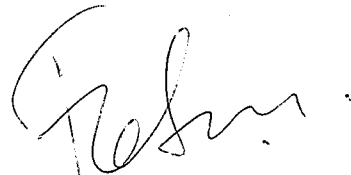


UM ANALISADOR SINTÁTICO COM REPARADOR DE
ERROS PARA UM SUBCONJUNTO DO ALGOL B/6700

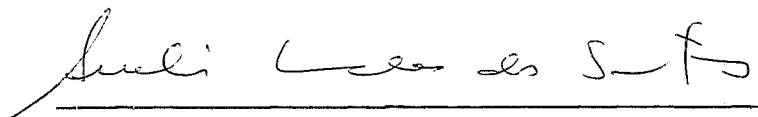
Rita Silva de Almeida

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 CIÊNCIAS (M.Sc.).

Aprovada por:



Estevam Gilberto De Simone - COPPE/UFRJ
Presidente



Sueli Mendes dos Santos - COPPE/UFRJ



Emmanuel P. Lopes Passos - IME

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 1982

ALMEIDA, RITA SILVA DE

Um Analisador Sintático com Reparador de Erros para um Subconjunto do Algol B/6700 [Rio de Janeiro] 1982.

VIII, 121 p., 29,7 cm (COPPE/UFRJ, M.Sc. Engenharia de Sistemas e Computação, 1982).

Tese - Univ. Fed. Rio de Janeiro. Centro de Tecnologia.

I. Compiladores I. COPPE/UFRJ II. Título (série).

À

Lígia Alves Barros

AGRADECIMENTOS

À Universidade Federal de Alagoas e à Coordenação dos Programas de Pós-Graduação em Engenharia da Universidade Federal do Rio de Janeiro, pela oportunidade da realização deste trabalho.

A todas as pessoas que de uma maneira ou de outra contribuíram para o desenvolvimento do trabalho. Em especial, Lígia Alves Barros, Miguel de Teive e Argollo Jr., Sérgio de Mello Schneider, Sueli Mendes dos Santos e a Renê Javier Uzeda Leon.

RESUMO

Neste trabalho se apresenta um analisador de sentenças pertencentes a um subconjunto da linguagem Algol B/6700, com tratamento de erros e emissão de mensagens em português.

A análise é feita através do método LL(1) e qualquer erro de sintaxe é reparado, incluindo, retirando, ou trocando-se símbolos da sentença, no local do erro. A escolha de uma dessas formas de reparo, baseia-se no custo mínimo local das mesmas.

Tratamentos também são efetuados na fase léxica de compilação e por algumas ações semânticas que complementam a análise das sentenças.

ABSTRACT

A parser of a B6700 Algol Language Subset is presented. It includes error handling and error messages in portuguese.

The parser uses LL(1) Parsing Method and any synthatical error is repaired by including, deleting or exchanging symbols of the wrong sentence. The choice of repairing action is based on locally least-cost criteria.

Error handling is also made in the lexical level of compilation as well as through some additional semantic actions.

ÍNDICE

	Pág.
INTRODUÇÃO	1
CAPÍTULO 1 - CONCEITUAÇÃO BÁSICA	4
I. Conceito de erro léxico, sintático e semântico	5
II. Recuperador de erros	8
III. Reparador de erros	9
IV. Corretor de erros	10
CAPÍTULO 2 - TÓPICOS PRELIMINARES	12
I. Análise estatística sobre frequência de erros	12
II. Erros de sintaxe em programas escri- tos em Algol	15
III. Resumo da liteiratura pesquisada sobre tratamento de erros	18
CAPÍTULO 3 - O ANALISADOR SINTÁTICO	23
I. Gramáticas LL(k)	24
II. Gramáticas LL(1)	27
III. Restrições de uma gramática LL(k)	28
IV. O analisador sintático para gramã- ticas LL(1)	29
V. O momento da ocorrência de um erro e o momento do seu descobrimento em ana- lisadores sintáticos LL(1)	31

	Pág.
VI. Uma gramática não anulável	34
VII. Opções para implementação	38
CAPÍTULO 4 - O REPARADOR DE ERROS	40
I. Gramática aumentada	41
II. Custo de um Símbolo terminal	42
III. Cadeia de símbolos terminais com custo mínimo	43
IV. Cadeia de inserção com custo mínimo	46
V. O reparo utilizando somente cadeia de inserção com custo mínimo	50
VI. O reparo utilizando inserção e retirada de cadeias de símbolos terminais	54
VII. Testes de implementação	59
CAPÍTULO 5 - A IMPLEMENTAÇÃO FINAL	74
I. Especificações léxicas da linguagem	74
II. Especificações do analisador léxico	76
III. Tratamento de erros descobertos na fase léxica de compilação e mensa gens de advertência	77
IV. Especificações do procedimento semânti co	80
V. O algoritmo do procedimento semântico ...	81
VI. Tratamento e mensagens de erro do pro- cedimento semântico	87
VII. Opções, recursos e mensagens de erro do analisador sintático	88

	Pág.
CONCLUSÕES	90
APÊNDICE A	92
APÊNCIDE B	94
APÊNCIDE C	103
REFERÊNCIAS BIBLIOGRÁFICAS	106
ANEXO 1	108

INTRODUÇÃO

A teoria sobre análise sintática de sentenças de uma linguagem, já atingiu um bom desenvolvimento, ou seja, os mé todos existentes que verificam se uma sentença pertence ou não a uma linguagem determinada, são teoricamente satisfatō - rios. Porém, o mesmo não acontece quando se trata da análise sintática de sentenças com a preocupação de reparos de erros, devido aos vários problemas que aparecem quando tentamos cor rigir uma sentença e nem todos foram teorica ou praticamente resolvidos.

Não é razoável, no entanto, construir um compilador sem um tratamento especial para sentenças incorretas, visto que a compilação será interrompida na primeira ocorrência de um erro e assim, várias compilações serão necessárias até que todos os erros sejam encontrados, fato que onera consideravell mente o tempo de compilação.

Apesar de já existirem boas propostas sobre tratamento de erros, a maioria dos compiladores lançam mão apenas em um método simples que permite a continuidade da análise até o final da sentença mas não realiza um tratamento propriamente dito.

O compilador Algol do B/6700 possui um mecanismo tal que analisa o programa do usuário até o final, encontra um número razoável de erros mas não todos e emite mensagens de erro escritas em inglês que nem sempre correspondem a um diagnóstico correto.

Com a finalidade de facilitar e melhorar a aprendizagem do Algol, resolvemos contruir um analisador sintático que além de analisar o programa até o final, execute um tratamento para cada erro encontrado através de um critério prático e lógico, encontre o maior número de erros possível em uma única compilação e edite mensagens de erros claras, precisas e em português.

De acordo com o objetivo do trabalho, achamos desnecessário construir o analisador para toda a linguagem mas apenas para um subconjunto que deverá ser aquele comumente utilizado pelos iniciantes e pela maioria dos usuários.

No capítulo 1, conceituamos os erros segundo a fase de compilação em que são encontrados e diferenciamos os tipos de tratamento de erros de acordo com a política adotada. No capítulo 2, escolhemos o subconjunto da linguagem, fizemos uma análise sobre frequência de erros e um resumo bibliográfico sobre métodos de tratamento de erros, com o objetivo de analisar e escolher um método que melhor se enquadre no nosso trabalho. O capítulo 3 é uma revisão sobre o método de análise sintática LL e a determinação da solução para o problema da defasagem entre o momento da ocorrência de um erro e o momento do seu descobrimento que aparece nesta técnica de análise. O capítulo 4 descreve o funcionamento do método utilizado no tratamento de erros, algumas alterações a serem feitas no sentido de simplificar a técnica e os resultados de testes realizados antes da implementação final. Finalmente, no capítulo 5, falamos sobre os tratamentos realizados pelo analisador léxico e pelas ações semânticas, das mensagens de erro e das

opções e recursos utilizados na implementação do analisador.

CAPÍTULO 1

CONCEITUAÇÃO BÁSICA

Chama-se "erro em fase de compilação", qualquer falha cometida no programa, impedindo uma compilação satisfatória de acordo com a estrutura lógica do compilador.

Um problema considerado mais difícil e longe de ser resolvido que aparece ao se elaborar um método de tratamento de erros, diz respeito à intenção do programador pois as técnicas existentes e em desenvolvimento, não garantem que o tratamento dado à sentença incorreta, a transforma na sentença tal qual foi imaginada pelo programador. Os outros problemas possíveis de serem resolvidos, podem ser vistos sob os seguintes aspectos:

- a) o momento da ocorrência do erro em relação ao momento do seu descobrimento;
- b) o tipo de erro ocorrido: léxico, sintático ou semântico;
- c) geração das condições necessárias ao analisador para continuar a análise do resto da sentença;
- d) como transformar o texto errado em outro sintaticamente correto (e possível de ser executado).

Nos próximos capítulos, veremos como a técnica utilizada pelo analisador satisfaz a estes pontos. Neste capítulo, conceituaremos os três tipos básicos de erro e apresentamos os tipos clássicos de tratamento de erros.

I. CONCEITO DE ERRO LÉXICO, SINTÁTICO E SEMÂNTICO

A princípio poderemos dizer que um "erro léxico" é um erro cometido na fase léxica de compilação mas isto é um conceito impreciso visto que na maioria dos compiladores, as fases ou formas de análise acontecem paralelamente como se fossem sem co-rotinas, além disso, muitos erros ditos léxicos só poderão ser descobertos na fase de análise sintática ou na fase de análise semântica.

Os erros léxicos assim conceituados podem ser classificados em duas categorias:

1. Erros reconhecidos na fase léxica de compilação: uso de carácter inválido para a linguagem, digitação incorreta de números e se o analisador léxico for constituído com mais opções de testes: número de caracteres superior ao permitido para um tipo de símbolo da linguagem e valor de um número superior ou inferior ao permitido.
2. Erros reconhecidos pelas outras fases do processo de compilação: digitação incorreta de palavras reservadas vista pelo analisador sintático como um "identificador não esperado" e digitação incorreta de um identificador vista pelo analisador semântico como um "identificador não declarado".

Os analisadores sintático e semântico devem reconhecer os erros que se enquadram na categoria 2, como possíveis erros léxi

cos, invocando assim um procedimento de correção ortográfica.

Diante do que vimos, podemos concluir que um "erro l \bar{e} xico" \bar{e} essencialmente um erro de digita \bar{c} o ou erro de ortografia cometido pelo programador e |10| 80% deles s \bar{a} o devidos a um caracter errado, suprimido ou excedente ou ainda , troca de dois caracteres adjacentes.

Quando um s \bar{i} mbolo n \bar{a} o pode ocupar um determinado espaço em uma frase ou uma frase n \bar{a} o pode ocupar um determinado espa \bar{c} o em uma senten \bar{c} a pertencente a uma linguagem e o s \bar{i} mbolo ou a frase est \bar{a} o ocupando este espa \bar{c} o, ou ainda, quando um s \bar{i} mbolo n \bar{a} o encaixa na \bar{a} rvore sint \bar{a} tica da gram \bar{a} tica geradora de uma linguagem, dizemos que isto \bar{e} um "erro sint \bar{a} tico".

Quando tal acontece, o analisador sint \bar{a} tico rejeita o s \bar{i} mbolo causador do erro na frase ou na senten \bar{c} a, interrompendo seu processo natural e invocando um procedimento de tratamento de erros.

Os erros de sintaxe s \bar{a} o descobertos realmente na fase sint \bar{a} tica de compila \bar{c} o, por \bar{e} m, n \bar{a} o podemos perder de vista a possibilidade de serem erros l \bar{e} xicos, os causadores de erros sint \bar{a} ticos.

Os erros sint \bar{a} ticos tamb \bar{e} m s \bar{a} o causados na sua maioria, pela aus \bar{e} ncia, troca ou excesso de elementos, s \bar{o} que aqui o elemento passa a ser um s \bar{i} mbolo ao inv \bar{e} s de um car \bar{a} cter. Um s \bar{i} mbolo mal digitado tamb \bar{e} m provoca um erro sint \bar{a} tico.

Um problema t \bar{i} pico no tratamento de erros sint \bar{a} ticos \bar{e} o momento da ocorr \bar{e} ncia dos mesmos. Em muitos casos,

quando se descobre que um determinado símbolos não se encaixa na especificação sintática do texto em análise, este símbolo passa a ser apontado como errôneo. Entretanto, uma ausência, troca ou excesso de um outro símbolo, ou mesmo um outro símbolo com erro de digitação poderão ter sido o real causador do erro, não descoberto anteriormente pelo analisador sintático.

A fase semântica de compilação verifica o que é possível em termos de semântica, antes do programa entrar na fase de geração de códigos e execução. A semântica analisável nesta fase de compilação, chamamos de semântica estática e quando o compilador descobre incompatibilidades entre as definições previamente feitas para uma certa entidade e a utilização desta no programa, dizemos que ocorreu um "erro semântico".

Quando incompatibilidades semânticas são sentidas na fase de geração de códigos ou de execução, dizemos que ocorreu um "erro de execução".

Nas linguagens "fortemente tipificadas" são comuns os seguintes tipos de erro semântico:

1. o uso de identificador não definido anteriormente;
2. identificador definido como um tipo determinado e referenciado como outro;
3. variável fora do escopo.

Entende-se como linguagens "fortemente tipificadas" aquelas cujos identificadores são definidos inicialmente, ou seja, antes de serem referenciados.

De maneira sucinta, fizemos uma abordagem dos tipos de erros cometidos e dos problemas advindos destes. Faremos agora uma breve descrição dos tipos clássicos de tratamento de erro, segundo os objetivos propostos por cada um deles.

Os termos utilizados para nomear cada tipo de tratamento, tem uma finalidade didática e não são encontrados na maior parte da literatura que trata o assunto. Entretanto, a chamamos interessante adotá-los.

II. RECUPERADOR DE ERROS

No momento em que não é mais possível continuar a análise sintática devido à ocorrência de um erro, um tipo de tratamento deverá entrar em ação. Um deles chama-se "recuperador de erros" cuja ação é utilizar algum artifício para permitir a continuação da análise. Neste ponto, a geração de códigos é suspensa e a análise sintática continua graças a uma modificação feita na estrutura de dados do analisador sintático pelo recuperador de erros.

Na realidade, um recuperador de erros não recupera os erros mas permite ao analisador sintático se recuperar do impasse sofrido pelo aparecimento do erro e assim continuar a análise até o fim do programa, o que significa descobrir um número maior de erros.

A vantagem está na minimização do tempo de compilação pois ao invés de uma compilação para cada erro, passaremos a uma compilação para vários erros.

A desvantagem consiste no fato do próprio recupera -

dor de erros poder eventualmente, gerar outros erros que na realidade não existiram se a solução proposta por ele fosse a correta, isto é, correspondesse às intenções do programador.

Esses erros típicos, gerados por um tratamento inadequado são comumente chamados de "erros em cascata".

III. REPARADOR DE ERROS

Este outro tipo de tratamento de erros chamado "reparador de erros", não só possibilita a continuação da análise sintática como também, transforma o texto errado em outro sintaticamente correto mesmo que, muitas vezes, não corresponda ao texto original, ou seja, ao texto imaginado pelo programador.

Com a "reparação do erro", a suspensão da análise semântica e da geração de códigos torna-se desnecessária.

A vantagem está na possibilidade de descobrir um número de erros maior que os descobertos pelo recuperador e na capacidade de deixar o programa completo e sintaticamente correto.

A desvantagem é a incapacidade de transformar o texto incorreto no texto imaginado pelo programador, podendo por isso, ainda gerar alguns erros em cascata.

É conveniente observar que, em geral, um método de tratamento de erros onera muito o tempo de compilação e algumas vezes interfere até na análise de programas corretos, tornando-a mais lenta do que quando feita por um compilador desprovido de um método de tratamento de erros. Além disso, exis

te a possibilidade de textos muito longos serem inseridos no ato da reparação de erros sintáticos.

Em face a essas observações, torna-se importante salientar algumas propriedades que um reparador de erros deve possuir ou pelo menos, tê-las próximas do seu projeto e que chamaremos de "propriedades ideais de um reparador de erros".

São elas:

1. não provocar a geração de novos erros;
2. descobrir todos os erros existentes no programa ou pelo menos, o maior número possível;
3. o tempo de compilação de um programa correto em um compilador com reparador de erros, não deve ser superior ao tempo gasto por um compilador sem tratamento de erro;
4. possuir um bom desempenho em termos de tempo e espaço;
5. sempre ser possível se chegar ao final da análise;
6. os diagnósticos comunicados aos usuários através das mensagens de erro, sejam precisos e reais.

IV. CORRETOR DE ERROS

A um tipo especial de tratamento de erros capaz de substituir um texto incorreto por outro sintaticamente correto, imaginado pelo programador e possível de ser executado, chamaremos de "corretor de erros".

Este tipo de tratamento é impossível e um corretor de erros é por isso desejável mas não factível.

Finalizando o capítulo, queremos salientar que o nosso trabalho consiste num analisador sintático com "reparador de erros" e que as propriedades citadas no ítem III foram levadas em consideração.

CAPÍTULO 2

TÓPICOS PRELIMINARES

De acôrdo com a finalidade do trabalho, retiramos da gramática que especifica a sintaxe da linguagem Algol B/6700, um subconjunto que define apenas as declarações e comandos comumente utilizados pelo conjunto de alunos e maioria dos usuários. Para o aluno iniciante, este subconjunto é suficiente para iniciar o aprendizado e ter condições de escrever programas para uma grande variedade de problemas. Daqui por diante, chamaremos ao subconjunto escolhido da B6700/7700 Extended Algol Language de SAE e os seus componentes se encontram no Apêndice A.

Faremos neste capítulo, uma análise estatística sobre frequência de erros e um resumo bibliográfico sobre técnicas de tratamento de erros.

I. ANÁLISE ESTATÍSTICA SOBRE FREQUÊNCIA DE ERROS

A maioria dos artigos sobre tratamento de erros, contém alguns dados estatísticos sobre frequência de erros estudados empiricamente. Os que tratam o assunto com maior profundidade são poucos e os encontrados foram: Youngs [15] Litecky and Davis [11] e Ripley and Druseikis [12].

Tentaremos selecionar os dados mais importantes de maneira a sintetizar o assunto.

Alguns tipos de erro de sintaxe são:

1. Erro de retirada/ausência,
Ex.: ausência de parênteses direito;
2. Erro de troca,
Ex.: dois pontos no lugar de ponto e vírgula;
3. Erro de inserção,
Ex.: vírgula estranha, brancos extras;
4. Erro de transposição,
Ex.: palavra reservada mal escrita; transposição de letras.

Nem sempre é tão fácil identificar um erro como nos exemplos acima. Às vezes é difícil saber quantos são os erros e onde eles ocorrem.

Ex.: A:= B - C * D + E)

- haverá um parênteses direito a mais?
- faltará um parênteses esquerdo?
- se falta um parênteses esquerdo, em que lugar?

A maioria dos erros de sintaxe é devido somente à omissão ou uso errado de símbolos que trazem informações sobre propriedades estruturais do programa, tais como ponto e vírgula e os vários tipos de parênteses. Omissão de elementos que denotam explicitamente atividades do programa, tais como, operadores e operandos, são raros.

De uma análise de erros sintáticos feita com uma grande amostra (589) de programas, em PASCAL, foram obtidos os seguintes resultados:

1. 40,2% dos programas estavam sintática ou semanticamente incorretos;
2. 79,2% das declarações e comandos continham um único erro;
3. 12,9% das declarações e comandos continham apenas dois erros.

Os erros foram devidos a:

1. 41% - falta de um único símbolo;
2. 8% - símbolo extra;
3. 38,7% - escrita errada do símbolo;
4. 12,1% - outros casos envolvendo mais de um símbolo.

Consequentemente, com relação à maneira de tratamento, 85% dos erros poderiam ser corrigidos das seguintes maneiras:

1. 50% - inserindo-se um único símbolo;
2. 14% - retirando-se um único símbolo;
3. 21% - trocando-se um único símbolo.

Concluimos deste parágrafo que:

1. a maioria das declarações e comandos contém um único erro;
2. a maioria dos erros são devidos à falta de um único símbolo ou escrita errada de símbolo;
3. a maioria dos erros podem ser reparados através da inserção de um único símbolo.

Estas informações servirão de base para o nosso trabalho visto que, o PASCAL é uma linguagem estruturada semelhante ao ALGOL.

II. ERROS DE SINTAXE EM PROGRAMAS ESCRITOS EM ALGOL

Devido à falta de dados estatísticos de erros de sintaxe envolvendo a linguagem ALGOL, resolvemos efetuar uma pequena pesquisa com a finalidade de melhorar o nosso embasamento.

Numa amostra de 50 programas usando o compilador ALGOL do B/6700, obtivemos os seguintes resultados:

1. 38% dos programas estavam sintática ou semanticamente incorretos;
2. dos erros acusados pelo compilador:
 - 2.1. 36% eram erros já acusados anteriormente cujas mensagens emitidas para estes erros, chamamos de "mensagens repetidas";
 - 2.2. 30% foram causados por um símbolo estranho, mal digitado, trocado ou faltoso;
 - 2.3. 16% eram erros em cascata;
 - 2.4. 7% foram causados pelo uso de identificadores não declarados anteriormente;
 - 2.5. 11% outros.

Observando estes resultados e os programas analisados, chegamos às seguintes conclusões:

1. A maioria dos erros cometidos pelo programador são causados por um único símbolo, seja estranho, mal digitado, trocado ou faltoso e por isso, um tratamento de erros que inclua, retire ou troque símbolos, torna-se conveniente;
2. Um número considerável de erros apontados pelo compilador faz referência a erros já encontrados anteriormente os quais se tivessem sido devidamente reparados, não apareceriam novamente, onerando o tempo de compilação e emitindo mensagens desnecessárias.
3. Uma boa parte dos erros apontados, na realidade não existem, ou seja, são erros em cascata.
4. Além dos erros contabilizados, haviam outros que não foram apontados pelo compilador.
5. Uma boa parte das mensagens emitidas não eram adequadas ao erro encontrado.

Mostraremos alguns diagnósticos feitos pelo compilador:

```
BEGIN
FILE .....;
INTEGER .....;
INTEGER PROCEDURE FIBO (N);
INTEGER N;
    BEGIN
```

```

        INTEGER K;
        FIBO(1):= 0;
> > >                * SEMICOLON EXPECTED(1)
        FIBO(2):= 1;
> > >                * SEMICOLON EXPECTED(2)
        FOR .....
        STEP .....
        UNTIL .....
        DO FIBO(K):= FIBO(K-1)
> > >                * SEMICOLON EXPECTED(3)
        END OF PROC
BEGIN
INTEGER N;
READ (NUMERO, <I2>, N);
> > >                * UNDECLARED IDENTIFIER(4)
        :
        WRITE(....., N, FIBO(N)
                * UNDECLARED IDENTIFIER(5)
        END
        END.
> > >                * FINAL END NOT FOLLOW BY PERIOD(6)

```

As mensagens 1 , 2 e 3 registram a ocorrência de um mesmo erro e além disso, o diagnóstico não é real.

As mensagens 4 e 5 também registram a ocorrência de um mesmo erro sō que, neste caso, o erro não existe.

A mensagem 6 se refere ao ũltimo END com sendo um sĩmbolo extra onde na realidade, estĀ fechando o bloco princi-

pal.

O ";" seguidor do END OF PROC está faltando, erro que não foi acusado.

No comando final (WRITE) está faltando um parênteses do lado direito, erro que também não foi acusado.

A pesquisa que acabamos de fazer, ajuda bastante no sentido de reforçar a validade da nossa preocupação em fornecer ao programador iniciante um analisador sintático que descubra um maior número de erros, minimize a ocorrência de erros em cascata, evite a multiplicidade desnecessária de mensagens e forneça diagnósticos com maior precisão.

Faremos agora um resumo da literatura pesquisada sobre o assunto com a finalidade de selecionar um método de tratamento de erros simples, eficiente e que satisfaça os objetivos do nosso trabalho.

III. RESUMO DA LITERATURA PESQUISADA SOBRE TRATAMENTO DE ERROS.

Apresentaremos o resumo de algumas técnicas que julgamos mais interessantes:

1. Estado de pânico (panic mode).

É uma técnica intuitiva e a mais amplamente utilizada. Consiste em ignorar parte da cadeia de entrada a partir do símbolo errôneo, até que seja encontrado um símbolo de sincronismo. Geralmente considera-se um símbolo de sincronismo como sendo um delimitador de comando tal como ";" ou "END".

Uma alteração é realizada na configuração do analisador sintático de maneira a aceitar o símbolo de sincronismo e com isso, tornar-se possível a continuação da análise. Este método garante a continuação da análise em qualquer tipo de analisador sintático, no entanto, nenhum reparo é efetuado no local do erro e a parte do texto ignorado traz uma série de desvantagens tais como:

- a) os erros que eventualmente existam no texto ignorado, não serão descobertos;
- b) provoca o aparecimento de erros em cascata;
- c) nem sempre será possível a realização de um diagnóstico verdadeiro.

2. Recuperação pré-programada.

É outra técnica intuitiva e fácil de ser aplicada em analisadores sintáticos que utilizem tabelas.

Consiste em preencher as entradas vazias da tabela, ou seja, as entradas que indicam pontos de erro, com ponteiros para rotinas de erro. Essas rotinas deverão ser construídas manualmente, observando-se cada caso de erro, escolhendo-se uma solução tal como inclusão, retirada ou troca de símbolos de entrada e modificando-se a configuração da pilha sintática de acordo com a solução escolhida. Isto porém é bastante trabalhoso e já existem propostas que automatizam este processo.

3. Produções de erro.

É também uma técnica intuitiva e consiste em aumentar a gramática original com "produções de erro" tais que, algumas construções sintáticas incorretas passariam a ser aceitas pelo analisador sintático o qual seria induzido a executar ações especiais em casos de erro. A escolha das produções de erro é feita em função dos erros mais frequentes ocorridos na linguagem e deve-se tomar cuidado no sentido de não tornar a gramática ambígua, muito grande e não mais analisável sintaticamente.

Fischer | 5 | utiliza este método na análise sintática por precedência simples e sugere seu uso como uma "segunda linha de defesa" para melhorar o desempenho de um reparador de erros.

4. Inserção e retirada de símbolos localmente corretos e com custo mínimo.

Diferente das anteriores, esta é uma técnica formalizada que consiste em gerar uma cadeia de símbolos terminais da gramática com a finalidade de substituir zero ou mais símbolos da cadeia de entrada ainda não analisada, partindo-se do primeiro símbolo de entrada e errôneo. Desta forma, símbolos são inseridos, retirados ou trocados no local do erro e o critério de seleção de uma destas formas baseia-se nos custos dos símbolos terminais componentes da cadeia gerada, os quais são definidos de uma maneira experimental. A forma de reparo para cada caso será aquela de menor custo em relação às outras

possíveis formas que possam reparar o mesmo erro. O texto assim modificado estará sintaticamente correto e isto garante a continuação da análise até o final da sentença.

A técnica foi proposta e implementada por Fischer^[8] para o método de análise sintática LL(1) e exige que o analisador sintático possua as propriedades de descobrimento imediato do erro e de prefixo correto.

O algoritmo é simples e fácil de ser implementado e não aumenta significativamente o tempo e o espaço requisitado pelo analisador sintático.

A desvantagem é efetuar apenas correções locais e em casos onde se torne necessário alterar o texto anterior ao local do erro ou analisar a cadeia de entrada mais adiante com a finalidade de efetuar reparos mais precisos, os reparos oferecidos pelo método se tornam custosos com possibilidade de inserção de cadeias muito longas e distantes da correção natural.

5. Construção automática de analisadores corretores de erro.

É uma técnica formalizada e baseia-se na teoria de autômatos. Consiste na determinação de uma cadeia de continuação através de um autômato de continuação do autômato original. Isto significa que sempre será possível a determinação de uma cadeia de continuação desde que, o autômato original seja continuável. Para os autômatos dos analisadores sintáticos do tipo LL(k) sempre será possível a construção de um au-

tômato de continuação e para os analisadores dos tipos LR(k), SLR(k) e LALR(k), isto será possível com algumas modificações no algoritmo. Desta forma, sempre existirá uma cadeia de símbolos terminais w tal que:

- Se uv é uma cadeia de entrada, u foi aceita pelo analisador e está sendo apontado um erro no primeiro símbolo de v , então:
 - $uw'v''$ será uma sentença sintaticamente correta, onde: w' é o menor prefixo de w e substitui o prefixo v' de v .
 - v'' é o maior sufixo de v ,
 - $w = w'w''$, $v = v'v''$ e vw também pertence à linguagem.

Esta técnica foi proposta por Röhrich [13] e tem a propriedade de reparar sintaticamente qualquer sentença incorreta pertencente à linguagem. Entretanto, os reparos efetuados não correspondem ao melhor reparo local e são por isso distantes da realidade na maioria das vezes.

Nos itens anteriores deste capítulo, concluímos que a grande maioria dos erros de sintaxe cometidos, poderão ser reparados, incluindo-se, retirando-se ou trocando-se um ou mais símbolos no local do erro. Considerando-se esta conclusão, utilizaremos o método 4 que além de simples e eficiente, possui uma filosofia tal que satisfaz os objetivos do nosso trabalho.

CAPÍTULO 3O ANALISADOR SINTÁTICO

As técnicas de análise sintática existentes estão baseadas, de uma maneira geral, em duas formas de análise: a análise sintática descendente (TOP DOWN) e a análise sintática ascendente (BOTTOM UP).

O método de análise sintática recursivo descendente é um dos mais antigos e precursor do método de análise sintática LL o qual automatizou o processo de análise, através de tabelas e passou a ser utilizado durante muito tempo.

Com a formalização de método LR (ascendente), o método LL entrou um pouco em desuso, em virtude da classe de gramáticas aceita pelo primeiro ser maior que a classe aceita pelo segundo. No entanto, o espaço requisitado para armazenar as tabelas geradas pelo método LR é grande e dificulta sua implementação a nível comercial e apesar de existirem boas propostas de compactação das tabelas, a preferência é dada ao método LL, com bastante frequência.

As linguagens estruturadas tipo Algol cujas declarações e comandos são geralmente iniciados por uma palavra reservada, facilitam uma análise tipo descendente visto que, essas palavras reservadas já informam sobre a derivação a ser efetuada. Por outro lado, o método de tratamento de erros escolhido no capítulo 2, foi implementado pelo autor, num analisador sin

tático LL(1) com sucesso.

