gradiente dos vínculos, que fazem parte integrante do método de direções viáveis a ser apresentado a seguir. Ademais, é de se esperar que a solução ótima obtida correspondente à resolução do subproblema $D_{\lambda_j}^i$ ($j=0,1,\dots$) ($1 \leq i \leq k$) pertença à fronteira do politopo correspondente, ou seja, o processo de otimização poderá eventualmente terminar em um número finito de passos.

Direção viável ótima de deslocamento

Dado um ponto viável, a direção de deslocamento à qual corresponda a maior taxa de ganho inicial (no sentido incremental) em valor da função-objetivo é dada pelo valor local do gradiente dessa mesma função-objetivo. Todavia, é necessário assegurar-se que tal direção de deslocamento seja viável, ou seja, é necessário levar-se também em conta informações locais

àcerca dos vínculos que definem a região viável do problema de otimização. Zoutendijk [50] observou que a sequência de pontos viáveis gerada por um algoritmo de direções viáveis pode eventualmente "engarrafar" em um ponto não-desejável quando levamos em conta, na determinação da direção viável de deslocamento, apenas os vínculos localmente ativos. Esse fenômeno pode ser evitado levando-se em conta os vínculos localmente ε -ativos, isto é, os vínculos ativos mais aquêles prestes a tornarem-se ativos.

A direção viável de deslocamento é determinada resolvendo-se um problema de programação linear chamado Minimax [37], que passamos a apresentar. Será necessário definir:

$x_i \in X_i$ - ponto viável inicial

$A_i^l \in R^{n_i}$ - l-ésima ($0 \leq l \leq m_i$) linha da matriz A_i

$B_i^l \in R$ - l-ésima ($0 \leq l \leq m_i$) componente do vetor B_i

$h_i \in R^{n_i}$ - direção de deslocamento a partir de x_i

$J_i^\varepsilon(x_i)$ - conjunto de vínculos ε -ativos em x_i , dado por:

$$J_i^\varepsilon(x_i) = \left\{ l \in N \mid (0 \leq l \leq m_i) \quad (A_i^l \cdot x_i + B_i^l + \varepsilon \geq 0) \right\}$$

A direção viável ótima de deslocamento a partir de x_i , que representaremos por h_i^* ($1 \leq i \leq k$), é obtida como solução ótima do seguinte problema de programação linear [37]:

$$\text{Minimax} \quad \text{maximize } (\nabla f_i(x_i) - \lambda \cdot \nabla g_i(x_i)) \cdot h_i \\ \text{sujeito a:}$$

$$A_i^l \cdot h_i \leq 0 \quad (l \in J_i^\varepsilon(x_i))$$

$$|h_i^j| \leq 1 \quad (0 \leq j \leq n_i)$$

O primeiro conjunto de restrições serve para garantir a viabilidade da solução ótima procurada. O último con-

junto de restrições serve apenas como um fator de normalização da direção ótima procurada.

Definindo-se h_{io}^* como o valor ótimo da função-objetivo do minimax, prova-se [37] que $h_{io}^* = 0$ para todo x_i viável e todo $\epsilon \geq 0$. Mais ainda, se $x_i \in X_i$ é uma solução ótima para o problema D_λ^i , e tomando-se $\epsilon = 0$, então $h_{io}^* = 0$.

Deslocamento máximo viável

Dado um ponto viável inicial x_i , e calculada uma direção viável ótima de deslocamento h_i^* através da resolução do minimax, o método de direções viáveis prevê, como etapa seguinte, a identificação do traço da semi-reta de origem x_i e direção h_i^* na fronteira do conjunto viável X_i . Ou seja, devemos calcular o deslocamento máximo viável ρ^* a partir do ponto inicial x_i :

$$\rho^* = \max \left\{ \rho \in \mathbb{R} \mid (x_i + \rho \cdot h_i^*) \in X_i \right\}$$

A determinação de ρ^* é feita por um processo iterativo de tentativa-e-erro. Tipicamente, este processo iterativo consta de um número infinito de passos, necessitando-se, pois, truncá-lo. Esse truncamento, sob certas condições [37], não afeta as propriedades de convergência do método de direções viáveis.

Otimização unidirecional

Uma vez determinado o segmento:

$$[x_i, x_i + \rho^* \cdot h_i^*] \in X_i$$

resta determinar uma solução ótima x_i^* do seguinte problema de optimização unidirecional:

$$\begin{aligned} & \text{maximize } [f_i(x_i) - \lambda \cdot g_i(x_i)] \\ & \text{sujeito a:} \\ & x_i \in [\bar{x}_i, x_i + C^* \cdot h_i^*] \end{aligned}$$

o que se faz pelo método da busca pelo segmento áureo [37]. Esse processo iterativo, de tentativa-e-erro, desenvolve-se tipicamente em um número infinito de passos. Também aqui surge a necessidade de truncar o processo iterativo, o que, sob certas condições [37], não afeta as propriedades de convergência do método de direções viáveis.

O método de direções viáveis por nós implementado é uma versão do algoritmo 4.3.20 de [37]. Uma observação importante a ser feita é que, neste trabalho, não foram feitos estudos teóricos sobre as consequências do truncamento, limitando-nos a introduzir mecanismos heurísticos com essa finalidade.

O diagrama de blocos completo do método de direções viáveis faz parte integrante do apêndice B.

3. Adição-e-abandono de vínculos

Reportando-nos a (IV.4) e (IV.5), vimos que cada cortante C_i^j ($1 \leq i \leq k$) é caracterizado por um semi-espacô suporte dado por:

$$C_i^j \triangleq \left\{ (\tau, \lambda) \in R^{k+m} \mid \sigma_i + \lambda \cdot g_i(\bar{x}_i^{j+1}) \geq f_i(\bar{x}_i^{j+1}) \right\} \cap X$$

onde $\bar{x}_i^{j+1} \in X_i$ é uma solução ótima do subproblema D_{λ}^i , obtida pelo método de direções viáveis apresentado no item anterior.

Em cada caso, o conjunto viável Z^j correspondente à j -ésima iteração do algoritmo (IV.3) é dado pela in-

tersecção de uma coleção de cortantes, ou seja, Z° é claramente um politopo, definido por inequações lineares. Esse conjunto de inequações lineares pode, evidentemente, ser colocado sob a forma:

$$Z^{\circ} \triangleq \left\{ z \in \mathbb{R}^{k+m} \mid A^{\circ} \cdot z \leq B^{\circ} \right\}$$

onde A° é uma matriz retangular e B° é um vetor coluna. O número de linhas da matriz A° é igual ao número de cortantes correspondentes à i -ésima iteração do algoritmo, por sua vez igual ao número de elementos do vetor B° . A matriz A° tem, evidentemente $(k+m)$ colunas.

Tipicamente, uma linha da matriz A° seria expressa por:

$$\begin{bmatrix} 0 & \dots & 1 & \dots & 0 & g_i^1(\bar{x}_i^j) & \dots & g_i^m(\bar{x}_i^j) \end{bmatrix}$$

\swarrow
i-ésima coluna

e o correspondente elemento do vetor B° por:

$$\begin{bmatrix} f_i(\bar{x}_i^j) \end{bmatrix}$$

para $(1 \leq i \leq k)$.

Percebemos já que o mecanismo de adição-e-abandono de vínculos nada mais deverá ser que um mecanismo de redefinição da matriz A° e do vetor B° , por adição-e-abandono, respectivamente, de linhas e elemento, de iteração para iteração do algoritmo.

A determinação dos novos cortantes c_i° ($1 \leq i \leq k$) corresponderá a adicionar automaticamente à matriz A° e ao vetor B° , respectivamente, k novas linhas e k novos elementos. Mais ainda, o mecanismo computacional proposto, a crité

rio do valor corrente da função-separador definida em (IV.4), deverá proceder à identificação e abandono de linhas e elementos que correspondam a restrições inativas para a solução ótima corrente.

Desta maneira obtemos uma nova matriz \vec{A}^{j+1} e um novo vetor \vec{B}^{j+1} , que definem o novo conjunto viável Z^{j+1} . No programa de computador por nós desenvolvido essa tarefa é realizada pelas subrotinas MTRX1 e MTRX2 (ver apêndice B).

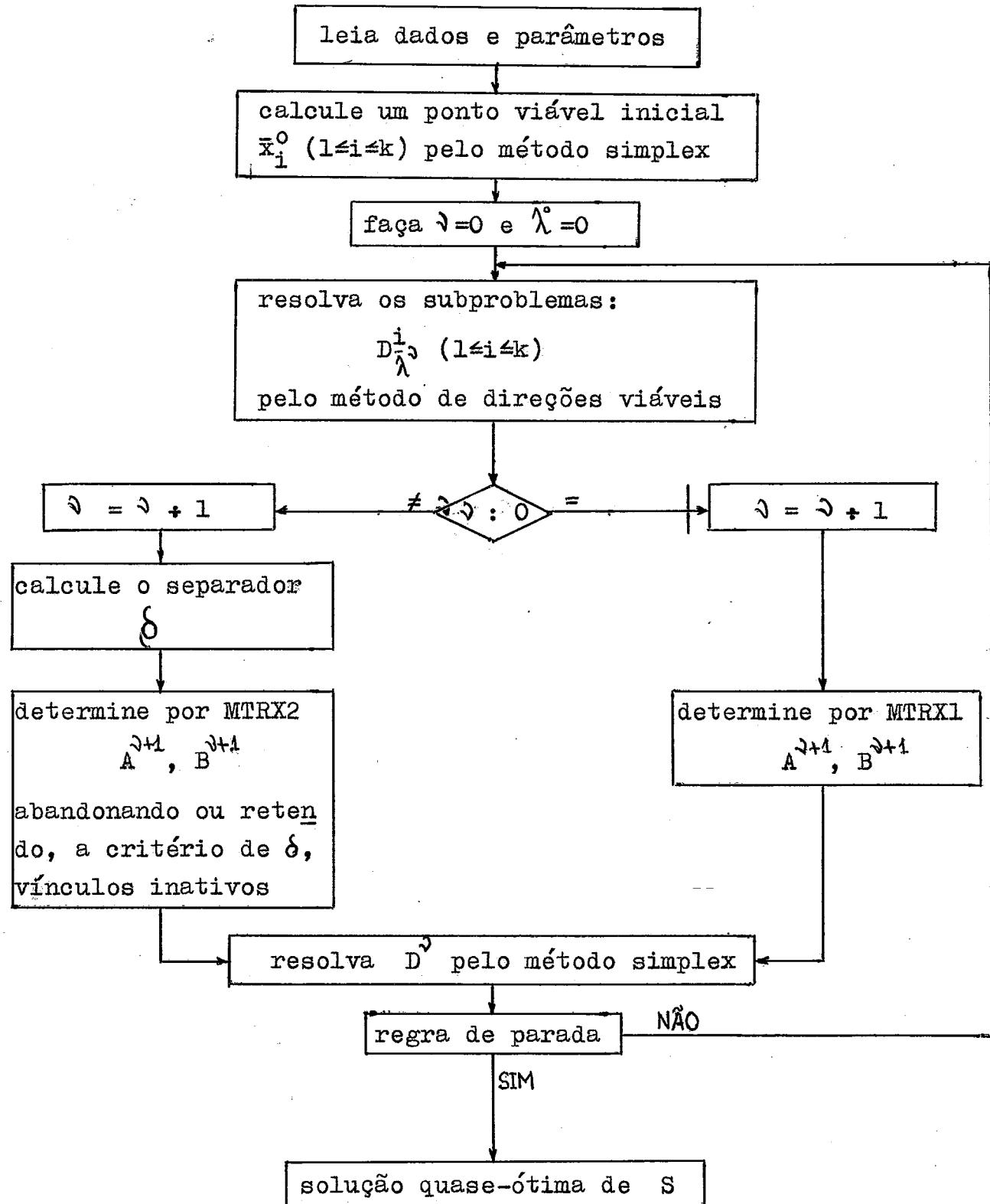
É importante notar que as dimensões da matriz \vec{A}^j e do vetor \vec{B}^j variam de iteração para iteração do algoritmo, e que se faz necessário adicionar variáveis de folgas desigualdades matriciais:

$$\vec{A}^j \cdot z \leq \vec{B}^j \quad (j=0,1,\dots)$$

para transformá-las em igualdades matriciais, com vistas à utilização do método simplex para programação linear (ver apêndice B).

4. Fluxograma final

O seguinte fluxograma simplificado corresponde ao algoritmo dual implementado para resolução do problema S (I.1), sendo o resultado final deste trabalho. Neste fluxograma identificamos as principais etapas do processo iterativo, deixando o detalhamento completo por conta dos diagramas de bloco apresentados no apêndice B.



5. Exemplo proposto

O seguinte exemplo foi por nós proposto para ilustrar a aplicação do algoritmo dual (V.4). Assim sendo, tomando como modelo a formulação do problema S (I.1), formularemos o nosso exemplo como:

$$\text{maximize } f_1(x_1) + f_2(x_2)$$

sujeito a:

$$A_1 \cdot x_1 + B_1 = 0$$

$$A_2 \cdot x_2 + B_2 = 0$$

$$g_1(x_1) + g_2(x_2) \leq b$$

onde:

$$f_1(x_1) = -(x_1^1 - 2)^2 - (x_1^2)^2$$

$$f_2(x_2) = -(x_2^1 - 1)^2 - (x_2^2 - 2)^2 + 6$$

$$g_1^1(x_1) = (x_1^1)^2$$

$$g_1^2(x_1) = (x_1^1 + x_1^2)^2$$

$$g_2^1(x_2) = x_2^1$$

$$g_2^2(x_2) = x_2^1 + x_2^1$$

$$A_1 = \begin{bmatrix} -1 & 1 \\ 1 & 1 \\ 0.6 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad B_1 = \begin{bmatrix} -1 \\ -4 \\ -3 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned} A_2 &= \begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} & B_2 &= \begin{bmatrix} -5 \\ 0 \\ 0 \end{bmatrix} \\ b &= \begin{bmatrix} 6 \\ 8 \end{bmatrix} \end{aligned}$$

APÊNDICE A

1. Teoria de dualidade

Transcrevemos, a seguir, alguns resultados de [7] sobre teoria de dualidade, baseado nos quais demonstra-se o teorema de dualidade forte, empregado em (III.3).

Definição: o problema canônico primal P é:

$$\begin{aligned}
 P \quad & \text{maximize} \quad f(x) \\
 & x \in X \\
 & \text{sujeito a:} \\
 & g(x) \leq 0
 \end{aligned}$$

onde:

$f: X \rightarrow \mathbb{R}$ - é uma função-critério côncava explicitamente conhecida, definida em X e com valores em \mathbb{R} ;

$g: X \rightarrow \mathbb{R}^m$ - é uma função-demanda convexa explicitamente conhecida, definida em X e com valores em \mathbb{R}^m ;

$X \subseteq \mathbb{R}^n$ - é um conjunto convexo não-vazio.

Definição: o problema canônico dual D de P é:

$$D \quad \underset{\lambda \geq 0}{\text{minimize}} \quad \sup_{x \in X} \quad f(x) - \lambda \cdot g(x)$$

onde $\lambda \in \mathbb{R}^m$ é o vetor de variáveis duais.

Definição: o valor ótimo de P é o supremo de $f(x)$ sujeito a $x \in X$ e $g(x) \leq 0$.

Definição: o valor ótimo de D é o supremo de seu maximando sujeito a $\lambda \geq 0$.

Definição: um vetor λ é um vetor multiplicador ótimo para P se o par (x, λ) satisfaz as condições de optimidade:

- (a) x maximiza $f(x) - \lambda \cdot g(x)$ em X ,
- (b) $\lambda \cdot g(x) = 0$
- (c) $\lambda \geq 0$
- (d) $g(x) \leq 0$

para algum $x \in X$.

Definição: a função-perturbação v associada a P definida em \mathbb{R}^m e com valores em $\mathbb{R} \cup \{\pm\infty\}$ é calculada como:

$$v(y) = \sup_{x \in X} f(x)$$

sujeito a:

$$g(x) \leq y$$

onde $y \in \mathbb{R}^m$ é o vetor perturbação.

Definição: seja $\bar{y} \in \mathbb{R}^m$ um ponto onde v é finita. Um vetor $\bar{\lambda} \in \mathbb{R}^m$ é subgradiente de v em y se:

$$v(y) \geq v(\bar{y}) + \bar{\lambda} \cdot (y - \bar{y}) \quad (\forall y \in \mathbb{R}^m)$$

Definição: o problema P é estável se $v(0)$ é finito e:

$$\lim_{\theta \rightarrow +0} \frac{v(0) - v(\theta y)}{\theta \|y\|} < \infty \quad (\forall y \neq 0)$$

Definição: $Y \subseteq \mathbb{R}^m$ é o conjunto de todos os vetores perturbação para os quais o correspondente problema perturbado é viável, isto é:

$$Y \triangleq \left\{ y \in \mathbb{R}^m \mid g(x) \leq y \text{ para algum } x \in X \right\}$$

Lema 1: Y é um conjunto convexo, e v é uma função convexa em Y.

Demonstração: [17].

Lema 2: Seja θ uma função convexa definida em um conjunto convexo $Y \subseteq \mathbb{R}^m$ e com valores em $\mathbb{R} \cup \{-\infty\}$. Seja $\|\cdot\|$ uma norma definida em \mathbb{R}^m , e seja y um ponto no qual θ é finita. Então θ possui um subgradiente em $\bar{y} \in Y$ se e somente se existe um escalar positivo M tal que:

$$\frac{\theta(\bar{y}) - \theta(y)}{\|y - \bar{y}\|} \leq M$$

para todo $y \in Y$ tal que $y \neq \bar{y}$.

Demonstração: [17].

Lema 3: Seja $v(0)$ finito. Então λ é uma solução ótima do problema D e os valores ótimos de D e P são iguais se e somente se $-\lambda$ é subgradiente de v em $y=0$.

Demonstração: [17].

Lema 4: Seja $v(0)$ finito e λ um subgradiente de v em $y=0$. En-

tão x é uma solução ótima para o problema P se e somente se $(x, -\lambda)$ satisfaz as condições de optimidade (a), (b), (c) e (d) para P .

Demonstração: [17].

Teorema: (dualidade forte)

Se o problema P é estável, então:

- (a) o problema D possui uma solução ótima;
- (b) os valores ótimos dos problemas P e D são iguais;
- (c) λ é uma solução ótima do problema D se e somente se $-\lambda$ é subgradiente de v em $y=0$;
- (d) qualquer solução ótima λ do problema D caracteriza o conjunto de todas as soluções ótimas do problema P como soluções do sub-problema:

$$\text{maximize}_{x \in X} f(x) - \lambda \cdot g(x)$$

sujeito a:

$$g(x) \leq 0$$

$$\lambda \cdot g(x) = 0$$

Demonstração: como P é estável, $v(0)$ é finito e concluimos dos lemas 1 e 2 que v possui um subgradiente em $y=0$. As partes (a), (b) e (c) decorrem por aplicação imediata dos lemas 3. A parte (d) decorre imediatamente do lema 4 com o auxílio da parte (c). Com isso completamos a prova do teorema da dualidade forte.

2. Mapeamentos ponto-conjunto

Apresentamos a seguir alguns resultados extraídos de [2], [22] e [23] concernentes a mapeamentos ponto-conjunto. Estes resultados são utilizados na demonstração do lema 8 de 10, citado como prova do teorema , apresentado no capítulo IV.

No que se segue, consideraremos X e Y como dois espaços topológicos [23].

Definição: um mapeamento ponto-conjunto $\Omega : X \rightarrow Y$ é aberto em $x_0 \in X$ se, qualquer que seja $G \subseteq Y$ aberto, com $G \cap \Omega(x_0) \neq \emptyset$, existe uma vizinhança U de x_0 tal que:

$$(\forall x \in U) \implies \Omega(x) \cap G \neq \emptyset$$

Definição: um mapeamento ponto-conjunto $\Omega : X \rightarrow Y$ é fechado em $x_0 \in X$ se, qualquer que seja $G \subseteq Y$ aberto, com $G \ni \Omega(x_0)$ existe uma vizinhança U de x_0 tal que

$$(\forall x \in U) \implies \Omega(x) \subseteq G$$

Definição: um mapeamento ponto-conjunto $\Omega : X \rightarrow Y$ é contínuo em $x_0 \in X$ se é aberto e fechado em x_0 .

Definição: um mapeamento ponto-conjunto $\Omega : X \rightarrow Y$ é uniformemente compacto nas vizinhanças de $x_0 \in X$ se existe uma vizinhança U de x_0 tal que o fecho de:

$$\bigcup_{x \in U} \Omega(x)$$

é compacto.

Nos dois teoremas que se seguem, extraídos de [22], consideraremos:

$$f: X \times Y \longrightarrow [-\infty, +\infty]$$

$$v(x) \triangleq \sup_{y \in \Omega(x)} f(x, y)$$

onde $\Omega: X \rightarrow Y$ é um mapeamento ponto-conjunto que associa a cada ponto de X um subconjunto de Y .

Teorema: Se Ω é contínuo em $x_0 \in X$ e uniformemente compacto nas vizinhanças de x_0 , e se f é contínua em $x_0 \times \Omega(x_0)$ então v é contínua em x_0 .

Demonstração: [22].

Teorema: Se Ω é contínuo em $x_0 \in X$ e f é contínua em $x_0 \times \Omega(x_0)$, então o mapeamento M definido em X como:

$$M(x) \triangleq \{ y \in \Omega(x) \mid v(x) \leq f(x, y) \} \subseteq Y$$

é fechado em x_0 .

Demonstração: [22].

APENDICE B

Neste apêndice apresentamos uma descrição do algoritmo implementado de resolução do problema S por coordenação dual. Esta descrição não pretende ser exaustiva, mas tão só - mente apresentar aspectos que interessem diretamente ao usuário do programa desenvolvido.

I. Entradas e saídas do programa

Parâmetros

INTMP	número máximo de iterações do algoritmo
INTSS	número máximo de iterações do algoritmo de resolução dos subproblemas D_{λ}^i (7)
EPSON	fator que determina, em função do valor corrente do separador, o valor inicial da precisão de trabalho do algoritmo de resolução dos subproblemas D_{λ}^i (7)
EPS2N	precisão utilizada na regra de parada do processo iterativo, conforme (III.5)
EPS3N	idem, ibidem
DIMAX	número máximo de colunas da matriz A

Dados fornecidos

KSYST	número de subsistemas
M	número de recursos produtivos comuns oferecidos aos subsistemas

BPRIM	vetor cujas M componentes representam as disponibilidades dos recursos produtivos comuns oferecidos aos subsistemas
MSYST(I)	número total de restrições que definem o conjunto X_i de decisões viáveis associado ao i-ésimo subsistema
NSYST(I)	dimensão do vetor de decisão associado ao i-ésimo subsistema
AR(I)	matrizes que definem o conjunto X_i de decisões viáveis associado ao i-ésimo subsistema:
$X_i = X(I) R^{NSYST(I)} AR(I) \cdot X(I) + BR(I) = 0$	

Expressões analíticas a serem fornecidas:

deverão ser fornecidas ao programa diversas expressões analíticas, diretamente ligadas às expressões das funções-critério f_i e das funções-demanda g_i ($l=i=k$), nos seguintes lugares:

fluxograma página

Lembramos que, ao fornecer estas expressões, é necessário proceder a alterações nos comandos GO TO(...), I associados, em função do valor de KSYST.

Valores computados:

OPVDL	valor quase-ótimo da função-objetivo do problema S
X(I)	vetor de decisão quase-ótimo associado ao i-ésimo subsistema

XIAMB	vetor de preços quase-ótimo
A	matriz cujas KSYST colunas representam os valores quase-ótimos das funções-demanda g_i ($1=i=k$) associadas aos respectivos subsistemas
USAGE	vetor cujas M componentes representam a diferença entre a disponibilidade de um dado recurso produtivo e a demanda total quase-ótima desse mesmo recurso produtivo.

2. Mensagens de erro

INVIABILIDADE

SUBSISTEMA I = ...

Houve erro no fornecimento dos valores dos elementos das matrizes AR(I) e BR(I). O programa apontou o conjunto X_i como sendo vazio.

AUMENTAR PRECISÃO

Erro excessivo na determinação das aproximações lineares. Necessário diminuir o valor do parâmetro EPSON.

DIMENSIONAMENTO ULTRAPASSADO

Foi ultrapassado o valor máximo admitido para o número de colunas da matriz A. Necessário redefinir o dimensionamento (Bx1) da matriz A e o valor do parâmetro DIMAX.

3. Dimensionamento

A estratégia de relaxação com abandono de

vínculos, conforme apresentada no capítulo IV, resulta em trabalhar-se, a cada iteração do algoritmo, com um número variável de aproximações lineares. Desta maneira, é preciso levar em conta a variação no dimensionamento das matrizes correspondentes às aproximações lineares, não se permitindo seja ultrapassado o dimensionamento máximo previsto no início da corrida do computador.

Desta maneira, é possível tão somente definir um dimensionamento mínimo para diversas matrizes, e recomendar um valor máximo correspondente.

Será necessário definir os seguintes valores:

$$L1 = \max_I NSYST(I) + MSYST(I) + 3$$

$$L2 = KSYST + 2$$

$$C1 = \max_I 2 \cdot NSYST(I) + MSYST(I) + 2$$

$$C2 = 3 \cdot KSYST + M$$

com base nos quais computamos:

$$MSIMP = \max (L1 , L2)$$

$$NSIMP = \max (C1 , C2)$$

valores estes que servirão de base para definir dimensionamentos mínimos.

Dimensionamentos conhecidos a priori

$$AR(KSYST, \max_I MSYST(I), \max_I NSYST(I))$$

$$BR(KSYST, \max_I MSYST(I))$$

$$BPRIM(M)$$

$$ITOTI(KSYST)$$

```

F( (M+1)•máximo NSYST(I) )
I
MSYST(KSYST)
NSYST(KSYST)
NUMBR(KSYST)
S2GMA(KSYST)
USAGE(M)
X(KSYST, máximo NSYST(I))
I
XHZ( máximo NSYST(I) )
I
XLAMB(M)

```

Dimensionamentos mínimos

```

A(MSIMP, NSIMP)
B(MSIMP+2, MSIMP+2)
E(MSIMP+2, MSIMP+2)
EX(KSYST, M+1)
EM(MSIMP+2, MSIMP+2)
EX(KSYST, M+1)
INDEX(MSIMP)
KTIED(MSIMP)
SIML(MSIMP)
XX(MSIMP+2)
XMAXI(NSIMP+1)
X0(MSIMP+2)
X01(3•KSYST+1)

```

Todas as matrizes e vetores-coluna supracitados apresentam a propriedade seguinte: seus correspondentes números totais de linhas são de mesma ordem. Recomendamos adotar um valor múltiplo do dimensionamento mínimo acima definido, e lembramos que o valor do parâmetro DIMAX a ser fornecido representa o valor adotado para NSIMP.

4. Subrotinas

O algoritmo implementado para resolução do problema S incorpora três subrotinas, cujos argumentos e finalidades descrevemos a seguir.

SUBROUTINE MTRX1

argumentos de entrada: KSYST, M, I3172, BPRIM, X

argumentos de saída: MSIMP, NSIMP, NUMBR, A, XO, EX

Essa subrotina tem por finalidade inicializar o algoritmo implementado, construindo a primeira aproximação linear. Ao cabo do processo iterativo esta subrotina é novamente chamada, para computar os valores OPVD1 e USAGE, descritos no item 2.

SUBROUTINE MTRX2

argumentos de entrada: KSYST, M, T, BPRIM, NUMBR, EX,
X, X01

argumentos de saída: MSIMP, NSIMP, ITOTI, A, XO, EEX

Esta subrotina tem por finalidade gerar novas aproximações lineares e, a critério do argumento T, abandonar aproximações lineares antigas inativas. O argumento T é computado no programa principal em função do valor corrente do separador.