Neste capítulo, faremos uma revisão sobre o método LL e discutiremos o problema da defasagem entre o momento da ocorrência de um erro e o momento do descobrimento do mesmo.

I. GRAMÁTICAS LL(k)

Os analisadores sintáticos ditos eficientes, são caracterizados pelo fato da cadeia de símbolos de entrada ser explorada somente da esquerda para a direita e analisada através de um processo completamente determinístico.

É restrita a classe de gramáticas livres de contexto para a qual podemos construir analisadores sintáticos eficientes mas suficiente para especificar a sintaxe das linguagens de programação, sem considerar problemas de contexto.

A estrutura sintática de uma sentença pode ser determinada, pela sequência de produções usadas para derivar esta sentença.

O método LL trabalha deterministicamente, explorando a cadeia de símbolos de entrada da esquerda para a direita e verificando se existe uma árvore de derivação para esta cadeia, através de expansões associadas ao não terminal mais à esquerda da última forma sentencial construída, a partir do símbolo inicial.

As gramáticas LL(k) são aquelas para as quais pode-se construir um analisador sintático LL capaz de deduzir a produção a ser usada na derivação, olhando apenas os k primeiros símbolos de entrada.

Intuitivamente, podemos entender a idéia de uma gramática $LL(k)$, do seguinte modo:

Seja $G = (V_n, V_t, P, Z)$, uma gramática livre de contexto, onde:

V_n = conjunto dos símbolos não terminais,

V_t = conjunto dos símbolos terminais,

P = conjunto das produções,

Z = é o símbolo inicial e $V = V_n \cup V_t$.

Para a sentença $a_1 a_2 \dots a_n w \in L(G)$, onde $a_i, a_2 \dots a_n \in V_t$ e $w \in V_t^*$, existe uma única sequência de formas senten-
ciais mais à esquerda $\alpha_0, \alpha_1, \dots, \alpha_m$ tal que:

$$Z = \alpha_0, \alpha_i \implies \alpha_{i+1} \text{ para } 0 \leq i \leq m,$$

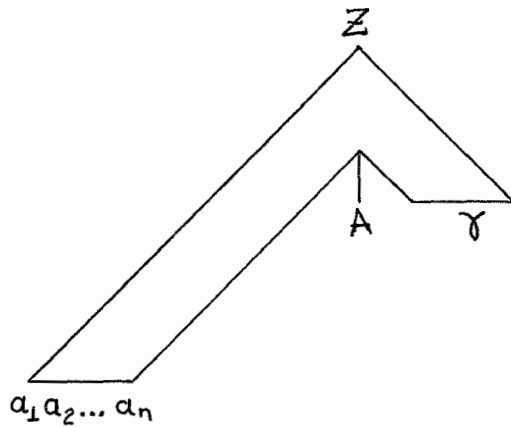
$$\alpha_i \in V^* \quad \text{e} \quad \alpha_m = a_1 a_2 \dots a_n w.$$

Com esta propriedade, podemos contruir uma árvore de derivação para a sentença, começando pela raiz e caminhando de cima para baixo (derivações mais a esquerda), deterministicamente, se conhecermos os k primeiros símbolos da parte ainda não processada, da cadeia de entrada.

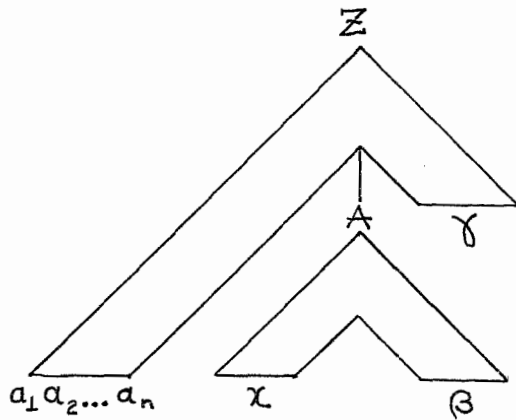
Assim, seja $w = xy$ onde $|x| = k$, a_1, a_2, \dots, a_n os símbolos da cadeia de entrada já processados que formam um prefixo de w .

Para algum $\alpha_i = a_1 a_2 \dots a_n A \gamma$, onde:

$A \in V_n$ e $\gamma \in V^*$, a árvore de derivação será:



Deve existir uma única sequência de produções tal que:
 $A \xRightarrow{*} x\beta$ e $\alpha_j = a_1 a_2 \dots a_n x\beta\gamma$, onde $\beta \in V^*$ e $i + 1 \leq j \leq m$. A
 árvore de derivação para α_j será:



Formalmente, podemos dizer que $G \in LL(k)$ se, dados uma cadeia $wA\beta \in V^*$ e os primeiros k símbolos terminais derivados de $A\beta$, existe no máximo uma produção que pode ser aplicada, a fim de que A derive uma cadeia de terminais iniciada por aqueles k símbolos e com prefixo w . Isto é, se

$$Z \xRightarrow{+} wA\beta \text{ e}$$

$$A \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_n \in P, \text{ então}$$

$FIRST_k(\alpha_1), FIRST_k(\alpha_2), \dots, FIRST_k(\alpha_n)$ são ao mesmo tempo disjuntos, onde:

$$FIRST_k(\alpha_i) = \{x \in Vt^* \mid |x| < k \text{ e } \alpha_i \xRightarrow{*} x \text{ ou}$$

$$\{|x| = k \text{ e } \alpha_i \xRightarrow{*} x \gamma \text{ para } \gamma \in V^*\}$$

II. GRAMÁTICA LL(1)

Dizemos que uma gramática \bar{G} é LL se \bar{G} é LL(k) para algum k. Dizemos que \bar{G} é LL(1) quando k=1.

Se, para todo $A \in V_n$, $A \rightarrow \epsilon \notin P$ e cada alternativa para A começa com um símbolo terminal distinto, então diremos que \bar{G} é uma "gramática simplesmente LL(1)". Isto significa que dado o par (A,a) onde $a \in V_t$, existe no máximo uma produção da forma $A \rightarrow a\alpha$, onde $\alpha \in V^*$.

Consideremos agora a gramática G com produções do tipo $A \rightarrow \epsilon$, para alguns $A \in V_n$. Então, diremos que \bar{G} é LL(1), se e somente se, para cada conjunto de produções de A, $A \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_n$, para todo $A \in V_n$, as seguintes condições são verdadeiras:

1. $FIRST_1(\alpha_1), FIRST_1(\alpha_2), \dots, FIRST_1(\alpha_n)$ são todos ao mesmo tempo disjuntos e, se $\alpha_i \xRightarrow{*} \epsilon$ para $1 \leq i \leq n$, então
2. $FIRST_1(\alpha_j) \cap FOLLOW_1(A) = \emptyset$, para $i \neq j$ e $1 \leq j \leq n$ e a função FOLLOW pode ser definida da seguinte maneira:

Se $\beta \in V^*$ então:

$$FOLLOW(\beta) = \{w \mid Z \xRightarrow{*} \alpha\beta\gamma \text{ e } w \in FIRST_k(\gamma)\}$$

onde k é uma constante inteira.

Se uma gramática G com produções do tipo $A \rightarrow \epsilon$ é tal que \bar{G} é LL(1), então diremos que G é uma "gramática fortemente LL(1)".

Se uma gramática G livre de contexto é tal que:

$A \rightarrow \beta$ e $A \rightarrow \gamma$ são produções distintas de A e,
 $\text{FIRST}_k(\beta \text{ FOLLOW}_k(A)) \cap \text{FIRST}_k(\gamma \text{ FOLLOW}_k(A)) = \emptyset$,

então diremos que esta gramática é uma "gramática fortemente $LL(k)$ ".

Toda gramática $LL(1)$ é uma gramática fortemente $LL(1)$ mas nem todas as gramáticas $LL(k)$ são fortemente $LL(k)$ para todo $k > 1$.

III. RESTRIÇÕES DE UMA GRAMÁTICA $LL(k)$

Em consequência da definição de gramática $LL(k)$, segue-se que:

1. Uma gramática livre de contexto, com recursividade à esquerda, não pode ser $LL(k)$ para algum k .
2. Uma gramática livre de contexto ambígua, não pode ser $LL(k)$ para algum k .

A recursividade à esquerda de uma gramática livre de contexto pode ser eliminada, da seguinte maneira:

Se $A \rightarrow Aa|b$, onde $A \in V_n$ e $a, b \in V_t$, então, é possível transformar essas produções em outras equivalentes:

$$A \rightarrow bA'$$

$$A' \rightarrow aA' | \epsilon$$

Estas novas produções pertencerão a uma gramática $G' = (V_n', V_t, P', Z)$ tal que V_n' é o conjunto dos símbolos não terminais de G e dos novos não terminais do tipo A' e P' é o conjunto das produções não modificadas de G e das novas produções que envolvem não terminais do tipo A' .

A ambiguidade de uma gramática livre de contexto pode ser eliminada através da fatoração das produções que produzem esta ambiguidade. Em geral, o processo de fatoração, envolve substituições de produções do tipo:

$$A \rightarrow \alpha\beta_1 | \alpha\beta_2 | \dots | \alpha\beta_n$$

por:

$$A \rightarrow \alpha A' \quad e$$

$$A' \rightarrow \beta_1 | \beta_2 | \dots | \beta_n$$

Estas novas produções pertencerão a uma gramática G' tal como foi definida anteriormente.

IV. O ANALISADOR SINTÁTICO COM GRAMÁTICAS LL(1)

Um analisador sintático construído para sentenças pertencentes a uma linguagem, $L(G)$ tal que G é uma gramática $LL(k)$ é também chamado de "analisador sintático preditivo" pelo fato de possuir a propriedade de "predizer" os k primeiros símbolos da parte não processada da cadeia de entrada.

Um analisador sintático para sentenças pertencentes a uma linguagem $L(G)$ tal que $G = (V_n, V_t, P, Z)$ é uma gramática livre de contexto $LL(1)$, pode ser visto em duas fases onde a pri

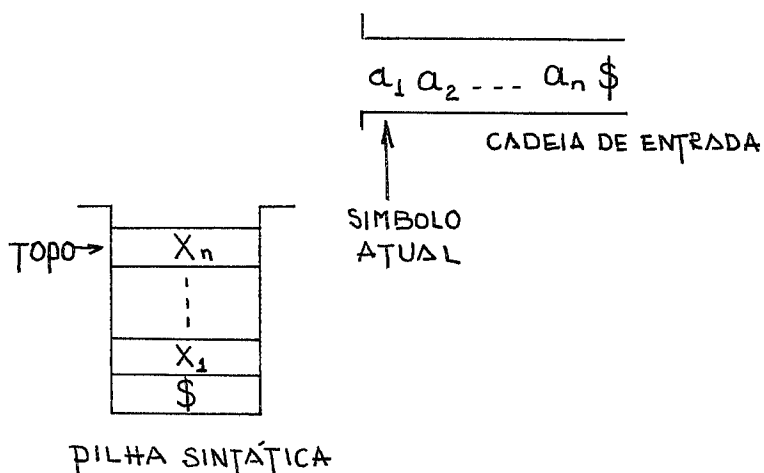
meira é uma função que recebe como entrada a gramática G e devolve como saída uma tabela T assim definida:

$$T = I \times i,$$

onde $I = (V \cup \{\$\})$, $i = (Vt \cup \{\epsilon\})$, $\$$ é um marco de fim de sentença que deverá ser acrescentado ao final da sentença a ser analisada e tal que:

1. Se $A \rightarrow \alpha$ é a n ésima produção em P , então $T(A,a) = (\alpha,n)$ para todo $a \in \text{FIRST}_1(\alpha)$ e $a \neq \epsilon$.
2. Se $\epsilon \in \text{FIRST}_1(\alpha)$, então $T(A,b) = (\alpha,n)$ para todo $b \in \text{FOLLOW}_1(A)$.
3. $T(a,a) = \text{"desempilhar"}$ para todo $a \in Vt$.
4. $T(\$, \epsilon) = \text{"aceitar"}$
5. $T(X,a) = \text{"erro"}$ para $X \in V \cup \{\$\}$ e $a \in Vt \cup \{\epsilon\}$ que não se incluam nos itens 1,2,3 e 4.

A segunda fase se refere à análise propriamente dita da sentença e pode ser representada pela seguinte configuração:



CADEIA DE ENTRADA guarda os símbolos a_i da sentença que ainda não foram processados onde $a_i \in I$.

PILHA SINTÁTICA guarda a última forma sentencial construída $X_n \dots X_2 X_1 \$$ tal que, se $Z \xrightarrow{*} w a_1 a_2 \dots a_n \$$ e w já foi processado então $X_n \dots X_2 X_1 \$ \xrightarrow{*} a_1 a_2 \dots a_n \$$ e $X_i \in I$

As ações a serem tomadas para o par (X_i, a_i) são definidas pela tabela T e sofrerão algumas modificações na consideração 5 quando ao analisador sintático for incorporado o reparador de erros.

V. O MOMENTO DA OCORRÊNCIA DE UM ERRO E O MOMENTO DO DESCOBRIMENTO DESTE EM ANALISADORES SINTÁTICOS LL(1).

A ação "erro" para o par (X_i, a_i) que se enquadre na consideração 5 do analisador sintático LL(1), significa uma incompatibilidade entre o símbolo atual de entrada e a estrutura sintática da sentença e portanto, não será mais possível continuar a análise. Neste ponto, um procedimento de reparo deve ser invocado para solucionar o problema, permitindo a continuação da análise.

A técnica de tratamento de erros a ser implementada exige, como pré-requisito, duas propriedades para o analisador sintático LL(1):

1. Propriedade do prefixo correto - os símbolos já aceitos pelo analisador sintático são garantidamente corretos e por isso formam um "prefixo correto" da cadeia de entrada ainda

não processada. Os analisadores sintáticos LL, possuem esta propriedade.

2. Propriedade do descobrimento imediato do erro - não existe defasagem entre o momento da ocorrência de um erro e o momento do descobrimento deste erro. Os analisadores sintáticos LL para gramáticas fortemente LL(1), não possuem esta propriedade.

A ocorrência de um erro e o não descobrimento imediato deste erro, pode induzir o analisador sintático a executar ações incorretas, alterando o conteúdo da pilha sintática e assim, deixando-a sem condições de informar a respeito de soluções adequadas para reparar o erro.

Em gramáticas fortemente LL(1), estas ações incorretas são devidas a produções do tipo $A \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_n$, onde $\alpha_1, \alpha_2, \dots, \alpha_n \in V^*$ e $\alpha_i \xRightarrow{*} \epsilon$ para algum $1 \leq i \leq n$ e que geralmente podem ser exemplificadas da seguinte maneira:

Seja a gramática G1:

1. $S \rightarrow E\$$
2. $E \rightarrow TE'$
3. $E' \rightarrow +T E' |$
4. ϵ
5. $T \rightarrow a |$
6. (E)

e a sentença: $a) \dots \$$, para a qual, os movimentos do analisador sintático seriam:

Nº DA AÇÃO	PILHA	ENTRADA	AÇÃO
1	\$E	a)...\$ Δ	expandir c/prod 2
2	\$E'T	a)...\$ Δ	expandir c/prod 5
3	\$E'a	a)...\$ Δ	aceitar simb entrada e desempilhar
4	\$E')...\$ Δ	expandir c/prod 4
5	\$)...\$ Δ	"erro"

"erro" deveria ter sido acusado já na ação 4 mas isto não aconteceu devido a $a \notin FOLLOW_1(E')$ e o símbolo E' foi retirado da pilha incorretamente.

Para solucionar o problema da defasagem entre o momento da ocorrência de um erro e o momento do seu descobrimento, existem algumas propostas.

Os analisadores sintáticos com gramáticas simplesmente LL(1), possuem a propriedade do descobrimento imediato do erro, visto que essas gramáticas não possuem produções do tipo $A \rightarrow \epsilon$ e por isso todo $A \in V_n$ é capaz de informar a respeito de expansões feitas anteriormente evitando com isso, movimentos incorretos do analisador. Isto significa que para todo $a \in V_t$, se $a \in FOLLOW_1(A)$, então certamente $Z \xRightarrow{+} \dots Aa \dots \$$. Fischer [7], adaptou aos não terminais de uma gramática fortemente LL(1), a característica dos não terminais de uma gramática simplesmente LL(1), modificando a estrutura da árvore sintática e obtendo com isso uma "gramática fracamente LL(1)" e uma tabela de tamanho consideravelmente maior que a anterior.

Ghezzi [9], também propõe uma transformação da gramã-

tica original em outra estruturalmente equivalente que chama de CR-LL(1) a qual propicia ao analisador sintático, a propriedade do descobrimento imediato do erro. A proposta não é muito atrativa, visto que:

- o algoritmo que transforma uma gramática LL(1) em uma gramática CR-LL(1) é semelhante ao algoritmo de construção do analisador sintático para gramáticas fracamente LL(1).
- a gramática CR-LL(1) é mais extensa que a gramática original e
- a tabela gerada pelo analisador sintático com gramáticas CR-LL(1), também é sensivelmente maior que a tabela gerada pelo analisador sintático anterior.

Uma proposta interessante é apresentada por Fischer [6] e consiste em algumas modificações na gramática fortemente LL(1), de maneira a transformá-la numa "gramática não anulável".

VI. UMA GRAMÁTICA NÃO ANULÁVEL

Dizemos que uma gramática livre de contexto é não anulável, se e somente se, cada produção é da forma:

$$A \rightarrow X_1 X_2 \dots X_n \quad (n \geq 1) \quad \text{ou}$$

$$A \rightarrow \epsilon, \quad \text{onde} \quad X_1 X_2 \dots X_n \neq \epsilon^+$$

Isto significa que ϵ sō poderā ser derivado em um passo.

Em um analisador sintātico construído com uma gramātica fortemente LL(1) nāo anulāvel, os ūnicos movimentos incorretos que poderāo ser executados devido a um sĭmbolo errōneo nāo descoberto imediatamente, serāo indūzidos por produções do tipo $A \rightarrow \epsilon$. Isto se deve ao fato de que, produções do tipo $A \rightarrow X_1 X_2 \dots X_n$ serāo preditivas de um sĭmbolo de entrada a , se e somente se, $a \in \text{FIRST}_1(X_1 X_2 \dots X_n)$, ou seja, se $X_1 X_2 \dots X_n \xRightarrow{*} a \dots$. Entretanto, esta prediçāo serā obviamente correta.

Baseando-se no que foi dito, o analisador sintātico farā uma avaliaçāo prēvia das ações a serem executadas, sem pre que estas ações forem induzidas por produções do tipo $A \rightarrow \epsilon$. A avaliaçāo serā feita do seguinte modo:

1. Observa-se a ação induzida pela produçāo ē desempilhar o sĭmbolo que se encontra no topo da pilha sintātica.
2. Se a ação seguinte ao desempilhamento ē aceitar um sĭmbolo de entrada, entāo a ação induzida pela produçāo deverā estar correta.
3. Se a ação seguinte ao desempilhamento ē empilhar uma nova produçāo diferente de $A \rightarrow \epsilon$, entāo a ação induzida pela produçāo anterior, tambēem deverā estar correta.
4. Se a ação seguinte ao desempilhamento nāo serā executada devido a ocorrēncia de um erro, entāo a ação induzida pela produçāo estarā obviamente incorreta. Neste ponto, os sĭmbolos desempilhados deverāo ser novamente colocados na pilha, restabelecendo-se a estrutura de dados anterior e o

procedimento de reparo deverá ser invocado.

5. Se a ação seguinte ao desempilhamento será induzida por uma nova produção do tipo $A \rightarrow \epsilon$, então serão feitas as considerações 1, 2, 3, 4 e 5 com relação ao próximo símbolo abaixo, na pilha sintática.

As pilhas sintáticas são geralmente implementadas em vetores e os símbolos desempilhados não serão perdidos, até que sejam sobrescritos por um novo empilhamento. Portanto, é fácil cancelar qualquer ação incorreta induzida por uma produção do tipo $A \rightarrow \epsilon$, incrementando-se o ponteiro do topo da pilha para a posição imediatamente abaixo do primeiro símbolo terminal encontrado. Deste modo, a pilha voltará a ter a estrutura relativa ao último símbolo de entrada correto e aceito pelo analisador sintático. Além disso, qualquer analisador sintático com gramáticas fortemente LL(1), construído desta maneira, terá a propriedade do descobrimento imediato do erro.

Toda gramática fortemente LL(1) pode ser algoritmicamente transformada numa gramática fortemente LL(1) não anulável equivalente. Para a classe de gramáticas fortemente LL(1), normalmente utilizadas na prática, a técnica de transformação obedecerá às seguintes regras:

- Se existe uma produção $A \rightarrow \alpha$, tal que $\alpha \xRightarrow{+} \epsilon$; isto significa que A deriva ϵ indiretamente. Então, criamos um conjunto alternado de produções que englobem derivações através de α , com as seguintes características:

1. $A \rightarrow A' \mid \varepsilon$
2. $A' \rightarrow \alpha$, onde $\alpha \not\stackrel{+}{\Rightarrow} \varepsilon$

Faz-se isto, observando-se as seguintes considerações:

1. Para toda produção $p = A \rightarrow X_1 X_2 \dots X_m$ para $m \geq 1$ e tal que $X_1 X_2 \dots X_m \stackrel{+}{\Rightarrow} \varepsilon$, faça:

Para todo não terminal B , do lado direito de p , faça:

- a) considere as produções

$$B \rightarrow \alpha_0 \mid \alpha_1 \mid \dots \mid \alpha_n \quad n \geq 0,$$

$$\text{onde } \alpha_0 \stackrel{*}{\Rightarrow} \varepsilon$$

- b) considere um novo não terminal B' tal que:

$$B' \rightarrow \alpha_1 \mid \alpha_2 \mid \dots \mid \alpha_n$$

- c) considere α_0 em toda sua extensão, ou seja, $\alpha_0 = Y_1 Y_2 \dots Y_m$ ($m \geq 0$)

- d) procure um Y_i , para $1 \leq i \leq m$, tal que $Y_i \stackrel{*}{\Rightarrow} a \dots$, onde a é o primeiro símbolo de qualquer sequência de símbolos terminais derivável de $Y_1 Y_2 \dots Y_m$.

- e) transforme então as produções $B \rightarrow \alpha_0 \mid \alpha_1 \mid \dots \mid \alpha_n$ iniciais nas seguintes produções:

$$B \rightarrow \varepsilon \mid Y_i Y_{i+1} \dots Y_m \mid B'$$

Efetuada essas alterações numa gramática G , a gramática G' resultante será tal que:

1. $L(G') = L(G)$
2. G' será uma gramática fortemente LL(1), não anulável.

3. A sequência de símbolos na pilha sintática, obtida durante a análise de qualquer cadeia $w \in L(G)$ através de G' , será idêntica àquela obtida durante a análise de w , através de G .
4. G' será ligeiramente mais extensa que G .
5. A tabela gerada pelo analisador sintático com G' não será consideravelmente maior que aquela gerada pelo analisador sintático com G .

VII. OPÇÕES PARA IMPLEMENTAÇÃO

Optamos por transformar a gramática que especifica o SAE em uma gramática fortemente $LL(1)$ não anulável que poderá ser vista no apêndice B.

Observamos porém que os exames efetuados com o conjunto de dados em toda ação induzida por uma produção do tipo $A \rightarrow \epsilon$, apesar de simples, são um pouco onerosos. Além disso, esses exames serão desnecessários em análise de programas corretos.

Partindo-se de que, os únicos movimentos incorretos executados pelo analisador sintático são desempilhamentos de não terminais que derivam ϵ , resolvemos porcessar a análise normalmente, auxiliada por um contador que será incrementado sempre que ocorra um desempilhamento deste tipo e zerado na primeira ação diferente da citada anteriormente que não seja "erro". No momento em que um erro for anunciado, verifica-se se o contador acusa algum desempilhamento e em caso afirmati-

vo, restabelece-se a pilha, incrementando-se o ponteiro do to po de tantas unidades quantas indicadas pelo contador. Na rea lidade, algumas verificações de erro poderão ser feitas inu tilmente. No caso de terem sido efetuados mais de um desempei- lhamento e como não sabe-se em qual deles estará contido o erro, torna-se necessário que a verificação seja feita a par- tir do primeiro desempilhamento. Isto poderá não ser ótimo mas é menos oneroso que a estratégia sugerida pelo autor.

Um problema que talvez se apresente ao se implementar esta técnica, diz respeito à destruição de ações semânticas associadas aos não terminais desempilhados. Isto no entanto, poderá ser contornado, não efetuando estas destruições até que a garantia seja dada a esses desempilhamentos ou evitando se associações de ações semânticas a não terminais que deri- vam ϵ .

CAPÍTULO 4

O REPARADOR DE ERROS

Neste capítulo, descreveremos o reparador de erros utilizando a técnica da inserção e retirada de símbolos localmente corretos e com custo mínimo.

O método foi desenvolvido por Fischer [8] em duas etapas. A primeira delas, lança mão apenas na inserção de uma cadeia de símbolos (um ou mais símbolos) imediatamente a esquerda do símbolo de entrada considerado errôneo, formando assim um prefixo correto tal que a sua inclusão na cadeia de símbolos de entrada garanta a aceitação sintática de pelo menos mais um símbolo da sentença e conseqüentemente, a continuidade do processo de análise até o final da sentença.

A segunda, acrescenta ao método uma outra forma de reparo que é a retirada de uma cadeia de símbolos. O método assim expandido torna-se satisfatório, visto que, as duas formas de reparo juntas implicam numa terceira que é a troca de símbolos (no caso de retirar e incluir símbolos ao mesmo tempo).

A função do método é avaliar cada caso e escolher uma solução para reparar o erro optando por uma simples inclusão, pela retirada ou pela troca de alguns símbolos.

Antes de entrarmos na descrição da técnica do reparador de erros, apresentaremos as definições básicas utilizadas como fontes de dados no processo de seleção da melhor solução para cada caso.

I- GRAMÁTICA AUMENTADA

Seja $G = (V_n, V_t, P, Z)$ uma gramática livre de contexto LL(1) onde,

V = conjunto de símbolos não terminais,

V_t = conjunto de símbolos terminais.

P = conjunto das produções,

Z = símbolo inicial e $V = V_n \cup V_t$

A técnica de análise sintática LL(1), reconhece o término de uma análise, através do marco "\$" que deverá ser acrescentado no final da sentença que se pretende analisar. Para que ocorra este reconhecimento, "\$" será colocado na base da pilha sintática e incluído na tabela gerente da análise T, da maneira como foi definido no capítulo 3. Portanto, "\$" não pertencerá (necessariamente), ao conjunto de símbolos terminais V_t , da gramática.

Com relação à técnica de análise sintática LL(1) com reparação de erros tal como foi definida, torna-se necessário a inclusão do marco "\$" no conjunto de símbolos terminais da gramática. Isto se deve à eventual necessidade de reparos no final de sentenças quando tentativas de inclusão de uma cadeia de símbolos ω serão efetuadas. Entretanto, isto só poderá ocorrer se, $Z \neq \omega$. Portanto, precisaremos de uma gramática aumentada $G' = (\widehat{V}_n, \widehat{V}_t, P', Z')$, equivalente a G , tal que:

$$\widehat{V}_n = V_n \cup \{Z'\},$$

$$\widehat{V}_t = V_t \cup \{\$, \}$$

$$P' = P \cup \{Z' \rightarrow Z \$\} \text{ onde,}$$

$$\$ \notin V_t, \quad Z' \notin V_n \quad \text{e} \quad \widehat{V} = \widehat{V}_n \cup \widehat{V}_t.$$

II - CUSTO DE UM SÍMBOLO TERMINAL

O custo de um símbolo terminal significa um valor previamente selecionado e associado ao símbolo terminal.

A seleção dos custos dos símbolos terminais de uma gramática, é feita experimentalmente e para isso é necessário um ótimo conhecimento da linguagem e da utilidade de cada símbolo terminal para os usuários desta linguagem. Portanto, a lista de custos pode variar, de acordo com a categoria dos usuários (iniciantes, com prática, etc).

O custo de um símbolo terminal será considerado ao incluir ou retirá-lo de uma sentença, no processo de reparação de um erro. Esta consideração se deve ao fato de que várias soluções serão testadas no mesmo local do erro, retirando-se e/ou incluindo-se cadeias de símbolos distintas. A melhor solução será aquela cuja soma dos custos dos símbolos terminais a serem incluídos e/ou dos custos dos símbolos terminais a serem retirados, é um valor mínimo com relação às outras soluções.

O custo de retirada de um símbolo terminal, é sempre maior do que o seu custo de inclusão, diminuindo a possibilidade da retirada indiscriminada de símbolos terminais que compõem a cadeia de entrada original (do usuário).

A seleção de custos para o reparador de erros, baseou-se no conjunto fornecido por Fischer [8], para os símbolos terminais do Algol 60. Aos demais terminais do SAE não constantes

no Algol 60, foram associados custos semelhantes, levando-se em consideração a proximidade dos tipos.

A lista de custos de inclusão e retirada encontra-se no apêndice C.

III - CADEIA DE SÍMBOLOS TERMINAIS COM CUSTO MÍNIMO

Baseando-se nos custos de inclusão dos símbolos terminais, associações de valores poderão ser feitas, a qualquer forma sentencial de G' através da função CUSTO DE INSERÇÃO, denotada por C e assim definida:

$$C(\epsilon) = 0,$$

$$C(a) = \text{custo de inclusão de } a \text{ para todo } a \in \hat{V}_t,$$

$$C(\$) \geq C(a) \text{ para todo } a \in V_t, \text{ a fim de evitar a inclusão de uma cadeia de símbolos terminais tal que provoque o fim da análise antes do final da sentença dada pelo usuário.}$$

$$C(x) = C(a_1) + C(a_2) + \dots + C(a_m) \text{ onde } x = a_1 a_2 \dots a_m \text{ para } m \geq 0 \text{ e } x \in \hat{V}_t^*,$$

$$C(\alpha) = C(X_1) + C(X_2) + \dots + C(X_n) \text{ onde } \alpha = X_1 X_2 \dots X_n \text{ para } n \geq 0 \text{ e } \alpha \in \hat{V}^*$$

Denominamos por cadeia de símbolos terminais com custo mínimo, a partir de um símbolo $X \in \hat{V}$; a cadeia de símbolos terminais x_1 , derivável de X e tal que $C(x_1)$ é mínimo em relação a qualquer outra cadeia de símbolos terminais que possa ser obtida de X .

A cadeia x_1 derivável de X será denotada por $S(X)$ e formalmente definida da seguinte maneira:

$$\text{Se: } X \stackrel{*}{\Rightarrow} x_1 | x_2 | \dots | x_m,$$

$$X \rightarrow x_1 x_2 \dots x_n \rightarrow \dots \rightarrow x_1 \text{ e}$$

$$\text{MIN } \{C(x_1), C(x_2), \dots, C(x_m)\} = C(x_1);$$

$$\text{onde: } X \in \widehat{V},$$

$$x_i \in \widehat{V}t^* \text{ e } m \geq 0,$$

$$x_j \in \widehat{V} \text{ e } n \geq 0;$$

$$\text{então: } S(X) = S(x_1 x_2 \dots x_n) \text{ ou}$$

$$S(X) = S(x_1) \text{ CAT } S(x_2) \text{ CAT } \dots \text{ CAT } S(x_n) \text{ ou}$$

$$S(X) = x_1, \text{ onde CAT} = \text{concatenar.}$$

$$\text{Consequentemente: } C(X) = C(S(X))$$

Passemos então, para o cálculo de $S(X)$ para todo $X \in \widehat{V}$. Em consequência direta da definição, temos:

$$S(\epsilon) = \epsilon,$$

$$S(a) = a \text{ para todo } a \in \widehat{V}t.$$

Para todo $A \in \widehat{V}_n$, $S(A)$ é calculada e armazenada numa tabela, identificada como TABELA S, através do seguinte algoritmo:

```

FOR todo  $A \in \widehat{V}_n$  DO  $C(A) \leftarrow \infty$ ;

REPEAT

    Pare  $\leftarrow$  TRUE;

    FOR toda produção  $p=(A \rightarrow X_1 X_2 \dots X_n)$ 

    DO IF  $C(X_1 X_2 \dots X_n) < C(A)$ 

        THEN BEGIN

             $C(A) \leftarrow C(X_1 X_2 \dots X_n)$ ;
             $S(A) \leftarrow S(X_1 X_2 \dots X_n)$ ;
            Pare  $\leftarrow$  FALSE
        END

UNTIL Pare

```

A atribuição $C(A) \leftarrow \infty$ feita inicialmente, significa que $S(A)$ é indefinida, ou seja, $S(A) = ?$ onde $? \notin \widehat{V}_t$.

Para todas as produções $p = (A \rightarrow X_1 X_2 \dots X_n)$ de G' , $C(X_1 X_2 \dots X_n)$ é calculado e comparado com $C(A)$. Se $C(X_1 X_2 \dots X_n) < C(A)$, então:

$$C(A) \leftarrow C(X_1 X_2 \dots X_n) \text{ e}$$

$$S(A) \leftarrow S(X_1 X_2 \dots X_n)$$

As iterações do laço REPEAT, significam vários percursos sobre as produções de G' , calculando, comparando e possivelmente trocando os valores de $C(A)$ e as cadeias $S(A)$. O laço se encerra quando $C(A)$ é garantidamente um valor mínimo para todo $A \in \widehat{V}_n$.

Na primeira iteração do laço REPEAT, serão efetuadas trocas para toda A que derive cadeias de símbolos terminais através de uma única derivação. Na segunda, para todo A que derive cadeias de símbolos terminais através de duas derivações e assim sucessivamente até que trocas possam ser efetuadas para todo A cujas as árvores de derivação das cadeias sejam as mais altas.

Todo $A \in \widehat{V}_n$, deriva uma cadeia de símbolos terminais com custo mínimo através de n derivações, onde $n \leq |\widehat{V}_n|$. Então serão necessárias pelo menos $|\widehat{V}_n|$ iterações do laço REPEAT para que se possa garantir que $C(A)$ é mínimo para todo $A \in \widehat{V}_n$.

IV - CADEIA DE INSERÇÃO COM CUSTO MÍNIMO

Seja: $xa \dots$, uma sentença a ser analisada

onde: $x \in Vt^*$ e $a \in \widehat{V}t$,

tal que: $Z' \xRightarrow{+} x \dots$ mas

$Z' \not\xRightarrow{+} xa \dots$

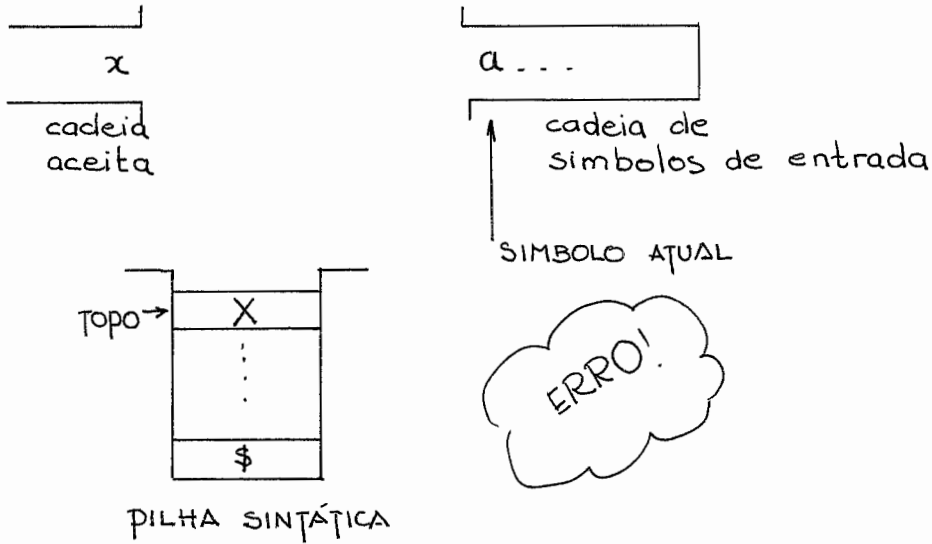
Suponhamos que para um certo $X \in \widehat{V}$, $y \in \widehat{V}t^*$ e $z \in \widehat{V}t^*$, as considerações abaixo, são verdadeiras:

$Z' \xRightarrow{+} x X a \dots$,

$X \xRightarrow{*} y \mid z$ e

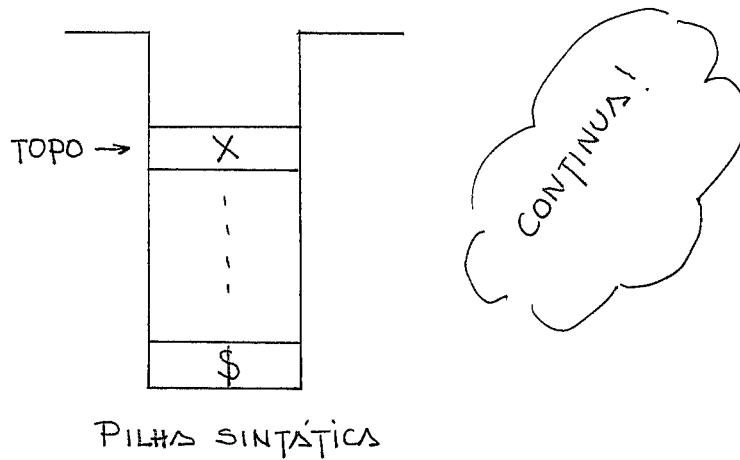
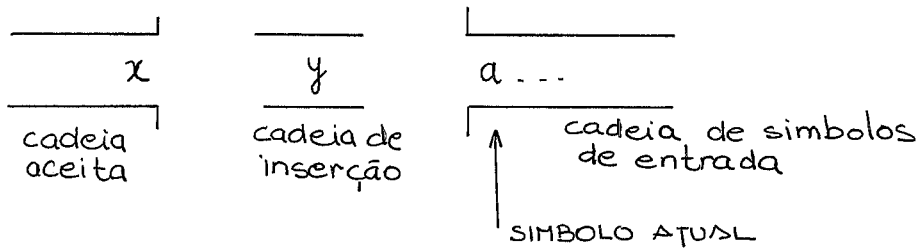
$\text{MIN} \{C(y), C(z)\} = C(y)$.

A configuração do analisador sintático, no momento em que x foi aceita e a está sendo apontado como um símbolo errôneo, é a seguinte:



Então, y será escolhida como sendo uma cadeia de símbolos terminais com custo mínimo tal que, incluída imediatamente à esquerda de a , permita a aceitação sintática deste a . Dizemos então que y é uma cadeia de inserção com custo mínimo para o par (X, a) onde, X é o símbolo do topo da pilha sintática e a é o símbolo atual e errôneo de entrada.

A situação do analisador sintático ao se inserir a cadeia y pode ser vista na seguinte configuração:



Denominamos de tabela E , a uma tabela que armazenará todas as cadeias de inserção com custo mínimo que poderão ser utilizadas na reparação de erros de sintaxe que possam ocorrer em uma sentença. Então, $E(X,a) = y$.

A tabela E será gerada automaticamente através do seguinte algoritmo:

```

FOR todo  $X \in \widehat{V}$ 
DO FOR todo  $a \in \widehat{V}t$  DO  $E(X,a) \leftarrow ?$ ;
FOR todo  $a \in \widehat{V}t$  DO  $E(a,a) \leftarrow \varepsilon$ ;
REPEAT

  Pare  $\leftarrow$  TRUE;
  FOR todo  $a \in \widehat{V}t$ 
  DO FOR toda produção  $p = (A \rightarrow X_1 X_2 \dots X_n)$ 
    DO BEGIN
       $K \leftarrow \text{MIN} (C(X_1 \dots X_{i-1}) + C(E(X_i, a)))$ ;
         $1 \leq i \leq n$ 

      Seja  $j$  tal que:
       $k = C(X_1 \dots X_{j-1}) + C(E(X_j, a))$ 

      IF  $K < C(E(A, a))$ 

      THEN BEGIN
         $E(A, a) \leftarrow S(X_1 \dots X_{j-1}) \text{ CAT } E(X_j, a)$ ;
        Pare  $\leftarrow$  FALSE
      END
    END
  END
UNTIL PARE

```

Seja novamente a sentença: $xa \dots$, onde $x \in \widehat{V}t^*$, $a \in \widehat{V}t$, $Z' \xRightarrow{+} x \dots$ mas $Z' \not\xRightarrow{+} xa \dots$.

Nem todo $X \in \widehat{V}$ é tal que $Z' \xRightarrow{+} X a \dots$ mas deve existir pelo menos um com esta característica.

A uma gramática $LL(1)$ com a propriedade acima, chamamos de gramática $LL(1)$ corrigível por inserção, ou seja, é uma gramática tal que, sempre será possível corrigir um erro, através da inserção de uma cadeia de símbolos terminais, imediatamente a esquerda do símbolo errôneo.

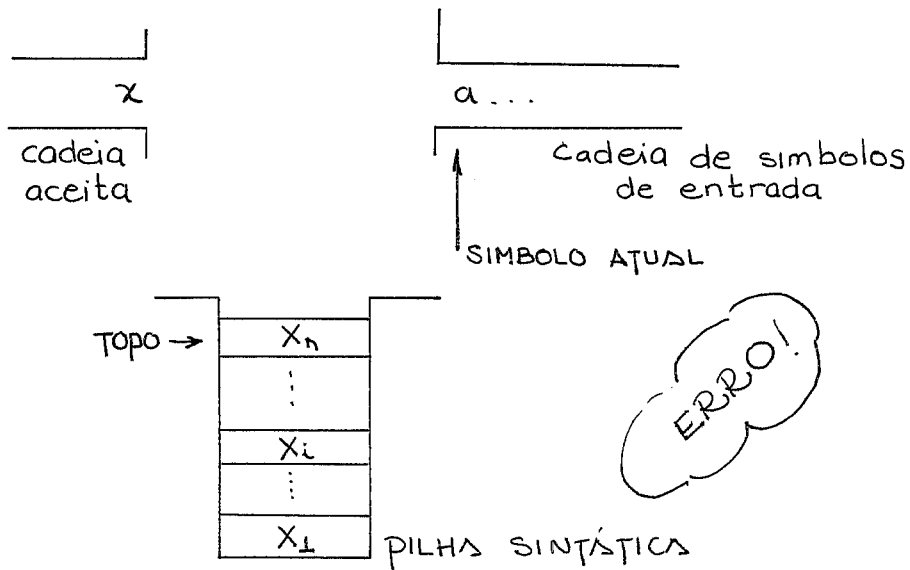
Com pequenas modificações, é possível transformar gramáticas que especificam linguagens tais, como o Algol e o Pascal, em gramáticas $LL(1)$ corrigíveis por inserção.

A inicialização do algoritmo gerador da TABELA E, considera todas as cadeias de inserção, como sendo ainda indefinidas e por isso $E(X,a) \leftarrow ?$ onde $C(E(X,a)) = \infty$, para todo $X \in \widehat{V}$ e $a \in \widehat{V}_t$, exceto para $X = a$, pois evidentemente, $E(a,a) = \epsilon$ onde $C(E(a,a)) = 0$. O algoritmo calcula as cadeias de inserção para todo par (A,a) , onde $A \in \widehat{V}_n$ e $a \in \widehat{V}_t$, desde que $Z' \xrightarrow{+} \dots Aa \dots$. Para $A \in \widehat{V}_n$ tal que $Z' \not\xrightarrow{+} \dots Aa \dots$, $E(A,a)$ permanecerá indefinida, assim como, $E(X,a)$ onde $X \in \widehat{V}_t$ e $X \neq a$, por ser impossível verificar se $Z' \xrightarrow{+} \dots Xa \dots$, partindo-se do próprio X.

O algoritmo tem um funcionamento semelhante ao do gerador da TABELA S, só que, os vários percursos sobre as produções de G' , serão efetuados para cada $a \in \widehat{V}_t$.

V - O REPARO UTILIZANDO SOMENTE CADEIA DE INSERÇÃO COM CUSTO MÍNIMO

Suponhamos que a configuração do analisador sintático, no momento da ocorrência de um erro, seja a seguinte:



Para esta ou qualquer outra situação particular do analisador sintático, é possível existir mais de uma cadeia de inserção com custo mínimo, ou seja, é possível que $E(X_n, a) = x$ e/ou $E(X_i, a) = y$ e/ou $E(X_{i-1}, a) = z$, para alguns X_i da pilha sintática.

Para considerarmos cadeias de inserção, relativas a um par (X_i, a) , onde X_i não é o símbolo do topo da pilha sintática, é necessário considerarmos também, o prefixo correto de $E(X_i, a)$ proveniente dos símbolos que se encontram empilhadas imediatamente acima de X_i .

Denominamos "cadeia de inserção local", a uma cadeia de símbolos terminais definidas por:

$$S(X_n X_{n-1} \dots X_{i+1}) \text{ CAT } E(X_i, a),$$

para $1 \leq i \leq n$,

onde $S(X_n X_{n-1} \dots X_{i+1})$ é a concatenação das cadeias de custo mínimo deriváveis de $X_n X_{n-1} \dots X_{i+1}$,

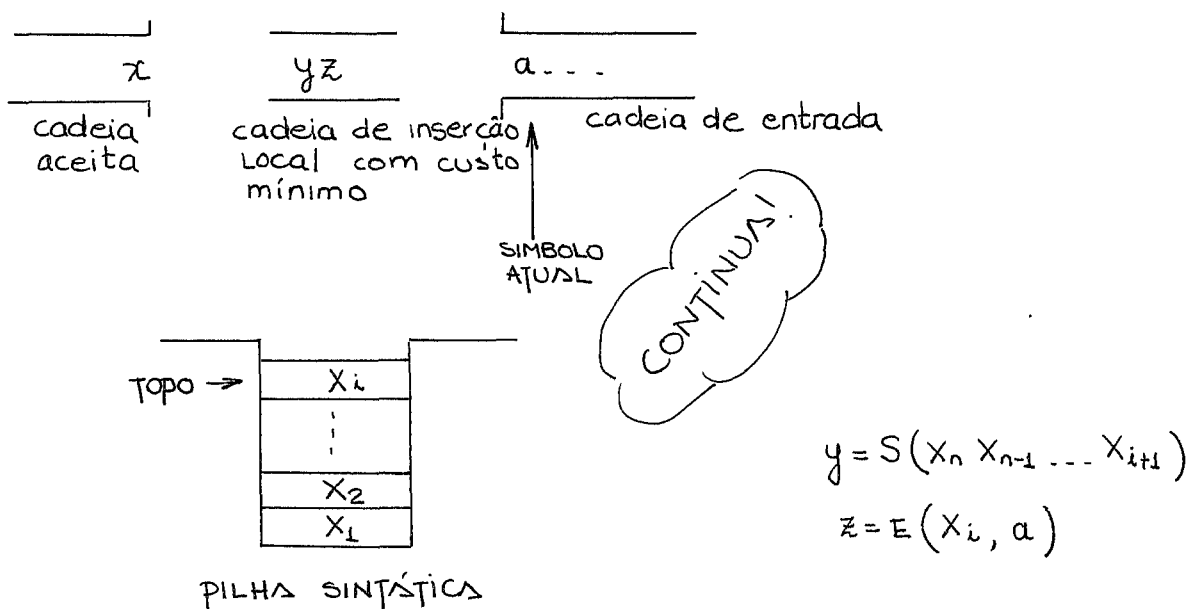
e $X_n X_{n-1} \dots X_{i+1}$ são todos os símbolos que se encontram em pilhados imediatamente acima de X_i , na pilha sintática.

Ou seja, $S(X_n X_{n-1} \dots X_{i+1})$ é um prefixo correto de $E(X_i, a)$. Portanto, podemos definir uma função chamada PREFIXO, como sendo:

$$\text{PREFIXO} = S(X_n X_{n-1} \dots X_{i+1}).$$

O reparador de erros percorrerá a pilha sintática de cima para baixo, até encontrar a "cadeia de inserção local com custo mínimo", que será escolhida como "solução ótima". Portanto, a solução considerada ótima será aquela para a qual $C(\text{PREFIXO}) + C(E(X_i, a))$ é um valor mínimo, para qualquer X_i da pilha sintática.

A situação de erro do analisador sintático, utilizando esta idéia está representada na figura abaixo:



Na prática, nem sempre será necessário percorrer a pilha sintática até a base. Algumas condições serão suficientes para garantir que a solução ótima já foi encontrada. São elas:

1. Quando X_i é um símbolo terminal igual ao símbolo atual de entrada, pois isto significa que $C(E(X_i, a)) = 0$ e daí por diante não será possível encontrar uma cadeia de inserção local com custo menor que as encontradas anteriormente.
2. Quando X_i é um símbolo não terminal tal que $C(E(X_i, a)) = 0$, o que significa que não haverá mais símbolos terminais a serem concatenados à cadeia de inserção local.
3. Quando $C(\text{PREFIXO})$ é igual ou maior ao custo da cadeia de inserção local, atualmente considerado mínimo.

O algoritmo do reparador de erros é uma função que recebe como entrada a pilha sintática e o símbolo atual e erro de entrada e devolve como saída, a cadeia de inserção local com custo mínimo que está representada no algoritmo por INSERI. É o seguinte:

```

Inseri ← ?;
Prefixo ← ε;
FOR I ← n STEP - 1 UNTIL 1
{percorrendo a pilha sintática de cima p/baixo}
DO BEGIN
    IF C(Prefixo) ≥ C(Inseri)
    THEN RETURN (Inseri);
    {não será possível encontrar solução mais barata}
    IF C(Prefixo CAT E(Xi,a)) < C(Inseri)
    THEN Inseri ← Prefixo CAT E(Xi,a);
    {foi encontrada uma solução mais barata}
    Prefixo ← Prefixo CAT S(Xi)
END
RETURN (Inseri)

```

VI - O REPARO UTILIZANDO INSERÇÃO E RETIRADA DE CADEIAS DE SÍMBOLOS TERMINAIS

O custo de retirada de uma cadeia de símbolos terminais será calculado através da função D , assim definida:

$$D(\epsilon) = 0,$$

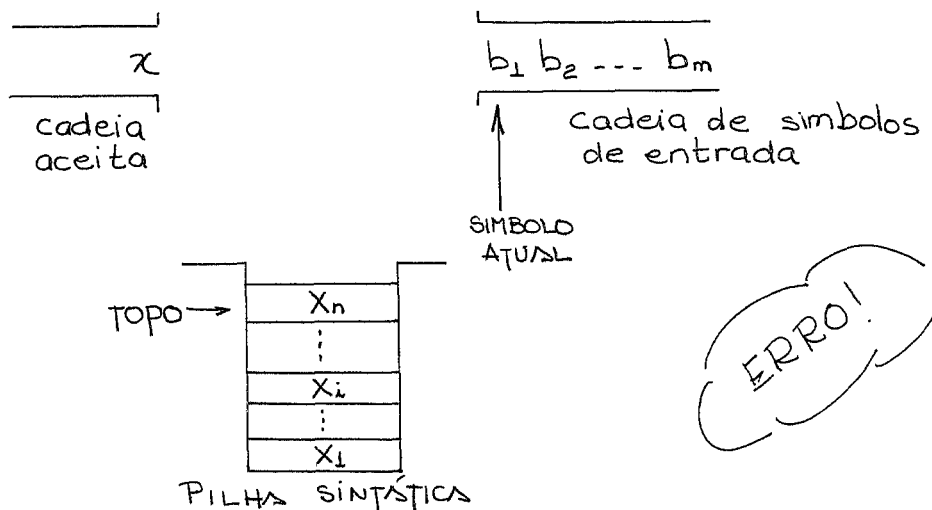
$$D(a) = \text{custo de retirada de } a \text{ para todo } a \in \hat{V}_t,$$

$$D(\$) = \infty, \text{ pois } \$ \text{ é um símbolo garantidamente correto.}$$

$$D(\omega) = D(a_1) + D(a_2) + \dots + D(a_m) \text{ onde}$$

$$\omega = a_1 a_2 \dots a_m \text{ para } m \geq 0 \text{ e } \omega \in \hat{V}t^*.$$

Seja a sentença: $\dots x b_1 b_2 \dots b_m$, onde $x \in Vt^*$ e $b_1 b_2 \dots b_m \in \hat{V}t$ para $m \geq 1$. Suponhamos que x já foi aceita pelo analisador sintático e que b_1 está sendo apontado como símbolo errôneo. A configuração do analisador sintático será:



Isto significa que $Z' \xRightarrow{*} \dots x \dots$
 mas $Z' \not\xRightarrow{*} \dots x b_1 b_2 \dots b_m$

Para esta ou qualquer outra situação particular do analisador sintático, poderemos ampliar o conceito de solução ótima, observando-se não só o símbolo de entrada atual e errôneo mas também os próximos primeiros símbolos de entrada imediatamente à direita do atual. Isto significa que passaremos a considerar várias soluções ótimas representada, pela cadeia de símbolos terminais:

PREFIXO CAT $E(x_i, b_i)$, para alguns b_i da cadeia de símbolos terminais de entrada.

Para considerarmos uma solução ótima representada pela expressão acima, onde b_i não é o primeiro símbolo da cadeia de entrada, é necessário considerarmos também a retirada dos símbolos $b_1 b_2 \dots b_{i-1}$ que se encontram imediatamente à esquerda de b_i na cadeia de símbolos de entrada. A solução ótima terá uma definição ampliada, como se segue:

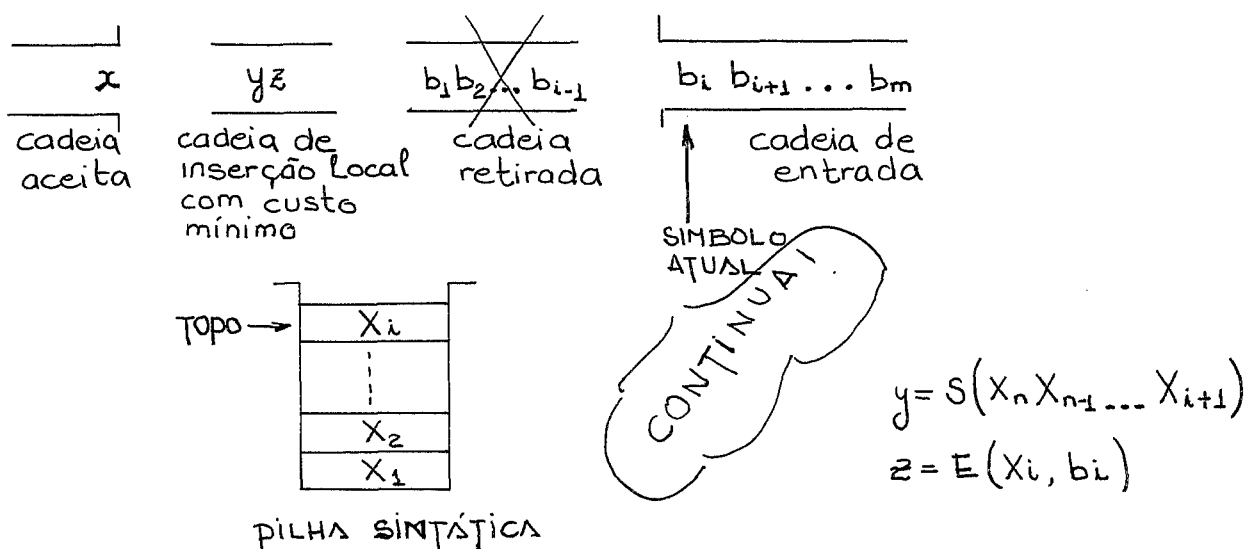
PREFIXO CAT $E(X_i, b_i)$ DEL($b_1 b_2 \dots b_{i-1}$), onde DEL($b_1 b_2 \dots b_{i-1}$) significa a retirada dos símbolos $b_1 b_2 \dots b_{i-1}$.

Denominamos de "solução local ótima" a uma "solução local" tal que:

$C(\text{PREFIXO}) + C(E(X_i, b_i)) + D(b_1 b_2 \dots b_{i-1})$ é um valor mínimo, para qualquer X_i da pilha sintática e qualquer b_i da cadeia de símbolos de entrada.

O reparador de erros, passará a percorrer a pilha sintática de cima para baixo, para cada símbolo da cadeia de entrada, a partir do primeiro, até encontrar uma solução local ótima.

A situação de erro do analisador sintático utilizando essa nova idéia pode ser vista na configuração abaixo:



O processo de seleção da solução local ótima terminará quando o último símbolo da cadeia de entrada - \$ - é chamado ou quando o custo de retirada acumulado é maior que o valor da solução local, atualmente considerada ótima.

O algoritmo do reparador de erros com inserção e retirada de símbolos é uma função que recebe como entrada a pilha sintática $X_n X_{n-1} \dots X_1$ e a cadeia de símbolos de entrada restante $b_1 b_2 \dots b_m$; devolve como saída, INSERI e RETIRA onde INSERI é a cadeia de inserção local com custo mínimo e RETIRA é o comprimento da cadeia de símbolos de entrada a ser retirada da sentença, a partir do símbolo atual e errôneo. O algoritmo do reparador de erros que utiliza somente cadeias de inserção local com custo mínimo, passa a ser a subrotina LL-INSERI $(X_n X_{n-1} \dots X_1, b_i)$. O algoritmo é o seguinte:

```

Inseri ← ?;
Retira ← 0;
FOR i ← 1 TO m
{chamando os símbolos da cadeia de entrada}
DO BEGIN
    IF  $D(b_1b_2 \dots b_{i-1}) \geq C(\text{Inseri}) + D(b_1b_2 \dots b_{\text{Retira}})$ 
    THEN RETURN (Retira, Inseri)
    {não é possível encontrar uma solução local + barata}
    IF  $C(\text{LL-INSERI}(X_n X_{n-1} \dots X_1, b_i) + D(b_1b_2 \dots b_{i-1})$ 
    <  $C(\text{Inseri}) + D(b_1b_2 \dots b_{\text{Retira}})$ 
    THEN BEGIN
        {foi encontrada uma solução local + barata}
        Inseri ← LL-INSERI ( $X_n X_{n-1} \dots X_1, b_i$ );
        Retira ← i-1
    END
END
END

```

O método de reparação de erros tal como foi projetado, torna o processo de análise sintática um pouco mais complexo visto que vários símbolos de entrada adiante do símbolo atual, poderão ser chamados sem a garantia de entrada na cadeia de símbolos a ser retirada. Entretanto, comprovou-se na prática que poucos serão esses símbolos e, além disso, jamais será preciso chamar símbolos além de b_i , quando $D(b_1b_2 \dots b_i) \geq C(\text{Inseri})$. Uma solução para o problema é guardar os símbolos

já chamados e não processados numa fila auxiliar. No momento em que o analisador sintático retomar suas atividades normais, a fila auxiliar terá prioridade sobre a fila de símbolos de entrada, durante as próximas n primeiras chamadas de símbolo de entrada, onde n é o número de símbolos que se encontram na fila auxiliar. No entanto, é possível simplificar o método, examinando apenas um símbolo de entrada além daquele que garantidamente pertencerá à cadeia de símbolos de entrada a ser retirada da sentença. Deste modo, restará apenas um símbolo de entrada já chamado e não processado. É claro que, ao se optar por esta simplificação, nem sempre será possível se conhecer a solução local ótima, tal como foi definida mas mesmo assim, acobertará a maioria das formas de reparo qualificadas como satisfatórias no capítulo 2 (inclusão, retirada ou troca de um único símbolo).

VII - TESTES DE IMPLEMENTAÇÃO

A simplificação sugerida no final do item anterior foi adotada e alguns testes foram realizados, antes da implementação final do analisador sintático com reparador de erros para o SAE.

1º TESTE:

Gramática Exemplo:

1. $S \rightarrow E \$$

2. $E \rightarrow T E'$

$$3. T \rightarrow F T'$$

$$4.5. E' \rightarrow + E \mid \xi$$

$$6.7. F \rightarrow (E) \mid a$$

$$8.9. T' \rightarrow * T \mid \xi$$

As tabelas S e E foram geradas separadamente. Os resultados foram dados de entrada para o programa principal, estruturados da seguinte maneira:

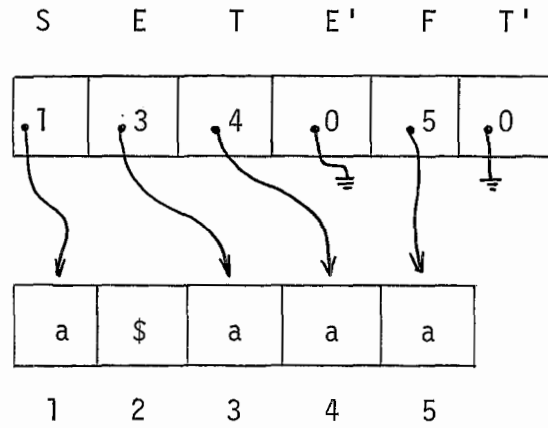
1. Vetor de custos de inserção dos símbolos terminais e não terminais da gramática, onde o custo de inserção de um não terminal significa o custo da cadeia de custo mínimo derivável deste não terminal.