SUBROUTINE SMPLX

argumentos de entrada: ISGNI, MSIMP, NSIMP, A, XK

argumentos de saída: IDIDI, XMAXI

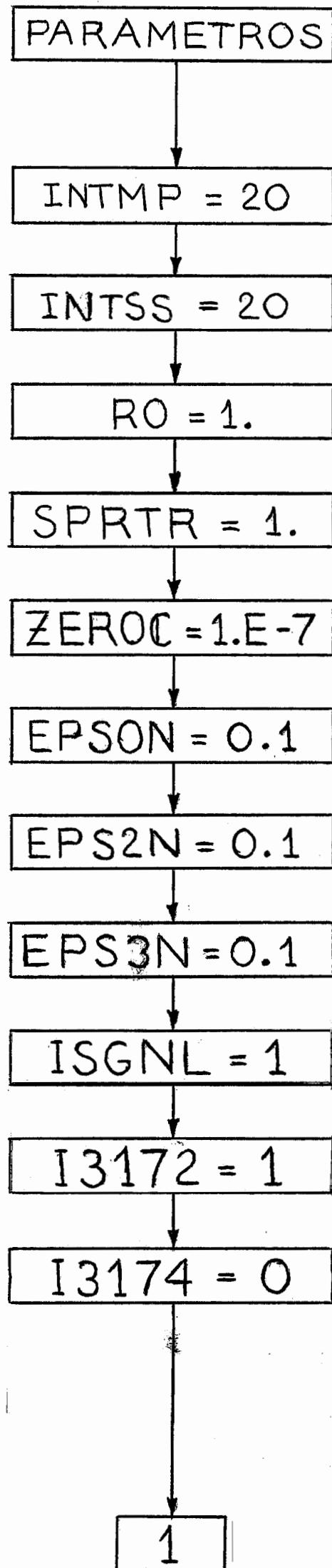
Esta subrotina tem por finalidade resolver problemas de programação linear pelo método simplex revisado [14]. Na inicialização do processo iterativo é chamada para de-

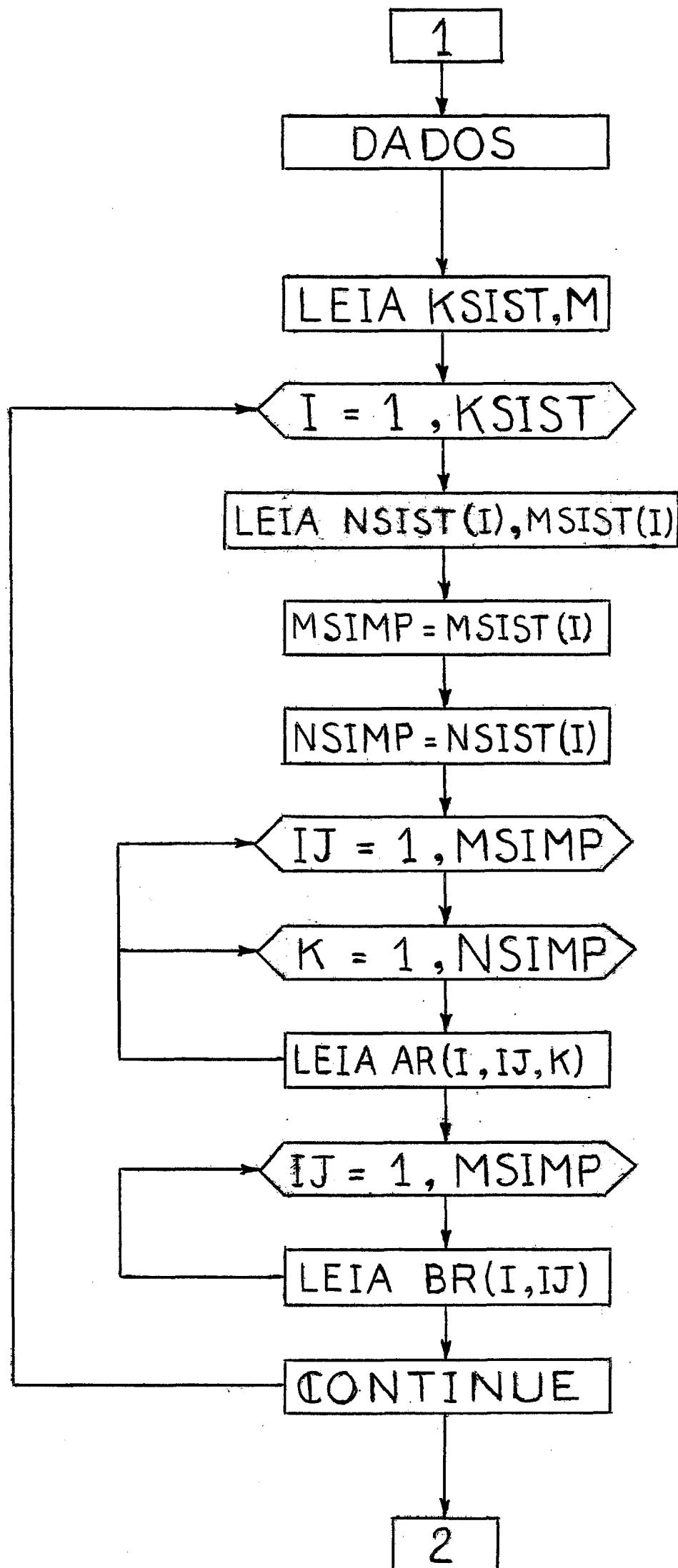
terminar uma decisão viável inicial para cada subsistema.

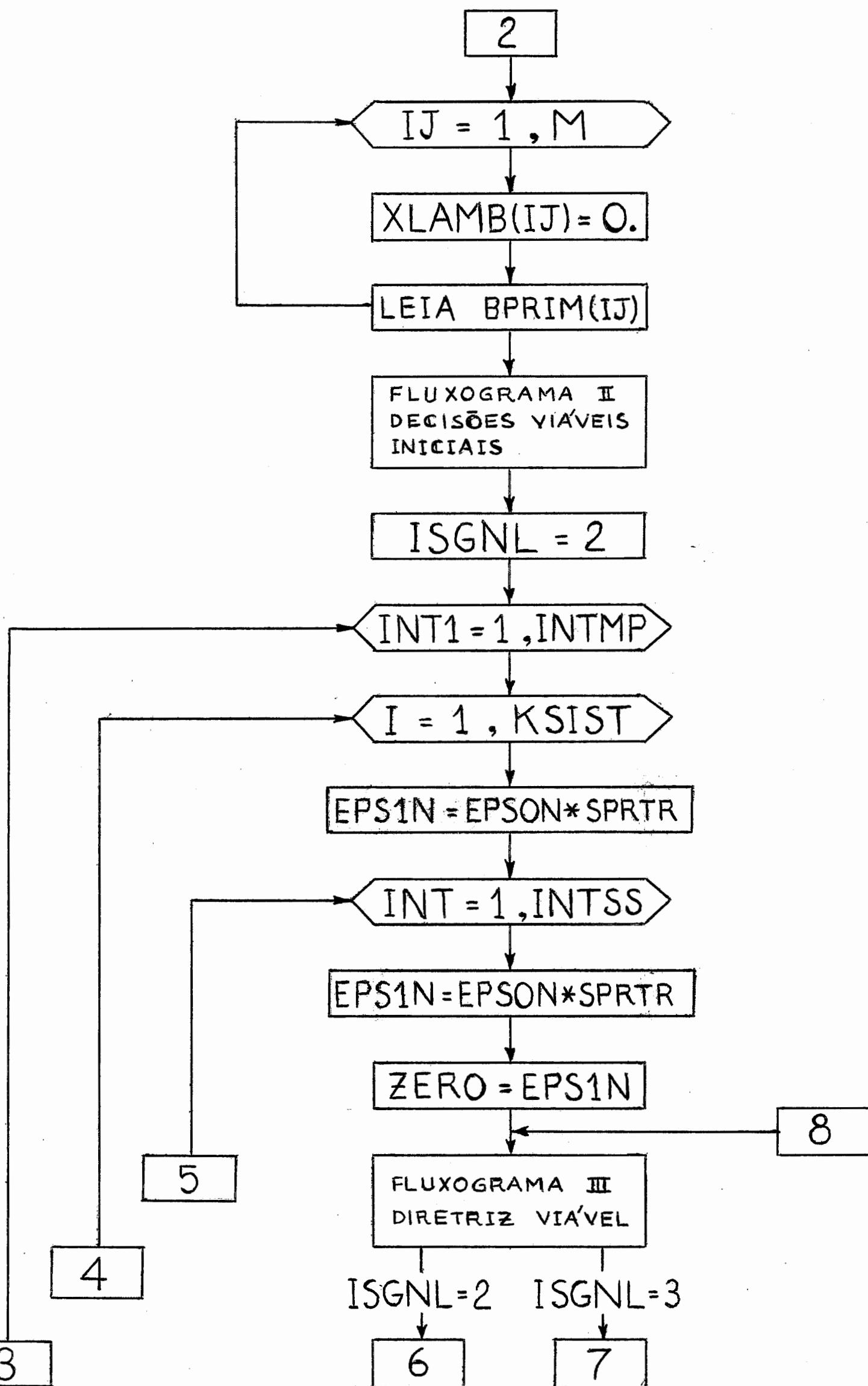
O argumento IDIDI pode tomar três valores, a saber, significando:

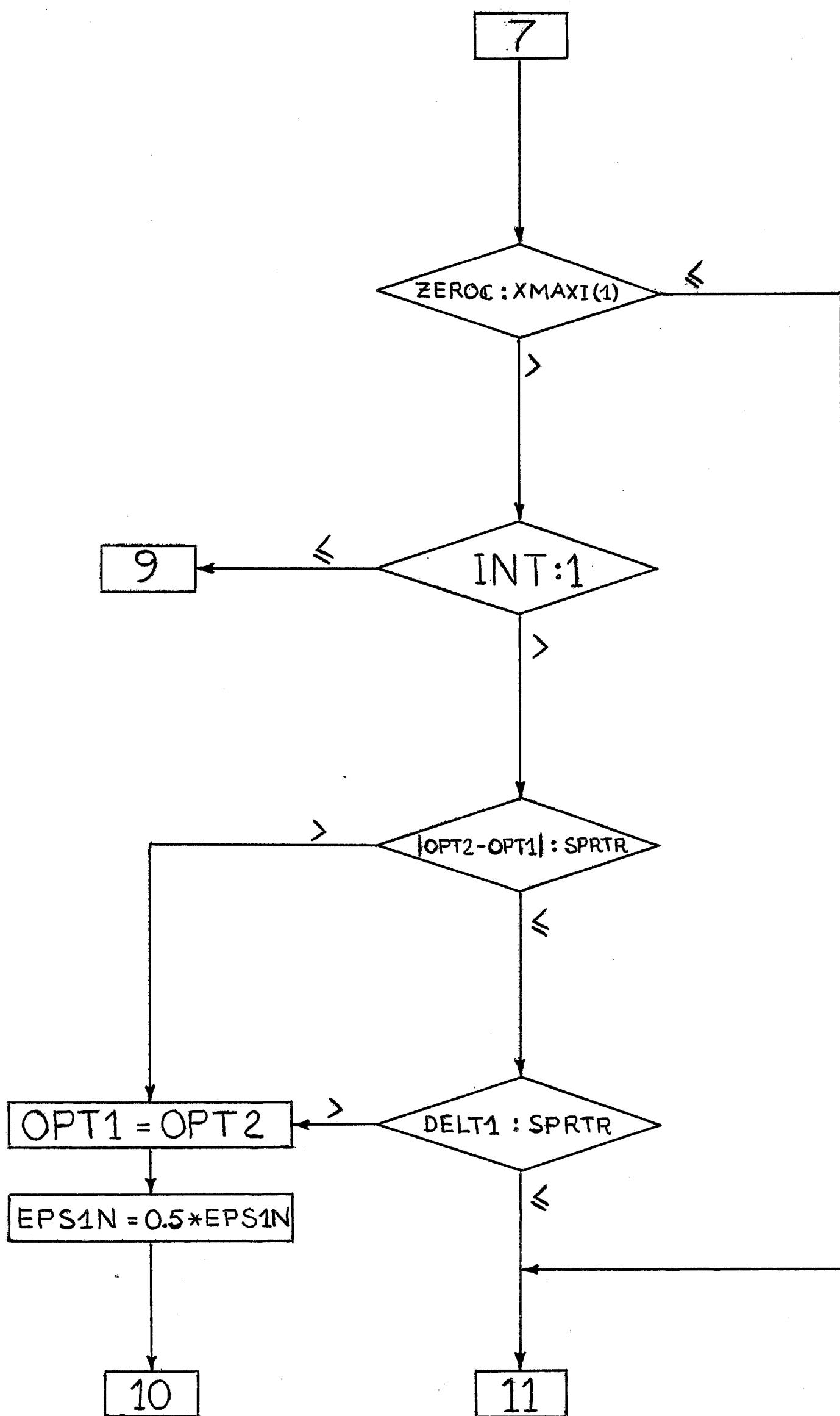
IDIDI = 1	problema consistente
IDIDI = 2	problema inconsistente, o conjunto viável proposto é vazio
IDIDI = 3	valor ótimo ilimitado; a subrotina fornece uma solução viável computada

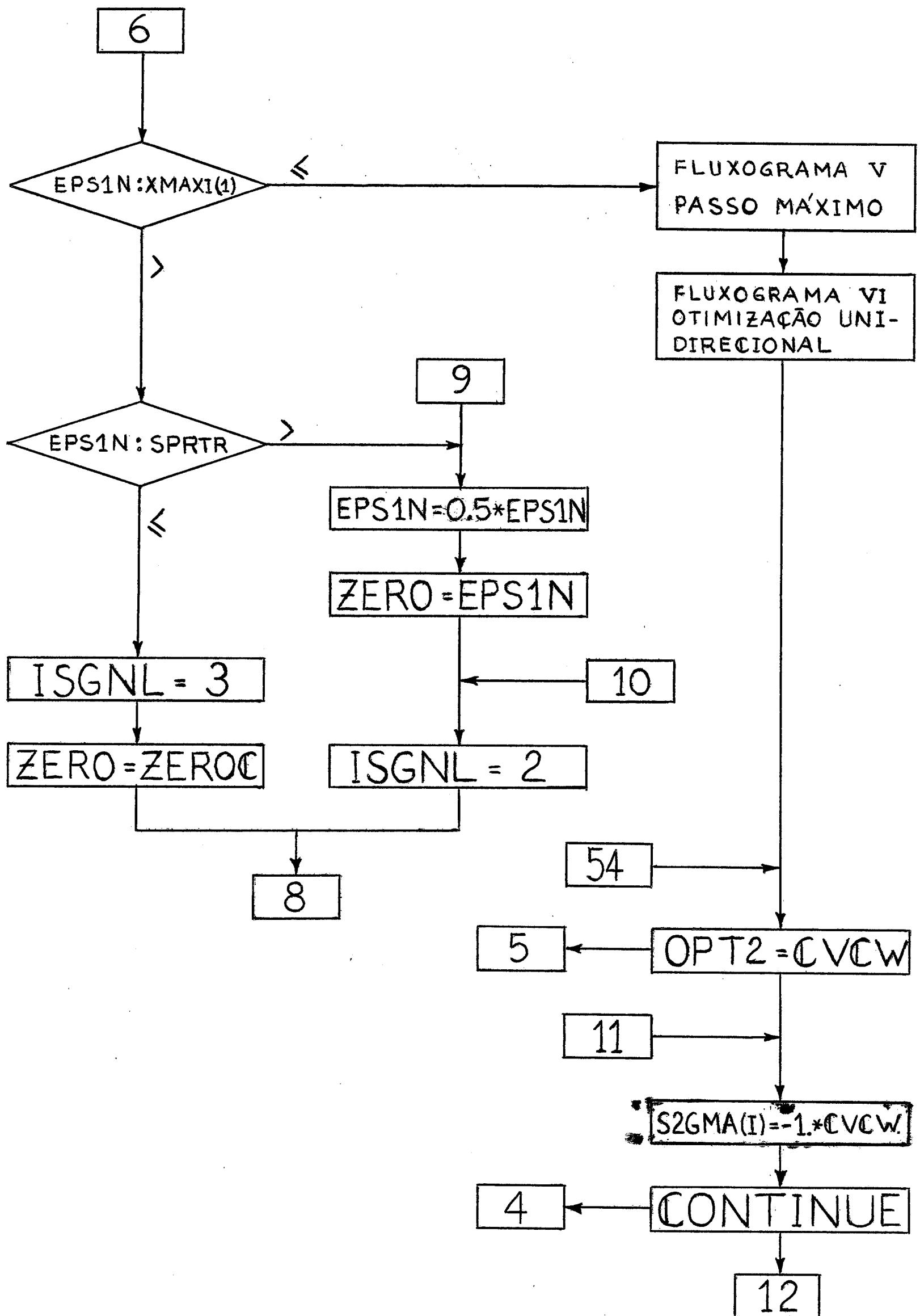
No primeiro caso, as NSIMP primeiras componentes do vetor XMAXI representam a solução ótima computada, e a $(NSIMP+1)$ -ésima componente representa o valor ótimo correspondente computado.

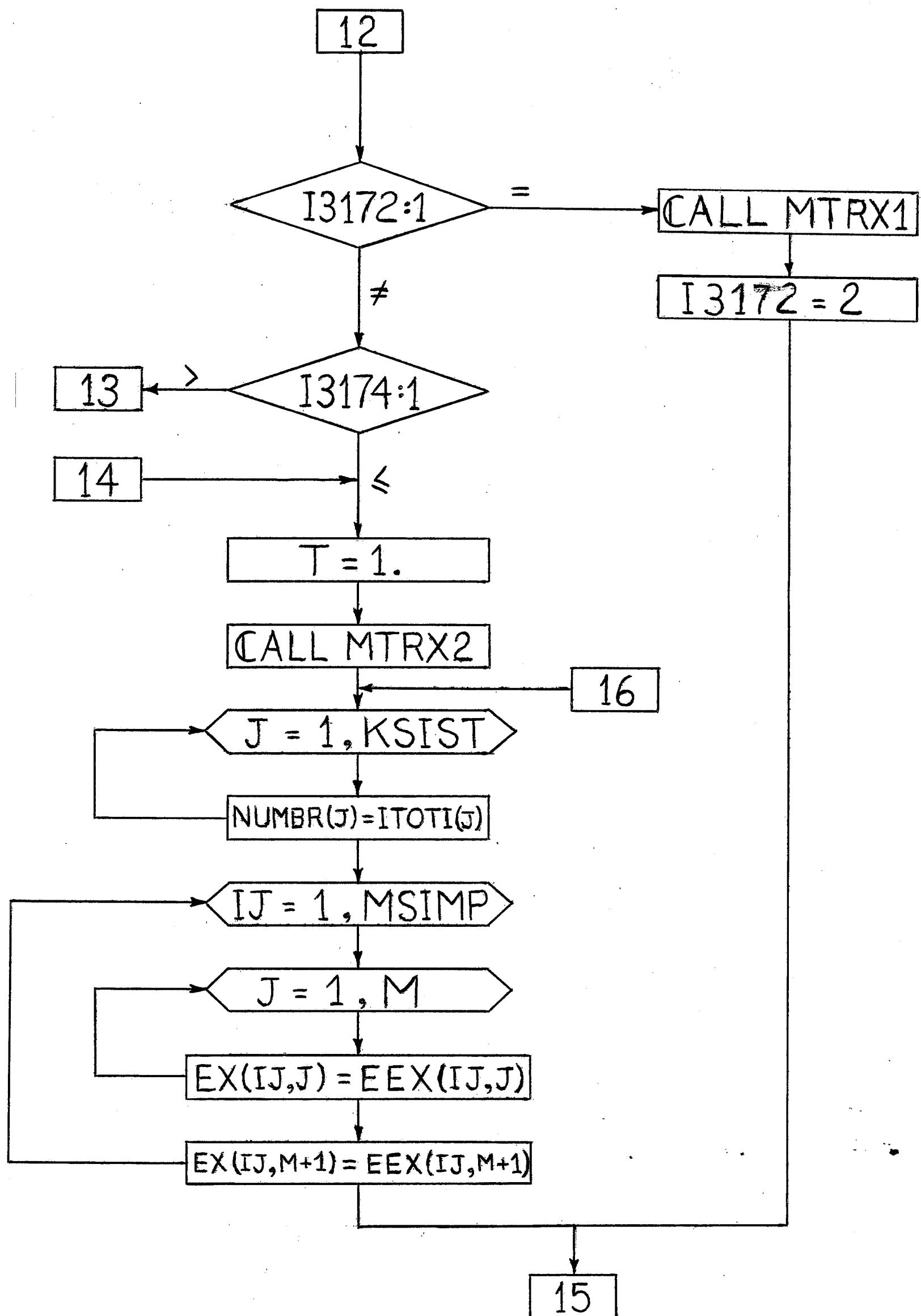


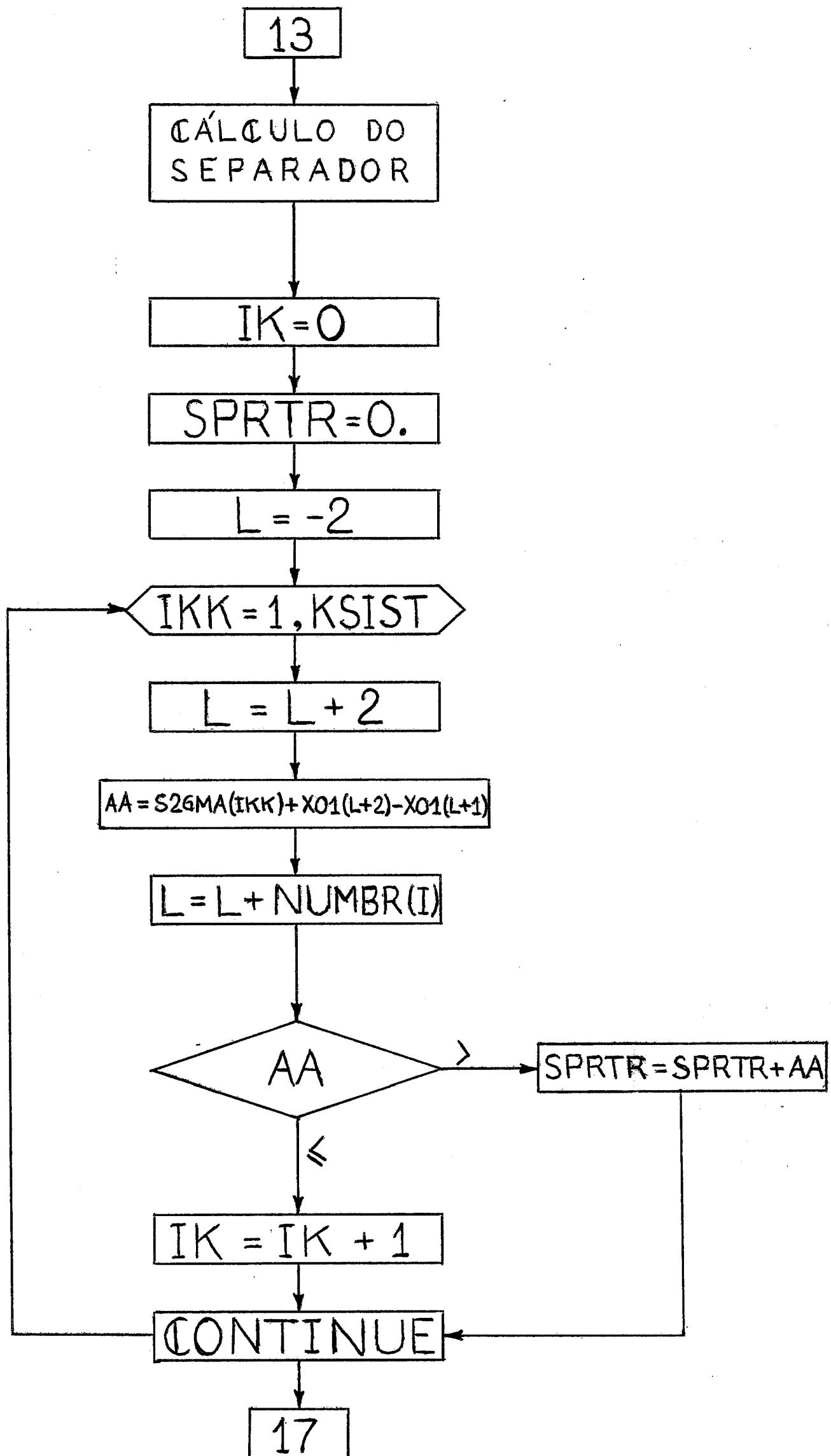


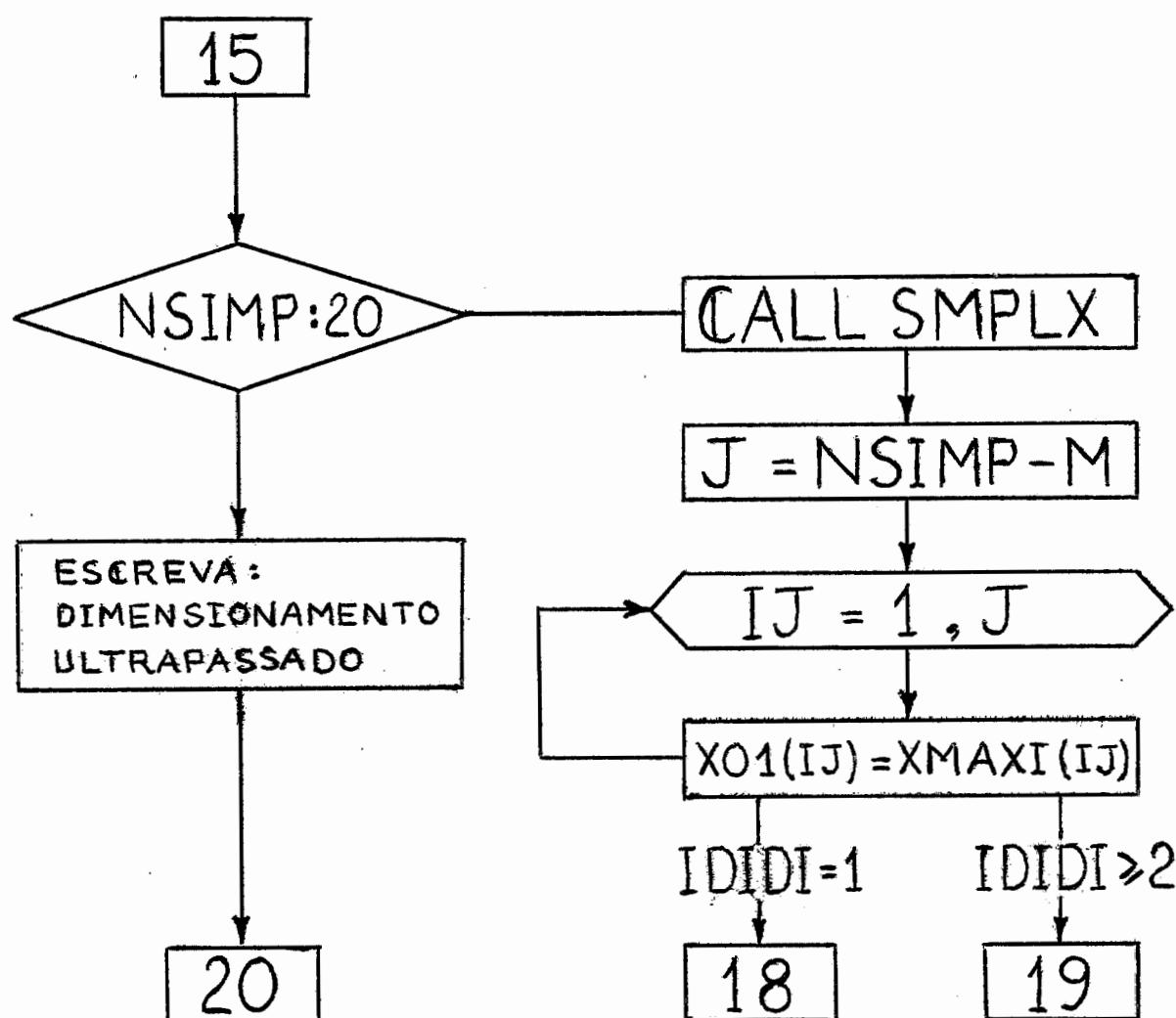
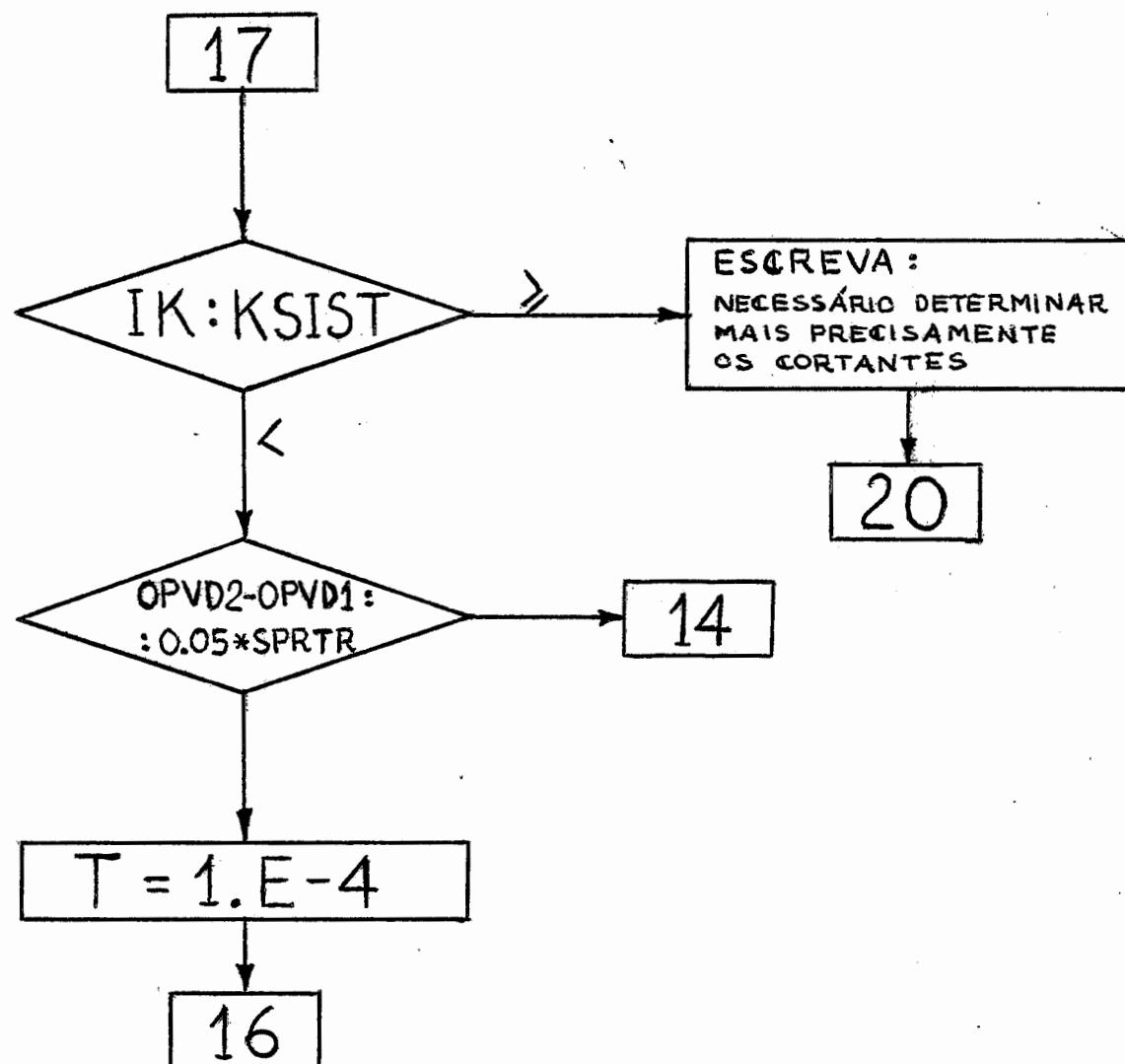