1	1	1	1	1	1	2	1	1	0	1	0
\$	+	()	a	*	S	E	T	E'	F	T'

2. Vetor dos tamanhos das cadeias de custo mínimo deriváveis dos não terminais.

2	1	1	0	1	0
S	E	T	E'	F	T'

3. Vetor de ponteiros para o início das cadeias de custo mínimos e vetor poço das cadeias de custo mínimo.



4. Tabela A de custos das cadeias de inserção com custo mínimo relativas aos pares (A,a) , onde A é um não terminal e a um terminal.

	\$	+	()	a	*
S	1	1	0	2	0	1
E	∞	1	0	2	0	1
T	∞	2	0	2	0	1
E'	∞	0	1	3	1	2
F	∞	2	0	2	0	2
T'	∞	3	1	3	1	0

5. Tabela B de produções a serem usadas nas expansões processadas pelo reparador de erros

	\$	+	()	a	*
S	1	1	1	1	1	1
E	0	2	2	2	2	2
T	0	3	3	3	3	3
E'	0	4	4	4	4	4
F	0	6	6	6	7	6
T'	0	8	8	8	8	8

6. Tabela C de ponteiros para o primeiro símbolo do lado direito da produção não incluído no conjunto de símbolos geradores das cadeias com custo mínimo que compõem a cadeia de inserção.

	\$	+	()	a	*
S	2	1	1	1	1	1
E	0	2	1	1	1	1
T	0	1	1	1	1	2
E'	0	1	2	2	2	2
F	0	2	1	3	1	2
T'	0	2	2	2	2	1

7. Tabela D dos tamanhos da cadeias de inserção

	\$	+	()	a	*
S	1	1	0	2	0	1
E	0	1	0	2	0	1
T	0	2	0	2	0	1
E'	0	0	1	3	1	2
F	0	2	0	2	0	2
T'	0	3	1	3	1	0

8. Tabela F dos ponteiros para o início das cadeias de inserção que se encontram num poço.

	\$	+	()	a	*
S	1	5	0	25	0	35
E	0	2	0	20	0	30
T	0	6	0	15	0	29
E'	0	0	12	22	28	31
F	0	3	0	13	0	33
T'	0	8	11	17	27	0

9. Vetor poço das cadeias de inserção

a	a	(a	a	(a	*	(a	*	+	(a	(a	*
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

(a	(a	+	(a	(a	*	+	a	a	+	a	(a	a
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35

10. Vetor de custos de retirada dos terminais

∞	1	1	1	1	1
----------	---	---	---	---	---

\$ + () a *

Baseando-se nestes dados, o processo de reparação de erros, seguirá os seguintes passos:

1. Escolhe o par (X_i, b_i) responsável pela solução local ótima.
2. Concatena as cadeias com custo mínimo deriváveis dos símbolos $X_n, X_{n-1}, \dots, X_{i+1}$ que se encontram acima de X_i na pilha sintática e também a cadeia de inserção relativa ao par (X_i, b_i) .

$$\text{Inseri} \leftarrow S(X_n, X_{n-1}, \dots, X_{i+1}) \text{ CAT } E(X_i, b_i).$$
3. Retira da sentença, os símbolos b_1, b_2, \dots, b_{i-1} que se encontram à esquerda de b_i .
4. Desempilha $X_n, X_{n-1}, \dots, X_{i+1}$.

Número de erros = 10

Reparos a serem feitos pelo programador, examinados pelo reparador de erros:

```
erro 1 : incluua "a"  
erro 2 : incluua "a"  
erro 3 : incluua "*" "  
erro 4 : incluua "*" "  
erro 5 : incluua "*" "  
erro 6 : incluua "a"  
erro 7 : incluua "a"  
erro 8 : incluua "a"  
erro 9 : incluua "*" "  
erro 10: incluua "a))"
```

Como mostra o exemplo, o reparador de erros evita sempre que possível, retirar símbolos que compõem a cadeia de entrada do usuário, mesmo neste caso, onde aos custos de retirada dos terminais foram associados valores iguais aos custos de inclusão.

Os resultados dos testes abaixo, mostram a eficiência do algoritmo em termos de tempo de execução:

Sentença correta: 0.46 seg

Sentença com erros de sintaxe: 0.50 seg

O tempo de execução de cada reparação foi de aproximadamente 0.008 seg. Para a gramática que especifica o SAE, onde

$|\widehat{V}_t| = 63$, $|\widehat{V}_n| = 83$ e $|P| = 180$, o espaço de memória requisitado para guardar as tabelas e vetores utilizados pelo reparador de erros, é muito grande e tendo em vista, a eventualidade dos erros em um programa, resolvemos eliminar as tabelas C, D, F e o vetor poço das cadeias de inserção. Para cada erro ocorrido, a cadeia de inserção necessária para a reparação deste erro, será gerada dentro do próprio algoritmo do reparador de erros que apesar de onerar um pouco o tempo de execução, é compensatório em termos de espaço de memória.

O processo de reparação de erros sofrerá as seguintes modificações:

1. Depois de escolher o par (X_i, b_i) , concatenará apenas as cadeias com custo mínimo relativas aos símbolos $X_n, X_{n-1}, \dots, X_{i+1}$ que se encontram acima de X_i na pilha sintática.

$$\text{Inseri} \leftarrow S(X_n, X_{n-1}, \dots, X_{i+1})$$
2. Depois de expandir um não terminal do topo da pilha, faz:
 - 2.1. Escolhe um símbolo X_j da pilha, tal que $C(E(X_j, b_i))$ é um valor mínimo para qualquer outro símbolo da pilha sintática.
 - 2.2. Concatena a Inseri , as cadeias com custo mínimo deriváveis dos símbolos que se encontram acima de X_j na pilha sintática.

$$\text{Inseri} \leftarrow \text{Inseri} \text{ CAT } S(\dots X_{j+1})$$

- 2.3. Desempilha os símbolos que se encontram acima de X_j na pilha sintática.
3. Se não é possível expandir o não terminal A que se encontra no topo da pilha, então concatena a $Inseri$, a cadeia de custo mínimo derivável deste não terminal e desempilha-o. $Inseri \leftarrow Inseri \text{ CAT } S(A)$
4. Se o símbolo que se encontra no topo da pilha é um terminal a diferente de b_i , então concatena-o a $Inseri$ e desempilha-o. $Inseri \leftarrow Inseri \text{ CAT } a$

TESTE 2: Uma pequena gramática que especifica alguns comandos e declaração do Algol, foi utilizada na observação de reparos efetuados, utilizando apenas cadeias de inserção local com custo mínimo. Foi processado um programa correto e um programa com erros de sintaxe e obtidos os seguintes resultados:

```
          Programa  correto:
BEGIN
INTEGER VAR, NUM, VALOR;
PROCEDURE SOMA (X,Y);
    BEGIN
        INTEGER AUX;
        AUX: = 100;
        X: = Y + AUX
    END SOMA;
NUM: = 25;
VAR: = NUM;
IF SOMA
THEN VALOR: = NUM + VAR
ELSE SOMA (NUM, VAR)
END PROGRAMA. $
```

Tempo de execução = 0.85 seg

Programa com erros de sintaxe:

```

BEGIN

INTEGER VAR, NUM, VALOR;

PROCEDURE SOMA (X + Y):
                ↑  ↑
                1  2

    BEGIN

    INTEGE AUX;
              ↑
              3

    AUX: = 100;

    X:= Y + AUX

    END SOMA,
          ↑
          4

NUM: 25;
  ↑  ↑
  5  6

VAR: = NUM;

IF SOMA VALOR: = NUM + VAR
   ↑  ↑
   7  8

ELSE SOMA (NUM, VAR

END PROGRAMA . $
↑          ↑
9          10

```

Número de erros encontrados = 10

Número de erros reais = 6

Número de erros em cascata = 4

Mensagens enviadas ao programador:

erro 1 : incluua ");ID:=1"
erro 2 : incluua "*(1"
erro 3 : incluua ";"
erro 4 : incluua ";ID(ID"
erro 5 : incluua "); BEGIN ARRAY ID [1"
erro 6 : incluua "] "
erro 7 : incluua "AND"
erro 8 : incluua "THEN ID"
erro 9 : incluua ")"
erro 10: incluua "END"

A penalização sofrida pelo programa ao considerarmos estes reparos, deve-se do fato de inclusão de cadeias em locais onde os erros foram causados pela troca de um símbolo - erros 1, 4 e 5. A solução para o erro 3 não foi correta devido a ausência do analisador semântico mas se olharmos do ponto de vista sintático, não poderemos dizer o mesmo. Os erros 2, 6, 8 e 10 foram consequências de um reparo incorreto para os erros 1, 5, 7 e 5 respectivamente. O reparo incorreto para o erro 7 deve-se a impossibilidade do reparador de erros conhecer as intenções do programador. O erro 9 foi devidamente reparado.

Tempo de execução : 0.88 seg

TESTE 3: Observação de reparos utilizando inserção e retirada, para os mesmos programas.

Tempo de execução para o programa correto : 0.81 seg

O programa com erros de sintaxe, recebeu o seguinte tratamento:

```

BEGIN

INTEGER VAR, NUM, VALOR;

PROCEDURE SOMA (X + Y);
                ↑
                1

    BEGIN

    INTEGE AUX;
              ↑
              2

    AUX: = 100;

    X:=  Y + AUX

    END SOMA;
          ↑
          3

NUM: 25;
  ↑
  4

VAR: = NUM;

IF SOMA VALOR: = NUM + VAR
   ↑      ↑
   5      6

ELSE SOMA (NUM, VAR

END PROGRAMA . $
↑
7

```

Número de erros encontrados = 7

Número de erros reais = 6

Número de erros em cascata = 1

Mensagens enviadas ao programador:

erro 1 : troque "+" por ","

erro 2 : inclua ";"

erro 3 : troque "," por ";"

erro 4 : troque ":" por "!="

erro 5 : inclua "AND"

erro 6 : inclua "THEN ID"

erro 7 : inclua ")"

Os erros 1, 3, 4 e 7 foram devidamente reparados.

Para o erro 2, as mesmas considerações do erro 3 no tratamento anterior.

Para o erro 5 e 6, as mesmas considerações dos erros 7 e 8 no tratamento anterior.

Tempo de execução : 0.93 seg

Tempo de execução de cada

reparação \approx 0.009 seg

CAPÍTULO 5

A IMPLEMENTAÇÃO FINAL

O analisador sintático com reparador de erros para o SAE, foi implementado na mesma linguagem Algol B/6700. Neste capítulo, falaremos dos recursos utilizados na implementação final, das opções do analisador léxico e procedimento semântico e das mensagens de erro.

O analisador léxico trabalhará deterministicamente, explorando os caracteres da cadeia de entrada e localizando os símbolos desta cadeia de acordo com as especificações léxicas da linguagem. Além disso, tratará erros encontrados nesta fase de análise e emitirá mensagens de advertência relativas a estes erros.

Um procedimento semântico complementar à análise da sentença, verificando uma parte da semântica estática (conceituada no capítulo 1) relativa aos identificadores criados pelo programador. Caso encontre alguma incompatibilidade entre um identificador e as especificações semânticas do programa para este identificador, tentará contorná-la e emitirá mensagem de erro relativa à esta incompatibilidade.

I. ESPECIFICAÇÕES LÉXICAS DA LINGUAGEM

Os símbolos pertencentes à cadeia de entrada, criados pelo usuário, deverão obedecer às seguintes especificações:

1. Número inteiro sem sinal - com no máximo, 12 dígitos e valor menor ou igual a 549 755 813 887. Não serão permitidos espaços em branco entre os dígitos.
2. Número real com precisão simples sem sinal - com no máximo, 12 dígitos na mantissa e 2 no expoente. Valor maior ou igual a 8.758 115 402 03 @ - 47 e menor ou igual a 4. 313 591 466 74 @ 68. Não serão permitidos espaços em branco entre:
 - a) os dígitos da parte inteira;
 - b) os dígitos da parte decimal;
 - c) o último dígito da parte inteira e o ponto;
 - d) o ponto e o primeiro dígito da parte decimal;
 - e) o sinal do expoente e o primeiro dígito da parte exponencial;
 - f) os dígitos da parte exponencial.

Para maior clareza do número, serão permitidos espaços em branco entre:

- a) o último dígito da parte inteira e "@" (caso não haja parte decimal);
- b) o ponto e "@" (caso a mantissa termine com ponto);
- c) o último dígito da parte decimal e "@";
- d) "@" e o sinal do expoente.

Os números não poderão terminar com "@" ou com o sinal do expoente.

3. Identificador - com no máximo 63 caracteres, sendo o primeiro uma letra e os outros, letra ou dígito. Não serão permitidos espaços em branco entre os caracteres.
4. Cadeia de caracteres - sem código e com no máximo, 6 caracteres EBCDIC.

II. ESPECIFICAÇÕES DO ANALISADOR LÉXICO

1. Receberá como entrada, um programa escrito na linguagem especificada pela gramática do SAE, perfurado em cartões que poderão ser utilizados da coluna 1 até a coluna 72, inclusive.
2. Devolverá como saída um conjunto de códigos correspondentes aos símbolos que compõem o programa, em sua ordem de entrada.
3. Os códigos corresponderão aos símbolos terminais da gramática previamente codificados pelo gerador de analisador sintático LL(1), construído na COPPE/UFRJ | 3 |.
4. Será constituído como um procedimento do analisador sintático chamado a cada entrada de um novo símbolo.
5. Será capaz de tratar erros descobertos nesta fase de análise e emitir advertências a respeito desses erros.

6. Saltará os comentários do tipo <END REMARK>, ignorando qualquer sequência de letras, dígitos e/ou brancos que apareça depois da palavra reservada END e antes das palavras reservadas END, ELSE , UNTIL ou de qualquer caracter EBCDIC diferente de letra, dígito ou branco.
7. Saltará os comentários do tipo <COMMENT REMARK> , ignorando qualquer sequência de caracteres EBCDIC, a partir da palavras reservada COMMENT até o primeiro ponto e vírgula encontrado.
8. Saltará os comentários do tipo <SCAPE REMARK>, ignorando qualquer texto a partir do sinal "%" até o final do cartão.
9. Considerará como texto de programa apenas aquele contido nas colunas 1 a 72.

III. TRATAMENTO DE ERROS DESCOBERTOS NA FASE LÉXICA DE COMPILAÇÃO E MENSAGENS DE ADVERTÊNCIA.

Os erros aqui tratados dizem respeito a símbolos criados pelo usuário que não obedecem às especificações léxicas da linguagem.

Os tipos de erro e seus respectivos tratamentos, estão assim definidos:

TIPO 1: Identificador com mais de 63 caracteres. Serão ignorados os caracteres a partir do 64º.

Mensagem: "NÃO É PERMITIDO IDENTIFICADOR COM MAIS DE 63 CARACTERES".

TIPO 2: Número inteiro com mais de 12 dígitos. Serão ignorados os dígitos a partir do 13º.

Mensagem: "NÃO É PERMITIDO NÚMERO INTEIRO COM MAIS DE 12 DÍGITOS".

TIPO 3: Número real com precisão simples com mais de 12 dígitos na mantissa. Serão ignorados os dígitos a partir do 13º.

Mensagem: "NÃO É PERMITIDO NÚMERO DECIMAL COM PRECISÃO SIMPLES, COM MAIS DE 12 DÍGITOS NA MANTISSA"

TIPO 4: Número real com precisão simples, com mais de 2 dígitos no expoente. Serão ignorados os dígitos a partir do 3º.

Mensagem: "NÃO É PERMITIDO NÚMERO REAL COM PRECISÃO SIMPLES, COM MAIS DE 2 DÍGITOS NO EXPOENTE"

TIPO 5: Número real com precisão simples terminado com "e" ou com o sinal do expoente. Será considerado um expoente igual a 1.

Mensagem: "NÃO É PERMITIDO TERMINAR UM NÚMERO REAL COM "e" OU COM O SINAL DO EXPOENTE".

TIPO 6: Número inteiro com valor superior ao permitido. Não será feito tratamento para este tipo de erro visto que

não necessitaremos do valor de um número na análise sintática.

Mensagem: "NÃO É PERMITIDO NÚMERO INTEIRO COM VALOR SUPERIOR A 549 755 813 887".

TIPO 7: Número real com precisão simples com valor menor que o permitido. Mesma consideração no erro tipo 6.

Mensagem: "NÃO É PERMITIDO NÚMERO REAL COM PRECISÃO SIMPLES COM VALOR INFERIOR A 8.758 115 402 03 @ - 47".

TIPO 8: Número real com precisão simples com valor maior que o permitido. Mesma consideração do erro tipo 6.

Mensagem: "NÃO É PERMITIDO NÚMERO REAL COM PRECISÃO SIMPLES COM VALOR SUPERIOR A 4.313 591 466 74 @ 68".

TIPO 9: Cadeia de caracteres com mais de 6 caracteres. Serão ignorados os caracteres a partir do 7º.

Mensagem: "NÃO É PERMITIDO CADEIA DE CARACTERES COM MAIS DE 6 CARACTERES".

TIPO 10: Caráter inválido para a linguagem. O carácter será ignorado.

Mensagem: "CARACTER INVÁLIDO".

O símbolo errôneo será apontado e uma mensagem será escrita para informar ao usuário, o número da advertência correspondente àquele erro. Os números que identifiquem os erros léxicos encontrados, serão guardados em um vetor e no final do programa do usuário, serão escritas todas as mensagens de advertência que se fizerem necessárias, evitando assim, multiplicidade de mensagens iguais.

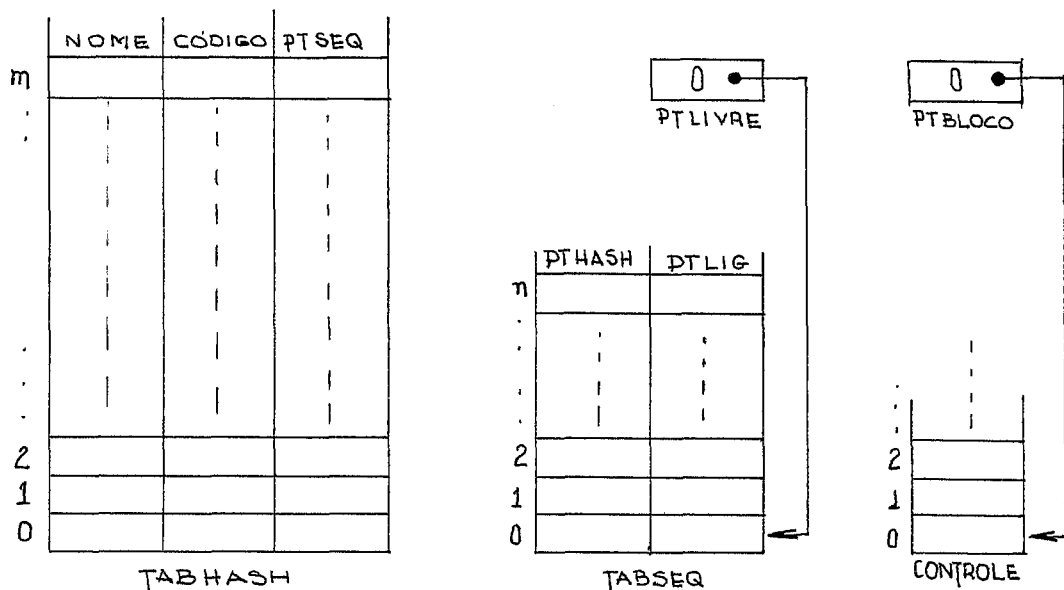
IV. ESPECIFICAÇÕES DO PROCEDIMENTO SEMÂNTICO.

1. Receberá como entrada, o nome de um identificador.
2. Devolverá como saída, o código deste identificador.
3. Será um procedimento do analisador léxico, chamado a cada entrada de um identificador.
4. Emitirá mensagem de erro, caso exista alguma incompatibilidade entre a definição e o uso de um identificador criado pelo programador.
5. Tratará um erro encontrado, de maneira a evitar que outras mensagens sejam emitidas referindo-se a este mesmo erro.
6. Armazenará um identificador de variável sendo atualmente definido em uma declaração.
7. Reconhecerá um identificador de variável contido num comando já definido anteriormente.

8. Reconhecerá a abertura de um bloco.
9. Reconhecerá o fechamento de um bloco.
10. Eliminará informações de identificadores de variáveis definidos em um bloco, no ato de fechamento deste bloco.

V. O ALGORITMO DO PROCEDIMENTO SEMÂNTICO

O procedimento semântico trabalhará junto a duas tabelas e terá a seguinte configuração inicial:



TABHASH é uma tabela que armazenará identificadores criados pelo usuário e palavras reservadas, sendo que todas as palavras reservadas pertencentes ao SAE, já se encontrarão inicialmente armazenadas. Terá 211 entradas e o acesso será determinado por

uma função hash. Cada entrada conterá 3 informações: NOME[ENDR], CÓDIGO[ENDR] e PTSEQ[ENDR], onde:

- Se I é o identificador atual de entrada, então:
ENDR = HASH (I);
- Se I é um identificador de variável contido numa declaração, então:
NOME[ENDR] = " ",
CODIGO[ENDR] = 0 e
PTSEQ[ENDR] = 0.
- Se I é um identificador de variável contido num comando, então:
NOME[ENDR] = nome de I,
CODIGO[ENDR] = código de I e
PTSEQ[ENDR] = ponteiro para o endereço das informações de I na TABSEQ;
- Se I é uma palavra reservada, então:
NOME[ENDR] = nome da palavra reservada,
CODIGO[ENDR] = código da palavra reservada e
PTSEQ[ENDR] = 0

Para o armazenamento inicial das 38 palavras reservadas pertencentes ao SAE (ver apêndice A), foram calculados os endereços através da função hash, ocorrendo uma única colisão. O fator de ocupação f , com palavras reservadas está em torno de 0.18, ou seja:

$$f = \frac{\text{nº de palavras reservadas}}{\text{nº de entradas na TABHASH}} = \frac{38}{211} \approx 0.18$$

Se o programador definir 50 identificadores no seu programa, o fator de ocupação final ficará em torno de 0.42, ou seja:

$$f = \frac{\text{nº de palavras reservadas} + \text{nº de identificadores de variáveis}}{\text{nº de entradas na TABHASH}} =$$

$$= \frac{38 + 50}{211} \approx 0.42$$

TABESQ é uma tabela auxiliar que armazenará informações a respeito dos identificadores de variáveis. As informações novas entrarão sequencialmente por ordem de chegada e as informações velhas serão acessadas através de PTSEQ[ENDR]. Terá 50 entradas e as n primeiras ($0 \leq n \leq 49$), conterão duas informações:

PTHASH [PTSEQ[ENDR]] e PTLIG [PTSEQ[ENDR]] ,

onde

PTHASH [PTSEQ[ENDR]] = ponteiro para o endereço de I na TABHASH e

PTLIG [PTSEQ[ENDR]] = ponteiro que encadeia informações de identificadores com mesmo nome, declarados em blocos distintos, ou seja:

PTLIG [PTSEQ[ENDR]] = x, se x é o endereço na TABESQ, de um identificador com mesmo nome, declarado em um bloco mais externo e

PTLIG [PTSEQ[ENDR]] = -1, se não existe outro identificador com mesmo nome.

PTLIVRE - é um ponteiro para a primeira posição livre de TABSEQ.

PTBLOCO - é um ponteiro para a primeira posição livre de CONTROLE.

CONTROLE - é um vetor de ponteiros para TABSEQ onde, cada ponteiro indica o endereço onde começam as informações sobre o conjunto de identificadores pertencentes a um bloco. Como as informações entrarão sequencialmente por ordem de chegada, aquelas relativas a identificadores de blocos mais internos se encontrarão nas últimas posições ocupadas na TABSEQ. Consequentemente, a última posição de CONTROLE ocupada, será um ponteiro para o endereço do início das informações sobre o conjunto de identificadores pertencentes ao último bloco mais interno, atualmente encontrado no programa.

Com base nessa estrutura, o procedimento semântico realizará as seguintes ações:

Ação 1: ABERTURA DE BLOCO

```
begin
  CONTROLE [PTBLOCO] ← PTLIVRE;
  PTBLOCO ← *+1
end
```

Ação 2: IDENTIFICADOR DE VARIÁVEL CONTIDO NUMA DECLARAÇÃO

```

begin
% seja I o identificador
HASH(I);
while not ALOCOU
do if NOME[ENDR] = " "
then begin
    NOME[ENDR] ← nome de I;
    CODIGO[ENDR] ← código de um identificador criado pelo
                    usuário;
    PTSEQ[ENDR] ← PTLIVRE;
    PTHASH[PTLIVRE] ← ENDR;
    PTLIG[PTLIVRE] ← -1;
    PTLIVRE ← *+1;
    ALOCOU ← true
end
else if NOME[ENDR] = nome de I
then begin
    if PTSEQ[ENDR] ≥ CONTROLE [PTBLOCO-1]
    then ERRO1
    else begin
        PTHASH[PTLIVRE] ← ENDR;
        PTLIG[PTLIVRE] ← PTSEQ[ENDR];
        PTSEQ[ENDR] ← PTLIVRE;
        PTLIVRE ← *+1
    end;
    ALOCOU ← true
end

```

```

else ENDR ← (ENDR + INCR) MOD 211
% INCR é calculado pela função HASH
end

```

Ação 3: IDENTIFICADOR DE VARIÁVEL CONTIDO NUM COMANDO.

```

begin
% seja I o identificador
HASH(I);
while not ENCONTROU
do if NOME[ENDR] = " "
then begin
    ERRO2;
    ENCONTROU ← true
end
else if NOME[ENDR] = nome de I
then begin
    if PTSEQ[ENDR] = -1 then ERRO3;
    ENCONTROU ← true
end
else ENDR ← (ENDR + INCR) MOD 211
end

```

Ação 4: FECHAMENTO DE BLOCO

```

begin
for i ← CONTROLE[PTBLOCO-1] step 1 until PTLIVRE - 1
do PTSEQ[PTHASH[i]] ← PTLIG[i];
PTLIVRE ← CONTROLE[PTBLOCO - 1];

```

```
PTBLCO ← *-1
```

```
end
```

VI. TRATAMENTO E MENSAGENS DE ERRO DO PROCEDIMENTO SEMÂNTICO

Nos pontos de erro do algoritmo, o procedimento semântico tomará as seguintes decisões:

ERRO 1: Apontará o identificador errôneo e emitirá a mensagem:
"NÃO É PERMITIDO IDENTIFICADORES COM NOMES IGUAIS EM
UM MESMO BLOCO".

ERRO 2: Apontará o identificador errôneo e emitirá a mensagem:
"IDENTIFICADOR NÃO DECLARADO".

O identificador será alocado na tabela, evitando a possibilidade de edição de outras mensagens relativas a este mesmo identificador.

ERRO 3: Apontará o identificador errôneo;
Emitirá a seguinte mensagem:
"VARIÁVEL FORA DE ESCOPO";

Acrescentará o identificador ao bloco atual, evitando também que outras possíveis mensagens relativas a este mesmo erro sejam editadas.

VII. OPÇÕES, RECURSOS E MENSAGENS DE ERRO DO ANALISADOR SINTÁTICO.

A tabela do analisador sintático foi gerada automaticamente através do gerador de analisador sintático LL(1).

A declaração FORMAT e os comandos READ e WRITE serão analisados fora do contexto da linguagem por rotinas especiais. Estas não foram implementadas e o analisador léxico, ao encontrar os símbolos "FORMAT", "READ" e "WRITE", saltará a declaração e os comandos citados.

A análise de expressões será semelhante àquela realizada para a gramática do teste 1 no capítulo 4 e poderá ser feita dentro do próprio contexto da linguagem. Isto também não está implementado e as expressões estão representadas pelos respectivos terminais abaixo:

<arithmetic expression>	- "EXPA"
<boolean expression>	- "EXPB"
<case expression>	- "EXPC"
<conditional expression>	- "EXPCD"
<designational expression>	- "EXPD"
<function expression>	- "EXPF"
<pointer expression>	- "EXPP"

As mensagens de erro emitidas pelo analisador sintático serão exatamente aquelas já vistas no capítulo 4, as quais podem ser classificadas em três tipos:

1. "INCLUA "
2. "TROQUE POR "
3. "RETIRE "

A listagem final se encontra no anexo 1.

CONCLUSÕES

1. A simplificação sugerida pelo autor do método de tratamento de erros que consiste em examinar apenas mais um símbolo, além daquele que garantidamente pertencerá à cadeia de símbolos a ser retirada da sentença, não induziu o analisador sintático a produzir diagnósticos imprecisos nos programas testados onde os erros introduzidos foram aqueles do tipo mais frequentes, de acordo com a pesquisa descrita no capítulo 2.
2. A decisão tomada durante a realização dos testes de implementação descrita no capítulo 4, que consiste em eliminar algumas tabelas e gerar as cadeias de inserção dentro do próprio algoritmo do reparador, não foi uma solução custosa, ou seja, onerou levemente o tempo de execução do algoritmo e economizou um espaço bastante significativo.
3. O algoritmo do reparador é simples, rápido, de fácil implementação e não penaliza o analisador sintático no processo de análise de um programa correto.
4. Os resultados obtidos satisfazem os objetivos do nosso trabalho.
5. Verificamos que o reparador de erros teria como um suporte mais adequado, um procedimento semântico com condições de

reconhecer em um identificador não declarado, uma possível palavra reservada mal digitada.

6. Para reparos mais sofisticados, sugerimos a utilização do algoritmo sem a simplificação descrita na conclusão 1 ou , até mesmo, de algoritmos que não só executem um reparo local mas um reparo global, o que significa utilizar informações de trechos já analisados e/ou de trechos da entrada ainda não processada, no exame dos reparos a serem feitos.

APÊNDICE A

Os componentes do SAE

1. Qualquer tipo de comentário.
2. Número inteiro sem sinal e real com precisão simples sem sinal.
3. Cadeia EBCDIC sem código e com no máximo 6 caracteres.
4. Palavras reservadas:

DO	FILE	WITH	WRITE
GO	FILL	ALPHA	FORMAT
IF	KIND	ARRAY	READER
OF	PACK	BEGIN	BOOLEAN
TO	READ	FALSE	INTEGER
END	REAL	LABEL	PRINTER
FOR	STEP	TITLE	DISKPACK
CASE	THEN	UNTIL	PROCEDURE
DISK	THRU	VALUE	
ELSE	TRUE	WHILE	

5. Demais símbolos:

:=	(-
.)	%
;	,	"
:	=	@
[*	\$
]	+	

6. Declarações:

Alpha	Integer
Array	Label
Boolean	Procedure
File	Real
Format	Value Array

7. Comandos:

Atribuição	Case	If
Bloco	Do	Read
Composto	Fill	Thru
Com rótulo	For	While
Vazio	Go To	Write

8. Todas as expressões representadas por seus respectivos "tokens" (ver Capítulo 5 - item VII)

OBS.: Nem todas as declarações e comandos foram considerados em toda sua extensão (ver gramática, apêndice B).