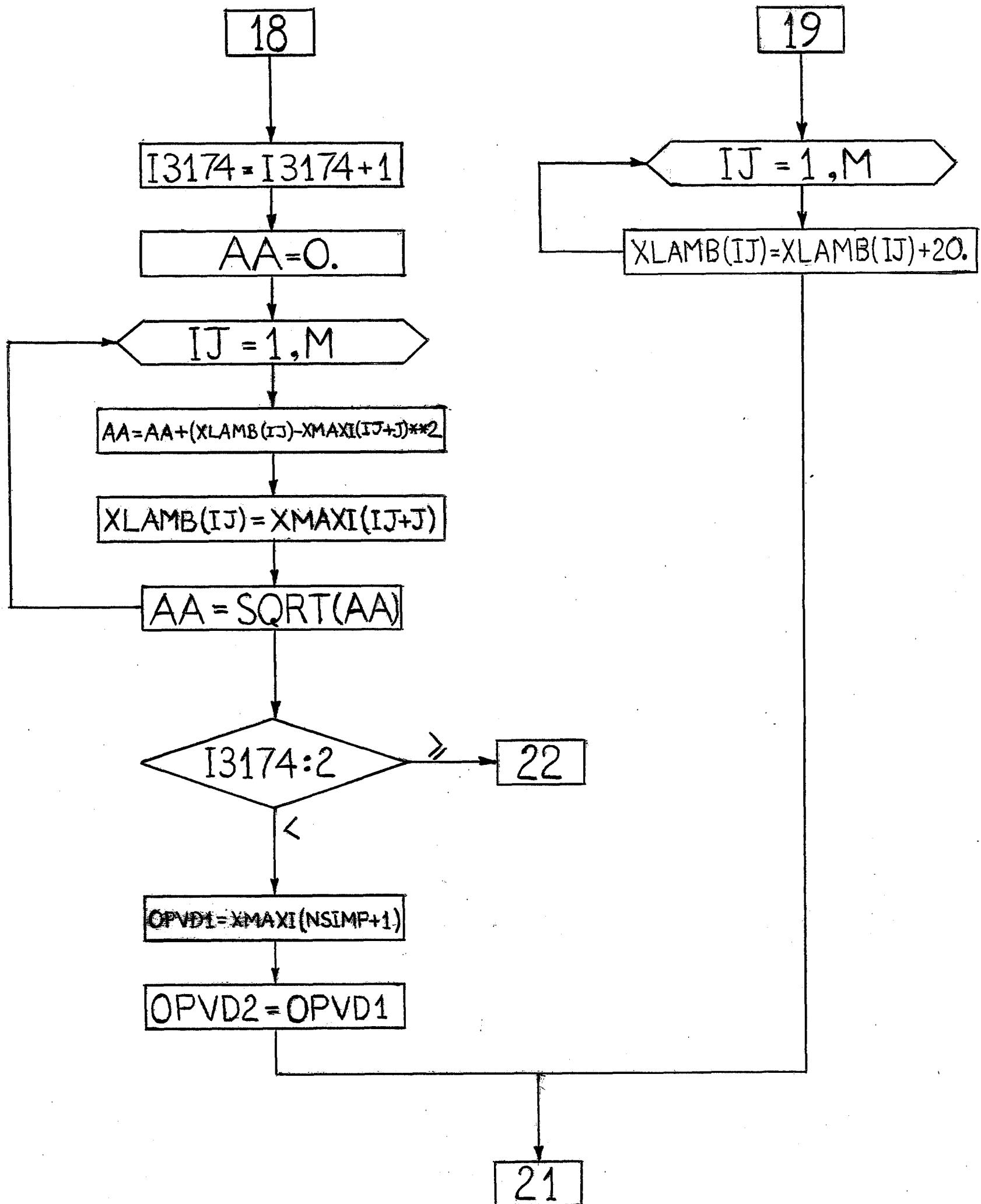


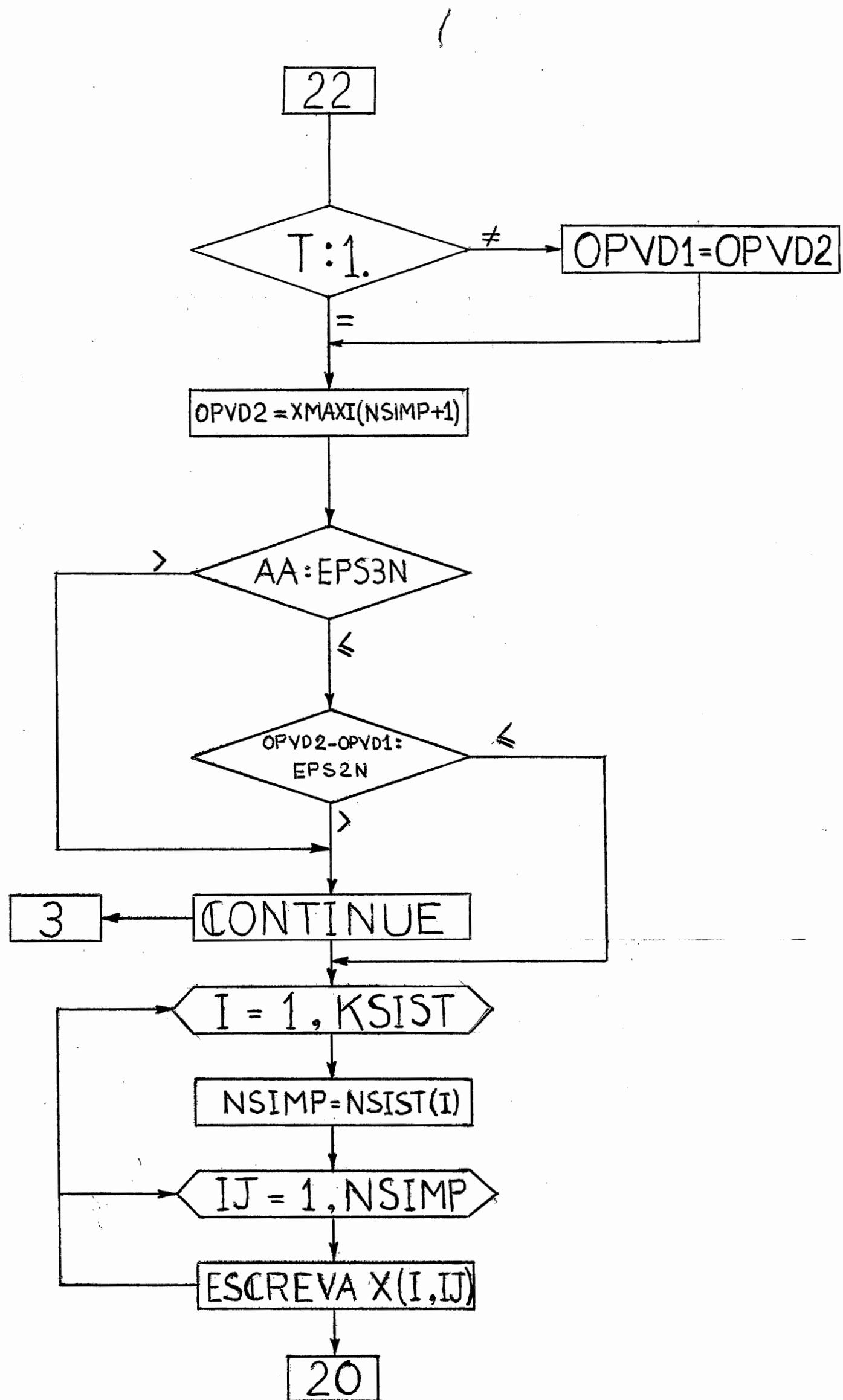


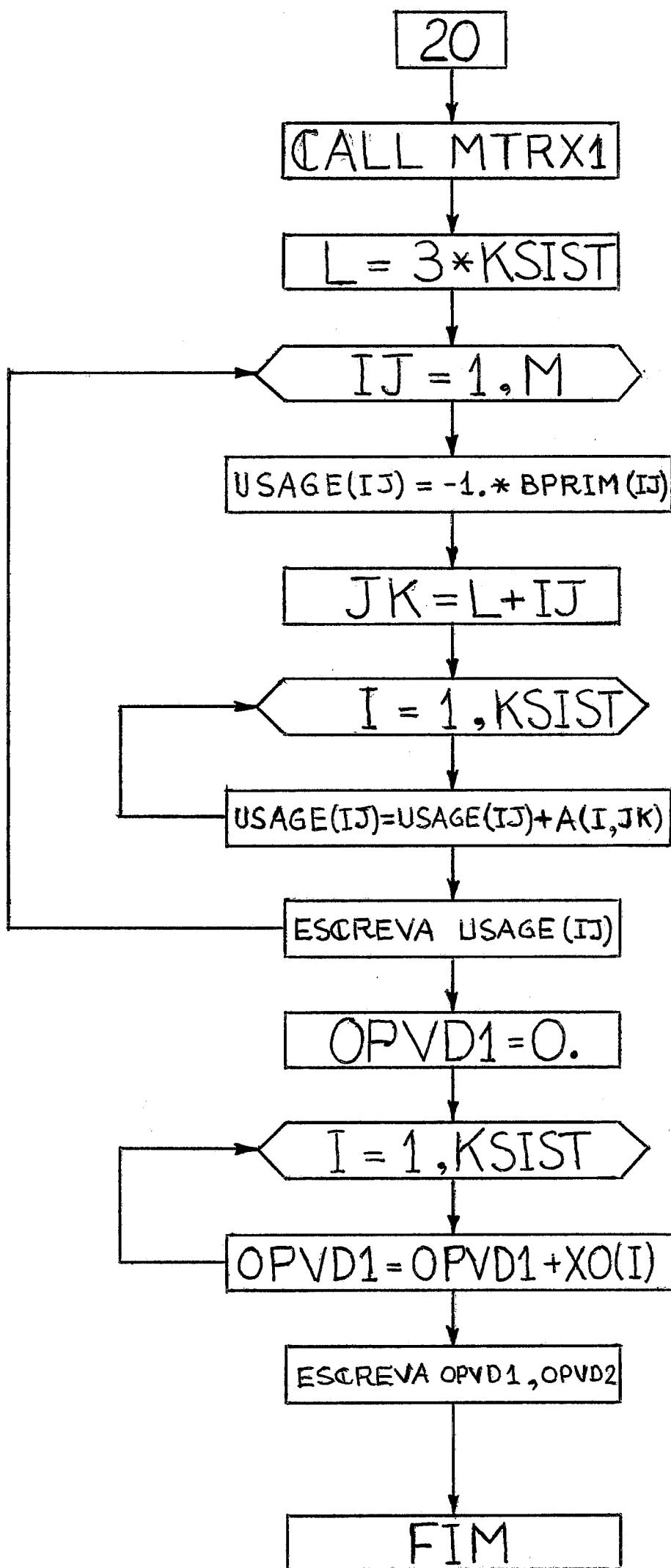


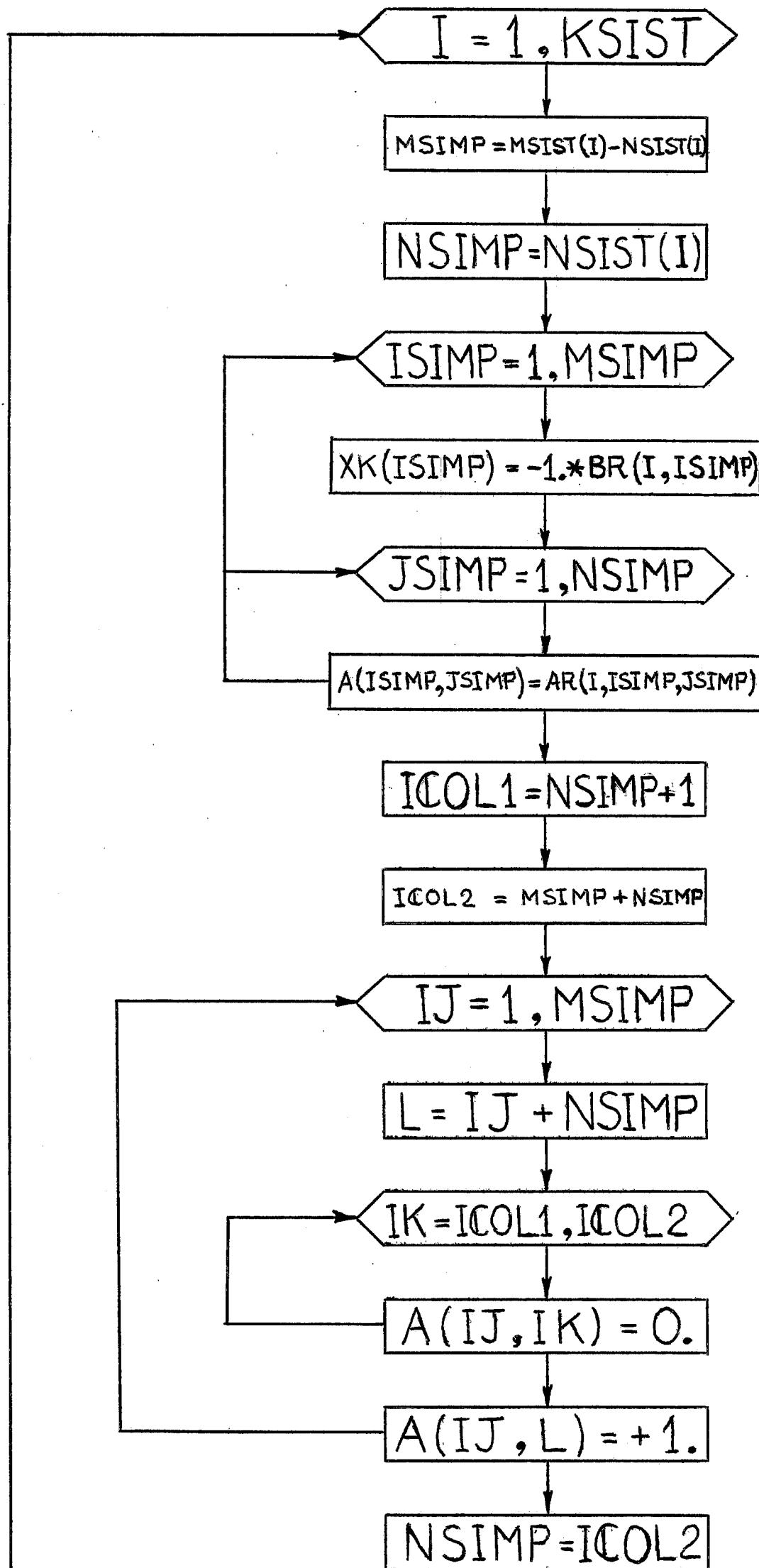


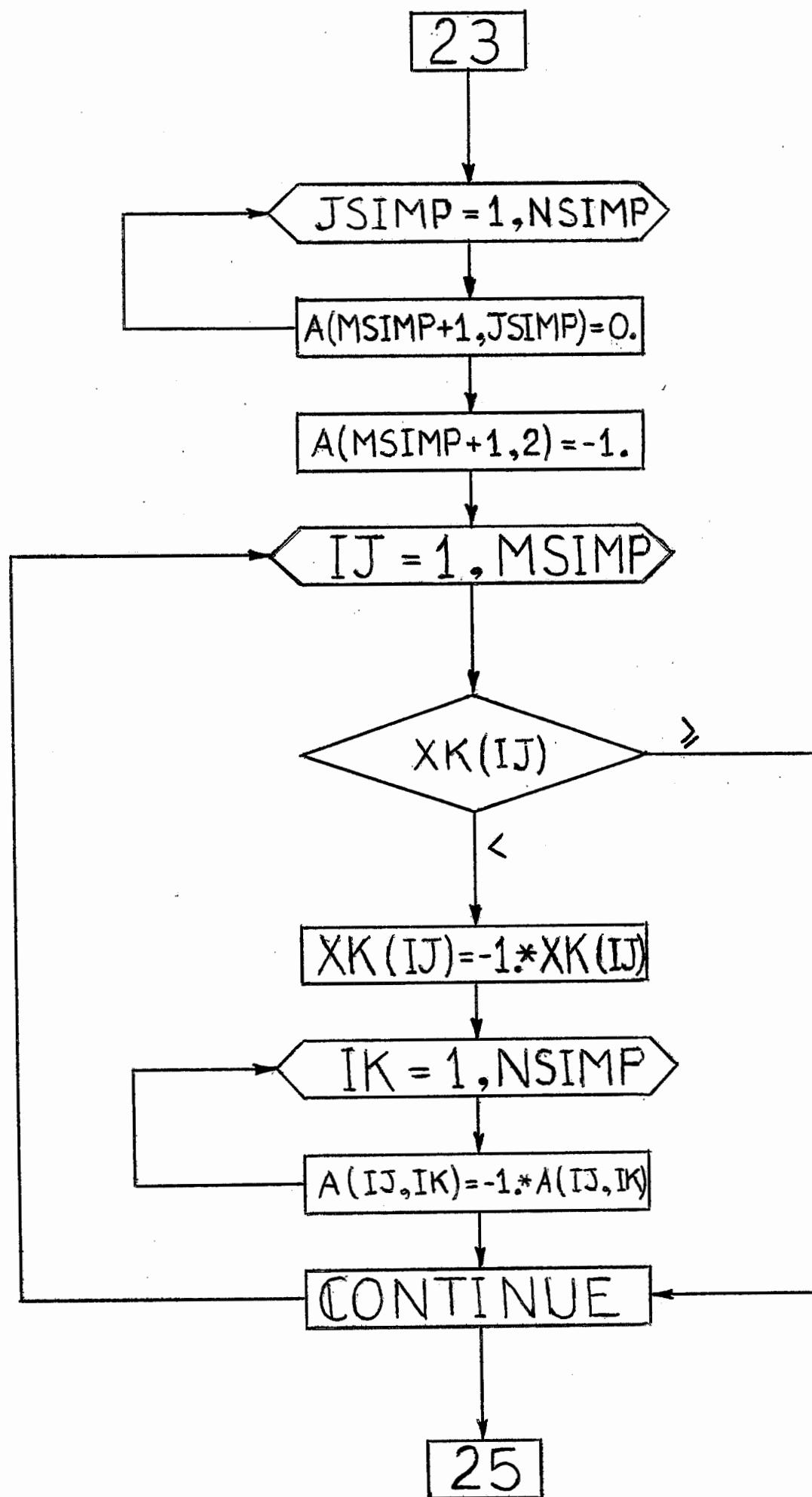


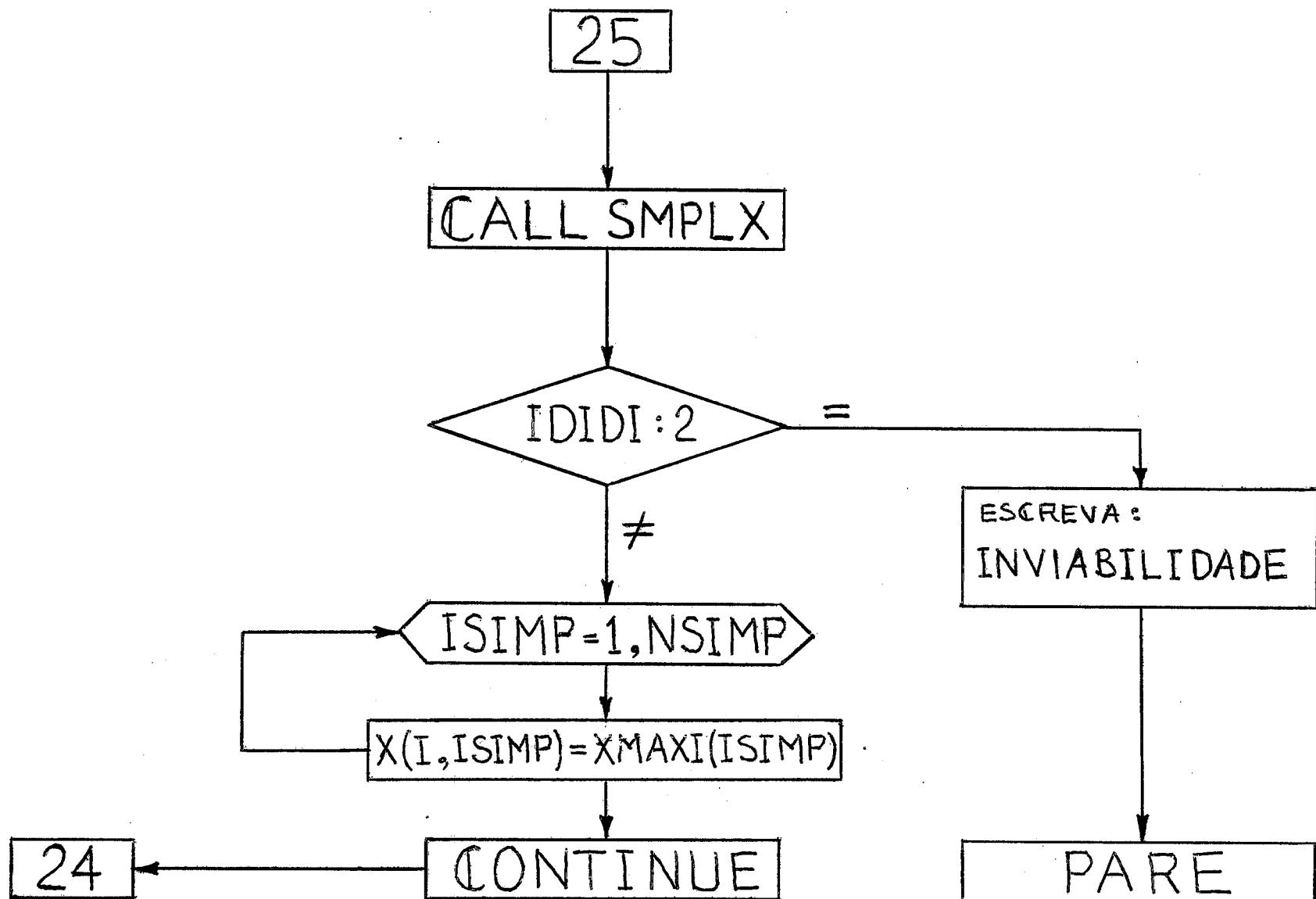


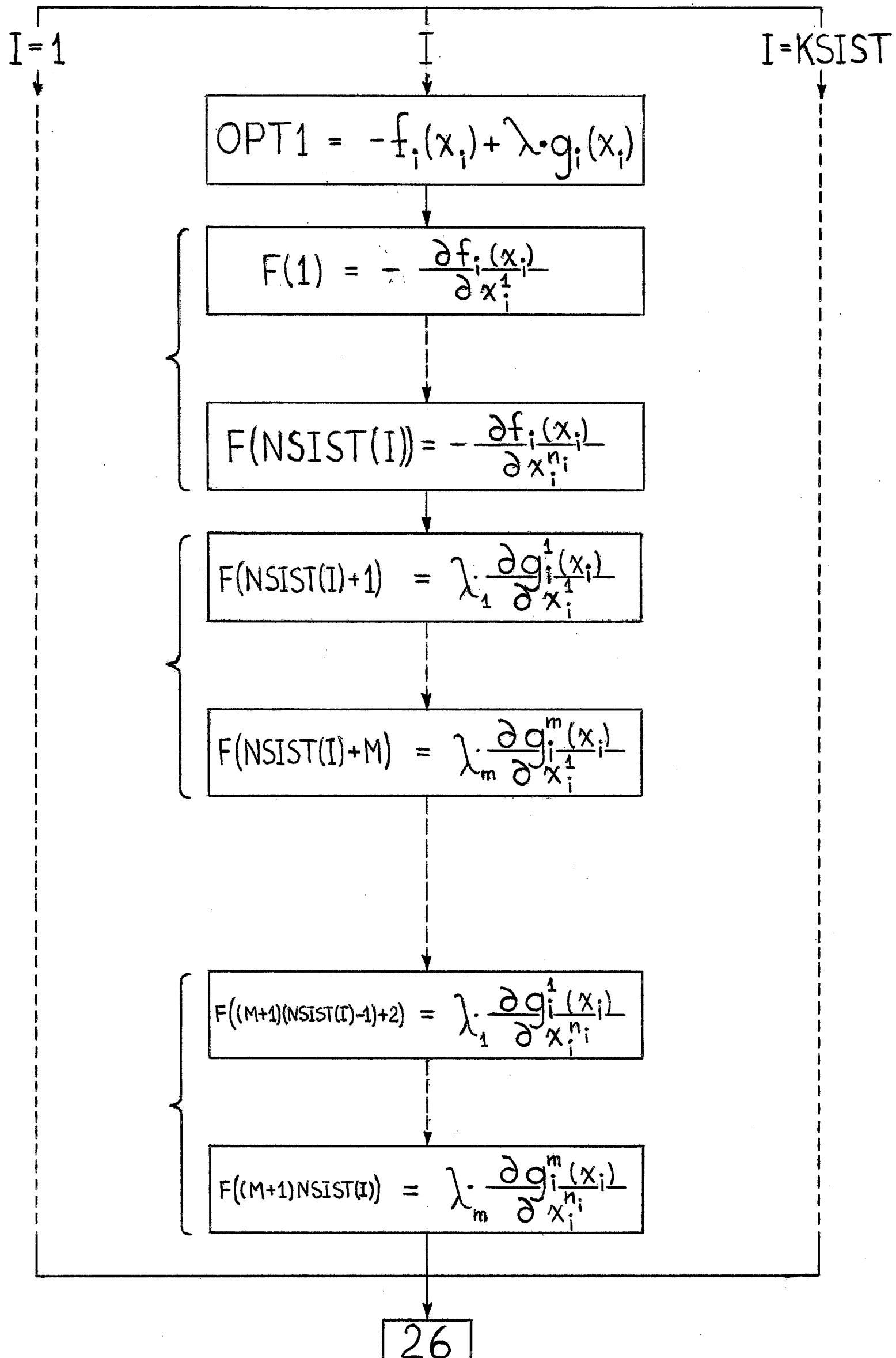


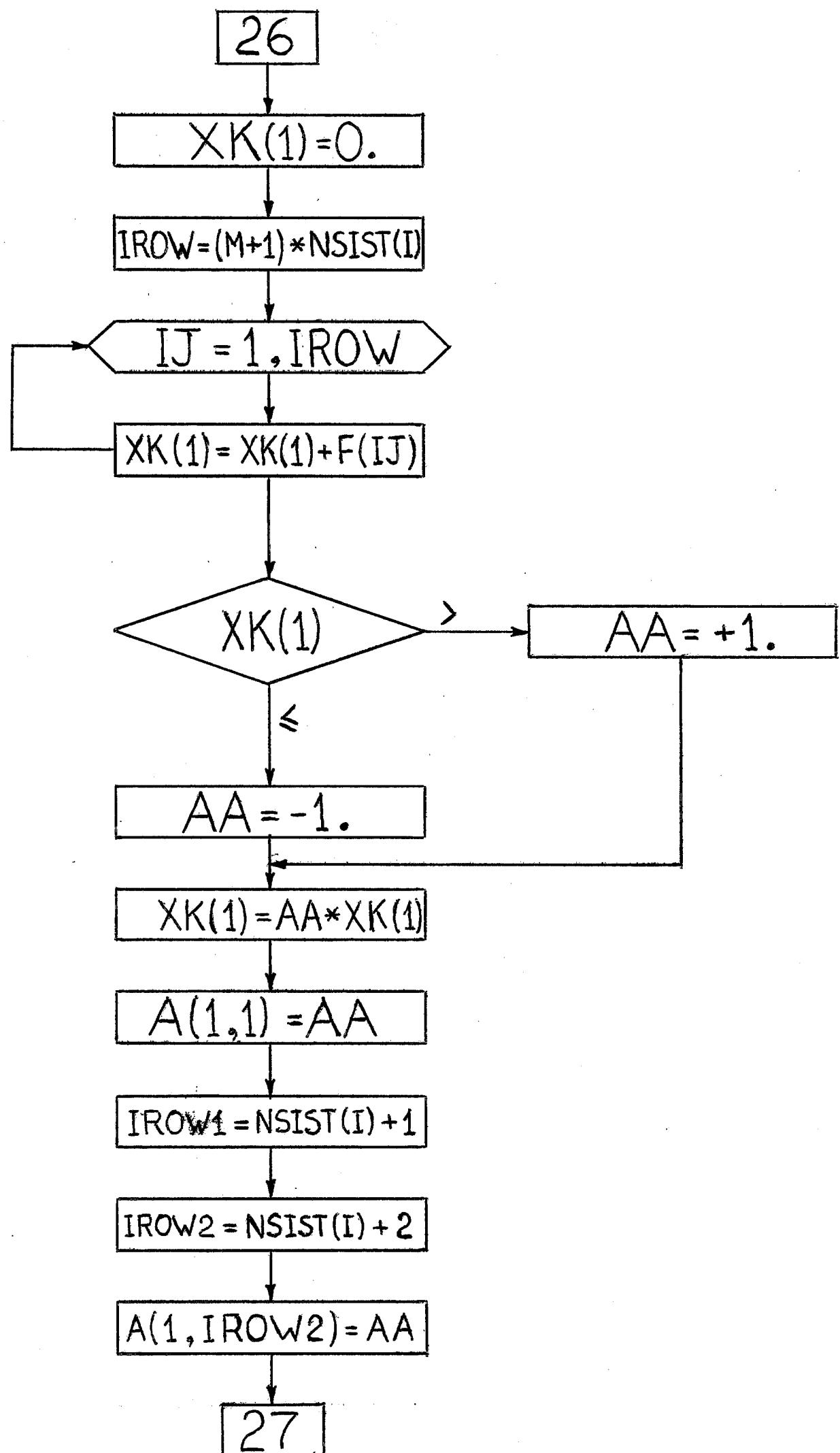


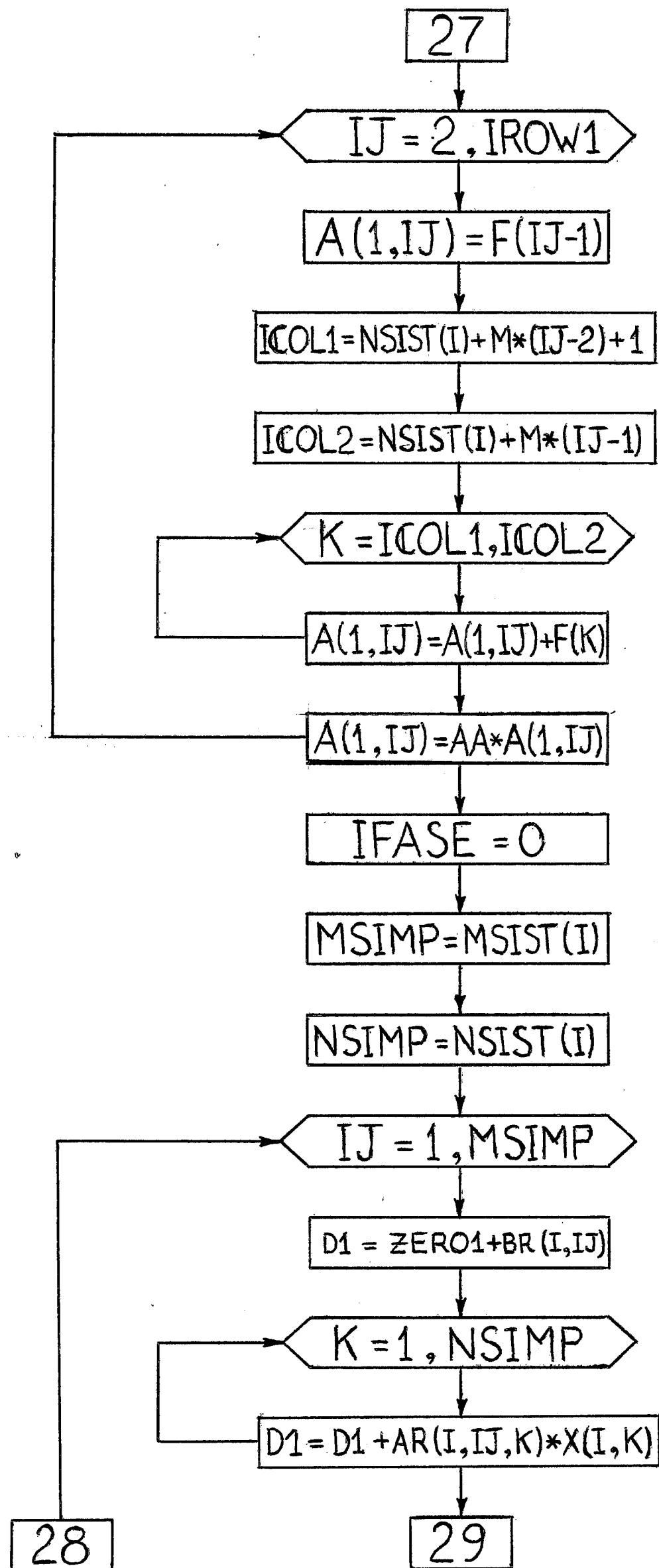


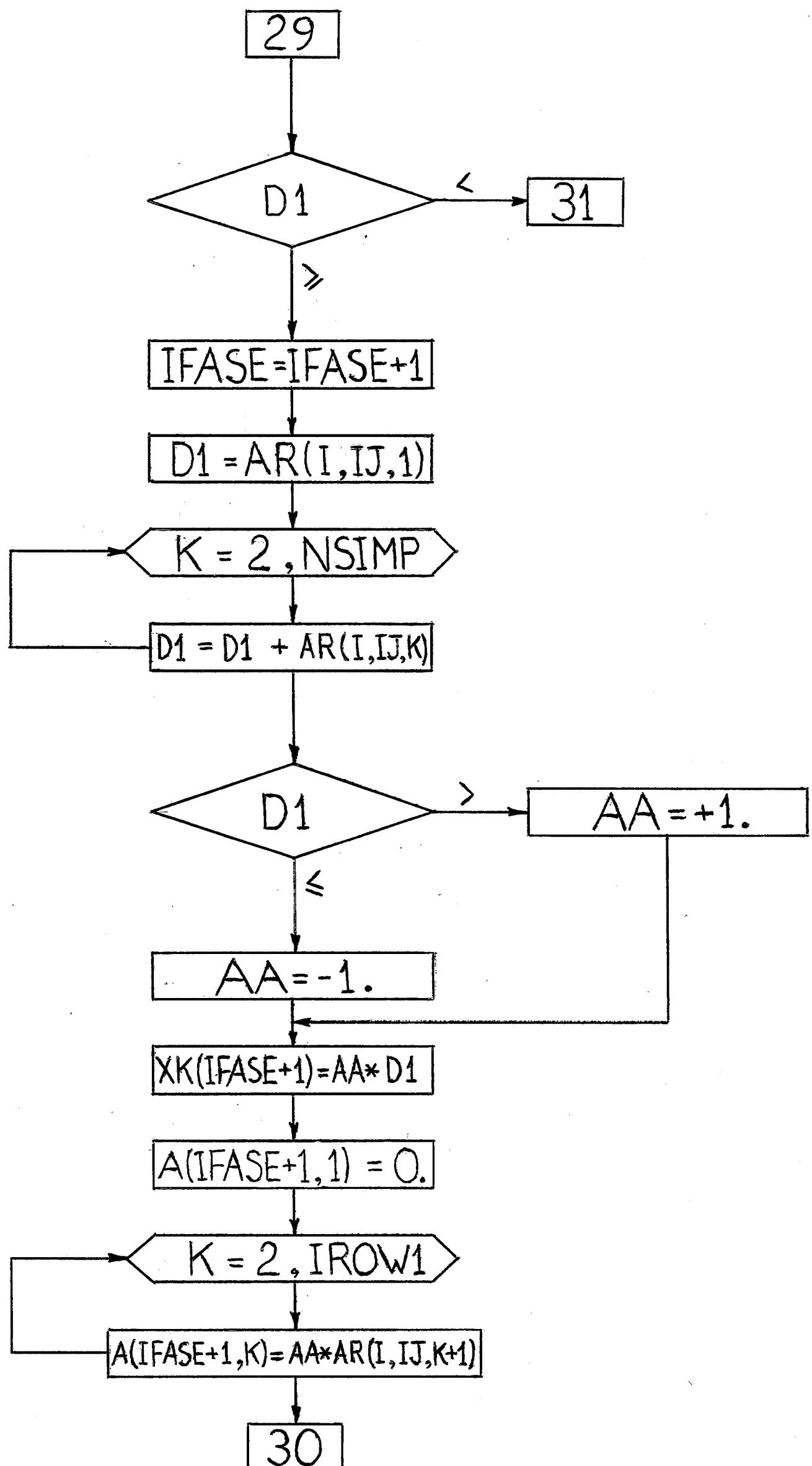