APÊNDICE B

Gramática que especifica a sintaxe do SAE, na forma fortemente LL(1) não anulável.

<PROGUNIT> ::= BEGIN <DECL*STAT> END . \$

<DECL*STAT> ::= <DECLLIST> <STATLIST> |
 <STATELIST>

<DECLLIST> ::= <DECL> ; <SEG*DECL>

<DECL> ::= <TYPE><TIPO*DECL> |
 <TIPO*GROUP> |
 FILE <FILELIST> |
 FORMAT |
 LABEL <IDENLIST>

<SEG*DECL> ::= <DECL> ; <SEG*DECL> |

ε

<TYPE> ::= ALPHA |
 BOOLEAN |
 INTEGER |
 REAL

<TIPO*DECL> ::= <IDENLIST> |
 <TIPO*GROUP>

```

<TIPO*GROUP> ::= ARRAY <ARRAYLIST> |
                PROCEDURE ID <PROCPART> |
                VALUE ARRAY <VALUELIST>

<IDENLIST> ::= ID <SEG*IDENT>

<SEG*IDENT> ::= , ID <SEG*IDENT> |
                ε

<ARRAYLIST> ::= <IDENLIST> [<BPAIRLIS>] <SEG*ARRAYLIST>

<SEG*ARRAYLIST> ::= , <IDENLIST> [<BPAIRLIS>] <SEG*ARRAYLIST> |
                ε

<BPAIRLIS> ::= EXPA : EXPA <SEG*BPAIR>

<SEG*BPAIR> ::= , EXPA : EXPA <SEG*BPAIR> |
                ε

<PROCPART> ::= (<IDENLIST>); <VALUEPART> |
                ; <UNLB*STAT>

<VALUEPART> ::= VALUE <IDENLIST> ; <SPEC> ; <SEG*SPEC> |
                <SPEC> ; <SEG*SPEC>

<SPEC> ::= <TYPE><IDENLIST> <SPECLIST> |
                <ARAYSPECLIST>

<SEG*SPEC> ::= ID <PARTBODY> |
                <TYPE> <IDENLIST> <SPECLIST> ; <SEG*SPEC> |
                <GROUP*STAT>

```

<SPECLIST> ::= [<LOBOLIST>] <SEG*ARAYSPEC> |

ε

<ARAYSPECLIST> ::= <IDENLIST> [<LOBOLIST>] <SEG*ARAYSPEC>

<LOBOLIST> ::= <SPECLOBO> <SEG*SPECLOBO>

<SEG*ARAYSPEC> ::= , <IDENLIST> [<LOBOLIST>] <SEG*ARAYSPEC> |

ε

<SPECLOBO> ::= <SIGN> INT |

INT |

*

<SEG*SPECLOBO> ::= , <SPECLOBO> <SEG*SPECLOBO> |

ε

<SIGN> ::= + |

-

<PARTBODY> ::= [<PARTSPEC> |

, <ARAYSPECLIST> ; <SEG*SPEC> |

<PART*ASIG>

<PARTSPEC> ::= <LOBOLIST>] <SEG*ARAYSPEC> ; <SEG*SPEC> |

<ARITLIST>] <PTWDPART> ::= <ARIT*BOOL>

<ARITLIST> ::= EXPA <SEG*ARITEXP>

<SEG*ARITEXP> ::= , EXPA <SEG*ARITEXP> |

ε

<PTWDPART> ::= . [EXPA : EXPA] |

ε

<ARIT*BOOL> ::= EXPA |
EXPB

<VALUELIST> ::= ID (<CONSTLIST>) <SEG*VALUELIST>

<SEG*VALUELIST> ::= , ID (<CONSTLIST>) <SEG*VALUELIST> |
ε

<CONSTLIST> ::= <CONST> <SEG*CONST>

<SEG*CONST> ::= , <CONST> <SEG*CONST> |
ε

<CONST> ::= INT <SEG*CONSTLIST> |
REALS |
<SIGN> <UNSIGNUM> |
<VALORLOG>

<SEG*CONSTLIST> ::= (<CONSTLIST>) |
ε

<UNSIGNUM> ::= INT |
REALS

<VALORLOG> ::= TRUE |
FALSE

<FILELIST> ::= ID <FILEPART> <SEG*FILELIST>

<SEG*FILELIST> ::= , ID <FILEPART> <SEG*FILELIST> |
ε

<FILEPART> ::= (<ATRIB> <SEG*FILEPART>) |

ε

<SEG*FILEPART> ::= , <ATRIB> <SEG*FILEPART> |

ε

<ATRIB > ::= KIND = <ARITVALUE> |

TITLE = CADEIA

<ARITVALUE> ::= EXPA |

DISK |

DISKPACK |

PACK |

PRINTER |

READER

<STATLIST> ::= <STAT> <SEG*STAT>

<SEG*STAT> ::= ; <STAT> <SEG*STAT> |

ε

<STAT> ::= ID <PARTSTAT> |

<GROUP*STAT>

<PART*STAT> ::= : <UNLB*STAT> |

<ASIG*PROC>

<UNLB*STAT> ::= ID <ASIG*PROC> |

<GROUP*STAT>

<ASIG*PROC> ::= <PART*ASIG> |

[<ARITLIST>] <PTWDPART> ::= <ARIT*BOOL>

$\langle \text{PART*ASIG} \rangle ::= \langle \text{AB*DESG} \rangle \mid$
 $\quad \cdot [\text{EXPA} : \text{EXPA}] := \langle \text{ARIT*BOOL} \rangle \mid$
 $\quad (\langle \text{PARMLIST} \rangle) \mid$
 $\quad \epsilon$

$\langle \text{AB*DESG} \rangle ::= \langle \text{ARIT*BOOL} \rangle \mid$
 $\quad \langle \text{ARRAYDESG} \rangle$
 $\langle \text{ARRAYDESG} \rangle ::= \text{ID} \langle \text{SUBARRAY} \rangle \mid \mid$
 $\langle \text{SUBARRAY} \rangle ::= [\langle \text{SUBSCRIPT} \rangle \langle \text{ARRAYPART} \rangle] \mid$
 $\quad \epsilon$

$\langle \text{SUBSCRIPT} \rangle ::= \langle \text{ARITLIST} \rangle \cdot \mid$
 $\quad \epsilon$

$\langle \text{ARRAYPART} \rangle ::= * \langle \text{SEG*ASTCO} \rangle$

$\langle \text{SEG*ASTCO} \rangle ::= , * \langle \text{SEG*ASTCO} \rangle \mid$
 $\quad \epsilon$

$\langle \text{PARMLIST} \rangle ::= \langle \text{ATUALPARG} \rangle \langle \text{SEG*PARG} \rangle$

$\langle \text{SEG*PARG} \rangle ::= , \langle \text{ATUALPARG} \rangle \langle \text{SEG*PARG} \rangle \mid$
 $\quad \epsilon$

$\langle \text{ATUALPARG} \rangle ::= \langle \text{EXPRESSION} \rangle \mid$
 $\quad \langle \text{ARRAYDESG} \rangle$

$\langle \text{EXPRESSION} \rangle ::= \text{EXPA} \mid$
 $\quad \text{EXPB} \mid$
 $\quad \text{EXPC} \mid$
 $\quad \text{EXPCD} \mid$
 $\quad \text{EXPD} \mid$

EXPF |

EXPP

```

<GROUP*STAT>::= BEGIN <DECL*STAT> END |
                CASE EXPA OF <CASEBODY>|
                DO <STAT> UNTIL EXPB|
                FILL <ARRAYROW> WITH <FILLLIST>|
                FOR <VARIAB>:= <FORLIST> DO <STAT>|
                GO <GOPART>|
                IF EXPB THEN <STAT> <CLELSE>|
                READ|
                THRU EXPA DO <STAT>|
                WHILE EXPB DO <STAT>|
                WRITE|
                ε

```

```

<CASEBODY>::= BEGIN <CASELIST> END

```

```

<CASELIST>::= <STATLIST>|
              INT : <STAT> <CASEPART>

```

```

<CASEPART>::= ; <SEG*CASELIST>|

```

ε

```

<SEG*CASELIST>::= <STAT> <CASEPART> |
                  INT : <STAT> <CASEPART>

```

```

<ARRAYROW>::= ID <SEG*ROW>

```

```

<SEG*ROW>::= [ <ROWDESG> ] |

```

ε

$\langle \text{ROWDESG} \rangle ::= * \mid$
 $\text{EXPA} , \langle \text{ROWDESG} \rangle$

$\langle \text{FILLLIST} \rangle ::= \langle \text{INITVALUE} \rangle \langle \text{SEG*VALUE} \rangle$

$\langle \text{SEG*VALUE} \rangle ::= , \langle \text{INITVALUE} \rangle \langle \text{SEG*VALUE} \rangle \mid$
 ϵ

$\langle \text{INITVALUE} \rangle ::= \text{INT} \langle \text{SEG*FILLLIST} \rangle \mid$
 $\text{REALS} \mid$
 $\langle \text{SIGN} \rangle \langle \text{UNSIGNUM} \rangle \mid$
 CADEIA

$\langle \text{SEG*FILLLIST} \rangle ::= (\langle \text{FILLLIST} \rangle) \mid$
 ϵ

$\langle \text{VARIAV} \rangle ::= \text{ID} \langle \text{SIMP*IND} \rangle$

$\langle \text{SIMP*IND} \rangle ::= [\langle \text{ARITLIST} \rangle] \mid$
 ϵ

$\langle \text{FORLIST} \rangle ::= \langle \text{FORELEM} \rangle \langle \text{SEG*FORLIST} \rangle$

$\langle \text{SEG*FORLIST} \rangle ::= , \langle \text{FORELEM} \rangle \langle \text{SEG*FORLIST} \rangle \mid$
 ϵ

$\langle \text{FORELEM} \rangle ::= \text{EXPA} \langle \text{ITERATION} \rangle$

$\langle \text{ITERATION} \rangle ::= \text{STEP} \text{EXPA} \langle \text{STEPPART} \rangle \mid$
 $\text{WHILE} \text{EXPB} \mid$
 ϵ

<STEPPART> ::= UNTIL EXPA |
 WHILE EXPB

<GOPART> ::= TO EXPD |
 EXPD

<CLELSE> ::= ELSE <STAT> |
 ϵ

APÊNDICE C

Custos de inclusão e retirada dos símbolos terminais da gramática que especifica o SAE.

TERMINAL	CUSTO DE INCLUSÃO	CUSTO DE RETIRADA
BEGIN	10	20
END	8	15
.	10	20
\$	50	200
;	5	14
FILE	12	25
FORMAT	50	50
LABEL	11	20
ID	8	15
ALPHA	10	20
BOOLEAN	9	20
INTEGER	8	20
REAL	8	20
ARRAY	11	20
PROCEDURE	12	25
VALUE	11	20
:	8	8
CASE	10	25
EXPA	6	50
OF	8	15

TERMINAL	CUSTO DE INCLUSÃO	CUSTO DE RETIRADA
DO	8	15
UNTIL	8	12
EXPB	7	50
FILL	10	25
WITH	8	15
FOR	10	25
:=	9	9
GO	9	5
IF	15	25
THEN	6	10
READ	50	50
THRU	10	20
WHILE	10	20
WRITE	50	50
[10	20
]	7	10
(10	20
)	4	10
,	4	8
TO	9	5
EXPD	8	50
ELSE	6	10
KIND	8	15
=	5	5
TITLE	8	15
CADEIA	7	15

TERMINAL	CUSTO DE INCLUSÃO	CUSTO DE RETIRADA
INT	6	15
REALS	7	15
DISK	8	15
DISKPACK	8	15
PACK	8	15
PRINTER	8	15
READER	8	15
*	5	5
+	5	5
-	5	5
STEP	8	12
TRUE	7	15
FALSE	7	15
EXPC	44	50
EXPCD	46	50
EXPF	8	50
EXPP	8	50

REFERÊNCIAS BIBLIOGRÁFICAS

- |1| AHO - ULLMAN (1972): "The Theory of Parsing, Translation and Compiling, Vol. I, Parsing". Prentice-Hall, Englewood Cliffs, N.S.

- |2| AHO - ULLMAN (1977): "Principles of Compiler Design", Addison-Wesley, Reading, Mass.

- |3| ARGOLLO-SIMONE (1981): "Projeto NHÃO NHÃO - Gerador de Analisador Sintático LL(1)", COPPE/UFRJ, Manuscrito Interno.

- |4| BAUER - EICKEL(1974): "Compiler Construction: An Advanced Course", Springer Verlag, New York, N.Y.

- |5| FISCHER - MAUNEY (1979): "On the role of error productions in syntatic errot corretion", University of Wisconsin - Madison, Computer Sciences Technical Report #364.

- |6| FISCHER - TAI - MILTON (1979): "Immediate error detection in strong LL(1) parsers", University of Wisconsin - Madison, Information Processing Letters, Vol.8, number 5.

- |7| FISCHER - MILTON - QUIRING (1979): "Efficient LL(1) error correcting and recovery using only insertions", University of Wisconsin - Madison, Acta Informatica, Vol. 13, Fasc. 2.

- |8| FISCHER - MAUNEY - MILTON (1979): "A locally least - cost LL(1) error corrector", University of Wisconsin - Madison, Computer Science Technical Report #371.

- |9| GHEZZI (1975): "LL(1) grammars supporting an efficient error handling", Information Processing Letters, Vol.3, 174-176.

- |10| GRIES (1971): "Compiler Construction for Digital Computers", John Wiley & Sons, inc., N.Y.

- |11| LITECKY - DAVIS (1976): "A study of errors, error - proneness, and error diagnosis in Cobol, CACM 19, 33-37.

- |12| RIPLEY - DRUSEIKIS (1978): "A statistical analysis of syntax errors", University of Arizona, Computer Languages, Vol. 3, 227-240.

- |13| RÖHRICH (1978): "Automatic construction of error correcting parsers", Fakultät für Informatik, Universität Karlsruhe.

- |14| WIRTH (1968): "PL360 - a programming language for the 360 computers", Stanford University, J. ACM 15:1, 37-74.

- |15| YOUNGS (1972): "Error - proneness in programming", University of North Carolina, Technical Report.

ANEXO 1

LISTAGEM FINAL


```

110,111,40,41,41,42,70,19,17,19,112,33,76,35,94,36,95,16,76,5,113,-4,5,
114,113,-4,5,114,115,116,39,9,37,97,33,98,43,44,117,45,44,46,39,99,100,
27,118,3,35,19,17,19,36,27,104,37,119,39,19,120,3,35,19,17,19,36,19,23,
57,47,17,70,121,35,122,36,47,123,48,124,125,46,39,107,108,35,102,36,19,
126,39,110,111,39,19,17,19,112,72,76,127,128,9,129,-3,72,76,127,-4,5,
114,78,47,130,48,124,125,131,39,115,116,19,49,50,51,52,53,104,132,133,
134,39,19,120,5,135,54,19,39,122,37,29,38,55,56,47,48,57,19,136,33,23,
35,137,36,138,76,35,137,36,138,35,139,39,128,5,114,101,37,97,38,58,59,
9,140,141,132,39,133,134,70,121,47,17,70,121,22,19,33,23,142,143,39,76,
35,137,36,138,137,36,138,5,114,102,36,103,27,104,35,144,145,36,19,23,60,
61,41,62,63,124,47,47,54,39,142,143,102,3,54,146,39,54,(146),

```

```

TAMCAO(O,
4,0,3,0,2,0,0,0,1,1,3,1,1,0,0,6,1,4,0,0,0,0,0,2,1,1,1,1,1,0,3,0,3,1,0,3,
0,0,1,0,1,0,0,1,0,0,1,0,0,2,0,1,0,1,1,1,0,0,1,0,1,1,0,0,4,0,0,1,1,1,0,0,
2,1,0,3,0,1,1,0,0,1,0),

```

```

PTCAO(O,
1,0,5,0,8,0,0,0,10,11,12,15,16,0,0,17,23,24,0,0,0,0,0,28,30,31,32,33,34,
0,35,0,38,41,0,42,0,0,45,0,46,0,0,47,0,0,48,0,0,49,0,51,0,52,53,54,0,0,
35,0,56,57,0,0,58,0,0,62,63,64,0,0,65,67,0,68,0,71,72,0,0,73,0),

```

```

PDCOAO(O,
1,2,3,4,12,9,5,12,9,12,9,15,9,5,9,9,9,35,19,17,19,36,5,9,37,47,38,1,2,9,
47,9,19,41,19,17,19,12,9,5,47,43,44,19,19,19,47,19,12,9,47,19,19,19,54,
55,47,9,35,54,36,58,9,19,22,19,54,54,36,5,19,54,54),

```

```

CUSTOINS(O,
10,8,10,50,5,12,50,11,8,10,9,8,3,11,12,11,8,10,6,8,8,6,7,10,8,10,9,9,
15,6,50,10,10,50,10,7,10,4,4,9,0,6,6,5,3,7,6,7,8,8,8,8,8,5,5,5,2,7,7,44,
46,8,8,78,0,21,0,16,0,0,0,8,3,25,8,8,0,0,45,5,28,0,0,0,0,0,18,8,6,8,6,8,
0,20,0,21,6,0,19,0,0,6,0,6,0,0,8,0,0,6,0,0,16,0,6,0,6,6,8,0,0,5,0,5,6,0,
0,30,0,0,7,8,6,0,0,14,5,0,17,0,6,5,0,0,5,0),

```

```

CUSTOJEL(O,
20,15,20,200,14,25,50,20,15,20,20,20,20,20,25,20,8,25,50,15,15,12,50,25,
15,25,9,5,25,10,50,20,20,50,20,10,20,10,8,5,50,10,15,5,15,15,15,15,15,
15,15,15,15,5,5,5,12,15,15,50,50,50,50),

```

```

TAMTRAO(O,
3,3,1,1,1,4,6,5,2,5,7,7,4,5,9,5,1,4,4,2,2,5,4,4,4,3,2,2,2,4,4,4,5,5,1,1,
1,1,1,2,4,4,4,1,5,4,1,3,4,8,4,7,6,1,1,1,4,4,5,4,5,4,4),

```

```

*****
PROCEDURE INICIALIZA;

```

```

BEGIN
  POINTER PTAUX;
  ALPHA ARRAY AUX(1:381);
  VALUE ARRAY
  TAMEXP(0,4(4),1(5),2(4)),
  TAMPR(0,5(2),2(3),14(4),10(5),2(6),3(7),1(8),1(9)),
  COJPR(0,21,28,29,20,40,2,26,18,49,42,6,24,43,51,31,13,57,30,32,58,25,
  10,14,1,59,8,45,22,16,33,34,7,53,11,12,52,50,15);

```

```

  FILL AJXI*1 WITH
  "EXPAEXP3EXPDEXPCEXPCEXPFXEXP";
  PTAUX:=POINTER(AUX);
  FOR I:=1 STEP 1 UNTIL 7
  DO BEGIN
    VETEXPII:=STRING(PTAUX,TAMEXP(I));
    PTAUX:=++TAMEXPII;
  END;

```

```

  FILL AUX*1 WITH
  "DDDDIFFOFIDENDFORCASEDJSKELSEFILEFILLKINDPACKREADREALSTEPHEN";
  "THRUTRUEINTHALPHAARRAYBEGINFALSELABELTITLEUNTILVALUE#HILEWRI";
  "IFORMATREADERBOOLEANINTEGERPRINTERDISKPACKPROCEDURE";
  FOR I:=0 STEP 1 UNTIL 210
  DO NOMEID(I):=" ";
  J:=0;
  PTAUX:=POINTER(AUX);
  FOR I:=172,96,170,(122,95,184,58,2,39,79,103,117,107,65,16,37,109,41,
  112,131,5,60,127,152,175,147,93,137,64,71,102,87,33,101,168,

```

```

DO BEGIN
  J:=+1;
  NOMEI:=STRNGCPTAUX, TAMPREJJ);
  CTOI:=COTI;
  PTAUX:=+TAMPREJJ;
END;

```

```

FILE AVAR*J, ATLE FORMLABELICALP#ARROULEFANINIEGERRALARRAYPRCCFOUREV";
VALUE:=CASEPHELESEKUNDELLLEABCOJY:IDISKDISKPACKPACKPRINTEPRRELADEP*+SFF;
FOR I:=1 STEP 1 UNTIL 53 ...
PDAUX:=PDAUX+STRNGCPTAUX);
DO BEGIN
  PDAUX:=+TAMPREJJ;
END;

```

```

FOR I:=1 STEP 1 UNTIL 72
DO MARCAADVERTI:=MARCASENLIJ:=MARCAERROUJ:=";
COUNMA:=1;
TOPU:=1;
PILHALIDPUI:=64;
END;

```

```

*****
PROCEDURE LECADETIACC,CTI);
INTEGER CC,CTI;

```

```

BEGIN
  INTEGER COD;
  CTI:=1;
  FJR KA:=1 STEP 1 UNTIL CC
  DO BEGIN
    THEN BEGIN
      STDEL:=+1;
      COD:=CADELLIPTDELI
    END
  ELSE
    BEGIN
      CTI:=+1;
      COD:=CAVINS(CPFINSI
    END;
  END;
  CADEIAT:=CADEIAT+";
END;

```

```

CTI:=+1;
CADEIAT:=CADEIAT+TRADUZTCOD);
THEN BEGIN
  CTI:=+1;
  CADEIAT:=CADEIAT+";
END

```

```

END;
CADEIAT:=CADEIAT+";
END;

```

```

*****
PROCEDURE ACUSAERRO;

```

```

BEGIN
  INTEGER COMPC,COMPD,COMPT;
  STRNG CADEIAD,CADEIAT;
  PDEL:=PINS:=0;
  FOR I:=1 STEP 1 UNTIL ERLIN
  DO BEGIN
    IF VEI DEL(I)>0
    THEN BEGIN
      J:=1;
      COMPC:=VEI DEL(I);
      LECADETIACC(CMPC,COMPD);
      CADEIAD:=CADEIAT
    END;
    IF VEI INS(I)>0
    THEN BEGIN
      J:=1;
      COMPC:=VEI INS(I);
      LECADETIACC(CMPC,COMPI);
      CADEIAT:=CADEIAT
    END;
  END;
  IF VEI DEL(I)=0
  THEN WRITE(SAIDA,<X7,>">>ERRO",I2,"INCLUA ",A*," <<<">,NUMER,

```

```

ELSE COMPI,CADEIAI)
  ELSE IF VETINS[I]=0
    THEN WRITE(SAIDA,<X7,">>ERRO",I2,"RETIRE ",A*," <<<">,
      NUMER,COMP,CADEIAI)
    ELSE WRITE(SAIDA,<X7,">>ERRO",I2,"TROQUE ",A*," POR ",A*,
      " <<<">,NUMER,COMP,CADEIAI,COMPI,CADEIAI)

```

```

END;
ERR:=FALSE;
ERLIN:=PTDEL:=PTINS:=0;
FOR I:=1 STEP 1 UNTIL 10 DO VEIDEL[I]:=VETINS[I]:=0
END;

```

```

*****
PROCEDURE IMPRIME;

```

```

BEGIN
  WRITE(SAIDA,<X10,A73>,LINHA);
  IF ADV
    THEN BEGIN
      WRITE(SAIDA,ADVERTENCIA,MARCAADV[*],A,FOR I:=1 STEP 1 UNTIL A
        DO TIP[C[I]]);
      ADV:=FALSE;
      A:=0;
      END;
  IF SEM
    THEN BEGIN
      WRITE(SAIDA,<X10,72A1>,MARCASEM[*]);
      IF ERSEM[1]=1
        THEN WRITE(SAIDA,<X7,">>ERRO",I2,"NÃO É PERMITIDO IDENTIFICA",
          "DOES COM NOMES IGUAIS NO MESMO BLOCO<<<">,NUMER);
      IF ERSEM[2]=1
        THEN WRITE(SAIDA,<X7,">>ERRO",I2,"IDENTIFICADOR NÃO DECLARA",
          "DO<<<">,NUMER);
      IF ERSEM[3]=1
        THEN WRITE(SAIDA,<X7,">>ERRO",I2,"VARIÁVEL FORA DE ESCOPO",
          " <<<">,NUMER);
      FOR I:=1 STEP 1 UNTIL 3 DO ERSEM[I]:=0;
      SEM:=FALSE;
      END;
  IF ERRO
    THEN BEGIN
      WRITE(SAIDA,<X10,72A1>,MARCAERRO[*]);
      IF PARE
        THEN ACUSAERRO
        ELSE FALTA:=TRUE;
      END;
  FOR I:=1 STEP 1 UNTIL 72
    DO MARCAADV[I]:=MARCASEM[I]:=MARCAERRO[I]:=0;
  END;

```

```

*****
PROCEDURE PROKLINHA;

```

```

BEGIN
  IMPRIME;
  READ(ENTRADA,<A73>,LINHA);
  COLUNA:=1;
  END;

```

```

*****
PROCEDURE ADVERT(X,MAXCAR);
  INTEGER X,MAXCAR;

```

```

BEGIN
  NUMADV:=X+1;
  ERLEXC[1]:=1;
  ADV:=TRUE;
  A:=X+1;
  TIP[C[1]]:=X;
  TAMANHO:=MAXCAR;
  MARCAADV[PTCOL]:=X;
  END;

```

```

*****
PROCEDURE HASH;

```

```

BEGIN
  INTEGER N,PAL;
  PAL:=1;
  COND:=FALSE;
  WHILE NOT COND
    DO BEGIN
      N:=X+6;
      IF TAMANHO>N
        THEN PAL:=X+1;
      ELSE COND:=TRUE;
    END;

```

```

END;
APONT:=PJINTER(VETCAR);
REPLACE APONT:APONT BY LINHA(PTCOD) FOR TAMANHO;
IF TAMANHO<N
THEN REPLACE APONT BY " " FOR N-TAMANHO;
IF PAL>1
THEN FOR I:=2 STEP 1 UNTIL PAL
DO VETCAR(I):=REAL(NOT(BOOLEAN(VETCAR(I)) EQV
BOOLEAN(VETCAR(I))));
VETCAR(I).[31:16]:=REAL(NOT(BOOLEAN(VETCAR(I).[47:16]) EQV
BOOLEAN(VETCAR(I).[31:16])));
VETCAR(I).[15:16]:=REAL(NOT(BOOLEAN(VETCAR(I).[31:16]) EQV
BOOLEAN(VETCAR(I).[15:16])));
ENR:=VETCAR(I).[15:16] MOD 211;
INCR:=VETCAR(I).[15:5]+1
END;

```

```

*****
PROCEDURE PROCSEM(S);
INTEGER S;

```

```

CASE S OF
BEGIN
0:PROCSEM(-PILHA(TOP));
TOP:=+-1;
1:CONTROLE(PTBLOC):=PTSEM;
PTBLOC:=++1;
2:PTBLOC:=+-1;
PTSEM:=+-1;
FOR I:=CONTROLE(PTBLOC) STEP 1 UNTIL PTSEM
DO PTSEQ(PHASH(I)):=PTLIG(I);
PTSEM:=CONTROLE(PTBLOC);
3:NOVO:=TRUE;
4:NOVO:=FALSE;
5:HASH;
COND:=FALSE;
WHILE NOT COND
DO IF NOMEID(ENR)=NOME
THEN BEGIN
IF CODID(ENR)=9
THEN IF NOT PARAM
THEN IF NOVO
THEN IF PTSEQ(ENR)>CONTROLE(PTBLOC-1)
THEN BEGIN
NUMR:=++1;
SEM:=TRUE;
MARCASEM(PTCOL):="*";
ERSEM(I)=1
END
ELSE PROCSEM(6)
ELSE IF PTSEQ(ENR)=-1
THEN BEGIN
NUMR:=++1;
SEM:=TRUE;
MARCASEM(PTCOL):="*";
ERSEM(I)=1;
PROCSEM(6)
END;
COND:=TRUE
END
ELSE IF NOMEID(ENR)=" "
THEN BEGIN
IF NOT PARAM
THEN BEGIN
PROCSEM(7);
IF NOT NOVO
THEN BEGIN
NUMR:=++1;
SEM:=TRUE;
MARCASEM(PTCOL):="*";
ERSEM(I)=1
END
END;
COND:=TRUE
END
ELSE ENR:=(ENR+INCR) MOD 211;
IF NOMEID(ENR)=9
THEN SIMBOL:=9
ELSE SIMBOL:=CODID(ENR);
6:PHASH(PTSEM):=ENR;
PTLIG(PTSEM):=PTSEQ(ENR);
PTSEQ(ENR):=PTSEM;
PTSEM:=++1;
7:NOMEID(ENR):=NOME;
CODID(ENR):=9;
PTSEQ(ENR):=PTSEM;
PHASH(PTSEM):=ENR;

```

```

PTLIGIPTSSEM):=-I;
PTSEM:==+1;
8:NOVO:=FALSE;
PARAM:=TRUE;
9:NOVO:=TRUE;
PARAM:=FALSE
END;

```

```

*****
PROCEDURE PROXSIMB;

```

```

BEGIN

```

```

% DECLARACOES
INTEGER COD,INT,MANT,EXP,PTEXP;

```

```

REAL DEC;
TRUTHSET DIGITOS("0123456789");

```

```

TRUTHSET LETRAS(ALPHA AND NOT DIGITOS);

```

```

% PROCEDIMENTO

```

```

% TRATAMENTO DE CARACTERES BRANCOS

```

```

COND:=FALSE;
WHILE NOT COND
DO BEGIN
  WHILE LINHA(COLUNA)=" " AND COLUNA<73 DO COLUNA:==+1;
  IF COLUNA=73
  THEN PROXLINHA
  ELSE COND:=TRUE
END;
PTCOL:=COLUNA;

```

```

% TRATAMENTO DE IDENTIFICADORES E PALAVRAS RESERVADAS
IF LINHA(COLUNA) IN LETRAS
THEN BEGIN

```

```

  WHILE LINHA(COLUNA) IN ALPHA AND COLUNA<73 DO COLUNA:==+1;
  TAMANHO:=COLUNA-PTCOL;
  IF TAMANHO>63 THEN ADVERT(1,63);
  NOME:=STRING(LINHA(PTCOL),TAMANHO);

```

```

% TRATAMENTO DE COMENTARIOS TIPO <COMMENT REMARK>

```

```

IF NOME="COMMENT"
THEN BEGIN
  COND:=FALSE;
  WHILE NOT COND
  DO BEGIN
    WHILE LINHA(COLUNA)!=";" AND COLUNA<73 DO COLUNA:==+1;
    IF COLUNA=73
    THEN PROXLINHA
    ELSE COND:=TRUE
  END;
  COLUNA:==+1;
  PROXSIMB
END

```

```

% ATRIBUICAO DE CODIGOS

```

```

ELSE BEGIN
  I:=1;
  COND:=FALSE;
  WHILE NOT COND AND I<8
  DO IF NOME=VETEXP(I)
  THEN COND:=TRUE
  ELSE I:==+1;
  IF COND
  THEN SIMBOLO:=CODEXP(I)
  ELSE PROCSEM(5)
END;

```

```

% TRATAMENTOS ESPECIAIS

```

```

CASE SIMBOLO OF
  BEGIN

```

```

    % TRATAMENTO DE COMENTARIOS TIPO <END REMARK>

```

```

    2:COND:=FALSE;
    WHILE NOT COND
    DO BEGIN
      WHILE ((LINHA(COLUNA)=" ") OR
        (LINHA(COLUNA) IN DIGITOS)) AND COLUNA<73
      DO COLUNA:==+1;
      IF COLUNA=73
      THEN PROXLINHA
      ELSE IF LINHA(COLUNA) IN ALPHA
        THEN IF LINHA(COLUNA)="E" OR LINHA(COLUNA)="U"
          THEN BEGIN

```

```

PTCOL:=COLUNA;
WHILE LINHA(PTCOL) IN ALPHA AND
PTCOL<73 DO PTCOL:=**+1;
TAMANHO:=PTCOL-COLUNA;
NOME:=STRING(LINHA(COLUNA),TAMANHO);
IF NOME="END" OR NOME="ELSE" OR
NOME="UNTIL"
THEN COND:=TRUE
ELSE COLUNA:=PTCOL
END
ELSE WHILE LINHA(COLUNA) IN ALPHA AND
COLUNA<73
DO COLUNA:=**+1
ELSE COND:=TRUE
END;

```

```

% ENTRADA PARA ROTINA DO "FORMAT"

```

```

7:COND:=FALSE;
WHILE NOT COND
DO BEGIN
WHILE LINHA(COLUNA)!=";" AND COLUNA<73
DO COLUNA:=**+1;
IF COLUNA=73
THEN PROXLINHA
ELSE COND:=TRUE
END;

```

```

% ENTRADA PARA ROTINA DO "READ"

```

```

31:COND:=FALSE;
WHILE NOT COND
DO BEGIN
WHILE LINHA(COLUNA)!=";" AND COLUNA<73
DO COLUNA:=**+1;
IF COLUNA=73
THEN PROXLINHA
ELSE COND:=TRUE
END;

```

```

% ENTRADA PARA ROTINA DO "WRITE"

```

```

34:COND:=FALSE;
WHILE NOT COND
DO BEGIN
WHILE LINHA(COLUNA)!=";" AND COLUNA<73
DO COLUNA:=**+1;
IF COLUNA=73
THEN PROXLINHA
ELSE COND:=TRUE
END
ELSE:END

```

```

END

```

```

% TRATAMENTO DE NUMEROS
ELSE IF (LINHA(COLUNA) IN DIGITOS) OR (LINHA(COLUNA)="3") OR
(LINHA(COLUNA)="." AND (LINHA(COLUNA+1) IN DIGITOS))
THEN BEGIN

```

```

NOME:=" ";
TAMANHO:=0;
COD:=1;
SIMBOLO:=47;
WHILE LINHA(COLUNA) IN DIGITOS AND COLUNA<73
DO COLUNA:=**+1;
IF LINHA(COLUNA)="."
THEN BEGIN
COD:=2;
SIMBOLO:=48;
COLUNA:=**+1;
WHILE LINHA(COLUNA) IN DIGITOS AND COLUNA<73
DO COLUNA:=**+1
END;

```

```

TAMANHO:=COLUNA-PTCOL;
IF COD=1
THEN BEGIN
IF TAMANHO>12 THEN ADVERT(2,12);
IVI:=TAMANHO
END

```

```

ELSE BEGIN
IF TAMANHO>13 THEN ADVERT(3,13);
MANI:=TAMANHO
END;

```

```

IF TAMANHO>0 THEN NOME:=STRING(LINHA(PTCOL),TAMANHO);
WHILE LINHA(COLUNA)!=" " AND COLUNA<73 DO COLUNA:=**+1;
IF LINHA(COLUNA)="3"
THEN BEGIN
SIMBOLO:=48;
IF NOME=" "
THEN NOME:="3"
ELSE NOME:=NOME||"3";
COLUNA:=**+1;

```



```

WHILE LINHA(COLUNA)=" " AND COLUNA<73 DO COLUNA:=+1;
IF LINHA(COLUNA)="+" OR LINHA(COLUNA)="-"
THEN BEGIN
  NOME:=NOME||STRING(LINHA(COLUNA),1);
  COLUNA:=+1;
END;
PTEXP:=COLUNA;
WHILE LINHA(COLUNA) IN DIGITOS AND COLUNA<73
DO COLUNA:=+1;
TAMANHO:=COLUNA-PTEXP;
IF TAMANHO>2 THEN ADVERT(4,2);
EXP:=TAMANHO;
IF EXP=0
THEN BEGIN
  ADVERT(5,0);
  NOME:=NOME||"1"
END
ELSE NOME:=NOME||STRING(LINHA(PTEXP),EXP)
END;
COD:=0;
IF SIMBOLQ=47 AND INT=12
THEN BEGIN
  ON INTEGEROVERFLOW(DEC):COD:=6;
  IF COD=0 THEN INT:=DECIMAL(NOME)
END;
IF SIMBOLQ=48 AND EXP=2
THEN BEGIN
  ON EXPONENTUNDERFLOW(DEC):COD:=7;
  ON EXPONENTOVERFLOW(DEC):COD:=8;
  IF COD=0 THEN DEC:=DECIMAL(NOME)
END;
IF COD=0 THEN ADVERT(COD,0)
END

% TRATAMENTO DE COMENTARIOS TIPO <SCAPE REMARK>
ELSE IF LINHA(COLUNA)="% "
THEN BEGIN
  PROXLINHA;
  PROXSIMB
END

% TRATAMENTO DE CADEIAS
ELSE IF LINHA(COLUNA)=""
THEN BEGIN
  SIMBOLQ:=46;
  PICOL:=COLUNA:=+1;
  WHILE LINHA(COLUNA)="" AND COLUNA<73 DO COLUNA:=+1;
  TAMANHO:=COLUNA-PICOL;
  IF TAMANHO>6 THEN ADVERT(9,6);
  COLUNA:=+1;
END

% TRATAMENTO DE OPER. DELIM, ATRIB E OUTROS
ELSE BEGIN
  APONT:=POINTR(VETCAR);
  REPLACE APONT BY LINHA(COLUNA) FOR 2;
  I:=1;
  COND:=FALSE;
  WHILE NOT COND AND I<15
  DO IF VETCAR[I].(47:81)=VETSINAL[I].(15:8)
  THEN I:=+1
  ELSE IF I>=2
  THEN COND:=TRUE
  ELSE IF VETCAR[I].(39:81)=VETSINAL[I].(7:8)
  THEN BEGIN
    COLUNA:=+1;
    COND:=TRUE;
  END
  ELSE I:=2;
  IF COND
  THEN BEGIN
    SIMBOLQ:=CODSINAL[I];
    COLUNA:=+1;
  END

% TRATAMENTO DE CARACTERES INVALIDOS
ELSE BEGIN
  ADVERT(10,0);
  COLUNA:=+1;
  PROXSIMB
END
END
END;

```

```

*****
PROCEDURE APLICAREGRA;

```

```

BEGIN
IF TAM[PROD]=0
THEN CTVZ:=*+1
ELSE BEGIN
CTVZ:=0;
FOR I:=P[POCO[PROD]+TAM[PROD]-1 STEP -1 UNTIL P[POCO[PROD]
DO BEGIN
PILHA[TOPO]:=P[COI];
TOPO:=*+1
END
END;
TOPO:=*-1
END;

```

```

*****
PROCEDURE INSER;

```

```

BEGIN
IF PILHA[TOPO]>63
THEN BEGIN
I:=P[CADEPI_HA[TOPO]-63];
J:=TAMCADEPI_HA[TOPO]-63];
K:=0
END
ELSE BEGIN
I:=J:=0;
K:=PILHA[TOPO]
END;
IF I=0 AND K=0
THEN COND:=TRUE
ELSE COND:=FALSE;
WHILE NOT COND
DO BEGIN
VETINS[ERLINI]:=*+1;
PTINS:=*+1;
IF I>0
THEN BEGIN
K:=P[COCA[COI];
IF I+1=I+J
THEN COND:=TRUE
ELSE I:=*+1
END
ELSE COND:=TRUE;
CADINS[PTINS]:=K
END;
TOPO:=*-1
END;

```

```

*****
PROCEDURE PREFIXO(SIMB,II,VI);
INTEGER SIMB,II,VI;

```

```

BEGIN
INTEGER PIPILHA,VALPREF,VALPAR,VAL;

II:=PIPILHA:=TOP;
VI:=200;
WHILE PIPILHA>0 DO
IF PILHA[PIPILHA]<0
THEN PIPILHA:=*-1
ELSE BEGIN
IF PILHA[PIPILHA]>63
THEN BEGIN
I:=(PILHA[PIPILHA]-64)*63+SIMB;
IF I<=2781
THEN VALPAR:=TABELA[II].[7:8]
ELSE BEGIN
I:=*-2781;
VALPAR:=TABELA[III].[7:8]
END
END
ELSE IF PILHA[PIPILHA]=SIMB
THEN VALPAR:=0
ELSE VALPAR:=200;
VAL:=VALPREF+VALPAR;
IF VAL<VI
THEN BEGIN
II:=PIPILHA;
VI:=VAL
END;
IF VALPAR=0
THEN PIPILHA:=0
ELSE BEGIN
VALPREF:=*+CUSTOINS[PILHA[PIPILHA]];
IF VALPREF>=VI
THEN PIPILHA:=0
ELSE PIPILHA:=*+1
END
END

```

```
END
END;
```

```
*****
PROCEDURE RETIFICA(SI);
INTEGER SI;

BEGIN
  INTEGER TOPDINS, TINOVO,
        VALDEL, VONOVO,
        VALINS, VINOVO,
        PTSIMB, DEL;
  SI:=SIMBOLD;
  TOPDINS:=TOPD;
  VALINS:=200;
  PARE:=FALSE;
  WHILE NOT PARE
  DO IF VONOVO>=VALDEL+VALINS
    THEN PARE:=TRUE
    ELSE BEGIN
      PTSIMB:=**+1;
      IF PTSIMB>1 THEN PROXSIMB;
      PREFIXO(SIMBOLD, TINOVO, VINOVO);
      IF VONOVO+VINOVO<VALDEL+VALINS
      THEN BEGIN
        IF PTSIMB>1
        THEN BEGIN
          PTOEL:=**+1;
          CADDELE(PTOEL):=SI;
          SI:=SIMBOLD;
          END;
          TOPDINS:=TINOVO;
          VALDEL:=VONOVO;
          VONOVO:=**+CUSFODEL(SIMBOLD);
          VALINS:=VINOVO;
          DEL:=PTSIMB-1;
          END
        ELSE PARE:=TRUE;
        END;
      VETDELE(LIN):=DEL;
      WHILE TOPD>TOPDINS
      DO IF PILHACTOPD<0
        THEN PROCSE(0)
        ELSE INSERT
      END;
    END;
```

```
*****
PROCEDURE CORRETUR;
BEGIN
  INTEGER SIMBINS, TOPDINS, VALINS;

  IF FALTA
  THEN BEGIN
    ACUSAERRO;
    FALTA:=FALSE;
    END;
  MARCAERRO(PTCOL):="*";
  ERRO:=TRUE;
  NUMER:=**+1;
  ER IN:=**+1;
  TOPD:=**+CIVZ;
  RETIFICA(SIMBINS);
  WHILE PILHACTOPD/=SIMBINS
  DO BEGIN
    WHILE PILHACTOPD>63
    DO BEGIN
      I:=(PILHACTOPD-64)*63+SIMBINS;
      IF I<=2781
      THEN PROD:=TABELAIII.[15:8]
      ELSE BEGIN
        I:=**+2781;
        PROD:=TABELAIII.[15:8]
        END;
      IF PROD>0
      THEN BEGIN
        APLICAREGRA;
        IF PILHACTOPD<0
        THEN PROCSE(0);
        PREFIXO(SIMBINS, TOPDINS, VALINS);
        WHILE TOPD>TOPDINS
        DO IF PILHACTOPD<0
          THEN PROCSE(0)
          ELSE INSERT
        END
      END
    END
  END
```

```

ELSE INSERI
END;
IF PILHACTOPUJ = SIMBINS THEN INSERI;
IF PILHACTOPUJ < 0
THEN PROCSEM(C)
END;
IF SIMBINS = SIMBOLO
THEN BEGIN
RESTA := TRUE;
TOPJ := *-1;
IF PILHACTOPUJ < 0
THEN PROCSEM(C)
END
END;

```

```

*****
PROCEDURE ANALISA;

```

```

BEGIN
WHILE TOPU > 0
DO BEGIN
IF PILHACTOPUJ < 0
THEN PROCSEM(C);
IF RESTA
THEN RESTA := FALSE
ELSE PROJXSIMB;
WHILE PILHACTOPUJ > 63
DO BEGIN
I := (PILHACTOPUJ - 64) * 63 + SIMBOLO;
IF I <= 2731
THEN PROJ := TABELAII[I].(23:8)
ELSE BEGIN
I := *-2781;
PROJ := TABELAIII[I].(23:8)
END;
IF PROJ > 0
THEN BEGIN
APLICAREGRA;
IF PILHACTOPUJ < 0
THEN PROCSEM(C)
END
ELSE CORRETOR
END;
IF PILHACTOPUJ = SIMBOLO
THEN CIVZ := 0
ELSE CORRETOR;
IF NOT RESTA THEN TOPU := *-1;
END;
IMPRIME
END;

```

```

*****
% PROGRAMA PRINCIPAL

```

```

INICIALIZA;
ANALISA;
WRITE(SAIDA, <///, X10, "*** SUMARIO ***">);
IF NUMADV = 0
THEN WRITE(SAIDA, </, X10, "NUMERO DE ADVERTENCIAS = 0">);
ELSE BEGIN
WRITE(SAIDA, </, X10, "ADVERTENCIAS: ">);
IF ERLEX[1] = 1
THEN WRITE(SAIDA, <X10, "TIPO 1: NAO E PERMITIDO IDENTIFICADOR COM ",
"MAIS DE 63 CARACTERES">);
IF ERLEX[2] = 1
THEN WRITE(SAIDA, <X10, "TIPO 2: NAO E PERMITIDO NUMERO INTEIRO ",
"COM MAIS DE 12 DIGITOS">);
IF ERLEX[3] = 1
THEN WRITE(SAIDA, <X10, "TIPO 3: NAO E PERMITIDO NUMERO REAL COM ",
"PRECISAO SIMPLES COM MAIS DE 12 DIGITOS NA MANTISSA">);
IF ERLEX[4] = 1
THEN WRITE(SAIDA, <X10, "TIPO 4: NAO E PERMITIDO NUMERO REAL COM ",
"PRECISAO SIMPLES COM MAIS DE 2 DIGITOS NO EXPOENTE">);
IF ERLEX[5] = 1
THEN WRITE(SAIDA, <X10, "TIPO 5: NAO E PERMITIDO TERMINAR NUMERO ",
"REAL COM 2 OU COM SINAL OU EXPOENTE">);
IF ERLEX[6] = 1
THEN WRITE(SAIDA, <X10, "TIPO 6: NAO E PERMITIDO NUMERO INTEIRO ",
"COM VALOR SUPERIOR A 549 755 813 387">);
IF ERLEX[7] = 1
THEN WRITE(SAIDA, <X10, "TIPO 7: NAO E PERMITIDO NUMERO REAL COM ",
"PRECISAO SIMPLES COM VALOR INFERIOR A 3.758 115 402 ",
"03 & -47">);
IF ERLEX[8] = 1
THEN WRITE(SAIDA, <X10, "TIPO 8: NAO E PERMITIDO NUMERO REAL COM ",
"PRECISAO SIMPLES COM VALOR SUPERIOR A 4.313 591 466 ",
"74 & 68">);

```

```
IF ERLEX(9)=1
THEN WRITE(SAIDA,<X10,"TIPO SIN USO CADEIA DE CARACTERES EBCDIC",
CM MAIS DE 6 CARACTERES">);
IF ERLEX(10)=1
THEN WRITE(SAIDA,<X10,"TIPO IO:CARACTER INVALIDO">);
END;
WRITE(SAIDA,<X10,"NUMERO DE ERROS ENCONTRADOS=",I2>," NUMERO);
WRITE(SAIDA,<X10,"ANALISE EXECUTADA">);
END.
```

```

BEGIN
FILE SAI(KIND=PRINTER);
INTEGER VAR,VUT;
REAL ARRAY VET(EXPA:EXPA);
BOOLEAN PARE;
PROCEDURE PROC(X,Y)
INTEGER X,Y;

```

```

***
>>>ERRO 1:INCLUA ";" <<<
BEGIN
INTEGER I;
FOR I=EXPA STEP EXPA UNTIL EXPA

```

```

*
>>>ERRO 2:TROQUE "=" POR "==" <<<
DO VET(EXPA):=EXPA;
X:=EXPA
END;

```

```

FILL VET[*] WITH 5(0),3(4),2(5));

```

```

*
>>>ERRO 3:RETIRE ")" <<<
I:=EXPA;

```

```

*
>>>ERRO 4:VARIAVEL FORA DE ESCOPO<<<
WHILE EXPB
DO IF EXPB
THEN PARE:=EXPB
ELSE BEGIN
VUT:=EXPA;
PROC(VAR,VUT)

```

```

*
>>>ERRO 5:IDENTIFICADOR NAO DECLARADO<<<
END;
WRITE(SAI,7,VET[*]);
END.3

```

```

*** SUMARIO ***

```

```

NUMERO DE ADVERTENCIAS=0

```

```

NUMERO DE ERROS ENCONTRADOS= 5

```

```

ANALISE EXECUTADA

```




BEG IN	FILE SAIRINDO=PRINTEND;
INTEGER VAR VUT;	REAL ARRAY VETEXPA:EXPA1;
BOOLEAN PARE;	PROCEDURE PROC(X,Y)
INTEGER X,Y;	
* >>>ERGC 1:INCLUA "J" <<<	
BEGIN	
INTEGER I;	FOR I=EXPA STEP EXPA UNTIL EXPA
* >>>ERRE 2:IRROQUE "N" POR "I" <<<	
EO VETEXPA1:=EXPA;	X:=EXPA
END;	
FILE VET[*] WITH 5(0),3(4),2(5));	*
>>>ERRT 3:RETIRE "J" <<<	
* I:=EXPA;	
>>>ERRO 4:VARIAVEL FORA DE ESCOPO<<<	
WHILE EXPB	DO IF EXPB
THEN PARE:=EXPB	ELSE BEGIN
VUT:=EXPA;	PROCVAR(VUT)
* >>>ERRE 5:IDENTIFICADOR NAQ DECLARADO<<<	
END;	
WRITE(SAI,/,VET[*]);	END.*
*** SUMARIO ***	
NUMERO DE ADVERTENCIAS=0	
NUMERO DE ERROS ENCONTRADOS= 5	
ANALISE EXECUTADA	

ELSE CORRETCR;	00877000	003:015A:4
IF NOT RESTA THEN TIPO:=**1)	00878000	003:015C:3
END;	00879000	003:015E:4
IMPRIME	00880000	003:015F:1
END)	00881000	003:015F:4
	00882000	003:0164:5
X*****	00883000	003:0164:5
X PROGRAMA PRINCIPAL	00884000	003:0164:5
	00885000	003:0164:5
INICIALIZA;	00886000	003:0164:5
ANALISA;	00887000	003:0165:3
WRITE(SAIDA,<///,X10,"*** SUMARIO ***">);	00888000	003:0166:1
IF NUMADV=0	00889000	003:016B:2
THEN WRITE(SAIDA,<///,X10,"NUMERO DE ADVERTENCIAS=0">)	00890000	003:016B:4
ELSE BEGIN	00891000	003:016F:3
WRITE(SAIDA,<///,X10,"ADVERTENCIAS:">);	00892000	003:0171:5
IF ERLEXI1=1	00893000	003:0176:2
THEN WRITE(SAIDA,<X10,"TIPO 1:NAO E PERMITIDO IDENTIFICADOR COM ",	00894000	003:0176:5
"MAIS DE 63 CARACTERES">);	00895000	003:0179:3
IF ERLEXI2=1	00896000	003:017C:2
THEN WRITE(SAIDA,<X10,"TIPO 2:NAO E PERMITIDO NUMERO INTEIRO ",	00897000	003:017C:5
"COM MAIS DE 12 DIGITOS">);	00898000	003:017F:3
IF ERLEXI3=1	00899000	003:0182:2
THEN WRITE(SAIDA,<X10,"TIPO 3:NAO E PERMITIDO NUMERO REAL COM ",	00900000	003:0183:0
"PRECISAO SIMPLES COM MAIS DE 12 DIGITOS NA MANTISSA">);	00901000	003:0185:4
IF ERLEXI4=1	00902000	003:0188:2
THEN WRITE(SAIDA,<X10,"TIPO 4:NAO E PERMITIDO NUMERO REAL COM ",	00903000	003:0189:0
"PRECISAO SIMPLES COM MAIS DE 2 DIGITOS NO EXPOENTE">);	00904000	003:018B:4
IF ERLEXI5=1	00905000	003:018E:2
THEN WRITE(SAIDA,<X10,"TIPO 5:NAO E PERMITIDO TERMINAR NUMERO ",	00906000	003:018F:0
"REAL COM 2 OU COM SINAL DO EXPOENTE">);	00907000	003:0191:4
IF ERLEXI6=1	00908000	003:0194:2
THEN WRITE(SAIDA,<X10,"TIPO 6:NAO E PERMITIDO NUMERO INTEIRO ",	00909000	003:0195:0
"COM VALOR SUPERIOR A 549 755 813 887">);	00910000	003:0197:4
IF ERLEXI7=1	00911000	003:019A:2
THEN WRITE(SAIDA,<X10,"TIPO 7:NAO E PERMITIDO NUMERO REAL COM ",	00912000	003:019B:0
"PRECISAO SIMPLES COM VALOR INFERIOR A 8.758 115 402 ",	00913000	003:0190:4
"C3 a -47">);	00914000	003:0190:4
IF ERLEXI8=1	00915000	003:01A0:2
THEN WRITE(SAIDA,<X10,"TIPO 8:NAO E PERMITIDO NUMERO REAL COM ",	00916000	003:01A1:0
"PRECISAO SIMPLES COM VALOR SUPERIOR A 4.313 591 466 ",	00917000	003:01A3:4
"74 a 68">);	00918000	003:01A3:4
		DATA IS 00A0 LONG
IF ERLEXI9=1	00919000	003:01A6:2
THEN WRITE(SAIDA,<X10,"TIPO 9:NAO USE CADEIA DE CARACTERES EBCDIC",	00920000	003:01A7:0
" COM MAIS DE 6 CARACTERES">);	00921000	003:01A9:4
IF ERLEXI10=1	00922000	003:01AC:2
THEN WRITE(SAIDA,<X10,"TIPO 10:CARACTER INVALIDO">)	00923000	003:01AD:0
END;	00924000	003:01B0:5
WRITE(SAIDA,<///,X10,"NUMERO DE ERROS ENCONTRADOS=",I2>,NUMER);	00925000	003:01B2:2
WRITE(SAIDA,<///,X10,"ANALISE EXECUTADA">)	00926000	003:01B9:2
END.	00927000	003:01BC:2
		B.0000(003) IS 01C0 LONG
		STACKCODE IS SEGMENT 002C
		STACKCODE(02C) IS 0040 LONG
		DATA IS 0045 LONG
=====		
NUMBER OF ERRORS DETECTED = 0.		
NUMBER OF SEGMENTS = 42. TOTAL SEGMENT SIZE = 8552 WORDS. CORE ESTIMATE = 10864 WORDS. STACK ESTIMATE = 120		
PROGRAM SIZE = 1685 CARCS, 23546 SYNTACTIC ITEMS, 401 DISK SEGMENTS.		
PROGRAM FILE NAME: (C0551001)REPSAE CA PACK.		
COMPILATION TIME = 83.772 SECONDS ELAPSED; 44.191 SECONDS PROCESSING; 13.437 SECONDS I/O.		
=====		



Line	Code	Address	Operation	Address	Operation	Address	Operation
2	PREFIXO(027) IS 0028 LONG	00750000	003:0130:5				
		00751000	003:0130:5				
		00752000	003:0130:5				
		00753000	003:0130:5				
		00754000	003:0130:5				
		00755000	003:0130:5				
		00756000	003:0130:5				
2	RETIFFICA IS SEGMENT 0028	00757000	028:0000:1				
		00758000	028:0000:1				
		00759000	028:0000:1				
		00760000	028:0000:1				
		00761000	028:0000:1				
		00762000	028:0002:2				
		00763000	028:0003:1				
		00764000	028:0003:5				
		00765000	028:0003:5				
		00766000	028:0005:2				
		00767000	028:0005:3				
		00768000	028:0007:4				
		00769000	028:0009:0				
		00770000	028:0009:3				
		00771000	028:0000:14				
		00772000	028:0000:15				
		00773000	028:0000:10				
		00774000	028:0000:12				
		00775000	028:0010:1				
		00776000	028:0011:3				
		00777000	028:0013:0				
		00778000	028:0013:3				
		00779000	028:0014:1				
		00780000	028:0015:1				
		00781000	028:0015:1				
		00782000	028:0018:0				
		00783000	028:0019:0				
		00784000	028:0019:2				
		00785000	028:0018:2				
		00786000	028:0018:5				
		00787000	028:0010:0				
		00788000	028:0010:5				
		00789000	028:0011:1				
		00790000	028:0020:1				
		00791000	028:0021:3				
		00792000	028:0024:0				
		00793000	003:0130:5				
		00794000	003:0130:5				
		00795000	003:0130:5				
		00796000	003:0130:5				
		00797000	003:0130:5				
		00798000	003:0130:5				
2	CORRETORES SEGMENT 0029	00799000	029:0000:1				
		00800000	029:0000:1				
		00801000	029:0001:0				
		00802000	029:0001:4				
		00803000	029:0001:4				
		00804000	029:0002:2				
		00805000	029:0004:0				
		00806000	029:0004:4				
		00807000	029:0006:0				
		00808000	029:0007:2				
		00809000	029:0008:5				
		00810000	029:0009:0				
		00811000	029:0009:0				

00750000 003:0130:5
00751000 003:0130:5
00752000 003:0130:5
00753000 003:0130:5
00754000 003:0130:5
00755000 003:0130:5
00756000 003:0130:5
00757000 028:0000:1
00758000 028:0000:1
00759000 028:0000:1
00760000 028:0000:1
00761000 028:0000:1
00762000 028:0002:2
00763000 028:0003:1
00764000 028:0003:5
00765000 028:0003:5
00766000 028:0005:2
00767000 028:0005:3
00768000 028:0007:4
00769000 028:0009:0
00770000 028:0009:3
00771000 028:0000:14
00772000 028:0000:15
00773000 028:0000:10
00774000 028:0000:12
00775000 028:0010:1
00776000 028:0011:3
00777000 028:0013:0
00778000 028:0013:3
00779000 028:0014:1
00780000 028:0015:1
00781000 028:0015:1
00782000 028:0018:0
00783000 028:0019:0
00784000 028:0019:2
00785000 028:0018:2
00786000 028:0018:5
00787000 028:0010:0
00788000 028:0010:5
00789000 028:0011:1
00790000 028:0020:1
00791000 028:0021:3
00792000 028:0024:0
00793000 003:0130:5
00794000 003:0130:5
00795000 003:0130:5
00796000 003:0130:5
00797000 003:0130:5
00798000 003:0130:5
00799000 029:0000:1
00800000 029:0000:1
00801000 029:0001:0
00802000 029:0001:4
00803000 029:0001:4
00804000 029:0002:2
00805000 029:0004:0
00806000 029:0004:4
00807000 029:0006:0
00808000 029:0007:2
00809000 029:0008:5
00810000 029:0009:0
00811000 029:0009:0