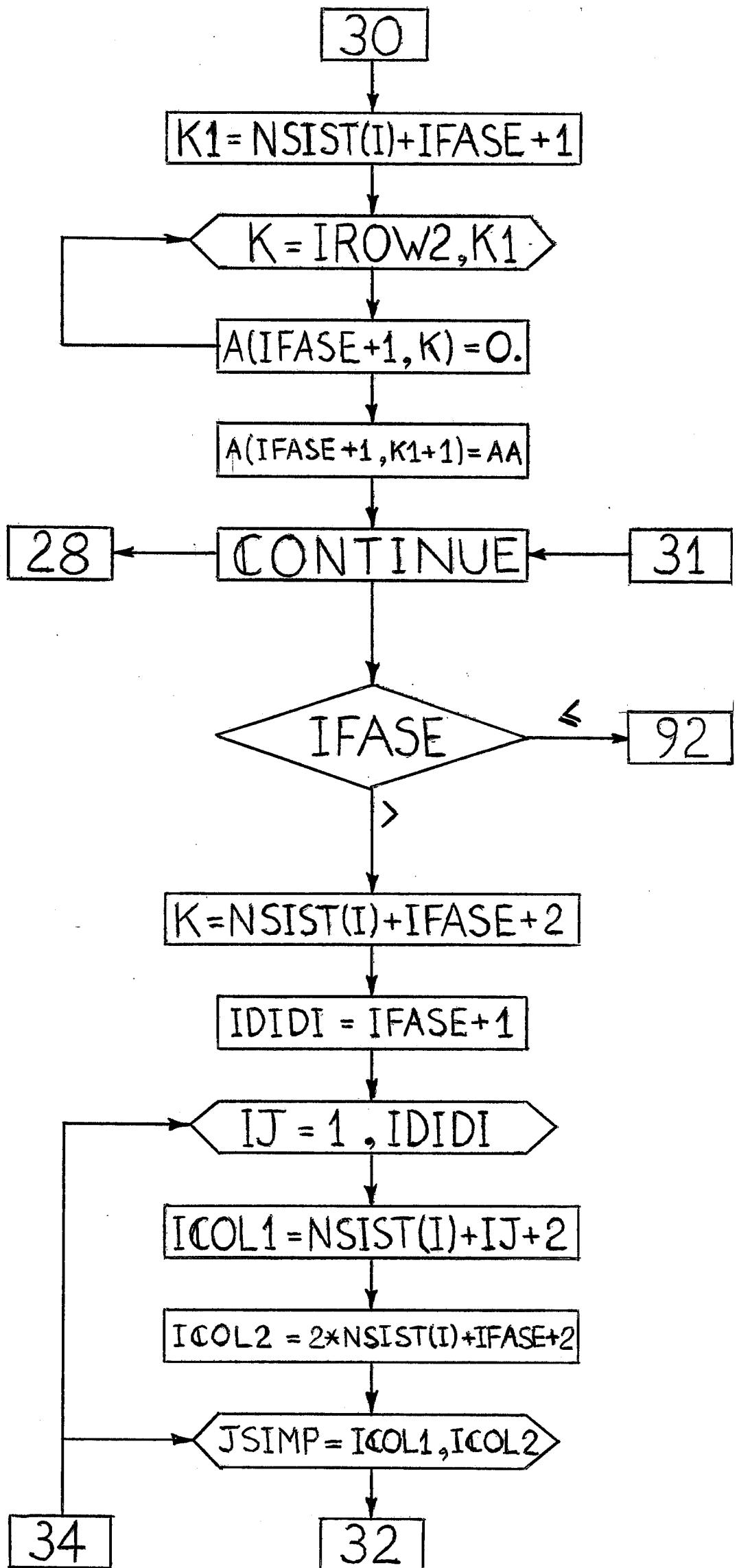


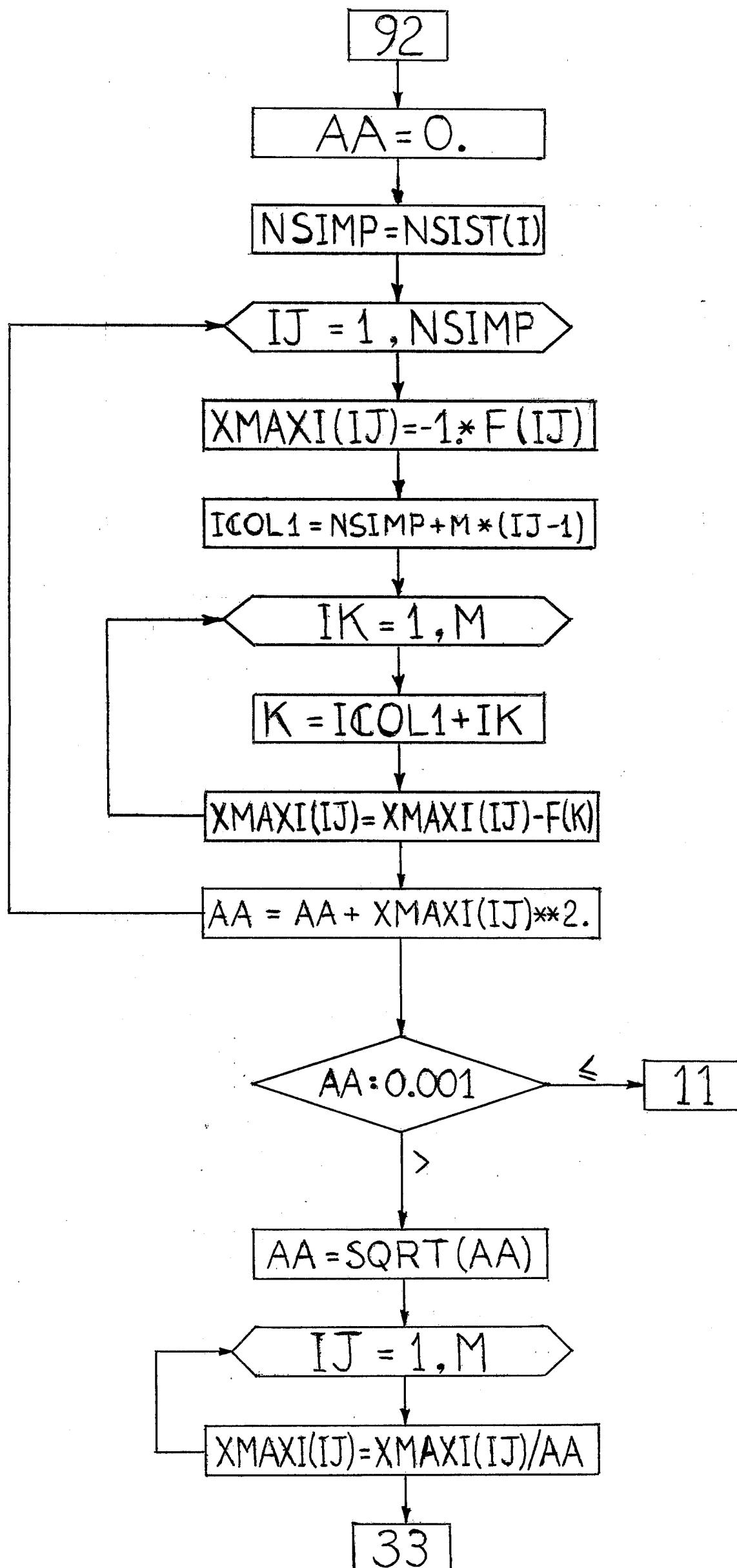


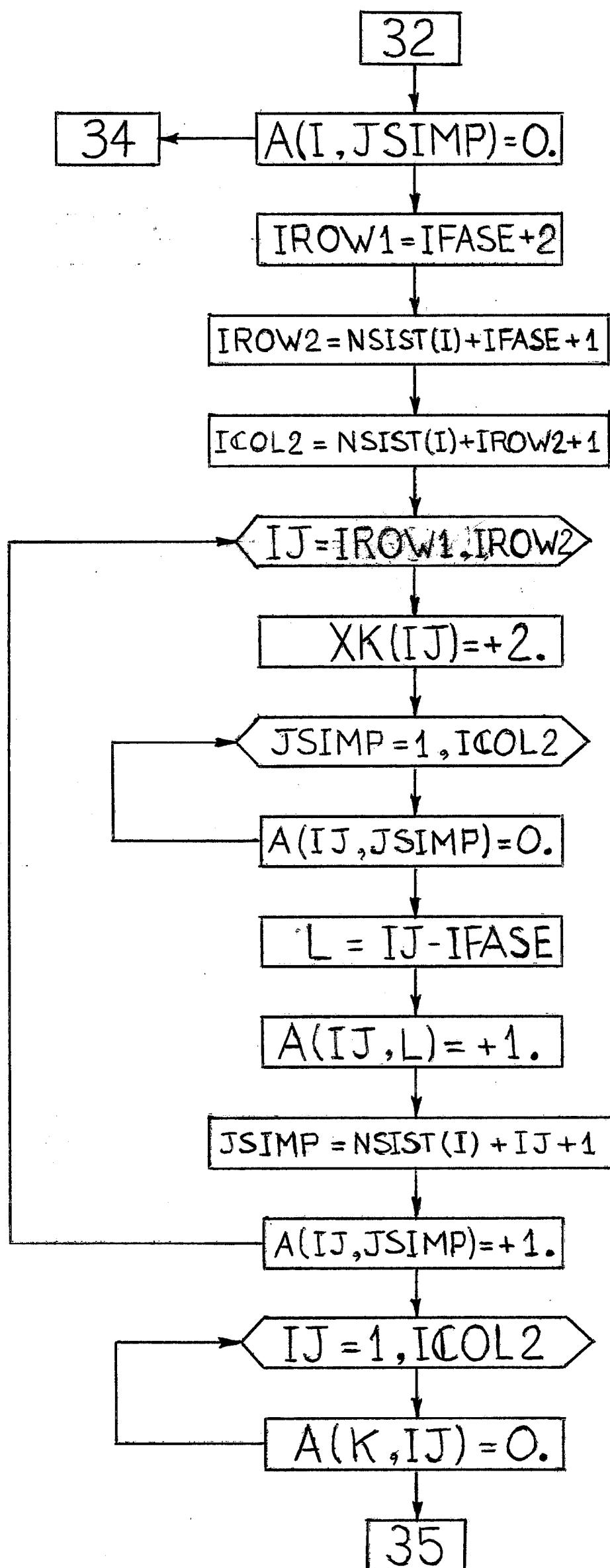


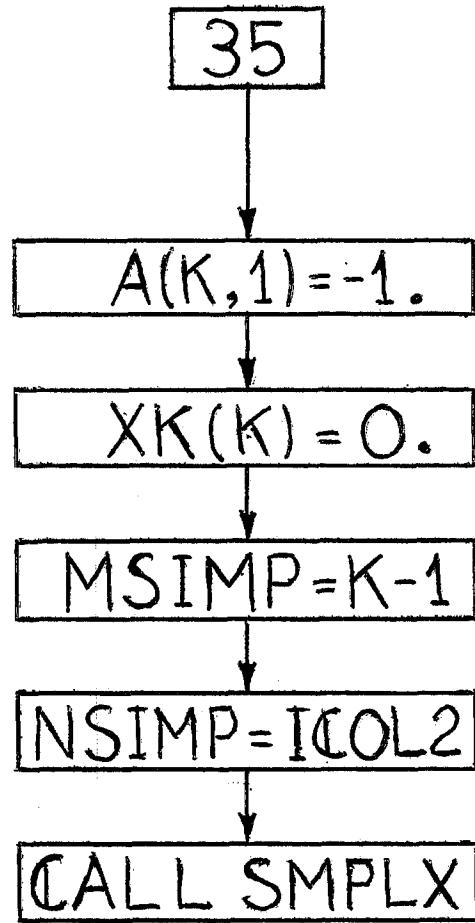


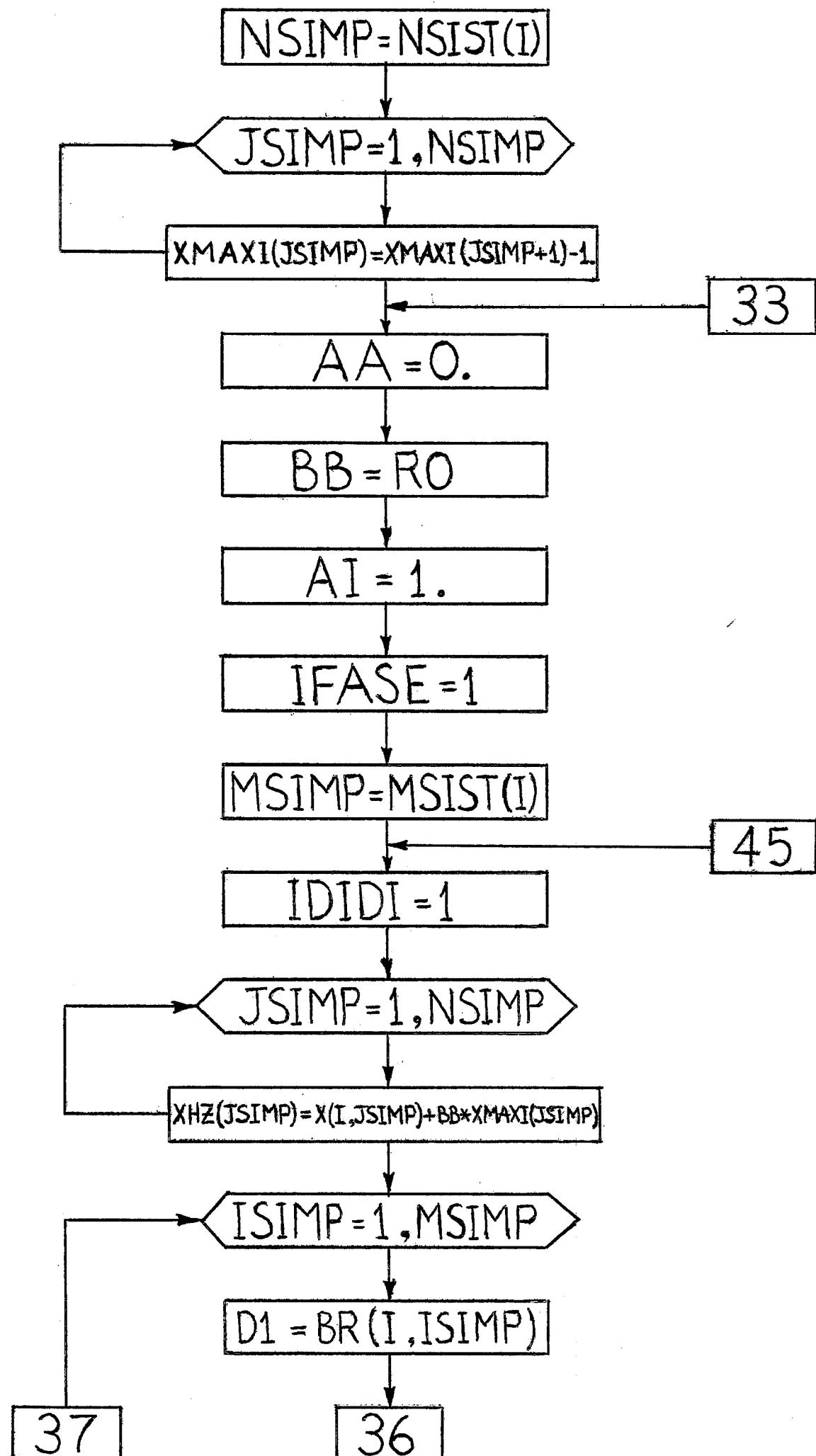


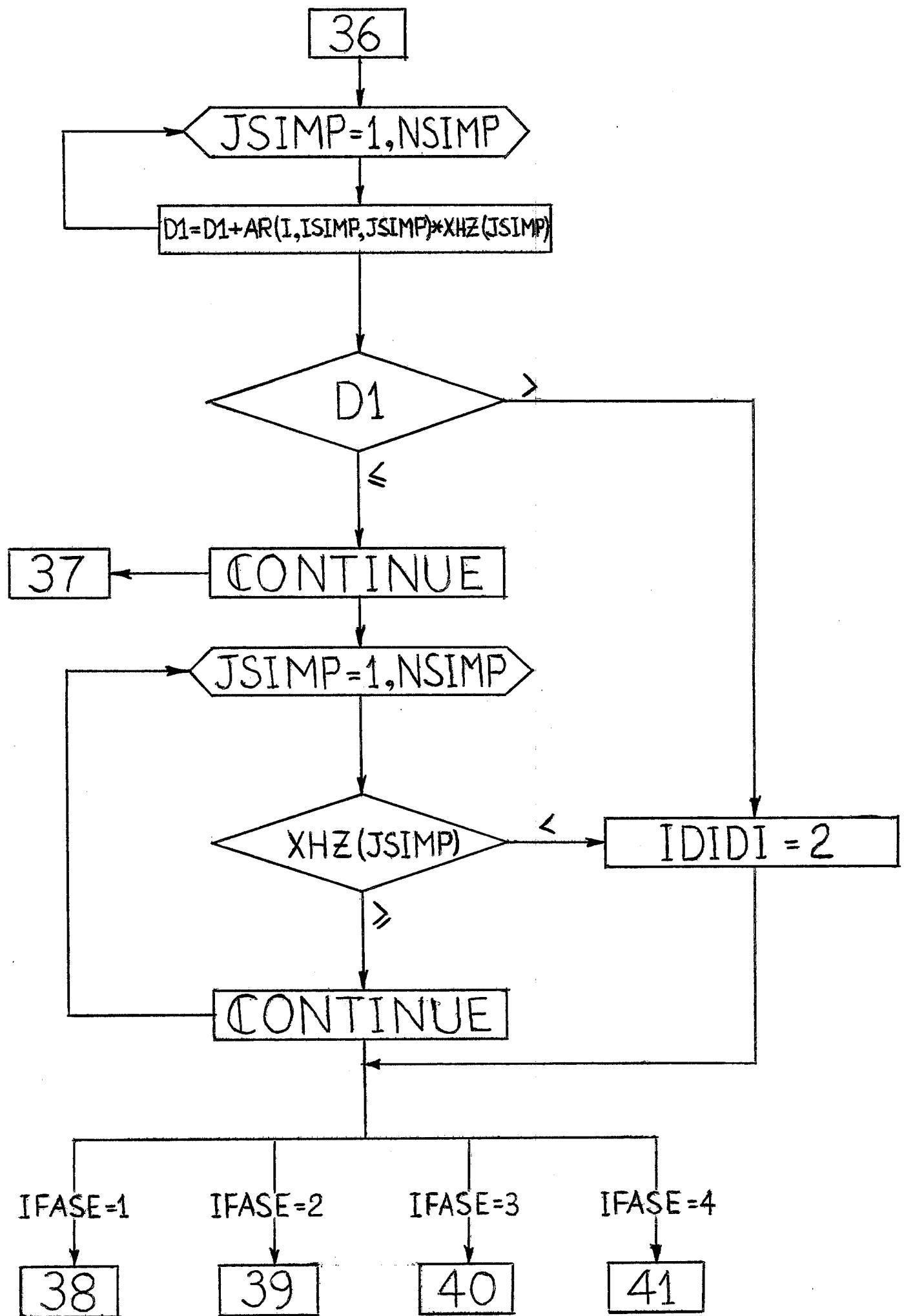


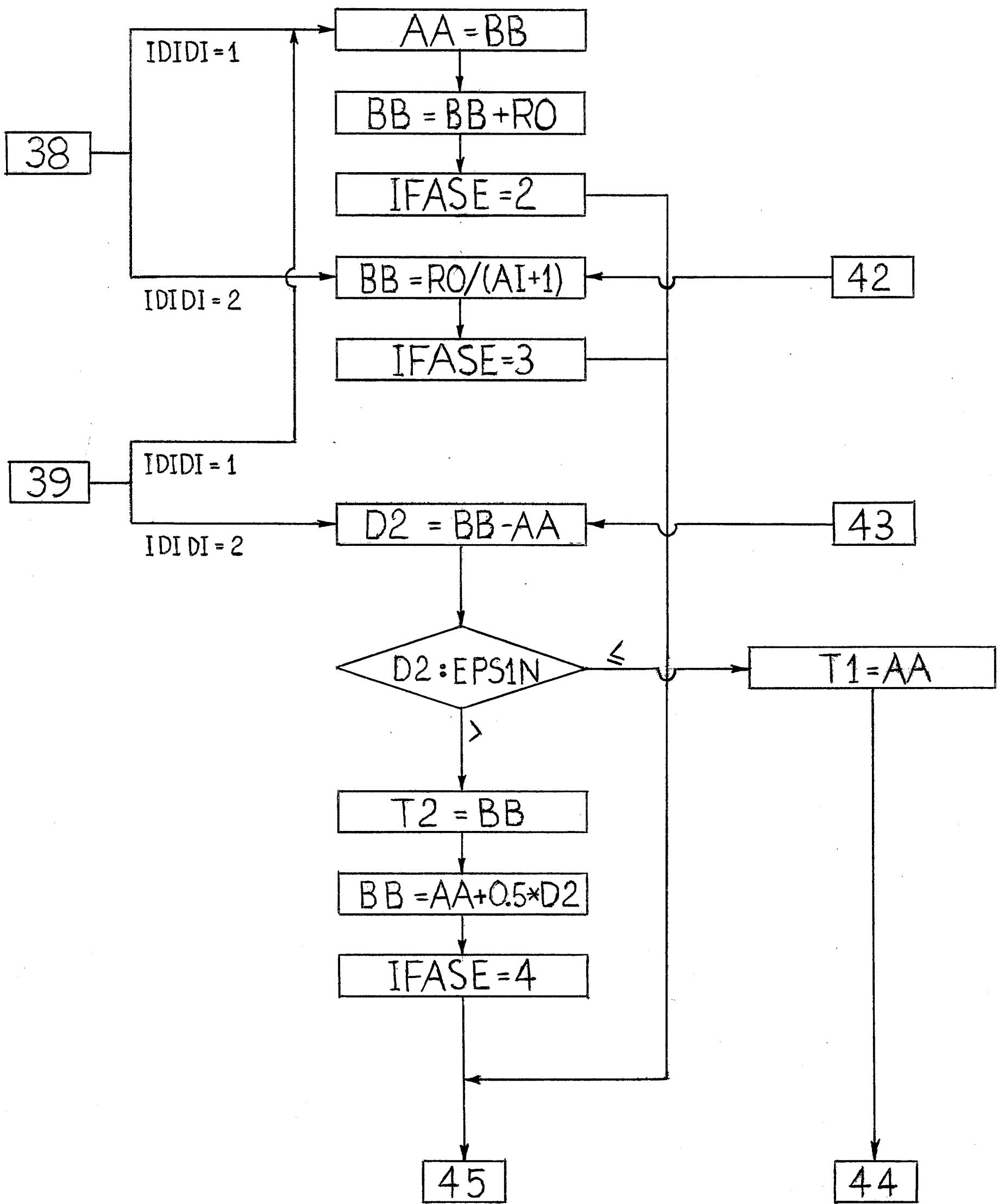


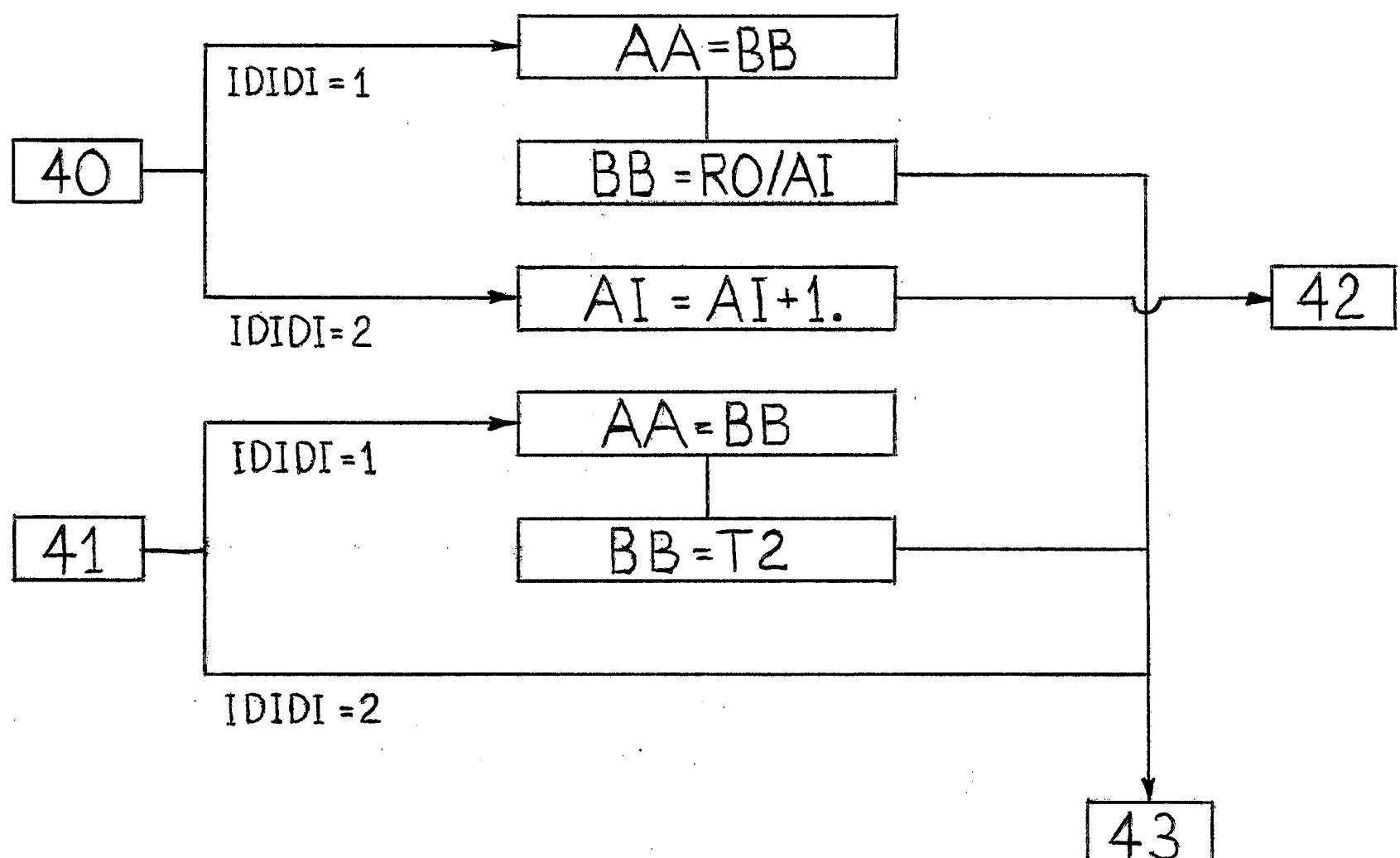


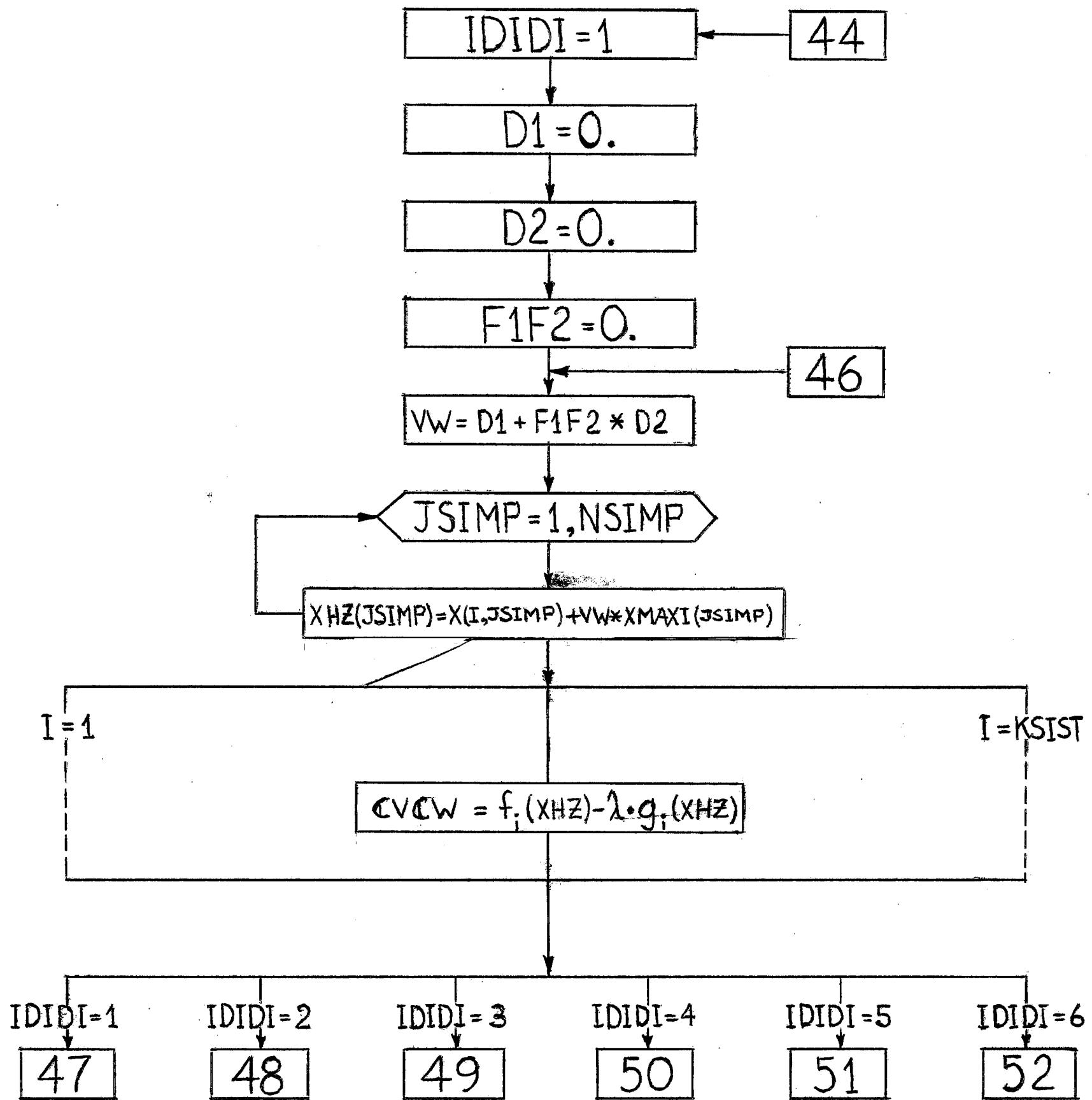


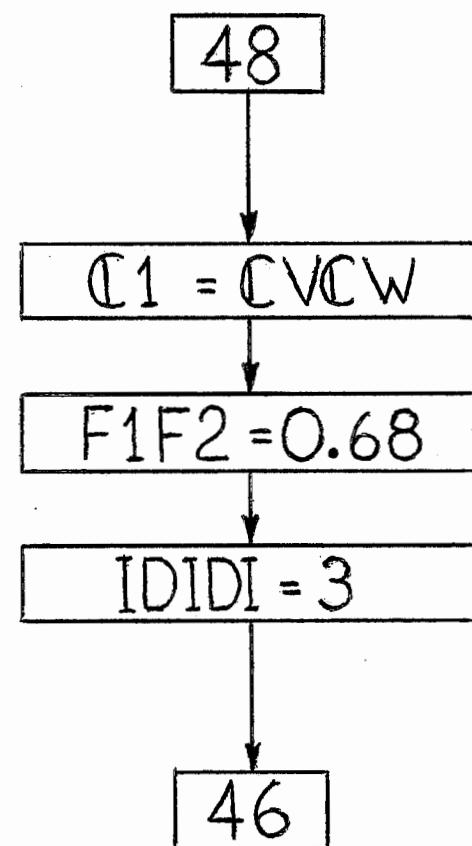
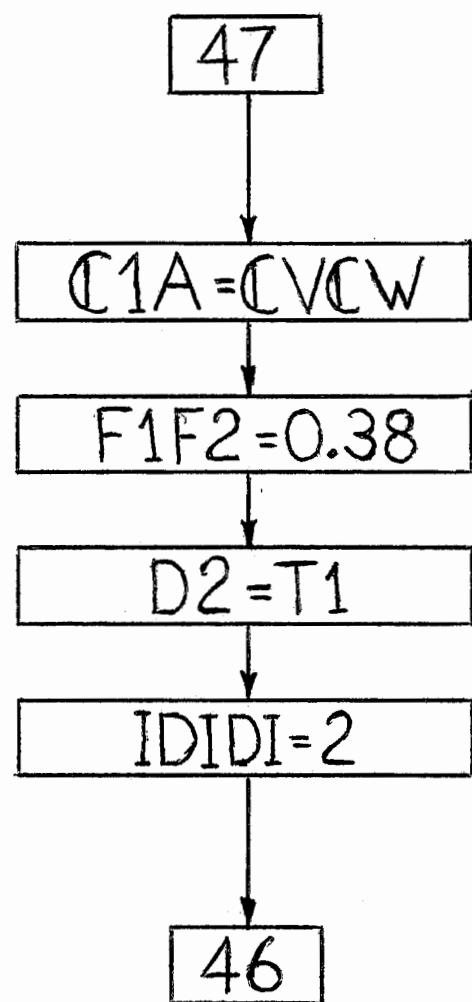