ELSE BEGIN									
I:=J:=0;	3	00655000	003:0126:0						
K:=PILHA(I*F0)	3	00686600	003:0126:3						
END;	3	00687000	003:0127:5						
IF I=0 AND K=0	3	00688800	003:0128:3						
THEN COND:=TRUE	3	00689000	003:0129:3						
ELSE COND:=FALSE;	3	00690000	003:012A:3						
DO BEGIN	3	00691000	003:012B:3						
WHILE NOT COND	3	00692000	003:012D:2						
DO BEGIN	3	00693000	003:012D:2						
VEIN(STERLIN):=**1;	3	00694000	003:012E:1						
PTINS:=**1;	3	00695000	003:0130:2						
IF I>0	3	00696000	003:0131:4						
THEN BEGIN	3	00697000	003:0132:0						
K:=PCTCAD(I);	3	00698000	003:0132:5						
IF I+1=I*J	3	00699000	003:0134:1						
THEN COND:=TRUE	3	00700000	003:0135:1						
ELSE I:=**1	3	00701000	003:0136:2						
END	3	00702000	003:0137:5						
ELSE COND:=TRUE;	3	00703000	003:0138:5						
CATINS(PTINS):=K	3	00704000	003:0138:5						
END;	3	00705000	003:0139:1						
TOPD:=**1	3	00706000	003:013C:2						
END;	3	00707000	003:013C:4						
*****	3	00708000	003:013D:5						
PROCEDURE PREFIXO(SIMB,TI,VI);	3	00709000	003:013D:5						
INTEGER SIMB, TI, VI;	3	00710000	003:013E:5						
BEGIN	3	00711000	003:013D:5						
INTEGER PTPILHA, VALPREF, VALPAR, VAL;	3	00712000	003:013D:5						
PTPILHA:=TPF0;	3	00713000	003:013D:5						
VI:=200;	3	00714000	003:013D:5						
WHILE PTPILHA>0 DO	3	00715000	027:0000:1						
IF PTPILHA(PTPILHA)<0	3	00716000	027:0002:0						
THEN PTPILHA:=**1	3	00717000	027:0003:0						
ELSE BEGIN	3	00718000	027:0003:4						
THEN BEGIN	3	00719000	027:0005:1						
I:=((FILHA(PTPILHA)-64)*63)*SIMB;	3	00720000	027:0006:2						
THEN BEGIN	3	00721000	027:0007:5						
I:=((FILHA(PTPILHA)-64)*63)*SIMB;	3	00722000	027:0008:5						
THEN BEGIN	3	00723000	027:0009:5						
IF I<=2781	3	00724000	027:0000:0						
THEN VALPAR:=TABELA(I):*(7:8)	3	00725000	027:0000:0						
ELSE BEGIN	3	00726000	027:0005:5						
I:=**2781;	3	00727000	027:0010:3						
VALPAR:=TABELA(I):*(7:8)	3	00728000	027:0012:1						
END	3	00729000	027:0012:3						
END	3	00730000	027:0013:4						
ELSE IF PTPILHA(PTPILHA)=SIMB	3	00731000	027:0013:4						
THEN VALPAR:=0	3	00732000	027:0015:1						
ELSE VALPAR:=200;	3	00733000	027:0015:1						
VAL:=VALPREF+VALPAR;	3	00734000	027:0018:1						
IF VAL<VI	3	00735000	027:0018:1						
THEN BEGIN	3	00736000	027:0019:4						
TI:=PTPILHA;	3	00737000	027:0020:4						
VI:=VAL	3	00738000	027:0018:5						
END;	3	00739000	027:001C:2						
IF VALPAR=0	3	00740000	027:003D:0						
THEN PTPILHA:=0	3	00741000	027:0010:2						
ELSE BEGIN	3	00742000	027:003E:1						
VALPREF:=**USTOINSTPILHA(PTPILHA);	3	00743000	027:003F:2						
IF VALPREF>VI	3	00744000	027:0021:5						
THEN PTPILHA:=0	3	00745000	027:0021:5						
ELSE PTPILHA:=**1	3	00746000	027:0022:5						
END	3	00747000	027:0024:2						
END	3	00748000	027:0025:2						
END;	3	00749000	027:0025:5						
END;	3	00750000	027:0025:5						

FN03

4

2

```

WHILE LINHA<COLUNA<=63 AND COLUNA<73 DO COLUNA:=++1;
TAMPAR:=COLUNA=PTCOL;
IF TAMPAR>6 THEN ADVERT(9,63);
COLUNA:=++1;
END

```

```

% TRATAMENTO DE OPER, DELIM, ATRIB E OUTROS
ELSE BEGIN
  APOINT:=PONTERR(VETCAR);
  REPLACE APOINT BY LINHA<COLUNA FOR 2;
  I:=1;
  COND:=FALSE;
  WHILE NOT COND AND I<45
  DO IF VETCAR(I)=47:83=VETSSINACT(I)=15:81
  THEN I:=++1
  ELSE IF I>=2
  THEN COND:=TRUE
  ELSE IF VETCAR(I)=139:80=VETSSINACT(I)=17:81
  THEN BEGIN
    COLUNA:=++1;
    COND:=TRUE;
    END
  ELSE I:=2;
  END

```

```

IF COND
THEN BEGIN
  SIMBOLD:=CODOSINACT(I);
  COLUNA:=++1;
  END
ELSE BEGIN
  ADVERT(10,0);
  COLUNA:=++1;
  PROXSIMB
  END

```

```

% TRATAMENTO DE CARACTERES INVALIDOS
ELSE BEGIN
  ADVERT(10,0);
  COLUNA:=++1;
  PROXSIMB
  END

```

```

END;
END

```

```

%*****
PROCEDURE APLICAREGRA;
BEGIN
  IF TAMPAR=0
  THEN CIVZ:=**1
  ELSE BEGIN
    CIVZ:=0;
    FOR I:=PTFCOIFROD]+TAMPAROD]-1 STEP -1 UNTIL PTFCOIFROD]
    TO BEGIN
      PIIHA(TOPQ):=PFCOIF(I);
      TOPQ:=++1;
      END;
    TGPQ:=*-1;
    END;
  %*****
  PROCEDURE INSENA;
  BEGIN
    IF PIIHA(TOPQ)>63
    THEN BEGIN
      I:=PTCABD(PIIHA(TOPQ)-63);
      J:=TAKCAD(PIIHA(TOPQ)-63);
      K:=0;
      FNO;
    END
  END

```

```

%*****
PROCEDURE APLICAREGRA;
BEGIN
  IF TAMPAR=0
  THEN CIVZ:=**1
  ELSE BEGIN
    CIVZ:=0;
    FOR I:=PTFCOIFROD]+TAMPAROD]-1 STEP -1 UNTIL PTFCOIFROD]
    TO BEGIN
      PIIHA(TOPQ):=PFCOIF(I);
      TOPQ:=++1;
      END;
    TGPQ:=*-1;
    END;
  %*****
  PROCEDURE INSENA;
  BEGIN
    IF PIIHA(TOPQ)>63
    THEN BEGIN
      I:=PTCABD(PIIHA(TOPQ)-63);
      J:=TAKCAD(PIIHA(TOPQ)-63);
      K:=0;
      FNO;
    END
  END

```

```

%*****
PROCEDURE APLICAREGRA;
BEGIN
  IF TAMPAR=0
  THEN CIVZ:=**1
  ELSE BEGIN
    CIVZ:=0;
    FOR I:=PTFCOIFROD]+TAMPAROD]-1 STEP -1 UNTIL PTFCOIFROD]
    TO BEGIN
      PIIHA(TOPQ):=PFCOIF(I);
      TOPQ:=++1;
      END;
    TGPQ:=*-1;
    END;
  %*****
  PROCEDURE INSENA;
  BEGIN
    IF PIIHA(TOPQ)>63
    THEN BEGIN
      I:=PTCABD(PIIHA(TOPQ)-63);
      J:=TAKCAD(PIIHA(TOPQ)-63);
      K:=0;
      FNO;
    END
  END

```

```

%*****
PROCEDURE APLICAREGRA;
BEGIN
  IF TAMPAR=0
  THEN CIVZ:=**1
  ELSE BEGIN
    CIVZ:=0;
    FOR I:=PTFCOIFROD]+TAMPAROD]-1 STEP -1 UNTIL PTFCOIFROD]
    TO BEGIN
      PIIHA(TOPQ):=PFCOIF(I);
      TOPQ:=++1;
      END;
    TGPQ:=*-1;
    END;
  %*****
  PROCEDURE INSENA;
  BEGIN
    IF PIIHA(TOPQ)>63
    THEN BEGIN
      I:=PTCABD(PIIHA(TOPQ)-63);
      J:=TAKCAD(PIIHA(TOPQ)-63);
      K:=0;
      FNO;
    END
  END

```

```

%*****
PROCEDURE APLICAREGRA;
BEGIN
  IF TAMPAR=0
  THEN CIVZ:=**1
  ELSE BEGIN
    CIVZ:=0;
    FOR I:=PTFCOIFROD]+TAMPAROD]-1 STEP -1 UNTIL PTFCOIFROD]
    TO BEGIN
      PIIHA(TOPQ):=PFCOIF(I);
      TOPQ:=++1;
      END;
    TGPQ:=*-1;
    END;
  %*****
  PROCEDURE INSENA;
  BEGIN
    IF PIIHA(TOPQ)>63
    THEN BEGIN
      I:=PTCABD(PIIHA(TOPQ)-63);
      J:=TAKCAD(PIIHA(TOPQ)-63);
      K:=0;
      FNO;
    END
  END

```

```

%*****
PROCEDURE APLICAREGRA;
BEGIN
  IF TAMPAR=0
  THEN CIVZ:=**1
  ELSE BEGIN
    CIVZ:=0;
    FOR I:=PTFCOIFROD]+TAMPAROD]-1 STEP -1 UNTIL PTFCOIFROD]
    TO BEGIN
      PIIHA(TOPQ):=PFCOIF(I);
      TOPQ:=++1;
      END;
    TGPQ:=*-1;
    END;
  %*****
  PROCEDURE INSENA;
  BEGIN
    IF PIIHA(TOPQ)>63
    THEN BEGIN
      I:=PTCABD(PIIHA(TOPQ)-63);
      J:=TAKCAD(PIIHA(TOPQ)-63);
      K:=0;
      FNO;
    END
  END

```

00622000 022:016D:4
00621000 022:0173:5
00622000 022:0175:2
00623000 022:017A:0

00624000 022:017A:2
00625000 022:017B:2
00626000 022:017B:2
00627000 022:017B:2
00628000 022:017B:5
00629000 022:017D:2

00630000 022:017F:0
00631000 022:017F:4
00632000 022:0180:2
00633000 022:0181:1
00634000 022:0184:0
00635000 022:0185:3
00636000 022:0187:2
00637000 022:018B:2
00638000 022:018B:1

00639000 022:018C:2
00640000 022:018D:4
00641000 022:018D:4
00642000 022:018E:2
00643000 022:0190:1
00644000 022:0190:1

00645000 022:0191:0
00646000 022:0192:2
00647000 022:0192:4
00648000 022:0193:4
00649000 022:0193:4
00650000 022:0193:4
00651000 022:0194:1
00652000 022:0197:4
00653000 022:0199:0

00654000 022:0199:3
00655000 022:0199:4
00656000 022:0199:4
00657000 022:0199:4
00658000 022:019F:1
00659000 022:019F:1
00660000 022:019F:1
00661000 022:019F:1
00662000 022:019F:1
00663000 022:019F:5
00664000 022:01A1:0

00665000 022:01A2:3
00666000 022:01A3:1
00667000 022:01A7:5
00668000 022:01B8:5
00669000 022:01B8:0
00670000 022:01B8:2
00671000 022:01B8:2
00672000 022:01A3:5
00673000 022:01A3:5
00674000 022:01A3:5
00675000 022:01A3:5
00676000 022:01A3:5
00677000 022:01A3:5
00678000 022:01A3:5
00679000 022:01A3:5

00680000 022:01A3:5
00681000 022:01A3:5
00682000 022:01A3:5
00683000 022:01A3:5
00684000 022:01A3:5
00685000 022:01A3:5
00686000 022:01A3:5
00687000 022:01A3:5
00688000 022:01A3:5
00689000 022:01A3:5
00690000 022:01A3:5

00691000 022:01A3:5
00692000 022:01A3:5
00693000 022:01A3:5
00694000 022:01A3:5
00695000 022:01A3:5
00696000 022:01A3:5
00697000 022:01A3:5
00698000 022:01A3:5
00699000 022:01A3:5
00700000 022:01A3:5

00701000 022:01A3:5
00702000 022:01A3:5
00703000 022:01A3:5
00704000 022:01A3:5
00705000 022:01A3:5
00706000 022:01A3:5
00707000 022:01A3:5
00708000 022:01A3:5
00709000 022:01A3:5
00710000 022:01A3:5

00711000 022:01A3:5
00712000 022:01A3:5
00713000 022:01A3:5
00714000 022:01A3:5
00715000 022:01A3:5
00716000 022:01A3:5
00717000 022:01A3:5
00718000 022:01A3:5
00719000 022:01A3:5
00720000 022:01A3:5

```

END;
TAMANHO:=COLUNA-PTCOL;
IF CTO=1
THEN BEGIN
  IF TAMANHO>12 THEN ADVERT(2,12);
  INT:=TAMANHO
END
ELSE BEGIN
  IF TAMANHO>13 THEN ADVERT(3,13);
  NANT:=TAMANHO
END;

```

```

00554000 022:0008:4
00555000 022:0000:1
00556000 022:000E:4
00557000 022:000F:0
00558000 022:000E:5
00559000 022:00E4:5
00560000 022:00E4:5
00561000 022:00E5:5
00562000 022:00E6:2
00563000 022:00E8:2
00564000 022:00EB:2
00565000 022:00EC:2
00566000 022:00F1:4
00567000 022:00F7:0
00568000 022:00F8:1
00569000 022:00FA:0
00570000 022:00FA:5
00571000 022:00FB:3
00572000 022:00FC:5
00573000 022:010A:1
00574000 022:010B:3
00575000 022:0110:5
00576000 022:0114:3
00577000 022:0116:3
00578000 022:011E:4
00579000 022:011F:0
00580000 022:0120:0
00581000 022:0121:0
00582000 022:0124:0
00583000 022:0127:0
00584000 022:0128:3
00585000 022:0120:1
00586000 022:012E:1
00587000 022:012E:1
00588000 022:012F:0
00589000 022:0132:3
00590000 022:0136:4
00591000 022:013A:2
00592000 022:0140:1
00593000 022:0144:3
00594000 022:0145:1
00595000 022:0146:2
00596000 022:0147:3
00597000 022:014E:0
00598000 022:014E:2
00599000 022:014F:4
00600000 022:0150:5
00601000 022:0152:0
00602000 022:0155:3
00603000 022:015A:5
00604000 022:0150:1
00605000 022:015E:3
00606000 022:0160:4
00607000 022:01E2:1
00608000 022:01E2:1
00609000 022:01E2:1
00610000 022:01E3:5
00611000 022:01E6:0
00612000 022:01E6:4
00613000 022:01E7:1
00614000 022:01E7:2
00615000 022:01E7:2
00616000 022:01E7:2
00617000 022:01E9:0
00618000 022:01E9:5
00619000 022:01E9:5
00620000 022:01E9:5

```

```

IF TAMANHO<0 THEN NOME:=STRING(LINHAPTCOL1,TAMANHO);
WHILE LINHACOLUNA1=" " AND COLUNA<73 DO COLUNA:=++1;
IF LINHAPTCLUNA1="2"
THEN BEGIN
  SINGELO:=48;
  IF NOME=" "
  THEN NOME:="q"
  ELSE NOME:=NOME1+"q";
  COLUNA:=++1;
  WHILE LINHACOLUNA1=" " AND COLUNA<73 DO COLUNA:=++1;
  IF LINHACOLUNA1="*" OR LINHACOLUNA1="-"
  THEN BEGIN
    NOME:=NOME1+STRING(LINHACOLUNA1,1);
    COLUNA:=++1
  END;
  PTEX:=COLUNA;
  WHILE LINHACOLUNA1 IN DIGITOS AND COLUNA<73
  DO COLUNA:=++1;
  TAMANHO:=COLUNA-PTEXP;
  IF TAMANHO>2 THEN ADVERT(4,2);
  EXP:=TAMANHO;
  IF EXP=0
  THEN BEGIN
    ADVERT(5,0);
    NOME:=NOME1+"1"
  END
ELSE NOME:=NOME1+STRING(LINHAPTEXP,EXP)
END;

```

```

00554000 022:0008:4
00555000 022:0000:1
00556000 022:000E:4
00557000 022:000F:0
00558000 022:000E:5
00559000 022:00E4:5
00560000 022:00E4:5
00561000 022:00E5:5
00562000 022:00E6:2
00563000 022:00E8:2
00564000 022:00EB:2
00565000 022:00EC:2
00566000 022:00F1:4
00567000 022:00F7:0
00568000 022:00F8:1
00569000 022:00FA:0
00570000 022:00FA:5
00571000 022:00FB:3
00572000 022:00FC:5
00573000 022:010A:1
00574000 022:010B:3
00575000 022:0110:5
00576000 022:0114:3
00577000 022:0116:3
00578000 022:011E:4
00579000 022:011F:0
00580000 022:0120:0
00581000 022:0121:0
00582000 022:0124:0
00583000 022:0127:0
00584000 022:0128:3
00585000 022:0120:1
00586000 022:012E:1
00587000 022:012E:1
00588000 022:012F:0
00589000 022:0132:3
00590000 022:0136:4
00591000 022:013A:2
00592000 022:0140:1
00593000 022:0144:3
00594000 022:0145:1
00595000 022:0146:2
00596000 022:0147:3
00597000 022:014E:0
00598000 022:014E:2
00599000 022:014F:4
00600000 022:0150:5
00601000 022:0152:0
00602000 022:0155:3
00603000 022:015A:5
00604000 022:0150:1
00605000 022:015E:3
00606000 022:0160:4
00607000 022:01E2:1
00608000 022:01E2:1
00609000 022:01E2:1
00610000 022:01E3:5
00611000 022:01E6:0
00612000 022:01E6:4
00613000 022:01E7:1
00614000 022:01E7:2
00615000 022:01E7:2
00616000 022:01E7:2
00617000 022:01E9:0
00618000 022:01E9:5
00619000 022:01E9:5
00620000 022:01E9:5

```

```

? I R A T A M E N H O D E C O M E N T A R I O S T I P O < S C A P E R E M A R K >
ELSE IF LINHAPTCLUNA1="X"
THEN BEGIN
  PROX1INHA;
  PROX1TIMB
END
? I R A T A M E N H O D E C A D E I A S
ELSE IF LINHACOLUNA1="***"
THEN BEGIN
  SIMBETO:=46;
  FTCL:=CON INHA+***;

```

```

00621000 022:01E9:5
00622000 022:01E9:5
00623000 022:01E9:5
00624000 022:01E9:5
00625000 022:01E9:5
00626000 022:01E9:5
00627000 022:01E9:5
00628000 022:01E9:5
00629000 022:01E9:5
00630000 022:01E9:5
00631000 022:01E9:5
00632000 022:01E9:5
00633000 022:01E9:5
00634000 022:01E9:5
00635000 022:01E9:5
00636000 022:01E9:5
00637000 022:01E9:5
00638000 022:01E9:5
00639000 022:01E9:5
00640000 022:01E9:5
00641000 022:01E9:5
00642000 022:01E9:5
00643000 022:01E9:5
00644000 022:01E9:5
00645000 022:01E9:5
00646000 022:01E9:5
00647000 022:01E9:5
00648000 022:01E9:5
00649000 022:01E9:5
00650000 022:01E9:5

```

```

00651000 022:01E9:5
00652000 022:01E9:5
00653000 022:01E9:5
00654000 022:01E9:5
00655000 022:01E9:5
00656000 022:01E9:5
00657000 022:01E9:5
00658000 022:01E9:5
00659000 022:01E9:5
00660000 022:01E9:5
00661000 022:01E9:5
00662000 022:01E9:5
00663000 022:01E9:5
00664000 022:01E9:5
00665000 022:01E9:5
00666000 022:01E9:5
00667000 022:01E9:5
00668000 022:01E9:5
00669000 022:01E9:5
00670000 022:01E9:5
00671000 022:01E9:5
00672000 022:01E9:5
00673000 022:01E9:5
00674000 022:01E9:5
00675000 022:01E9:5
00676000 022:01E9:5
00677000 022:01E9:5
00678000 022:01E9:5
00679000 022:01E9:5
00680000 022:01E9:5

```

```

00681000 022:01E9:5
00682000 022:01E9:5
00683000 022:01E9:5
00684000 022:01E9:5
00685000 022:01E9:5
00686000 022:01E9:5
00687000 022:01E9:5
00688000 022:01E9:5
00689000 022:01E9:5
00690000 022:01E9:5
00691000 022:01E9:5
00692000 022:01E9:5
00693000 022:01E9:5
00694000 022:01E9:5
00695000 022:01E9:5
00696000 022:01E9:5
00697000 022:01E9:5
00698000 022:01E9:5
00699000 022:01E9:5
00700000 022:01E9:5
00701000 022:01E9:5
00702000 022:01E9:5
00703000 022:01E9:5
00704000 022:01E9:5
00705000 022:01E9:5
00706000 022:01E9:5
00707000 022:01E9:5
00708000 022:01E9:5
00709000 022:01E9:5
00710000 022:01E9:5

```

```

00711000 022:01E9:5
00712000 022:01E9:5
00713000 022:01E9:5
00714000 022:01E9:5
00715000 022:01E9:5
00716000 022:01E9:5
00717000 022:01E9:5
00718000 022:01E9:5
00719000 022:01E9:5
00720000 022:01E9:5
00721000 022:01E9:5
00722000 022:01E9:5
00723000 022:01E9:5
00724000 022:01E9:5
00725000 022:01E9:5
00726000 022:01E9:5
00727000 022:01E9:5
00728000 022:01E9:5
00729000 022:01E9:5
00730000 022:01E9:5
00731000 022:01E9:5
00732000 022:01E9:5
00733000 022:01E9:5
00734000 022:01E9:5
00735000 022:01E9:5
00736000 022:01E9:5
00737000 022:01E9:5
00738000 022:01E9:5
00739000 022:01E9:5
00740000 022:01E9:5

```

PTCOL<73 DO PTCOL:=**1;	00488000	022:0064:2
TAMANHO:=PTCOL-COLUNA1	00489000	022:0064:4
NOME:=STRING(LINHACOLUNA1-TAMANHO);	00490000	022:0066:1
IF NOME="END" OR NOME="ELSE" OR	00491000	022:0068:2
NOME="UNTIL"	00492000	022:0075:1
THEN COND:=TRUE	00493000	022:0075:5
ELSE COLUNA:=PTCOL	00494000	022:007A:4
END	00495000	022:007B:5
ELSE WHILE (LINHACOLUNA1 IN ALPHA AND	00496000	022:007C:5
COLUNA<73	00497000	022:0080:0
DO COLUNA:=**1	00498000	022:0080:2
ELSE COND:=TRUE	00499000	022:0081:5
END;	00500000	022:0083:5
% ENTRADA PARA ROTINA DO "FORMAT"	00501000	022:0085:0
7:COND:=FALSE;	00502000	022:0085:0
WHILE NOT COND	00503000	022:0085:0
DO BEGIN	00504000	022:0086:1
WHILE (LINHACOLUNA1="f" AND COLUNA<73	00505000	022:0086:1
DO COLUNA:=**1;	00506000	022:0087:0
IF COLUNA=73	00507000	022:0089:5
THEN PROXLINHA	00508000	022:008C:5
ELSE COND:=TRUE	00509000	022:008D:1
END;	00510000	022:008E:2
% ENTRADA PARA ROTINA DO "HEAD"	00511000	022:008F:2
31:COND:=FALSE;	00512000	022:0090:3
WHILE NOT COND	00513000	022:0090:3
DO BEGIN	00514000	022:0090:3
WHILE (LINHACOLUNA1="f" AND COLUNA<73	00515000	022:0091:4
DO COLUNA:=**1;	00516000	022:0091:4
IF COLUNA=73	00517000	022:0092:3
THEN PROXLINHA	00518000	022:0095:2
ELSE COND:=TRUE	00519000	022:0098:2
END;	00520000	022:0098:4
% ENTRADA PARA ROTINA DO "WRITE"	00521000	022:009A:1
34:COND:=FALSE;	00522000	022:009A:5
WHILE NOT COND	00523000	022:009C:0
DO BEGIN	00524000	022:009C:0
WHILE (LINHACOLUNA1="f" AND COLUNA<73	00525000	022:009C:0
DO COLUNA:=**1;	00526000	022:009D:1
IF COLUNA=73	00527000	022:009D:1
THEN PROXLINHA	00528000	022:009E:0
ELSE COND:=TRUE	00529000	022:00A0:5
END	00530000	022:00A3:5
% TRATAMENTO DE NÚMEROS	00531000	022:00A4:1
ELSE IF (LINHACOLUNA1 IN DIGITOS) OR (LINHACOLUNA1="2") OR	00532000	022:00A5:4
(LINHACOLUNA1="." AND (LINHACOLUNA1+1 IN DIGITOS))	00533000	022:00A6:2
THEN BEGIN	00534000	022:00A7:0
NOME:=" ";	00535000	022:00B8:0
TAMANHO:=0;	00536000	022:00B8:0
COND:=1;	00537000	022:00B8:0
WHILE (LINHACOLUNA1 IN DIGITOS AND COLUNA<73	00538000	022:00B8:0
DO COLUNA:=**1;	00539000	022:00C0:5
IF (LINHACOLUNA1="." AND	00540000	022:00C5:3
COLUNA:=**1;	00541000	022:00C7:1
COND:=2;	00542000	022:00C9:0
WHILE (LINHACOLUNA1 IN DIGITOS AND COLUNA<73	00543000	022:00C9:4
DO COLUNA:=**1;	00544000	022:00CA:2
IF (LINHACOLUNA1="." AND	00545000	022:00CB:1
COLUNA:=**1;	00546000	022:00CE:1
COND:=3;	00547000	022:00D1:1
WHILE (LINHACOLUNA1 IN DIGITOS AND COLUNA<73	00548000	022:00D2:2
DO COLUNA:=**1;	00549000	022:00D4:1
COND:=4;	00550000	022:00D5:0
WHILE (LINHACOLUNA1 IN DIGITOS AND COLUNA<73	00551000	022:00D5:5
DO COLUNA:=**1;	00552000	022:00D7:1
COND:=5;	00553000	022:00D8:3

<pre> % TRATAMENTO DE CARACTERES BRANCOS COND:=FALSE; WHILE NOT COND DO BEGIN WHILE LINHA(COLUNA)= " " AND COLUNA<73 DO COLUNA:=++1; IF COLUNA=73 THEN PREXITIVA ELSE COND:=TRUE ENCL; PTCOL:=COLUNA; % TRATAMENTO DE IDENTIFICADORES E PALAVRAS RESERVADAS IF LINHA(COLUNA) IN LETRAS THEN BEGIN WHILE LINHA(COLUNA) IN ALPHA AND COLUNA<73 DO COLUNA:=++1; TAMANHO:=COLUNA-PTCOL; IF TAMANHO>63 THEN ADVERT(1,63); NOME:=STRING(LINHA(PTCOL),TAMANHO); % TRATAMENTO DE COMENTARIOS TIPO <COMMENT REMARK> IF NOME="<COMMENT>" THEN BEGIN COND:=FALSE; WHILE NOT COND DO BEGIN WHILE LINHA(COLUNA)="" AND COLUNA<73 DO COLUNA:=++1; IF COLUNA=73 THEN PROXIMA ELSE CKND:=TRUE ENCL; COLUNA:=++1; PROXSIMB END; % ATRIBUICAO DE CODIGOS ELSE BEGIN I:=1; COND:=FALSE; WHILE NOT COND AND I<8 DO IF NOME=VEGETI THEN CKND:=TRUE ELSE I:=++1; IF COND THEN SIMBLO:=CODEXP(1) ELSE PROCSEM(5) ENCL; % TRATAMENTOS ESPECIAIS CASE SIMBLO OF BEGIN % TRATAMENTO DE COMENTARIOS TIPO <END REMARK> 2:COND:=FALSE; WHILE NOT COND DO BEGIN WHILE ((LINHA(COLUNA)=" ") OR (LINHA(COLUNA) IN DIGITOS)) AND COLUNA<73 DO COLUNA:=++1; IF COLUNA=73 THEN PROXIMA ELSE IF LINHA(COLUNA) IN ALPHA THEN IF LINHA(COLUNA)="E" OR LINHA(COLUNA)="U" THEN BEGIN PTCOL:=COLUNA; WHILE LINHA(PTCOL) IN ALPHA AND </pre>	00422000	022:0000:1
	00423000	022:0000:1
	00424000	022:0000:1
	00425000	022:0000:5
	00426000	022:0000:5
	00427000	022:0001:4
	00428000	022:0007:0
	00429000	022:0007:2
	00430000	022:0008:5
	00431000	022:0009:3
	00432000	022:000A:4
	00433000	022:000B:4
	00434000	022:000B:4
	00435000	022:000B:4
	00436000	022:000D:3
	00437000	022:000E:5
	00438000	022:001:4:5
	00439000	022:0016:2
	00440000	022:001A:5
	00441000	022:001F:0
	00442000	022:001F:0
	00443000	022:001F:0
	00444000	022:001F:4
	00445000	022:0025:4
	00446000	022:0026:2
	00447000	022:0026:2
	00448000	022:0027:1
	00449000	022:002D:0
	00450000	022:002D:2
	00451000	022:002E:5
	00452000	022:002F:3
	00453000	022:0030:4
	00454000	022:0032:0
	00455000	022:0032:3
	00456000	022:0032:4
	00457000	022:0032:4
	00458000	022:0032:4
	00459000	022:0033:1
	00460000	022:0033:5
	00461000	022:0034:3
	00462000	022:0035:2
	00463000	022:0039:5
	00464000	022:0030:2
	00465000	022:0040:2
	00466000	022:0040:2
	00467000	022:0041:3
	00468000	022:0043:3
	00469000	022:0045:1
	00470000	022:0045:1
	00471000	022:0045:1
	00472000	022:0045:3
	00473000	022:0045:3
	00474000	022:0045:3
	00475000	022:0045:3
	00476000	022:0049:2
	00477000	022:0049:2
	00478000	022:004A:1
	00479000	022:004C:1
	00480000	022:004F:2
	00481000	022:0052:2
	00482000	022:0052:4
	00483000	022:0054:1
	00484000	022:0056:4
	00485000	022:0058:4
	00486000	022:0059:2
	00487000	022:0059:2
	00488000	022:0059:2
	00489000	022:0059:2
	00490000	022:0059:2
	00491000	022:0059:2
	00492000	022:0059:2
	00493000	022:0059:2
	00494000	022:0059:2
	00495000	022:0059:2
	00496000	022:0059:2
	00497000	022:0059:2
	00498000	022:0059:2
	00499000	022:0059:2
	00500000	022:0059:2


```

SEM:=TRUE;
MARCASEM:=PTCOLJ:=P*1;
ERSEM1J:=1
END
ELSE PROCSSEM(6)
ELSE IF PTSELENDR3=-1
THEN BEGIN
    NUMER:=*1;
    SEM:=TRUE;
    MARCASEM:=PTCOLJ:=*1;
    ERSEM1J:=1;
    PROCSSEM(6)
END;

```

```

COND:=TRUE
END
ELSE IF ACWEIENDR3=" "
THEN BEGIN
    IF NOT PARAM
    THEN BEGIN
        PROCSSEM(7);
        IF NOT NOVO
        THEN BEGIN
            NUMER:=*1;
            SEM:=TRUE;
            MARCASEM:=PTCOLJ:=*1;
            ERSEM1J:=1
            END;
        END;
        COND:=TRUE
        END
    ELSE ENDF:=(ENDR*INCR) MOD 211;