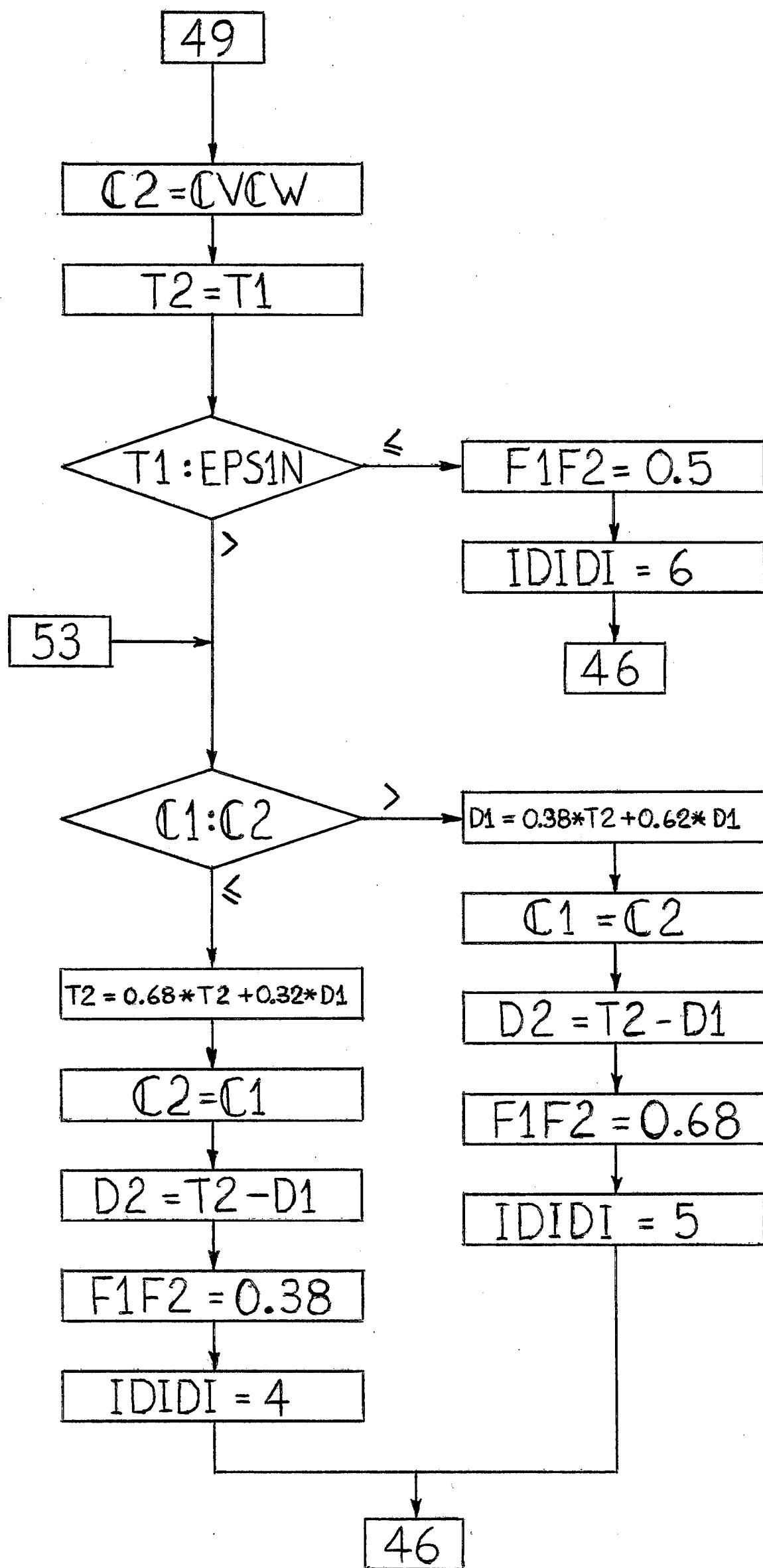


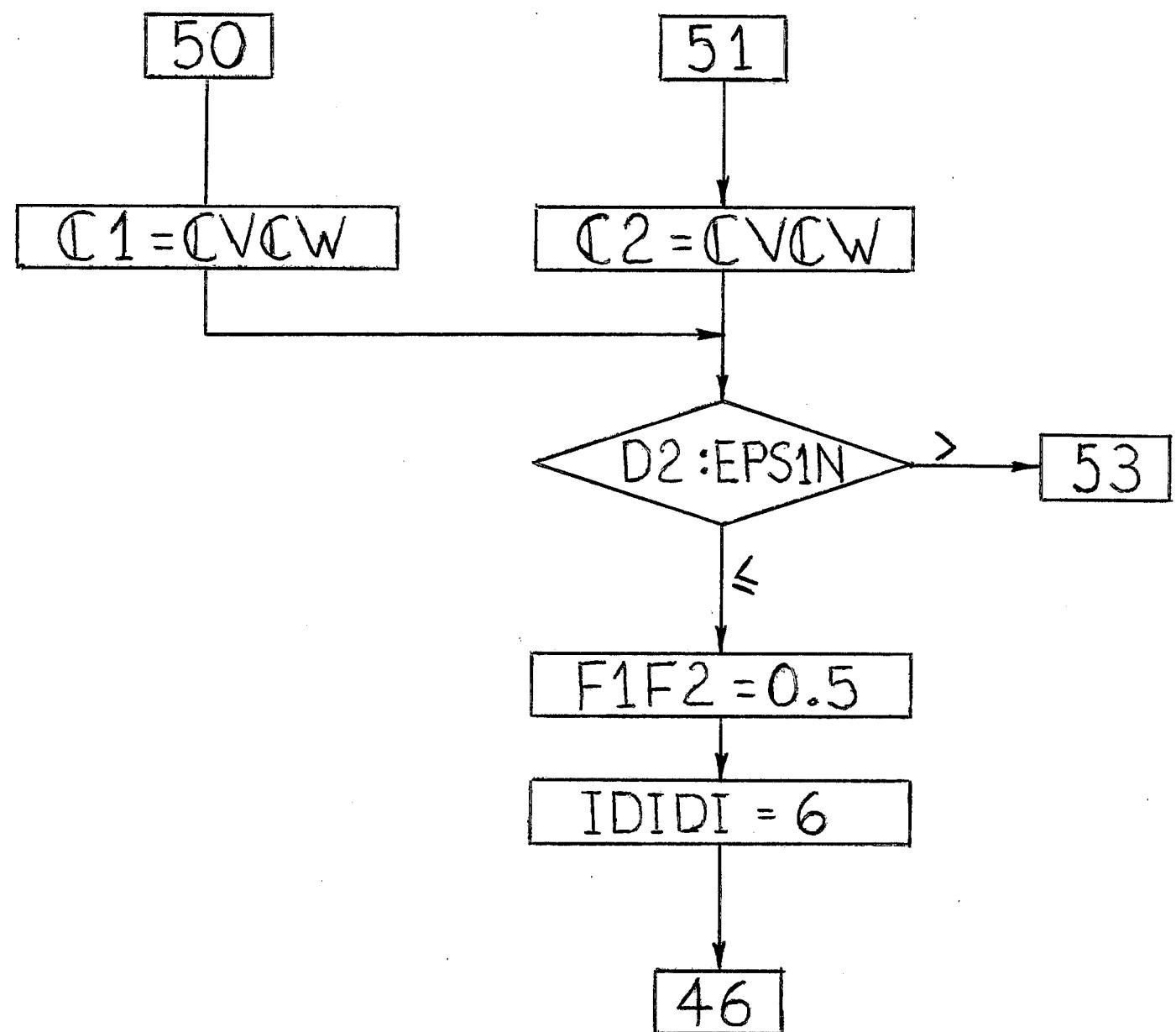


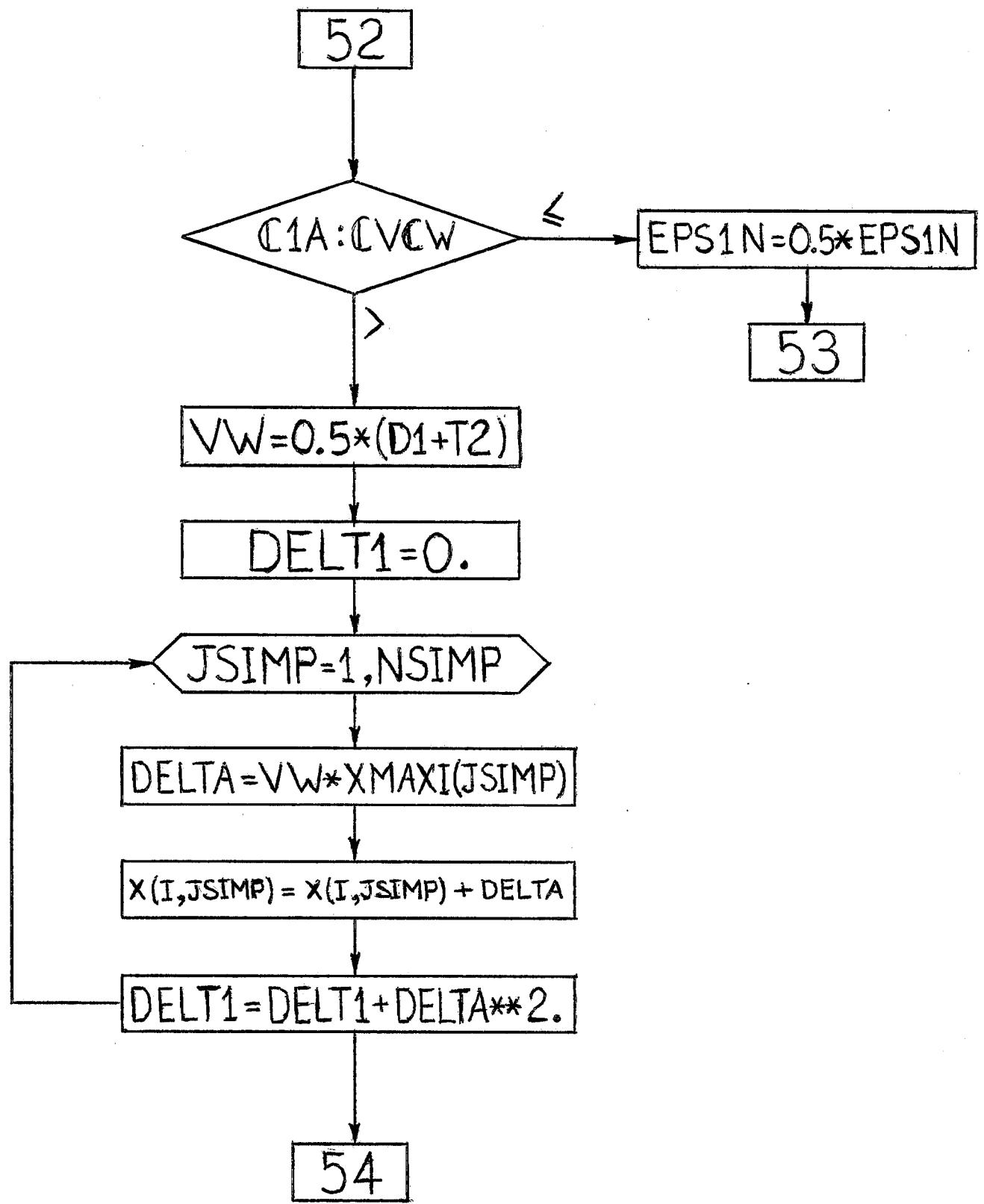


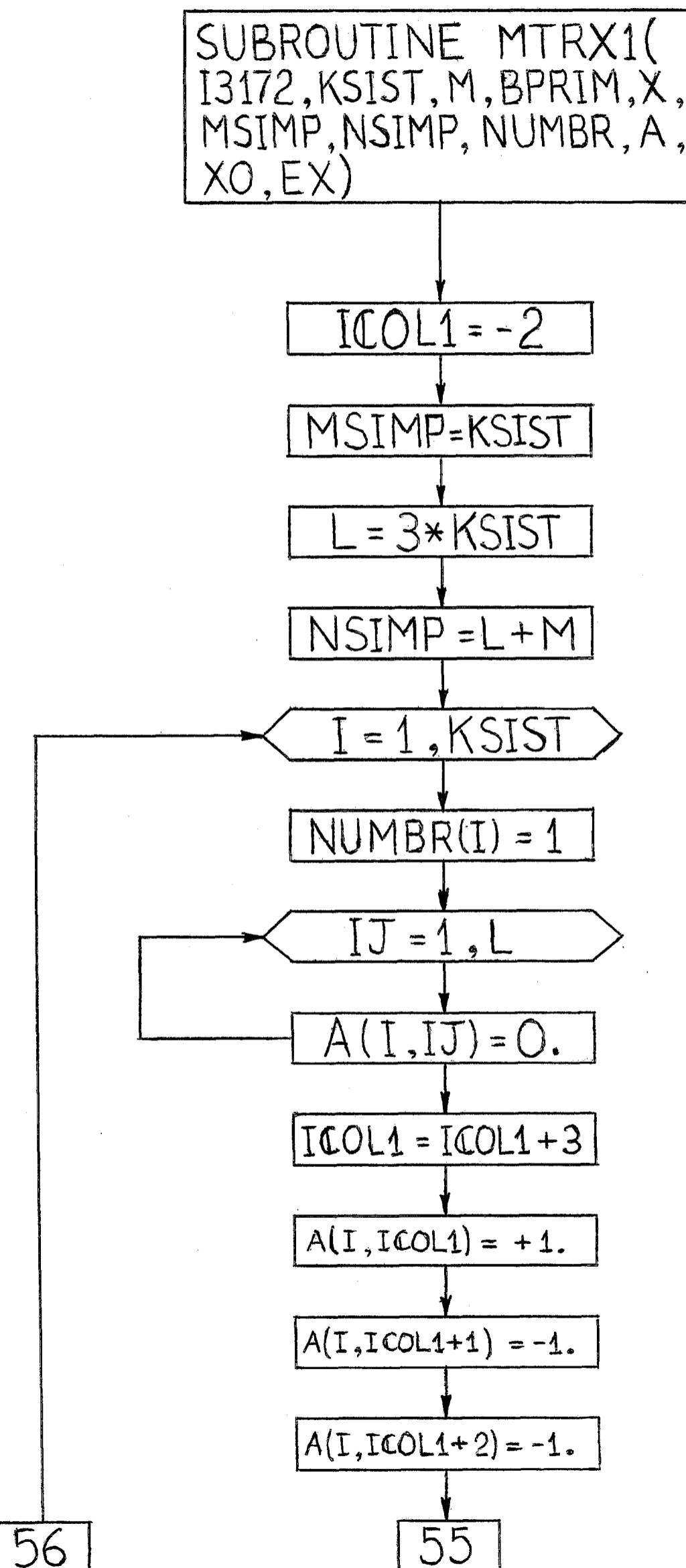


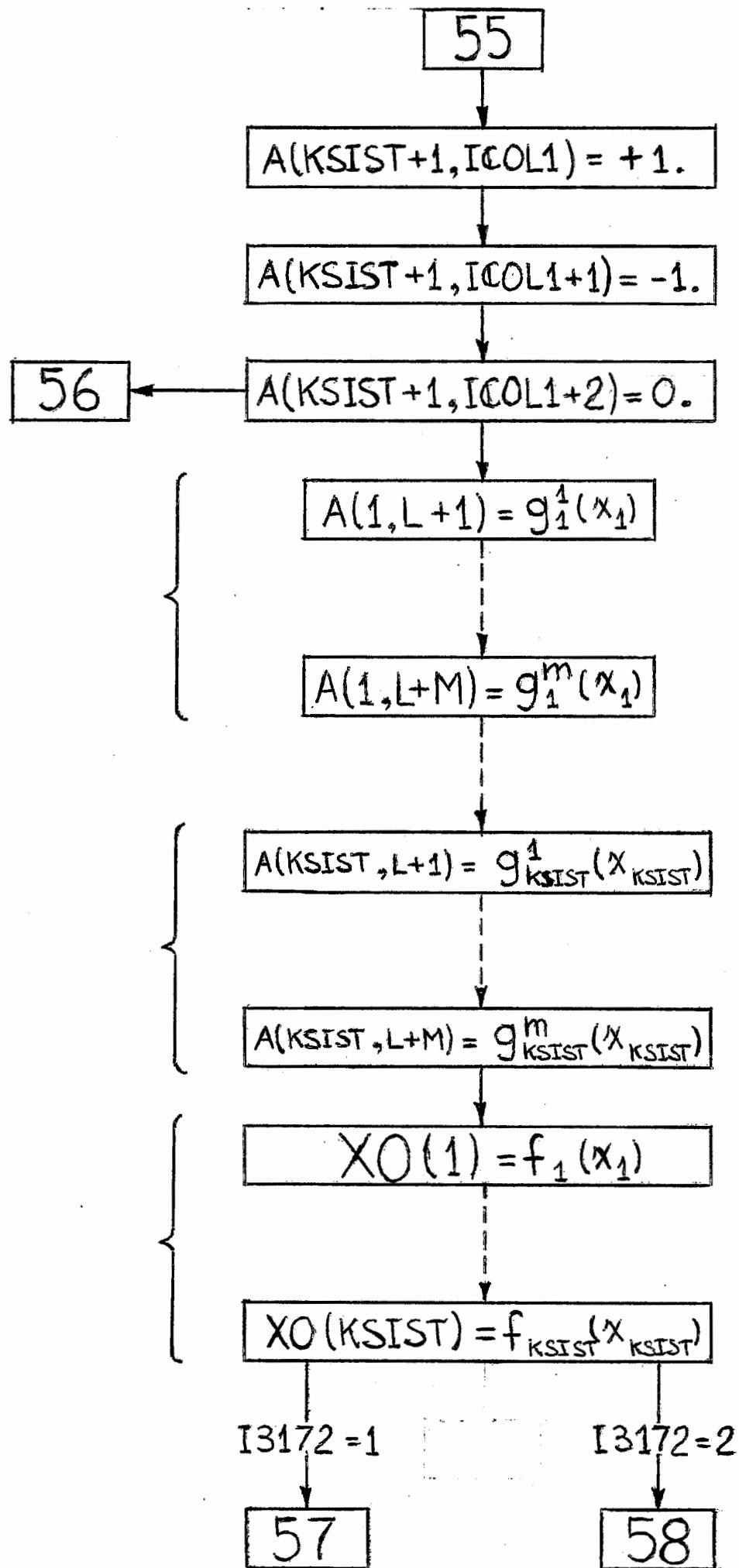


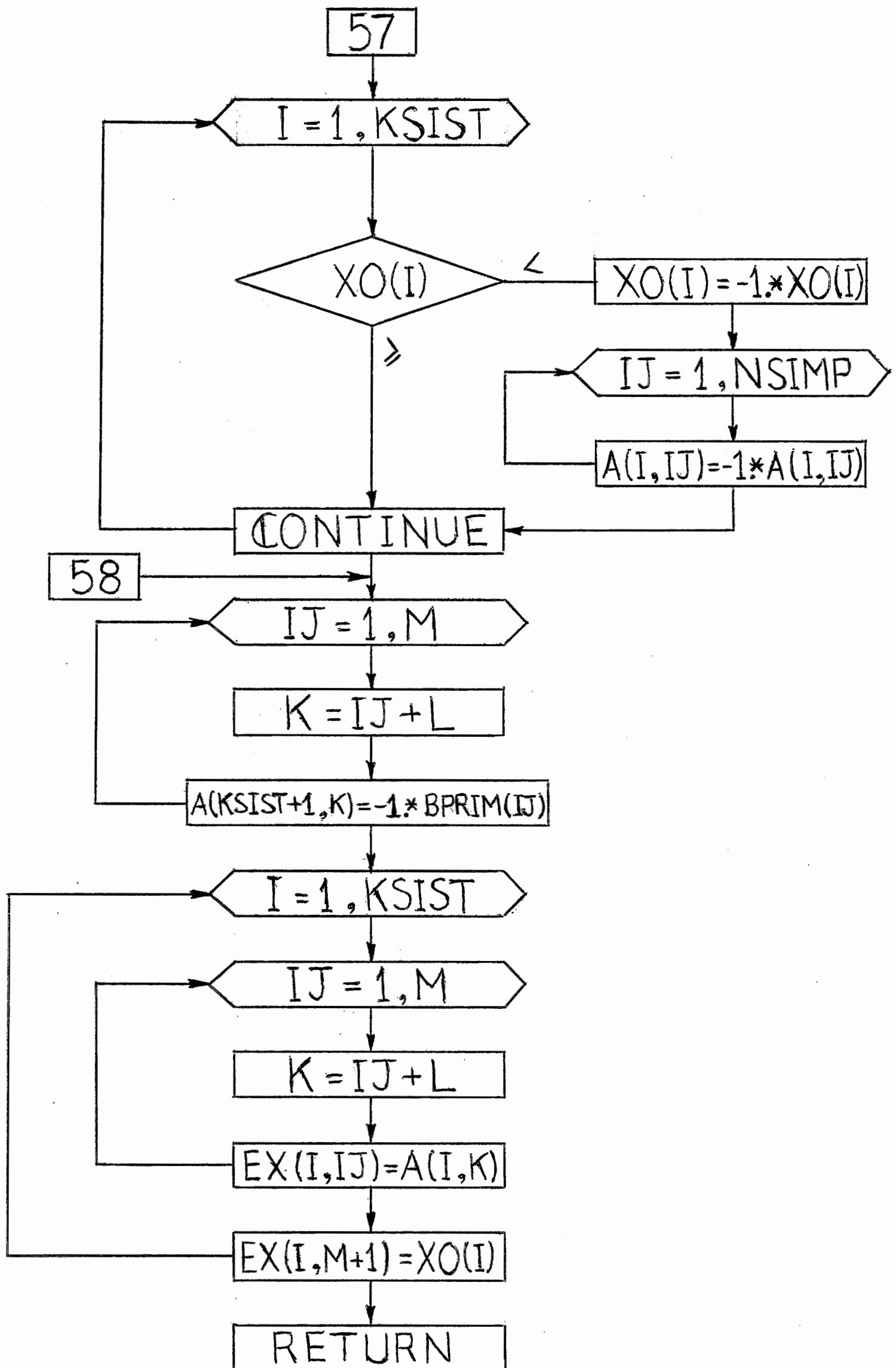


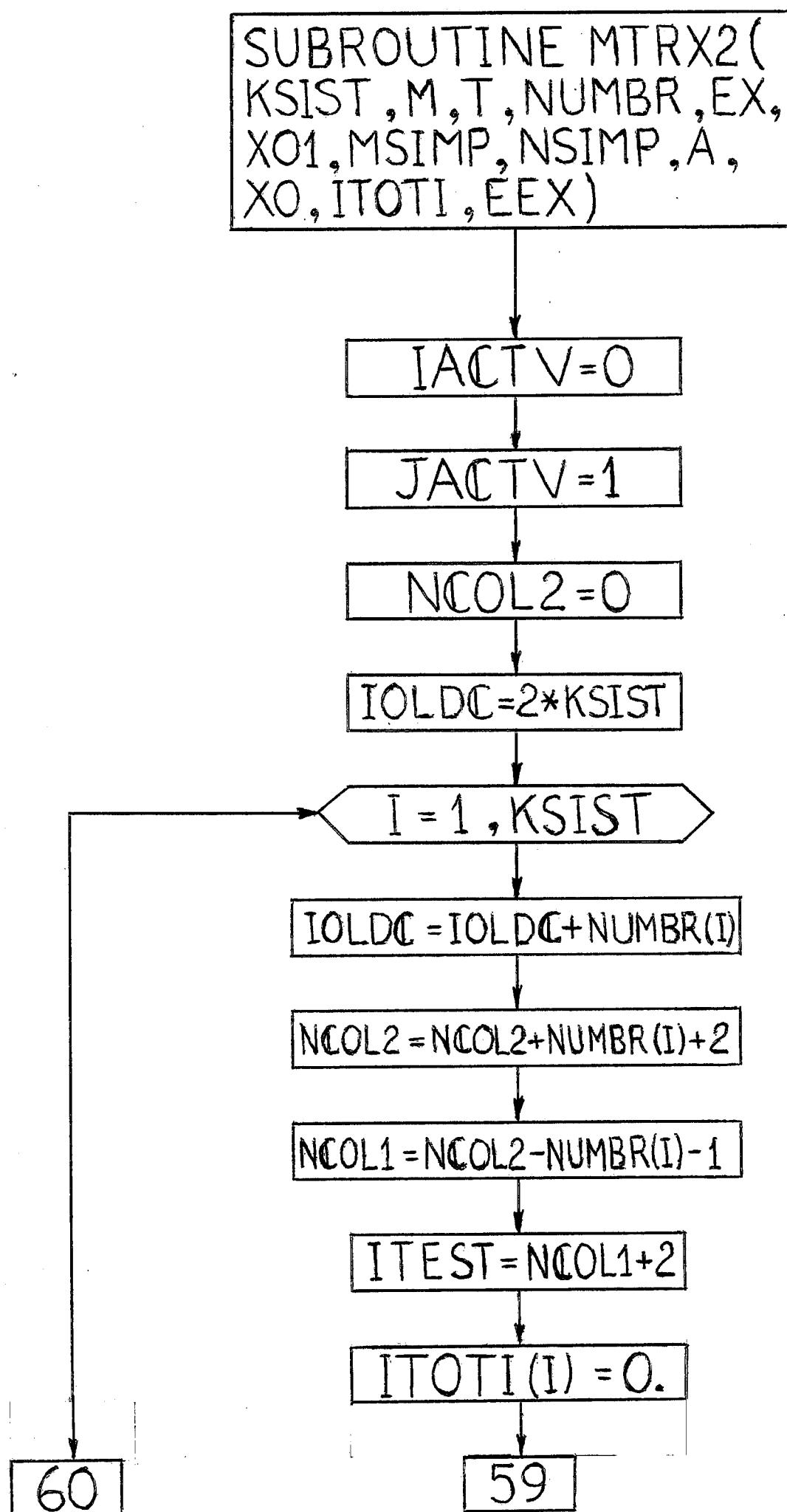


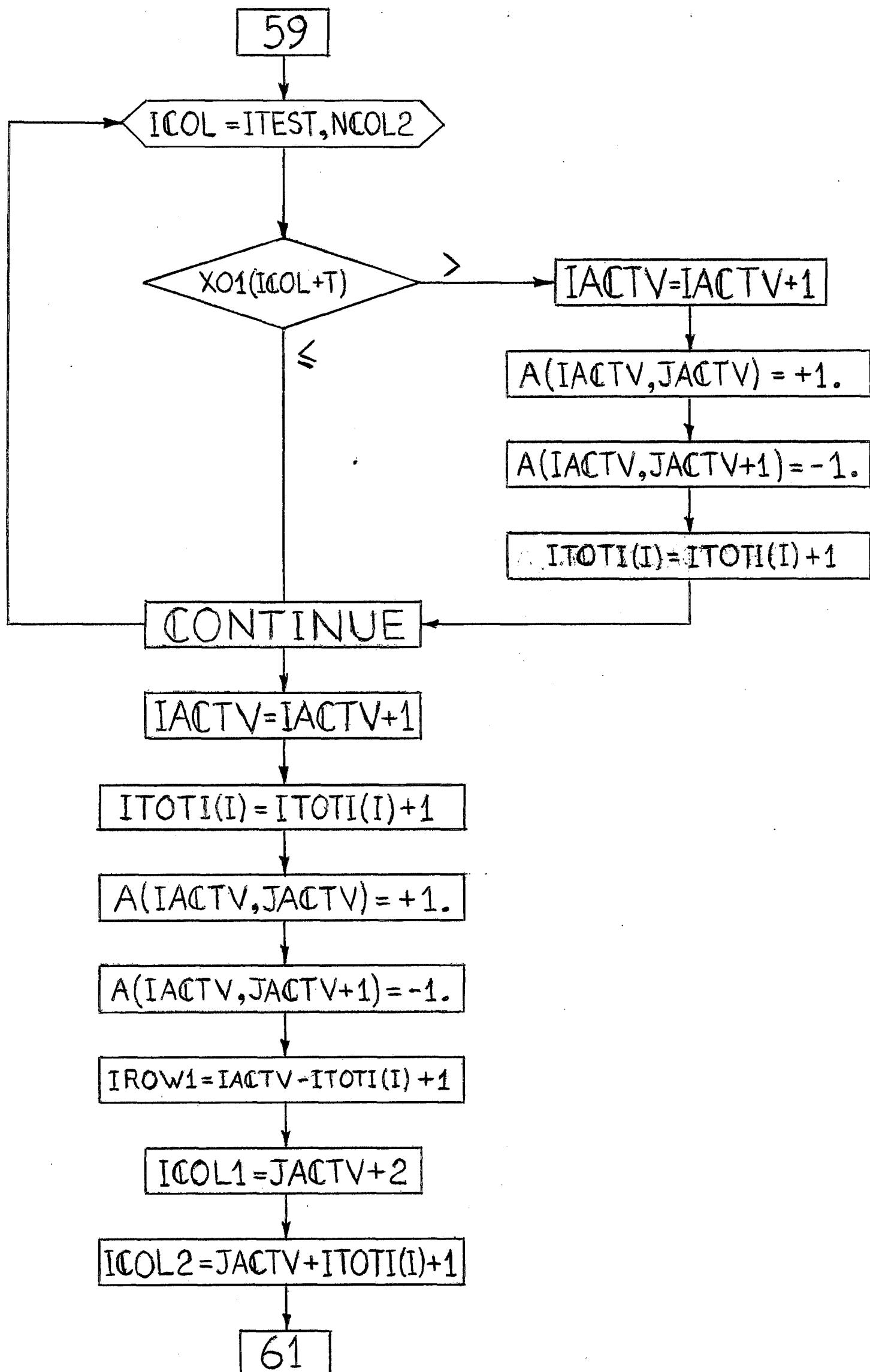


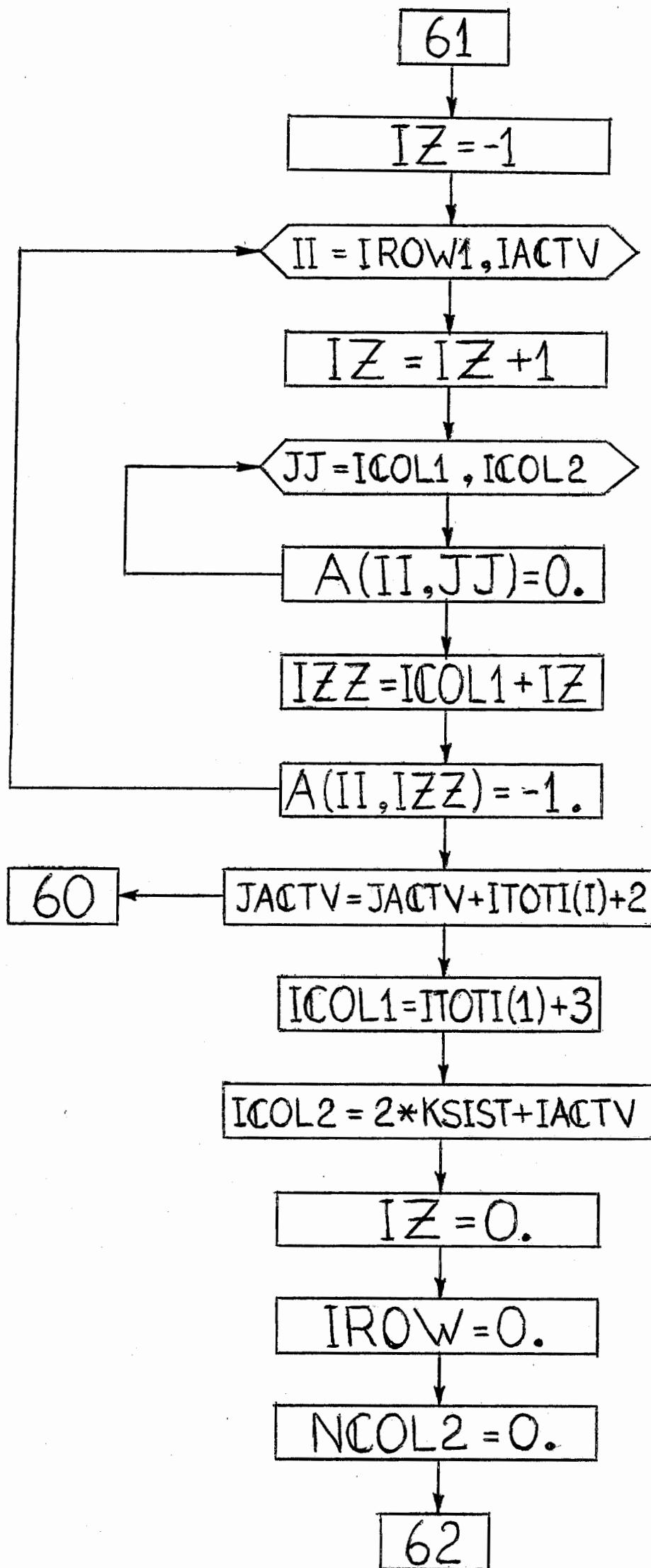


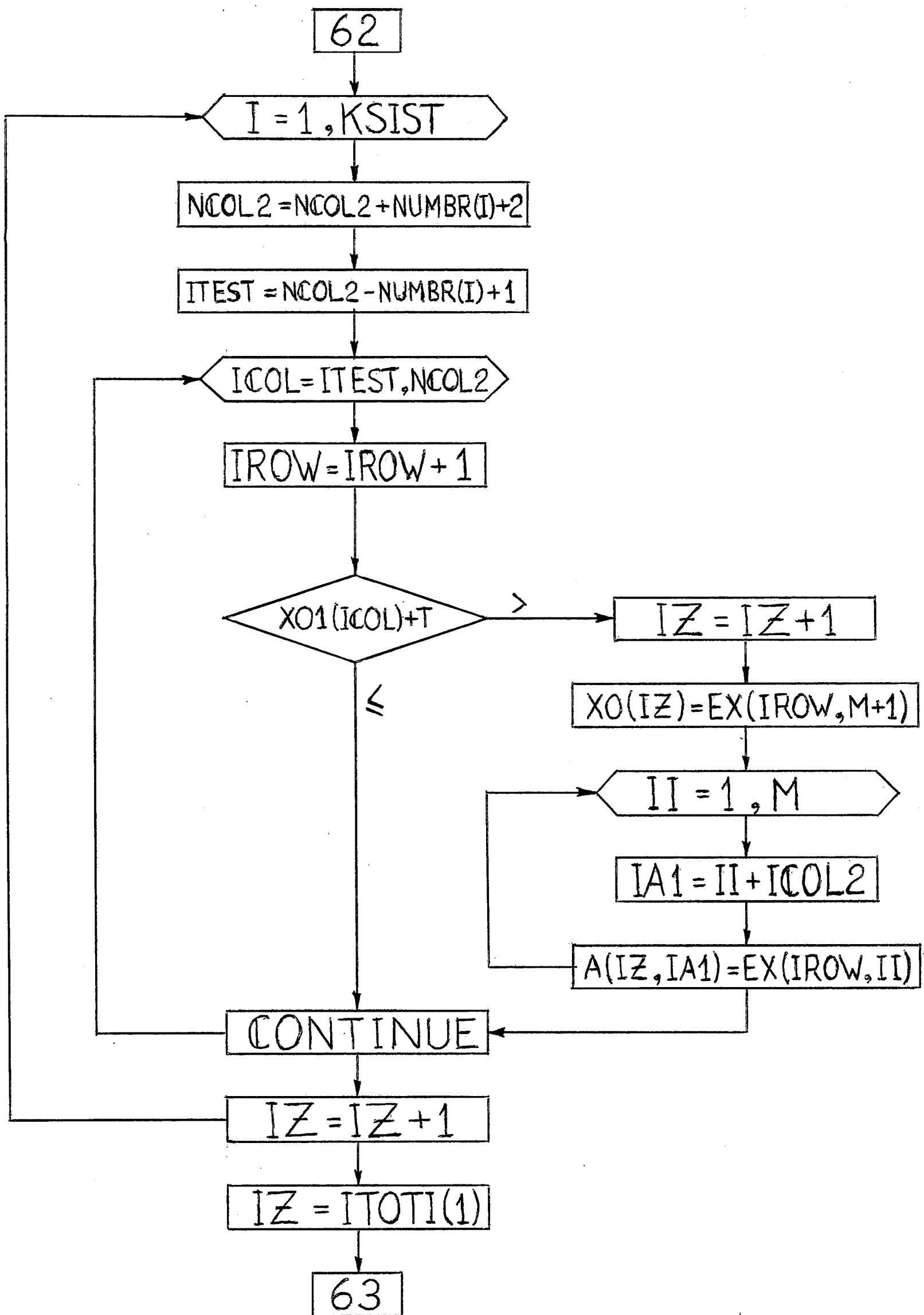


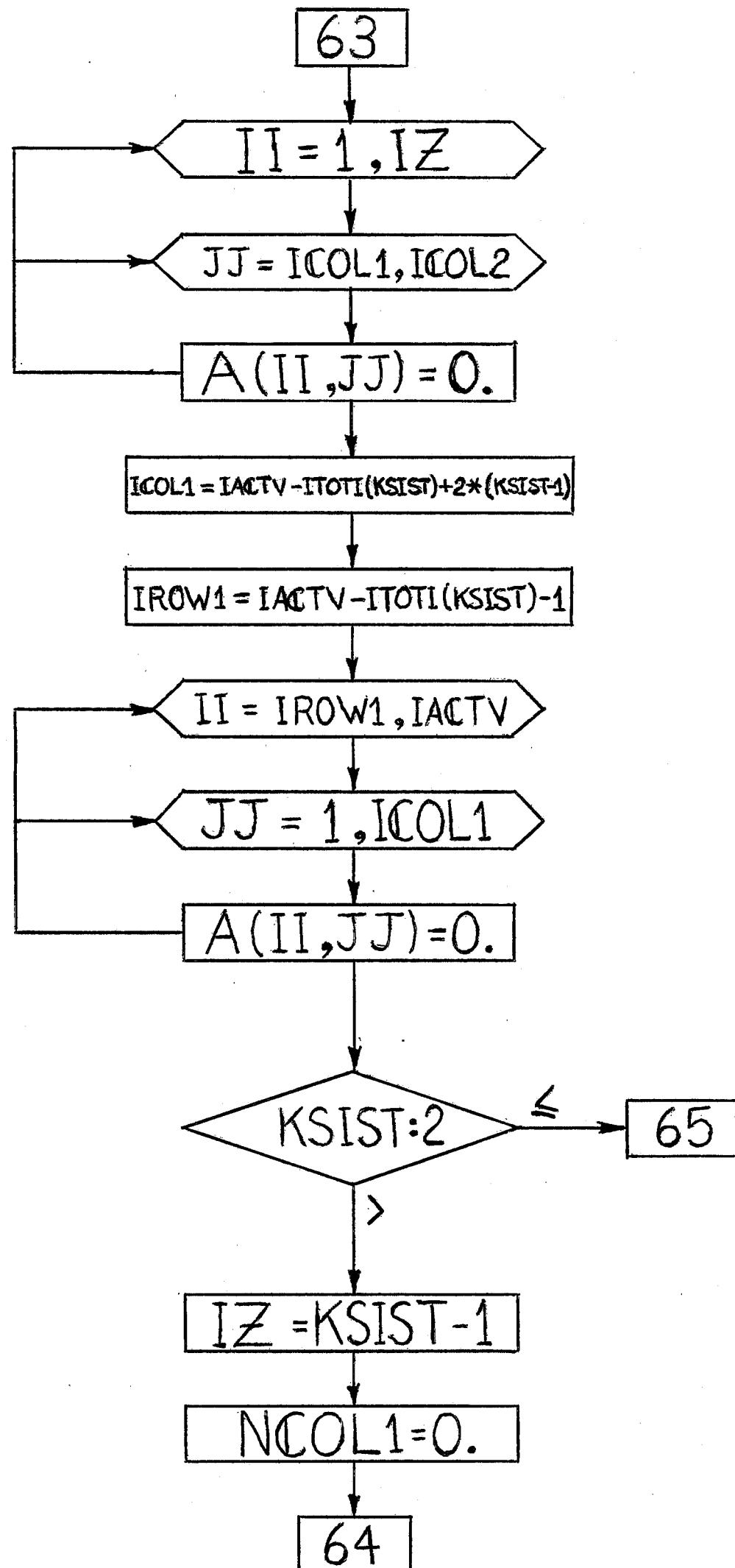


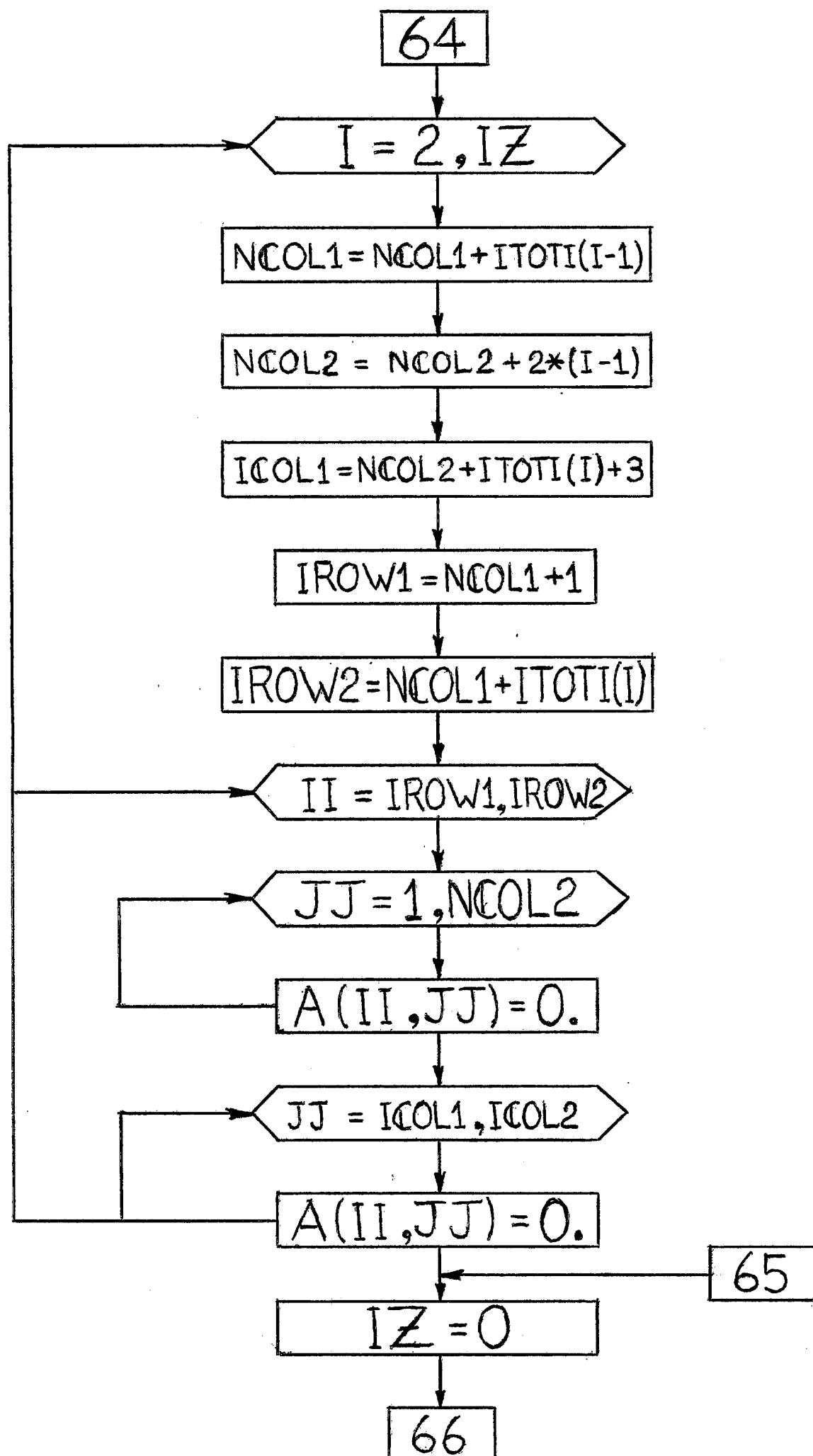


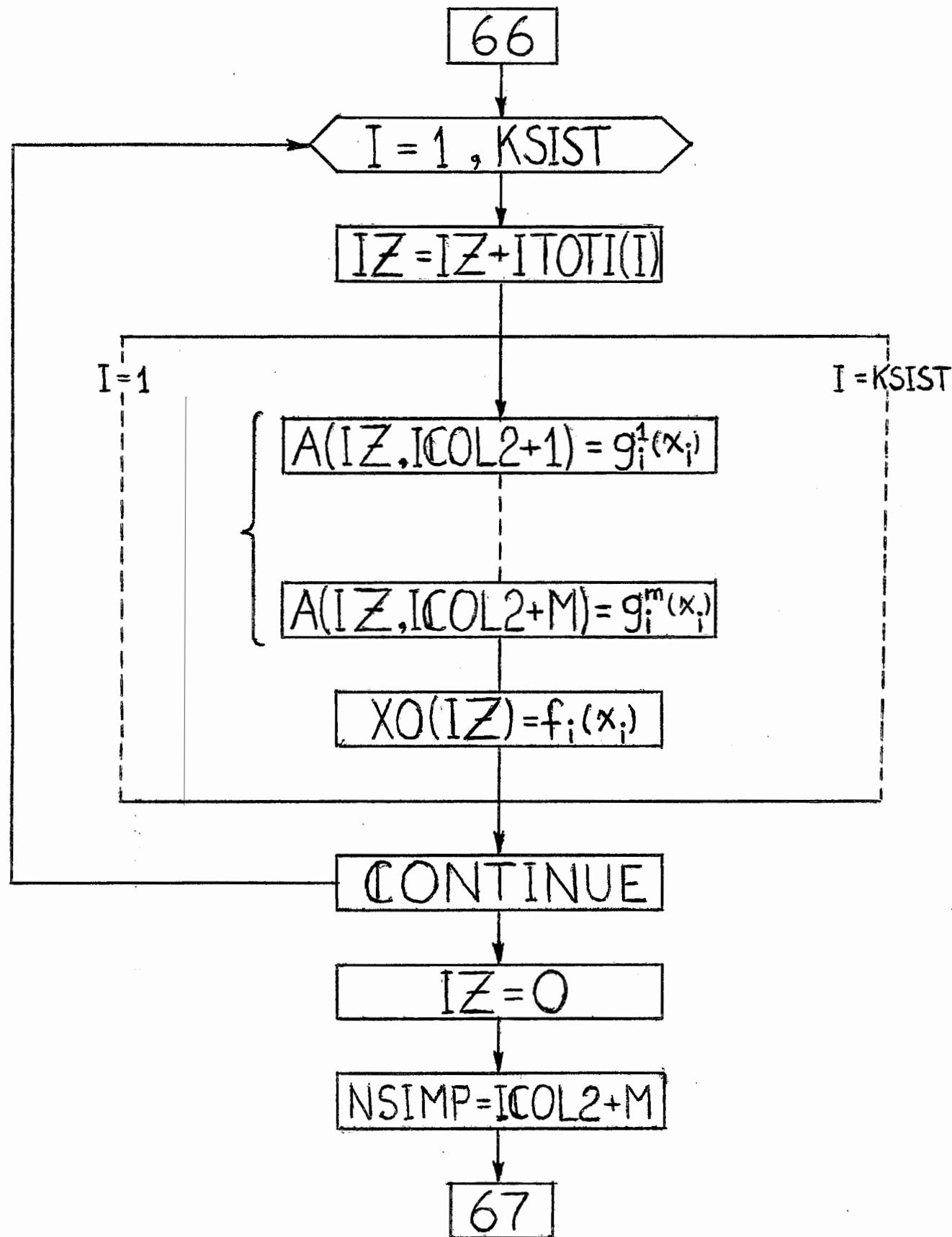


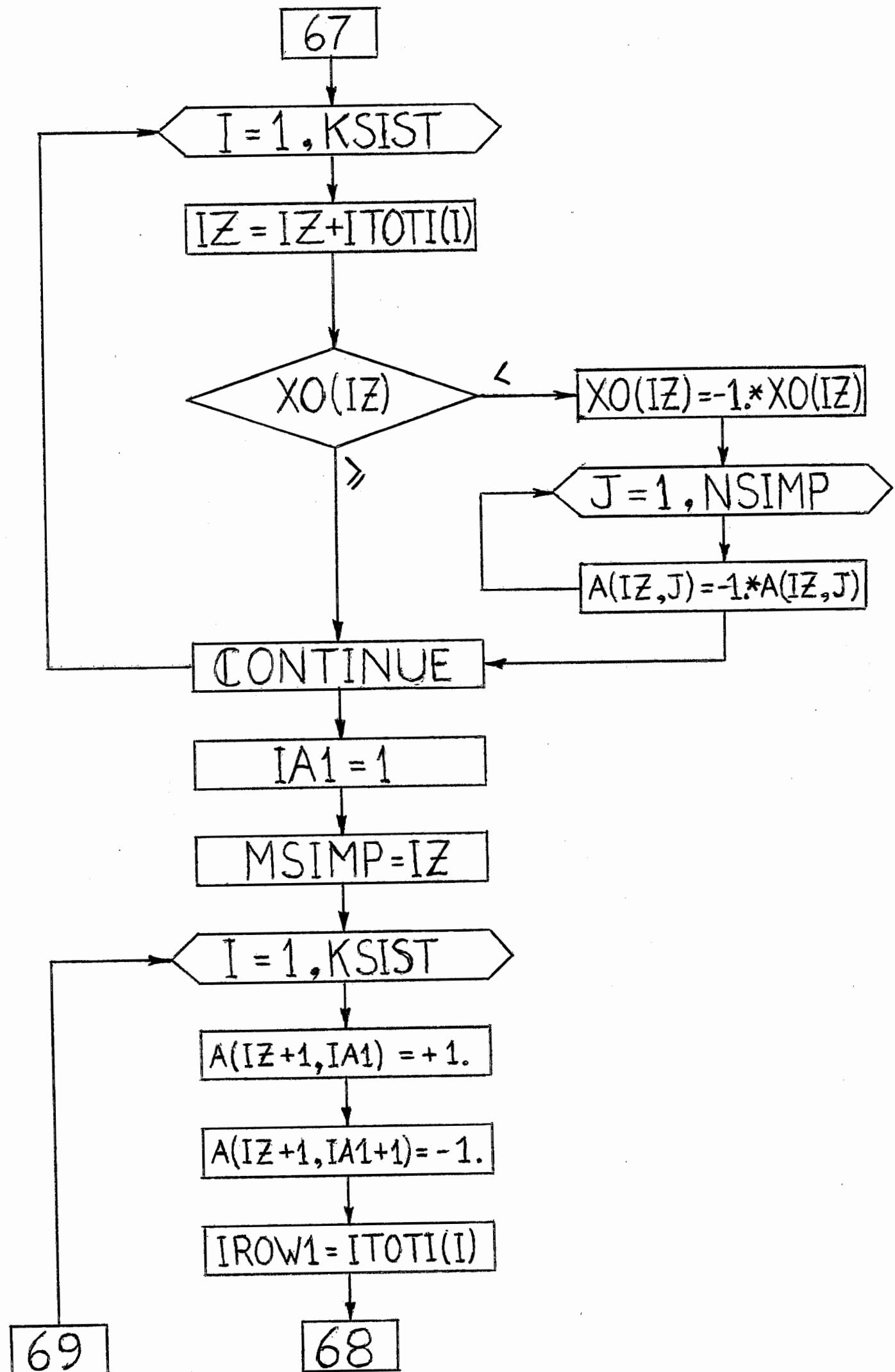


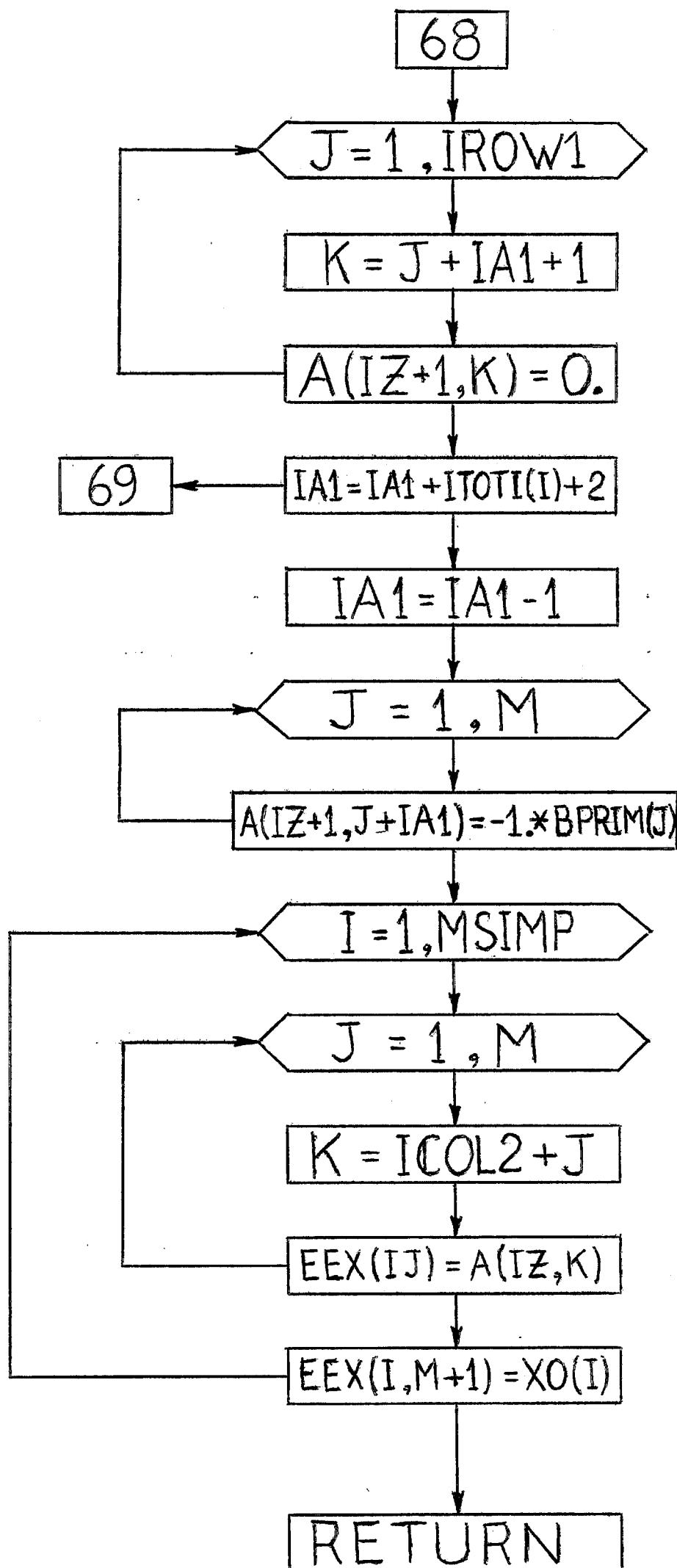


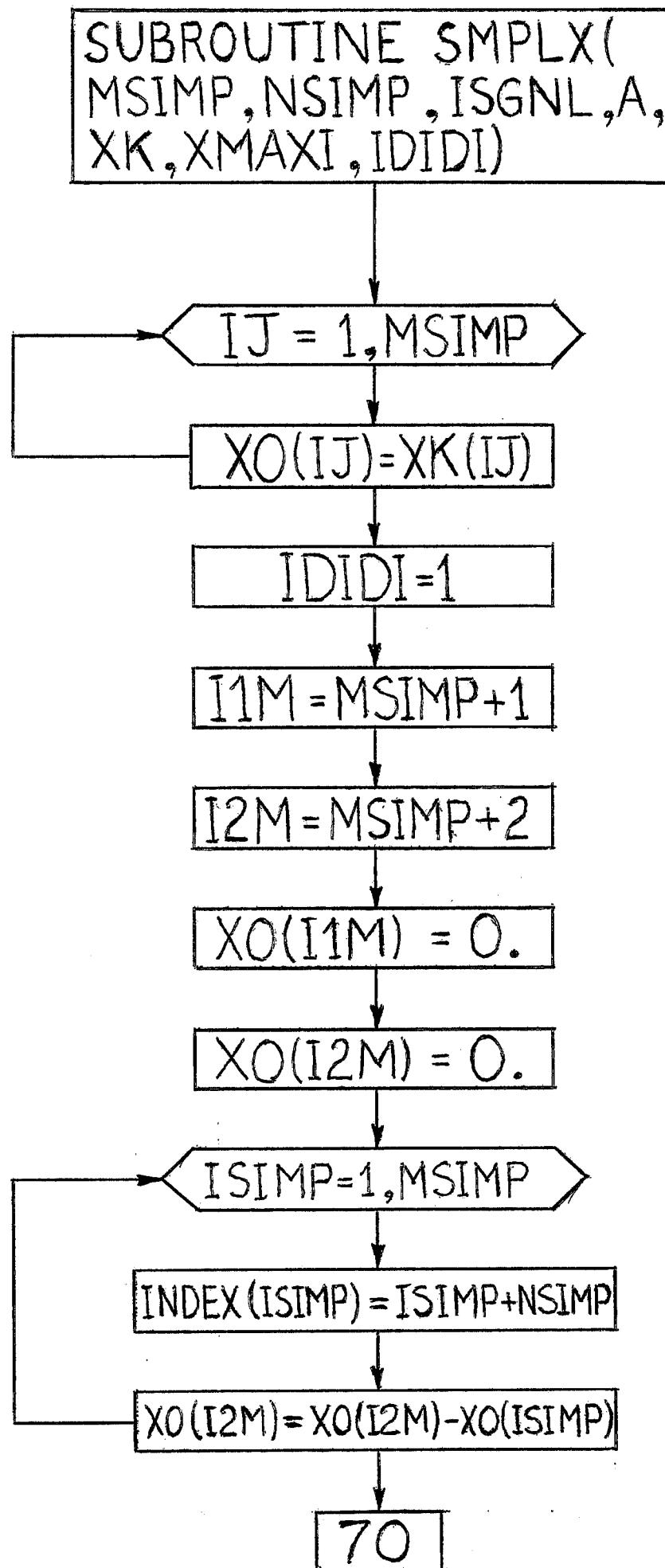


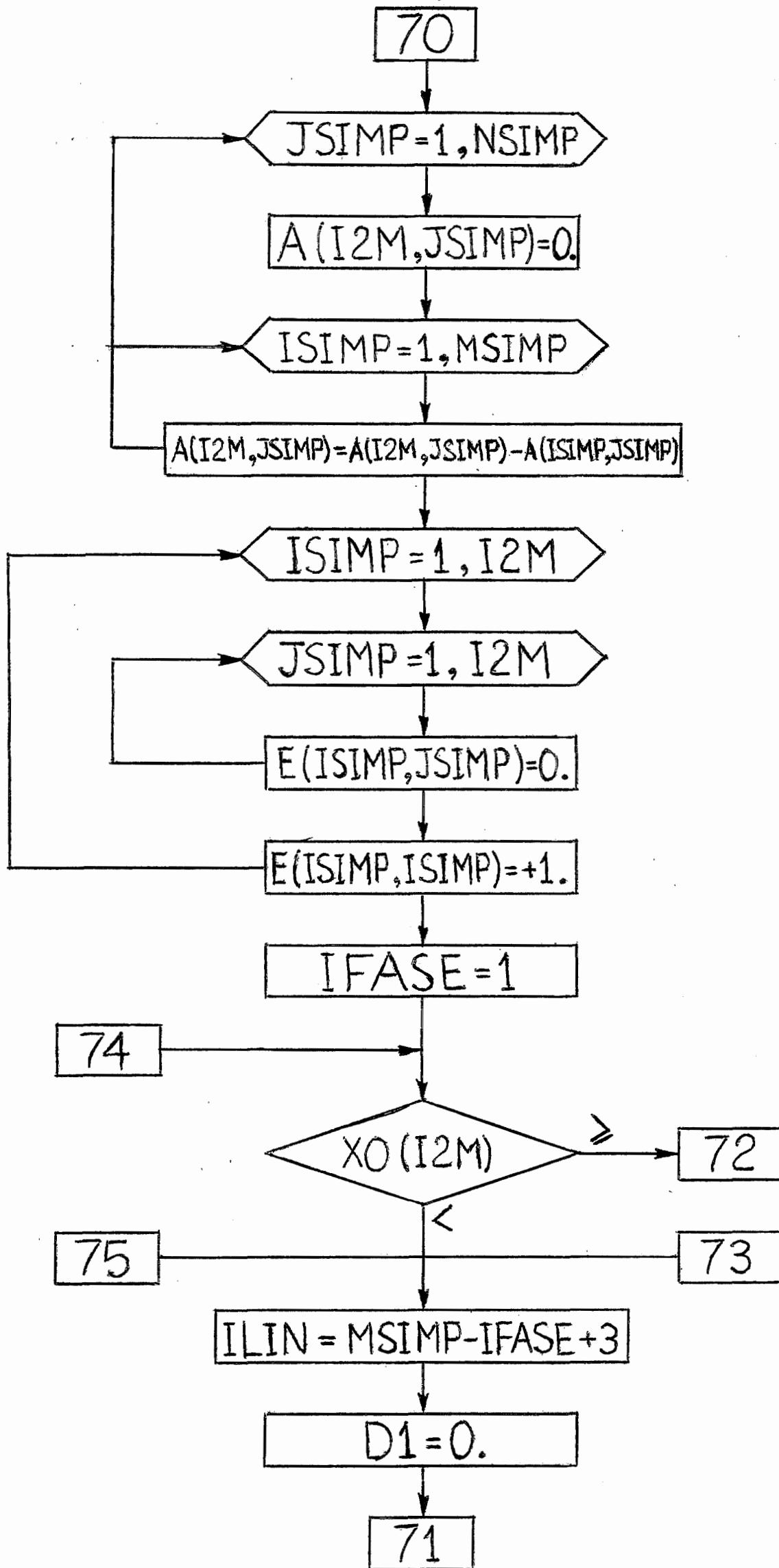


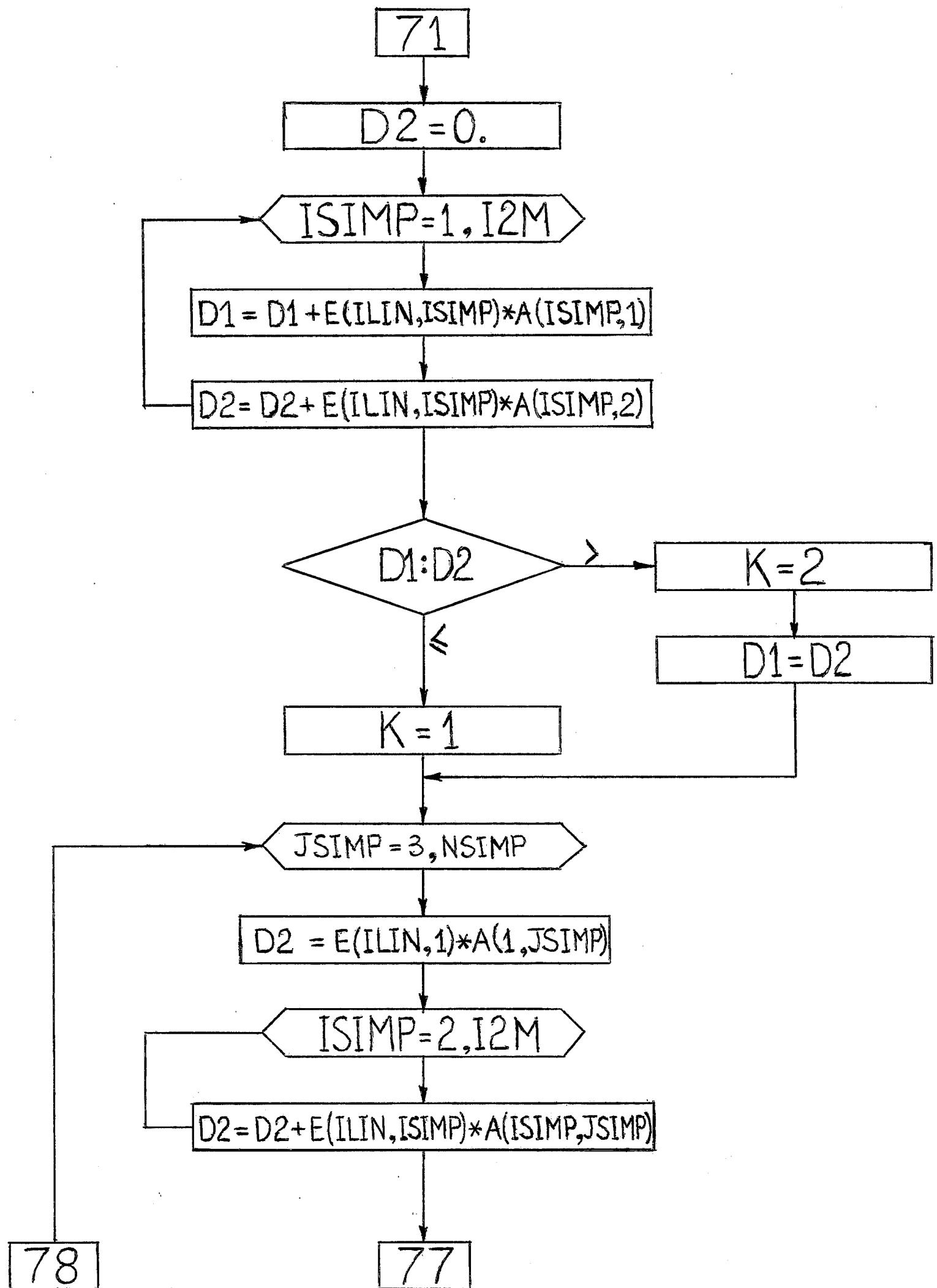


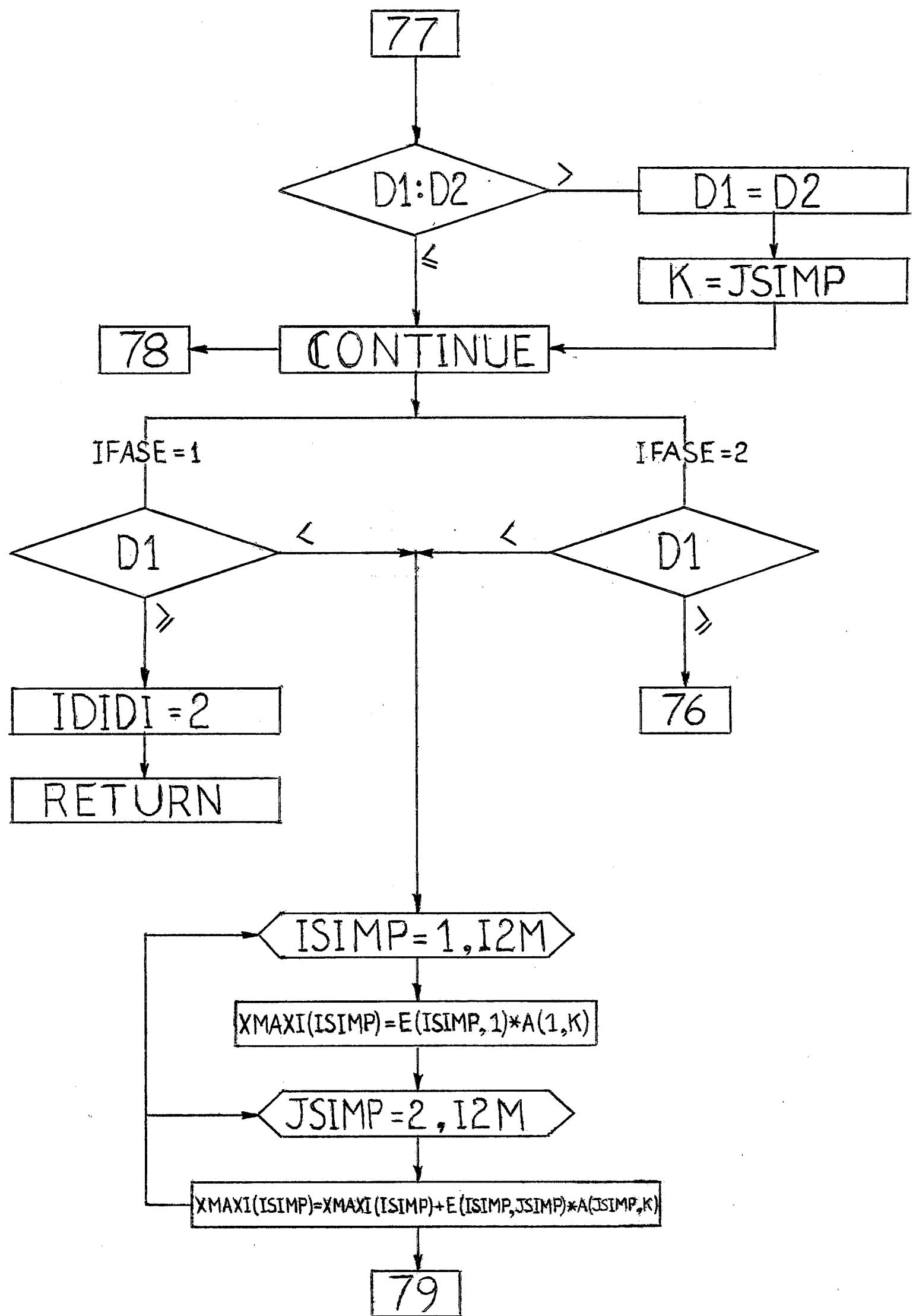


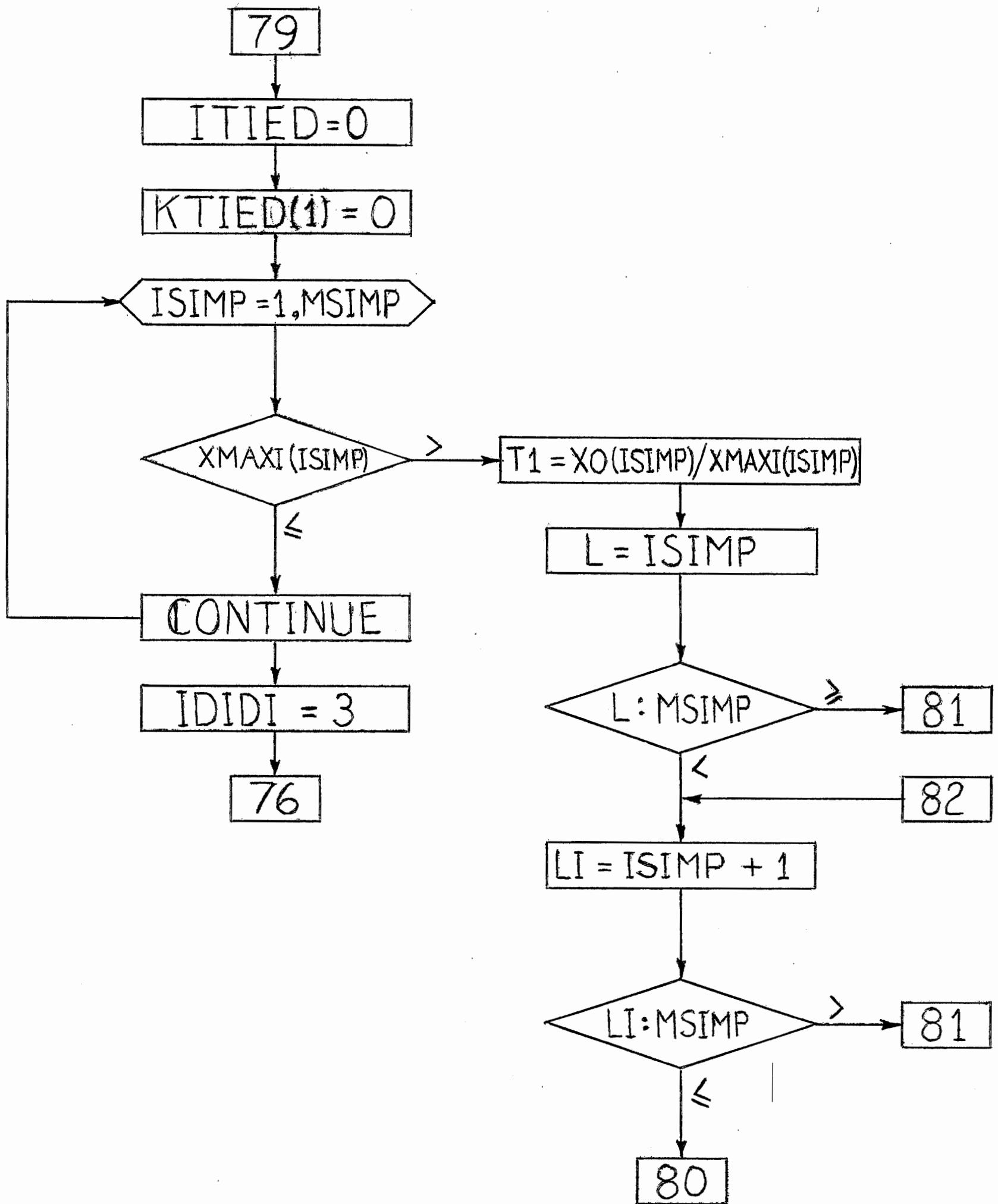


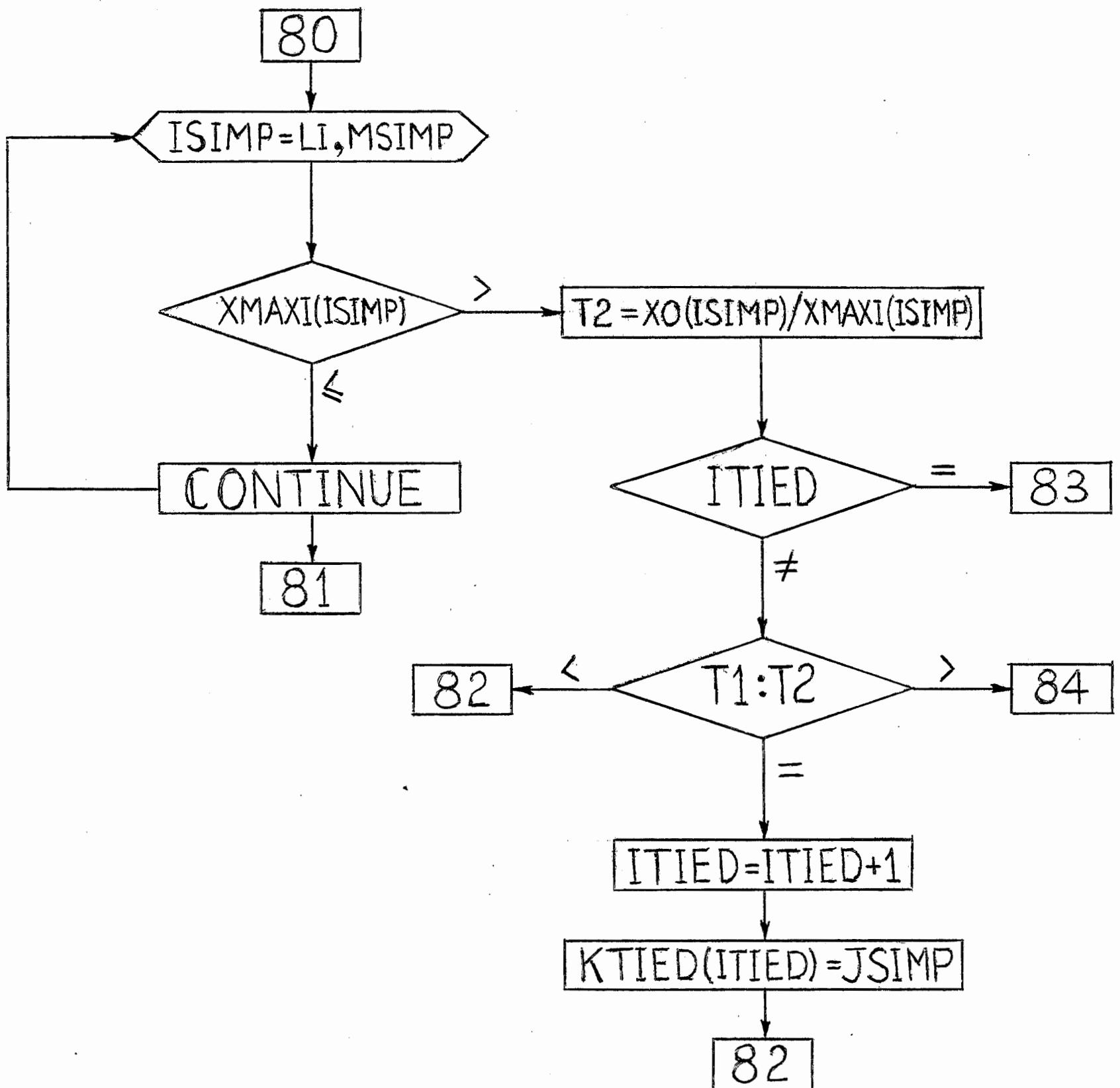


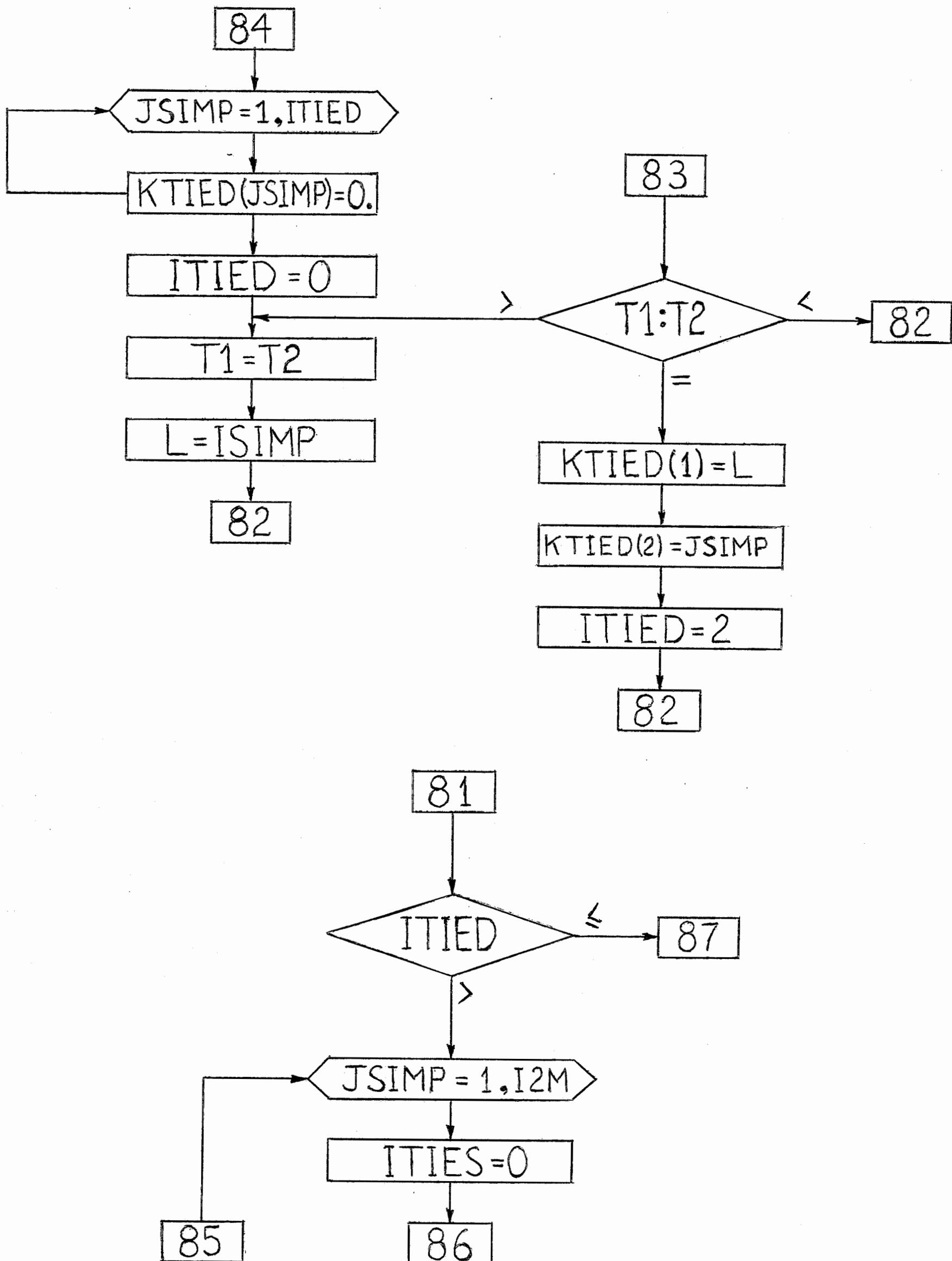


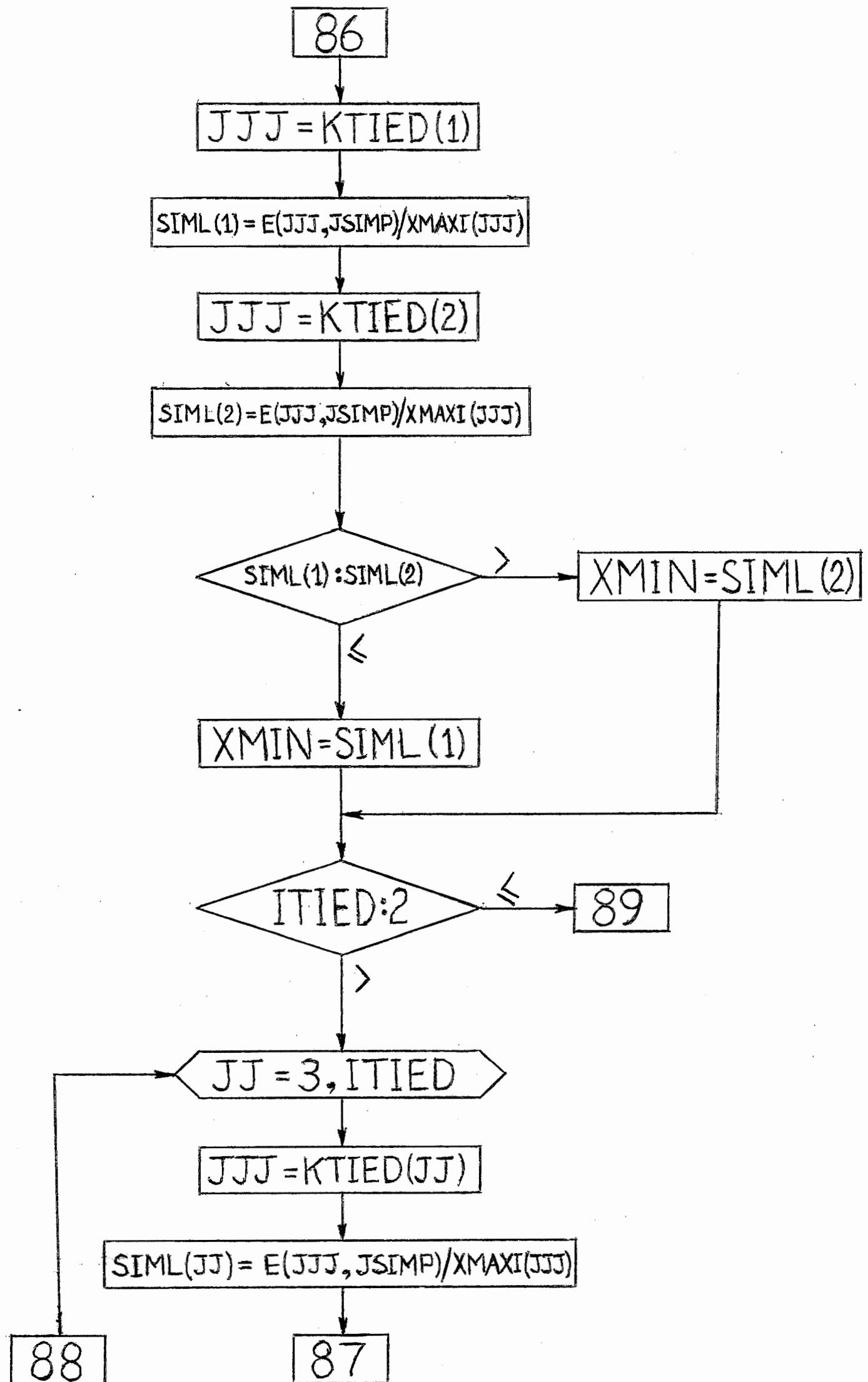


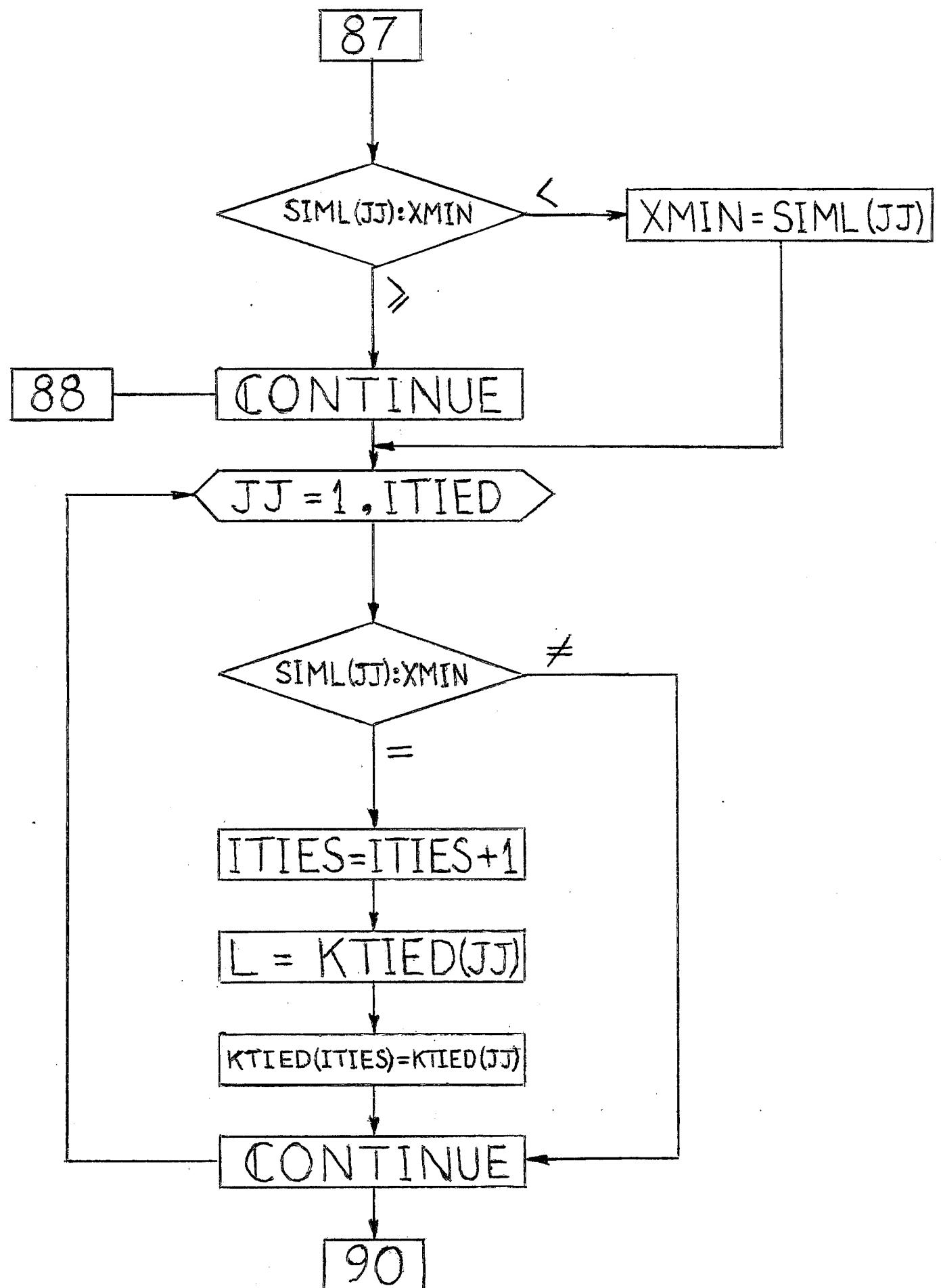


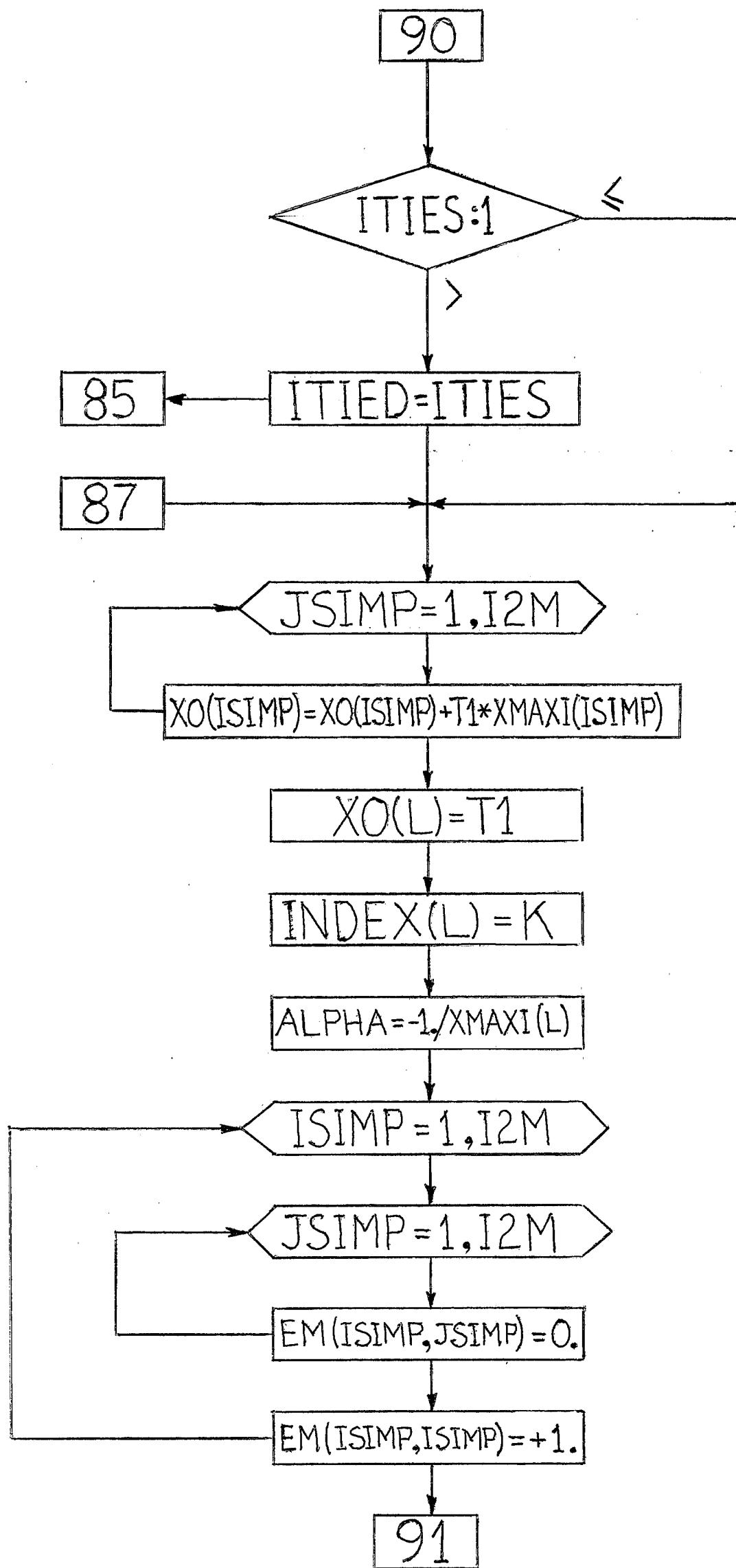


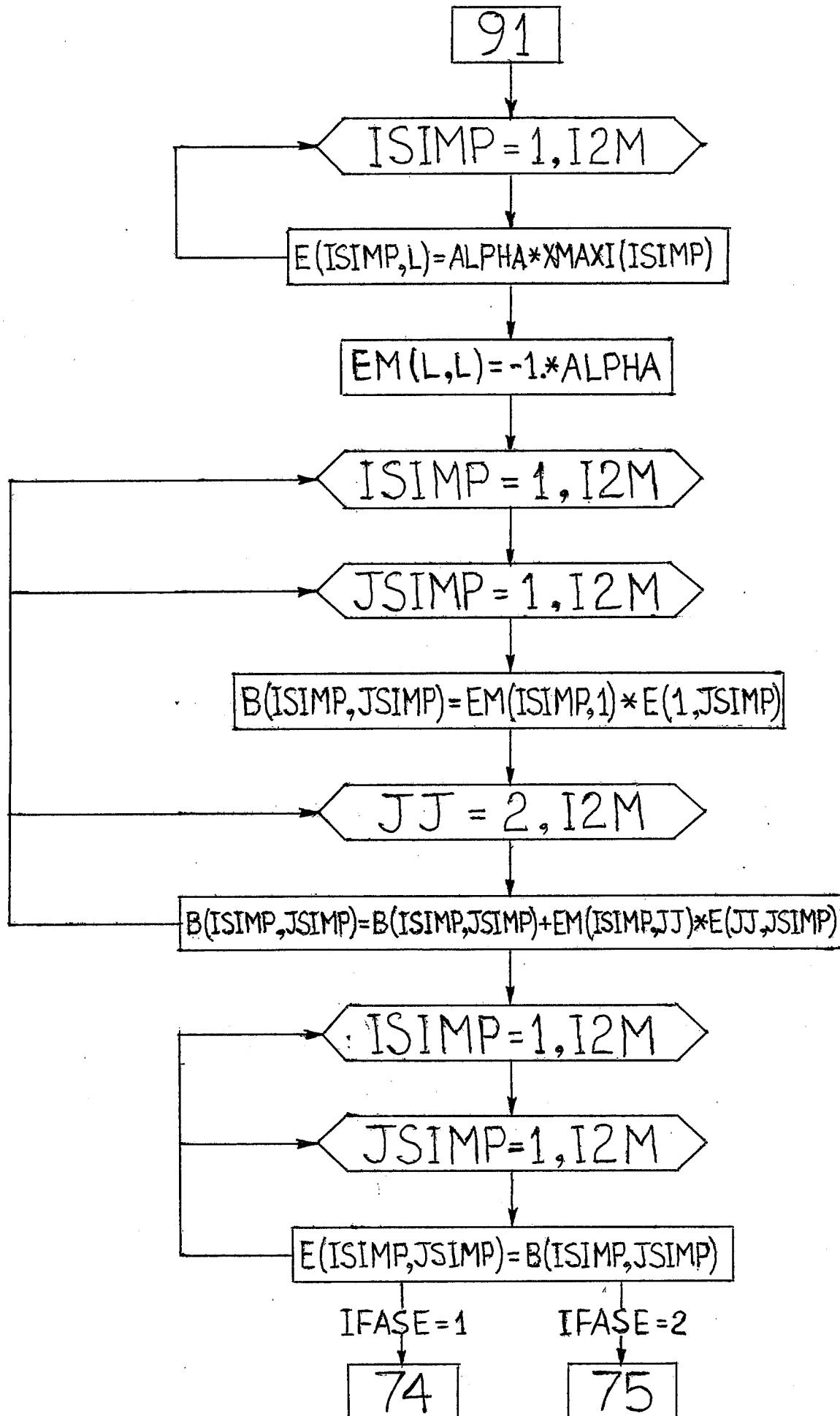


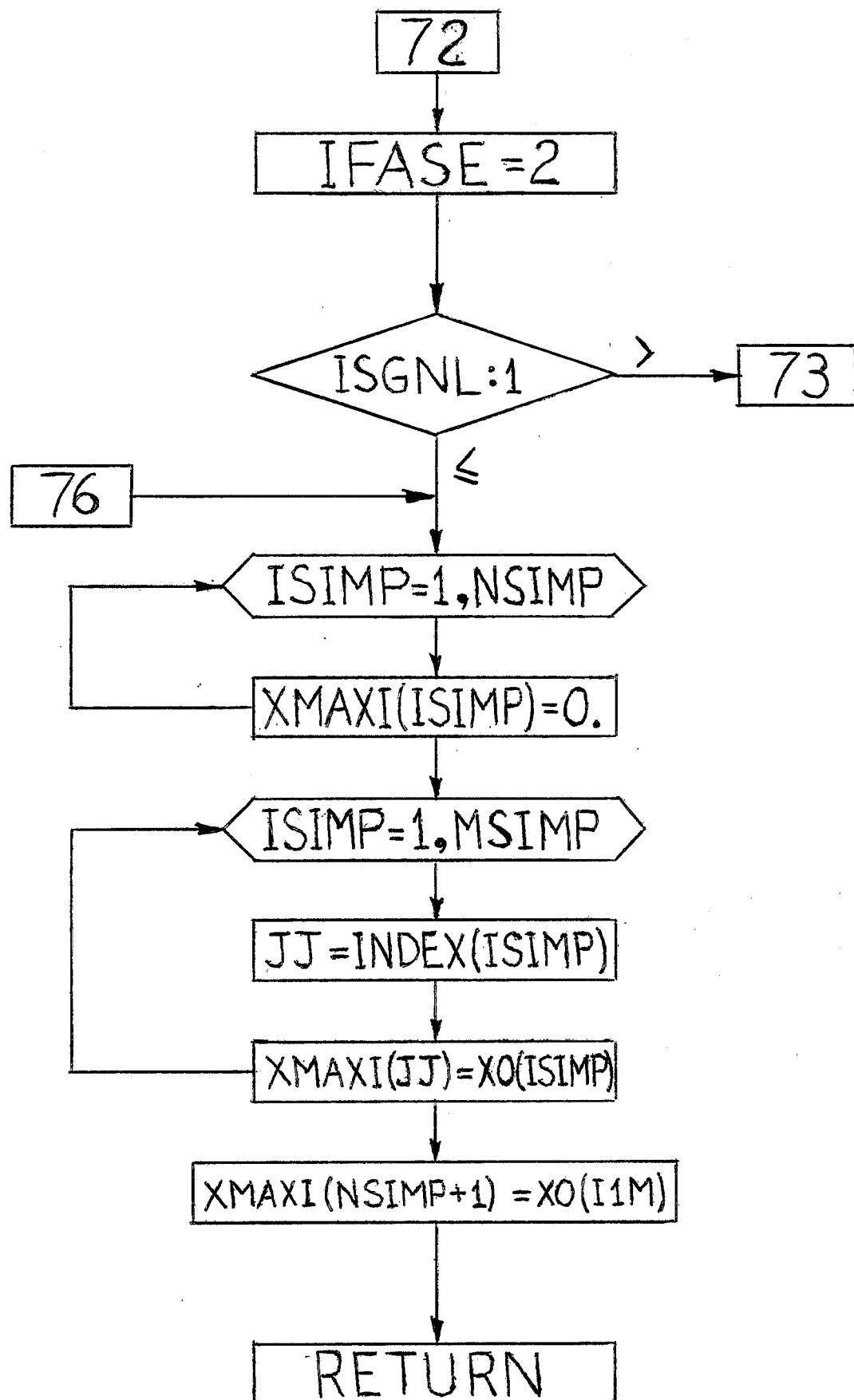












BIBLIOGRAFIA

- 1 Baumol, W. J. and Fabian, T.: Decomposition, Pricing for Decentralization and External Economies. Management Science, vol. 11, no. 1, September 1964.
- 2 Berge, C.: Spaces Topologiques. Editorial Dunod, Paris, 1966.
- 3 Bracken, J. and McCormick, G. P.: Selected Applications of Non-linear Programming. John Wiley & Sons, Inc., New York, 1968.
- 4 Campos, Odival Juliano de: Implementação de um algoritmo primal para otimização de sistemas decomponíveis. Tese apresentada à Coordenação dos Programas de Pós-Graduação de Engenharia da Universidade Federal do Rio de Janeiro, 1972.
- 5 Cheney, E. W. and Goldstein, A. A.: Newton's method of convex programming and Tchebycheff approximation. Numerische Mathematik, vol. 1, 1959.
- 6 Dantzig, G. B.: Linear Programming and Extensions. Princeton University Press, Princeton, New Jersey, 1963.
- 7 Dantzig, G. B.; Fulkerson, D. R. and Johnson, S.: Solution of a Large-Scale Traveling Salesman Problem.

- Operations Research, vol. 2, no. 4, November 1954.
- 8 Dantzig, G. B. and Madanski, A.: On the Solution of Two-Stage Linear Programs Under Uncertainty. Proceedings of the Fourth Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, 1961.
- 9 Dennis, J. B.: Mathematical Programming and Electrical Networks. John Wiley & Sons, Inc., New York, 1959.
- 10 Eaves, B. Curtis and Zangwill, W. I.: Generalized Cutting Plane Algorithms. SIAM Journal of Control, vol. 9, no. 4, November 1971.
- 11 Gale, D.: The theory of Linear Economic Models. McGraw-Hill, New York, 1960.
- 12 Gale, D.: A Geometric Duality Theorem with Economic Applications. Review of Economic Studies, vol. 34, no. 97, January 1967.
- 13 Gale, D.; Kuhn, H. W. and Tucker, A. W.: Linear Programming and the Theory of Games. in Koopman, T. J. (editor), Activity Analyses of Production and Allocation, John Wiley & Sons, Inc., New York, 1951.
- 14 Gass, Saul I.: Linear Programming - Methods and Applications. McGraw-Hill, New York, 1969.
- 15 Geoffrion, A. M.: Primal Resource Directive Approaches for Optimizing Non-Linear Decomposable Systems. RAND Report no. RM-5829-PR, 1968.
- 16 Geoffrion, A. M.: Elements of Large-Scale Mathematical Programming.

- Working Paper no. 144, Western Management Science Institute, UCLA, 1969.
- 17 Geoffrion, A. M.: Duality in Non-Linear Programming - a simplified applications-oriented development.
Working Paper no. 150, Western Management Science Institute, UCLA, 1969.
- 18 Geoffrion, A.M.; Dyer, J. S. and Feinberg, A.: An Interactive Approach for Multi-Criterion Optimization, with an application to the operation of an academic department.
Working Paper no. 176, Western Management Science Institute, UCLA, 1971.
- 19 Geoffrion, A. M. and Hogan, W. W.: Coordination of Two Level Organizations with Multiple Objectives.
Western Management Institute, UCLA, 1971.
- 20 Hausdorff, F.: Set Theory.
Chelsea, New York, 1962.
- 21 Hirshleifer, J.: On the Economics of Transfer Pricing.
Journal of Business, vol. 29, no. 3, July 1956.
- 22 Hogan, W. W.: Optimization and Convergence for Extre - mal Value Functions arising from Structured Non-Linear Programs.
Working Paper no. 180, Western Management Science Institute, UCLA, 1971.
- 23 Horváth, ..: Introduction a la Topología General.
Monografía Científica no.9, Série de Matemática, OEA.
- 24 Jennergren, L. P.: Studies in the Mathematical Theory of Decentralized Resource-Allocation.
Ph. D. dissertation, Stanford University, 1971.
- 25 Johannsen, H.; Robertson, A. and Brech, E. F. L.: Management Glossary.

- Longmans, Oxford, 1968.
- 26 Keley, J. E.: The Cutting-Plane Method for Solving Convex Programs.
Journal of the Society for Industrial and Applied Mathematics, vol. 8, no. 4, December 1960.
- 27 Koopmans, T. C. and Beckman, M.: Assignment Problems and the Location of Economic Activities.
Econometrica, vol. 25, no. 1, January 1957.
- 28 Kornai, J. and Liptak, Th.: Two-Level Planning.
Econometrica, vol. 33, no. 1, January 1965.
- 29 Lancaster, K.: Mathematical Economics.
The MacMillan Co., New York, 1968.
- 30 Lasdon, L. S.: Optimization Theory for Large-Scale Systems.
The MacMillan Co., New York, 1970.
- 31 Malinvaud, E.: Decentralized Procedures for Planning.
in Bacharach, M. O. L. and Malinvaud, E. (editors),
Activity Analyses in the Theory of Growth and Planning,
The MacMillan Co., London, 1967.
- 32 Mangasarian, O. L.: Non-Linear Programming.
McGraw-Hill, New York, 1969.
- 33 March, J. and Simon, H.: Organizations.
John Wiley & Sons, Inc., New York, 1958.
- 34 Marschak, J. and Radner, R.: The economic theory of teams.
Cowles Foundation Monograph, Yale University Press,
1972.
- 35 Mesarovic, M. V.; Macko, D. and Takahara, Y.: Theory of Hierarchical, Multilevel, Systems.
Academic Press, New York, 1970.

- 36 Nikaido, H.: Convex Structures, Economic Theory.
Academic Press, New York, 1968.
- 37 Polak, E.: Computational Methods in Optimization.
vol. 77 in Mathematics in Science and Engineering,
Bellman, R.,(editor), Academic Press, New York, 1971.
- 38 Stahl, I. and Ysander, B. -C.: General Principles for
Planning and Combat-Economic Evaluation.
Research Institute for National Defense, FOA P report
C-8171-11, Stockholm, 1967.
- 39 Stahl, I. and Ysander, B. -C.: System Analysis and
Long-Range Planning.
Research Institute for National Defense, FOA P report
C-8122-11, Stockholm, 1968.
- 40 Tcheng, T. -H.: Scheduling of a Large Forestry-Cutting
Problem by Linear Programming Decomposition.
Ph. D. dissertation, University of Iowa, 1966.
- 41 Topkis, Donald M.: Cutting Plane Methods without Nes-
ted Constraint Sets.
Operations Research, vol. 18, 1970.
- 42 Topkis, Donald M.: A Note on Cutting Planes without
Nested Constraint Sets.
Operations Research, vol. , 1970.
- 43 Varaiya, P. P.: Trends in the Theory of Decision-Mak-
ing in Large Systems.
University of California, Berkeley, 1972.
- 44 Veinott, A. F.: The Supporting Hyperplane Method for
Unimodal Programming.
Operations Research, vol. 15, 1967.
- 45 von Neumann, J.: Zur Theorie der Gesellschaftsspiele.
Matematische Annalen, vol. 100, 1928.

- 46 Weitzman, M.: Iterative Multilevel Planning with Production Targets.
Econometrica, vol. 38, no. 1, January 1970.
- 47 Zangwill, W. I.: Non-Linear Programming.
Prentice-Hall, Englewood Cliffs, New Jersey, 1969.
- 48 Zangwill, W. I.: Convergence Conditions for Non-Linear Programming Algorithms.
Management Science, vol. 16, no. 1, September 1969.
- 49 Zschau, E. V. W.: A Primal Decomposition Algorithm for Linear Programming.
Working Paper no. 91, Graduate School of Business,
Stanford University, 1967.
- 50 Zoutendijk, G.: Methods of Feasible Directions.
Elsevier, Amsterdam, 1960.