```

```

IF NUMEIDENDR3=" "
THEN SIMBRL(:=9
ELSE SIMBRL(:=CEIDENLENDR3;
6:PTTASH(PTSEM):=ETR;
PTIG(PTSEM):=PTSELENDR3;
PTSELENDR3:=PTSEM;
PTSEM:=*1;
7:NONEIDENDR3:=NCKE;
CCOIDELENDR3:=9;
PTSELENDR3:=PTSEM;
PTASH(PTSEM):=ENDR;
PTIG(PTSEM):=-1;
PTSEM:=*1;
0:NEVD:=FALSE;
PARAM:=TRUE;
9:NEVD:=TRUE;
PARAM:=FALSE
END;

```

```

*****
PROCEDURE PROXSIME;
BEGIN
% DECTARCODES
INTEGER COD,IM1,MANT,EXP,PTEXP;
REAL TEC;
TRUTHSET DIGIT(S(*0123456789*));
TRUTHSET LETRAS(ALFA AND NOT DIGITOS);

```

```

00360000 003:008:3
00361000 003:000:1
00362000 003:001:5
00363000 003:002:2
00364000 003:002:5
00365000 003:003:5
00366000 003:003:4
00367000 003:003:3
00368000 003:003:0
00369000 003:003:4
00370000 003:003:2
00371000 003:000:3
00372000 003:000:0
00373000 003:000:4
00374000 003:000:4
00375000 003:000:2
00376000 003:000:1
00377000 003:000:5
00378000 003:000:5
00379000 003:000:4
00380000 003:000:5
00381000 003:000:5
00382000 003:000:4
00383000 003:000:0
00384000 003:000:4
00385000 003:000:2
00386000 003:000:5
00387000 003:000:2
00388000 003:000:2
00389000 003:000:2
00390000 003:000:0
00391000 003:000:0
00392000 003:000:2
00393000 003:000:0
00394000 003:000:4
00395000 003:000:4
00396000 003:000:3
00397000 003:000:0
00398000 003:000:2
00399000 003:000:5
00400000 003:000:1
00401000 003:000:4
00402000 003:000:1
00403000 003:000:3
00404000 003:000:5
00405000 003:000:0
00406000 003:000:4
00407000 003:000:5
00408000 003:000:5
00409000 003:000:1
00410000 003:000:1
00411000 003:000:1
00412000 003:000:1
00413000 003:000:1
00414000 003:000:1
00415000 003:000:1
00416000 003:000:1
00417000 022:000:1
00418000 022:000:1
00419000 022:000:1
00420000 022:000:1
00421000 022:000:1

```

A:=**11;	00296000	003:009032
IPUTJ1:=X;	00297000	003:009114
TAMANHO:=MAXCAR;	00298000	003:009131
MARCAADV(LPITCOLJ):="**"	00299000	003:009141
END;	00300000	003:009151
*****	00301000	003:009160
PROCEDURE HASH;	00302000	003:009160
BEGIN	00303000	003:009160
INTER N,PAL;	00304000	003:009160
PAL:=1;	00305000	003:009160
COND:=FALSE;	00306000	003:009160
WHILE NOT COND	00307000	021:000001
DO BEGIN	00308000	021:000005
N:=**6;	00309000	021:000113
IF TAMANHO>N	00310000	021:000212
THEN PAL:=**1	00311000	021:000315
ELSE COND:=TRUE	00312000	021:000414
END;	00313000	021:000513
APONT:=PONT(VEICAR);	00314000	021:000710
REPLACE APONT:=APONT BY LINHA[PTCOLJ] FOR TAMANHO;	00315000	021:000710
IF TAMANHO<N	00316000	021:000811
THEN REPLACE A[PTNT BY " " FOR N-TAMANHO;	00317000	021:000914
IF P[PTNT]	00318000	021:000010
THEN FOR I:=2 STEP 1 UNTIL PAL	00319000	021:000012
DO VEICAR[I]:=FEAL(BOOLEAN(VEICAR[I]) EQV	00320000	021:001114
BOOLEAN(VEICAR[I+1]));	00321000	021:001110
VEICAR[I+1]:=FEAL(BOOLEAN(VEICAR[I]-I47:16J) EQV	00322000	021:001110
BOOLEAN(VEICAR[I+1]-I31:16J));	00323000	021:001110
VEICAR[I+1]:=FEAL(BOOLEAN(VEICAR[I]-I31:16J) EQV	00324000	021:001110
BOOLEAN(VEICAR[I+1]-I15:16J));	00325000	021:001110
END; INCH:=VEICAR[I+1]-I15:16J MOD 211;	00326000	021:001110
END;	00327000	021:001110
*****	00328000	021:002110
PROCEDURE P[ROCEM(S);	00329000	021:002311
INTEGER S;	00330000	021:002411
CASE S OF	00331000	003:009160
BEGIN	00332000	003:009160
0:P[ROCEM(-P[1]HAT(F[0]));	00333000	003:009160
T[0]:="**1";	00334000	003:009160
1:CONTROL[PT[LETT[CC[0]]:=PTSEM;	00335000	003:009160
P[1]L[0]:="**1";	00336000	003:009160
2:P[1]L[0]:="**1";	00337000	003:009160
PTSEM:=**1;	00338000	003:009160
FOR I:=CENTRAL[LETT[BL[0]] STEP 1 UNTIL PTSEM	00339000	003:009160
DO PTSEM[PT[ASH(I)]:=P[1]L[0];	00340000	003:009160
PTSEM:=CENTRAL[LETT[BL[0]]];	00341000	003:009160
3:NEVO:=TRUE;	00342000	003:009160
4:NEVO:=FALSE;	00343000	003:009160
5:H[SH];	00344000	003:009160
COND:=FALSE;	00345000	003:009160
WHILE NOT COND	00346000	003:009160
DO IF NOT[LETT[END[F]=NOME	00347000	003:009160
THEN BEGIN	00348000	003:009160
IF COD[LETT[END[R]=9	00349000	003:009160
THEN IF ACT[PARAM	00350000	003:009160
THEN IF NOVO	00351000	003:009160
THEN IF P[TE[LETT[END[R]=CONTROL[LETT[BL[0]-1]	00352000	003:009160
THEN BEGIN	00353000	003:009160
NUMERO:="**1";	00354000	003:009160
*****	00355000	003:009160
*****	00356000	003:009160
*****	00357000	003:009160
*****	00358000	003:009160
*****	00359000	003:009160
*****	00360000	003:009160

```

THEN WRITE(SAIDA,<X7,">>ERR0",I2,"REIRE "A**" <<<">,
NEMER,TEMPD,CADETAD)
ELSE WRITE(SAIDA,<X7,">>ERR0",I2,"TROQUE "A**" POR "A**
" <<<">,NEMER,CONPD,CADETAD,COMP1,CADETAD)
END;
ERR0:=FALSE;
ERLIN:=P TOEL:=TTINS:=0;
FOR I:=1 STEP 1 UNTIL 10 DO VEIDELT1:=VEINST1:=0
END;
ACUSAERRD(01F) IS 0058 LONG

```

```

*****
PROCEDURE IMPRIME;

```

```

BEGIN
WRITE(SAIDA,<X10,A73>,LINHA);

```

```

IF ADV
THEN BEGIN
WRITE(SAIDA,ADVERTENCIA,MARCAADV*1,A;FOR I:=1 STEP 1 UNTIL A
DU TIPORT1);
ADV:=FALSE;
A:=0
END;

```

```

IF SEM
THEN BEGIN

```

```

WRITE(SAIDA,<X10+72A1>,"MARCASEM*1");
IF ERSEM=1
THEN WRITE(SAIDA,<X7,">>ERR0",I2,"NAD E PERVAIIDD IDENTIFICA",
"UTRES COM NOMES 16VAIS NO RESMO BLOCDO<<<">,NLPER);
IF ERSEM=1
THEN WRITE(SAIDA,<X7,">>ERR0",I2,"IDENTIFICADOR MAD DECLARA",
"DC<<<">,NUMER);
IF ERSEM=1
THEN WRITE(SAIDA,<X7,">>ERR0",I2,"VARIIVEL FORA DE ESCOPD",
"DC<<<">,NUMER);

```

```

FOR I:=1 STEP 1 UNTIL 3 DO ERSEM=0;
SEM:=FALSE
END;

```

```

IF ERRO
THEN BEGIN

```

```

WRITE(SAIDA,<X10+72A1>,"MARCADERROT*1");
IF PARE
THEN ACUSAERRC
ELSE FALTA:=TRUE
END;

```

```

FOR I:=1 STEP 1 UNTIL 72
DO MARCAADVLT1:=PARTSEM=1:=MARCADERROT1:=0
END;

```

```

*****
PROCEDURE PROXLINHA;

```

```

BEGIN
IMPRIME;
READ(CENTRADA,<A73>,LINHA);
COLUNT:=1
END;

```

```

*****
PROCEDURE ADVERT(X,MAXCAR);
INTEGER Y,MAXCAR;

```

```

BEGIN
NUMADV:=1;
ERLINT1:=1;

```

00231000	01F:002425
00232000	01F:002833
00233000	01F:0020C5
00234000	01F:003334
00235000	01F:003030
00236000	01F:004115
00237000	01F:004233
00238000	01F:004433
00239000	01F:004730
ACUSAERRD(01F) IS 0058 LONG	
00240000	003:001932
00241000	003:001932
00242000	003:001932
00243000	003:001932
00244000	003:001932
00245000	003:001932
00246000	003:002132
00247000	003:002132
00248000	003:002231
00249000	003:002235
00250000	003:003332
00251000	003:003430
00252000	003:003430
00253000	003:003434
00254000	003:003434
00255000	003:003533
00256000	003:003532
00257000	003:003535
00258000	003:004133
00259000	003:004732
00260000	003:004735
00261000	003:007A33
00262000	003:005032
00263000	003:005130
00264000	003:005334
00265000	003:005932
00266000	003:005035
00267000	003:005035
00268000	003:005E33
00269000	003:005E33
00270000	003:005F32
00271000	003:006832
00272000	003:006832
00273000	003:006934
00274000	003:006A32
00275000	003:006B30
00276000	003:006E33
00277000	003:006E34
00278000	003:008035
00279000	003:008035
00280000	003:008035
00281000	003:008035
00282000	003:008035
00283000	003:008035
00284000	003:008133
00285000	003:008932
00286000	003:008932
00287000	003:008C35
00288000	003:008C35
00289000	003:008C35
00290000	003:008C35
00291000	003:008C35
00292000	003:008C35
00293000	003:008C35
00294000	003:008E31

TOP0:=1; PILHATIDP0J:=64; END;	00169600 014:00AB:0 00170000 014:00AB:4 00171000 014:00AD:0 INICIALIZA(Q14) IS 00B LONG
***** PROCEDURE LECACELAC(C,C,C1); INTEGER CC,C1; BEGIN INTEGER CODJ; CT:=1; CADET:=0; FOR K:=1 STEP 1 UNTIL CC DO BEGIN IF J<0 THEN BEGIN PTDEL:=**+1; CODJ:=C*VUDEL*PTDEL; END ELSE BEGIN PTINS:=**+1; CODJ:=CADINS*PTINS; END; CT:=**+1*AMIRAD*CODJ; CADEIA:=CADEIA11*ADUZ*CODJ; IF K<CC THEN BEGIN CT:=**+1; CADEIA:=CADEIA11* END END; CT:=**+1; CADEIA:=CADEIA11* END;	2 00172000 003:0019:2 00173000 003:0019:2 00174000 003:0019:2 00175000 003:0019:2 00176000 003:0019:2 00177000 003:0019:2 00178000 003:0019:2 LECADEIA IS SEGMENT 001B 2 00179000 01B:0000:1 00180000 01B:0001:0 00181000 01B:0003:2 00182000 01B:0003:5 00183000 01B:0007:5 00184000 01B:0008:1 00185000 01B:0009:0 00186000 01B:000A:2 00187000 01B:000A:4 00188000 01B:000B:4 00189000 01B:000C:1 00190000 01B:000D:3 00191000 01B:000D:5 00192000 01B:000E:5 00193000 01B:0030:5 00194000 01B:001A:4 00195000 01B:001B:0 00196000 01B:001C:0 00197000 01B:001D:3 00198000 01B:0021:4 00199000 01B:0024:5 00200000 01B:0025:2 00201000 01B:0026:5 00202000 01B:0028:0 LECADEIA(01B) IS 0039 LONG
***** PROCEDURE ACUSAERRO; BEGIN INTEGER COMPC,COMP0,COMP1; STRING CADEIAD,CADEIAI; PTDEL:=PTINS:=C; FOR J:=1 STEP 1 UNTIL ERLIN DO BEGIN IF VETDELLI1>0 THEN BEGIN J:=1; COMPC:=VETDELLI1; LECADEIAKCOMPC,COMP0; CADEIAIAC:=CADEIA END; IF VETINS11>0 THEN BEGIN J:=1; COMPC:=VETINS11; LECADEIAKCOMPC,COMP1; CADEIAI:=CADEIA END; IF VETDELLI1=0 THEN WRITE(SAIDA,<X7,">>>ERRO",I2," INCLUA "A",<<<<>,NUMER, COMPT:=CADEI11) CASE OF VETINS11=0	2 00203000 003:0019:2 00204000 003:0019:2 00205000 003:0019:2 00206000 003:0019:2 00207000 003:0019:2 00208000 003:0019:2 ACUSAERRO IS SEGMENT 001F 2 00209000 01F:0000:1 00210000 01F:0000:1 00211000 01F:0001:3 00212000 01F:0002:0 00213000 01F:0006:0 00214000 01F:0007:0 00215000 01F:0007:5 00216000 01F:0008:4 00217000 01F:000A:2 00218000 01F:000C:0 00219000 01F:000C:4 00220000 01F:000E:2 00221000 01F:000F:2 00222000 01F:0010:1 00223000 01F:0010:5 00224000 01F:0012:3 00225000 01F:0014:1 00226000 01F:0014:5 00227000 01F:0016:3 00228000 01F:0017:3 00229000 01F:001D:0 00230000 01F:001E:2

00051000	003:0019:2	
00052000	003:0019:2	
00053000	003:0019:2	
00054000	003:0019:2	
00055000	003:0019:2	
00056000	003:0019:2	
00057000	003:0019:2	
00058000	003:0019:2	
00059000	003:0019:2	
00060000	003:0019:2	
00061000	003:0019:2	
DATA IS 0085 LONG		
00062000	003:0019:2	
00063000	003:0019:2	
00064000	003:0019:2	
00065000	003:0019:2	
00066000	003:0019:2	
00067000	003:0019:2	
00068000	003:0019:2	
00069000	003:0019:2	
00070000	003:0019:2	
00071000	003:0019:2	
00072000	003:0019:2	
00073000	003:0019:2	
00074000	003:0019:2	
00075000	003:0019:2	
00076000	003:0019:2	
00077000	003:0019:2	
00078000	003:0019:2	
00079000	003:0019:2	
00080000	003:0019:2	
DATA IS 017E LONG		
00081000	003:0019:2	
00082000	003:0019:2	
00083000	003:0019:2	
00084000	003:0019:2	
00085000	003:0019:2	
DATA IS 0054 LONG		
00086000	003:0019:2	
00087000	003:0019:2	
00088000	003:0019:2	
00089000	003:0019:2	
00090000	003:0019:2	
DATA IS 0054 LONG		
00091000	003:0019:2	
00092000	003:0019:2	
00093000	003:0019:2	
00094000	003:0019:2	
00095000	003:0019:2	
DATA IS 004A LONG		
00096000	003:0019:2	
00097000	003:0019:2	
00098000	003:0019:2	
00099000	003:0019:2	
00100000	003:0019:2	
00101000	003:0019:2	
00102000	003:0019:2	
DATA IS 0093 LONG		
00103000	003:0019:2	
00104000	003:0019:2	
00105000	003:0019:2	
00106000	003:0019:2	
00107000	003:0019:2	
DATA IS 0040 LONG		
00108000	003:0019:2	
00109000	003:0019:2	

TAMRADCC

REP S A E U N O I S N
= = = = =

BEGIN

% DECLARACDES GLOBAIS

FILE ENTRADA(KIND=READER),SAIDA(KIND=PRINTER);

INTEGER I,J,K,
COLUNA,PTCCL,
SIMBOTO,TRANKTL,
ENDR,INCR,
TOP0,PR0D,CTVZ,
PTSEM,PTBUDCU,
NUMADV,ERLIN,NLWER,
FIINS,PIDEL;
00012000 003:0000:1
00013000 003:0000:1
00014000 003:0000:1
00015000 003:0000:1
00016000 003:0000:1
00017000 003:0000:1
00018000 003:0000:1
00019000 003:0000:1
00020000 003:0000:1
00021000 003:0000:1
00022000 003:0005:4
00023000 003:0008:1
00024000 003:0008:1
00025000 003:0008:1
00026000 003:0008:1
00027000 003:000A:4
00028000 003:000A:4
00029000 003:0000:1
00030000 003:0000:1
00031000 003:0000:1
00032000 003:0011:2
00033000 003:0015:2
00034000 003:0019:2
00035000 003:0019:2
00036000 003:0019:2
00037000 003:0019:2
DATA IS 000F LONG
DATA IS 0008 LONG
DATA IS 000F LONG
DATA IS 000F LONG
00043000 003:0019:2
00044000 003:0019:2
00045000 003:0019:2
00046000 003:0019:2
00047000 003:0019:2
00048000 003:0019:2
00049000 003:0019:2
00050000 003:0019:2

BOOLEAN COND,RESTP,
ADV,SEN,ERRO,
NGVD,PARAR,PAPE,FALIA;
00015000 003:0000:1
00016000 003:0000:1
00017000 003:0000:1
00018000 003:0000:1
00019000 003:0000:1
00020000 003:0000:1
00021000 003:0003:1
00022000 003:0005:4
00023000 003:0008:1
00024000 003:0008:1
00025000 003:0008:1
00026000 003:0008:1
00027000 003:000A:4
00028000 003:000A:4
00029000 003:0000:1
00030000 003:0000:1
00031000 003:0000:1
00032000 003:0011:2
00033000 003:0015:2
00034000 003:0019:2
00035000 003:0019:2
00036000 003:0019:2
00037000 003:0019:2
DATA IS 000F LONG
DATA IS 0008 LONG
DATA IS 000F LONG
DATA IS 000F LONG
00043000 003:0019:2
00044000 003:0019:2
00045000 003:0019:2
00046000 003:0019:2
00047000 003:0019:2
00048000 003:0019:2
00049000 003:0019:2
00050000 003:0019:2

POINTER PBDNT,
STRING NCNE,CADEIA;
00018000 003:0000:1
00019000 003:0000:1
00020000 003:0000:1
00021000 003:0003:1
00022000 003:0005:4
00023000 003:0008:1
00024000 003:0008:1
00025000 003:0008:1
00026000 003:0008:1
00027000 003:000A:4
00028000 003:000A:4
00029000 003:0000:1
00030000 003:0000:1
00031000 003:0000:1
00032000 003:0011:2
00033000 003:0015:2
00034000 003:0019:2
00035000 003:0019:2
00036000 003:0019:2
00037000 003:0019:2
DATA IS 000F LONG
DATA IS 0008 LONG
DATA IS 000F LONG
DATA IS 000F LONG
00043000 003:0019:2
00044000 003:0019:2
00045000 003:0019:2
00046000 003:0019:2
00047000 003:0019:2
00048000 003:0019:2
00049000 003:0019:2
00050000 003:0019:2

REAL ARRAY VETCAR(1:11);
00027000 003:000A:4
00028000 003:000A:4
00029000 003:0000:1
00030000 003:0000:1
00031000 003:0000:1
00032000 003:0011:2
00033000 003:0015:2
00034000 003:0019:2
00035000 003:0019:2
00036000 003:0019:2
00037000 003:0019:2
DATA IS 000F LONG
DATA IS 0008 LONG
DATA IS 000F LONG
DATA IS 000F LONG
00043000 003:0019:2
00044000 003:0019:2
00045000 003:0019:2
00046000 003:0019:2
00047000 003:0019:2
00048000 003:0019:2
00049000 003:0019:2
00050000 003:0019:2

STRING ARRAY VETPEL(1:7);
00027000 003:000A:4
00028000 003:000A:4
00029000 003:0000:1
00030000 003:0000:1
00031000 003:0000:1
00032000 003:0011:2
00033000 003:0015:2
00034000 003:0019:2
00035000 003:0019:2
00036000 003:0019:2
00037000 003:0019:2
DATA IS 000F LONG
DATA IS 0008 LONG
DATA IS 000F LONG
DATA IS 000F LONG
00043000 003:0019:2
00044000 003:0019:2
00045000 003:0019:2
00046000 003:0019:2
00047000 003:0019:2
00048000 003:0019:2
00049000 003:0019:2
00050000 003:0019:2

FORMAT ALVRENCIA(“(”>>LEVERT””,72A1,“(”<TIP0 “,(I2,X1),“(”<“);
00027000 003:000A:4
00028000 003:000A:4
00029000 003:0000:1
00030000 003:0000:1
00031000 003:0000:1
00032000 003:0011:2
00033000 003:0015:2
00034000 003:0019:2
00035000 003:0019:2
00036000 003:0019:2
00037000 003:0019:2
DATA IS 000F LONG
DATA IS 0008 LONG
DATA IS 000F LONG
DATA IS 000F LONG
00043000 003:0019:2
00044000 003:0019:2
00045000 003:0019:2
00046000 003:0019:2
00047000 003:0019:2
00048000 003:0019:2
00049000 003:0019:2
00050000 003:0019:2

VALUE ARRAY
00027000 003:000A:4
00028000 003:000A:4
00029000 003:0000:1
00030000 003:0000:1
00031000 003:0000:1
00032000 003:0011:2
00033000 003:0015:2
00034000 003:0019:2
00035000 003:0019:2
00036000 003:0019:2
00037000 003:0019:2
DATA IS 000F LONG
DATA IS 0008 LONG
DATA IS 000F LONG
DATA IS 000F LONG
00043000 003:0019:2
00044000 003:0019:2
00045000 003:0019:2
00046000 003:0019:2
00047000 003:0019:2
00048000 003:0019:2
00049000 003:0019:2
00050000 003:0019:2

VEISINAL(0,
00027000 003:000A:4
00028000 003:000A:4
00029000 003:0000:1
00030000 003:0000:1
00031000 003:0000:1
00032000 003:0011:2
00033000 003:0015:2
00034000 003:0019:2
00035000 003:0019:2
00036000 003:0019:2
00037000 003:0019:2
DATA IS 000F LONG
DATA IS 0008 LONG
DATA IS 000F LONG
DATA IS 000F LONG
00043000 003:0019:2
00044000 003:0019:2
00045000 003:0019:2
00046000 003:0019:2
00047000 003:0019:2
00048000 003:0019:2
00049000 003:0019:2
00050000 003:0019:2

COOSINAT(0,27,34,5,17,35,36,37,38,39,44,54,55,56);
00027000 003:000A:4
00028000 003:000A:4
00029000 003:0000:1
00030000 003:0000:1
00031000 003:0000:1
00032000 003:0011:2
00033000 003:0015:2
00034000 003:0019:2
00035000 003:0019:2
00036000 003:0019:2
00037000 003:0019:2
DATA IS 000F LONG
DATA IS 0008 LONG
DATA IS 000F LONG
DATA IS 000F LONG
00043000 003:0019:2
00044000 003:0019:2
00045000 003:0019:2
00046000 003:0019:2
00047000 003:0019:2
00048000 003:0019:2
00049000 003:0019:2
00050000 003:0019:2

LANGU,
7,2,1,5,2,2,1,2,5,6,2,1,3,0,1,1,1,1,1,2,3,3,3,2,2,1,5,4,4,4,6,2,5,
1,4,4,1,6,5,3,3,0,4,0,3,0,2,1,1,1,6,3,2,2,2,2,0,4,6,0,7,4,2,6,0,
3,3,5,0,2,8,5,0,2,5,0,1,7,1,4,3,0,2,1,2,1,3,0,3,0,2,3,0,5,0,3,1,2,7,1,2,
1,2,1,3,4,1,1,1,1,1,1,1,1,2,3,0,2,0,1,3,3,0,1,1,1,1,1,1,2,0,4,0,5,2,4,1,3,
0,1,1,2,1,1,3,0,2,4,2,2,2,6,0,5,5,4,0,1,1,1,1,1,1,2,1,1,3,0,2,0,2,3,
0);
00027000 003:0019:2
00028000 003:0019:2
00029000 003:0019:2
00030000 003:0019:2
00031000 003:0019:2
00032000 003:0019:2
00033000 003:0019:2
00034000 003:0019:2
00035000 003:0019:2
00036000 003:0019:2
00037000 003:0019:2
DATA IS 000F LONG
DATA IS 0008 LONG
DATA IS 000F LONG
DATA IS 000F LONG
00043000 003:0019:2
00044000 003:0019:2
00045000 003:0019:2
00046000 003:0019:2
00047000 003:0019:2
00048000 003:0019:2
00049000 003:0019:2
00050000 003:0019:2

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO



UNIVERSIDADE FEDERAL DO RIO DE JANEIRO



NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

```
*****          COS91001          *****
* TEMPO DE PROCESSADOR 0:00:47 (USC 170,85) SALDO DE PROCESSADOR USC 106,54 * (SALDO DEVEDOR)
* TEMPO DE E/S 0:00:18 (USC 35,29) SALDO DE LINHAS USC 1168,35 *
* LINHAS IMPRESSAS 1031 (USC 51,85) SALDO DE OUTROS USC 348,29 *
* INTEGRAL DE MEMORIA 2495 (USC 139,23) CUSTO DO JOB USC 397,24 *
*****
```




DATE: APR 2, 1982 17:53:55, SYSTEM SERIAL: 260, B6700 MCP: UFRJ/MCP3125. 31-280-1253

WORK FLOW STATEMENTS FOR JOB 7335

```

00000100 ? BEGIN JOB REPAE;
00001200 NAME="FEPSAE#82";
00000300 USER=CHS91001;
00001400 MAXPRECTIME=50;
00000500 MAXIDTIME=20;
00000600 MAXLINES=1500;
00000700 COMPITE-REPSAE ALGOL;
00000800 ALGOL FILE TABGRAVCRIND=PACK, TITLE=(COS00001)RITA/TABELA ON NCEPACK,
00000900 FAMILYNAME=NCEPACK);
00001000 ALGOL TAIA
00053500 $DATA ENTRADA
00056400 $END JFB
    
```

J O E S U M M A R Y

```

17:53:43 7336 USUARIO SUSPENSO: CPU
17:53:51 7343 USUARIO SUSPENSO: CPU
17:53:52 7333 USUARIO SUSPENSO: CPU
    
```

Job ID	User	System Serial	MCP	Serial	Flow Statement	Start	End	Status
7333	USUARIO	SUSPENSO	CPU					
7343	USUARIO	SUSPENSO	CPU					
7336	USUARIO	SUSPENSO	CPU					

APÊNDICE BTRAÇADO DE CURVAS EQUIPOTENCIAIS E LINHAS DE
CAMPO NA VIZINHANÇA DE UM FEIXE DE CONDUTORES

Para se efetuar o traçado de curvas equipotenciais e linhas de campo na vizinhança de um feixe de condutores, torna-se necessário o cálculo das interseções das curvas equipotenciais com as linhas de campo devidas ao feixe destes N condutores considerados, relativos a uma dada fase da linha de transmissão.

Este cálculo é realizado através dos Programas P-1 e P-2, implantados na calculadora HP-41C, com a utilização de cartões magnéticos.

No Programa P-1, os cálculos são relativos às interseções de curvas equipotenciais com linhas de força, ao longo destas curvas equipotenciais, enquanto no Programa P-2, as interseções são obtidas ao longo das linhas de força.

No ítem (B-1) deste apêndice, é apresentado o desenvolvimento teórico relativo às equações empregadas em ambos os programas, enquanto no ítem (B.2), são apresentados os Programas P-1 e P-2, incluindo seus respectivos modos de utilização e casos teste.

B.1 - DESENVOLVIMENTO TEÓRICO

Uma das vantagens da representação gráfica reside no fato de se prestar a uma primeira aproximação qualitativa para distribuições de campo elétrico mais complexas, possibilitando de certa forma verificar o quão são razoáveis os resultados obtidos através de um tratamento analítico eventualmente efetuado.

O fato de se tratar de uma geometria a duas dimensões permite uma simplificação em termos de representação gráfica, pois a solução encontrada para uma dada seção fornece as informações necessárias para seções paralelas próximas.

Valores quantitativos de campo elétrico relativo podem ser obtidos num ponto qualquer de uma dada seção plana, através do traçado de curvas equipotenciais e linhas de força.

Determinando-se o valor do campo elétrico máximo, considerado como de referência, obtém-se o valor aproximado do campo elétrico dos demais pontos. É o caso, por exemplo, quando se considera a relação entre o valor de campo elétrico em um ponto, e o valor de campo elétrico máximo na superfície de um dos N condutores do feixe de uma fase da linha de transmissão.

O método empregando "quadrados curvilíneos" permite aumentar a precisão desejada para se obter o valor de cam

po elétrico, subdividindo-se os quadrados traçados, em outros cada vez menores.

Pela interpolação de linhas equipotenciais suficientemente próximas, entre outras já existentes, obtem-se o valor médio da intensidade do campo elétrico \vec{E} , associado a um dado ponto, e para uma certa precisão, dividindo-se a diferença de potencial entre curvas equipotenciais, que envolvam o ponto considerado, pela distância entre elas, medida ao longo das linhas de força que também envolvam o ponto considerado.

Considerando-se entretanto o limite desta relação, o campo elétrico pode ser praticamente associado a um ponto, desde que os valores de módulo e direção obtidos para subdivisões menores de "quadrados curvilíneos", satisfaçam determinados valores de tolerância impostos.

Conforme deduzido no Apêndice A, a expressão [A.7], $\rho_k = \alpha e^{(\pi/m)k}$, é relativa a curvas equipotenciais devidas a um condutor cilíndrico de seção circular, cujo centro geométrico esteja localizado no ponto (1,0), do plano complexo W . Considerando-se cada condutor de um feixe de N condutores dispostos simetricamente, o ângulo que as linhas de força e a ele associadas fazem com o eixo real u , no plano W , são obtidas pela relação

$$\delta = (\pi/m)i,$$

onde

$2m$, é o número de linhas de força consideradas para cada condutor do feixe, para m inteiro positivo;

i , é um número inteiro não negativo;

δ , é o ângulo, em radiano, que uma dada linha de força faz com o eixo real u , no plano W , conforme mostrado na Fig. B.1.

Um ponto que pertença à interseção de uma curva equipotencial com uma linha de força, ambas devidas ao condutor localizado no ponto $(1,0)$ do plano W , pode ter suas coordenadas associadas ao vetor w tal que

$w = 1 + \rho \cdot e^{j\delta}$, sendo ρ dado pela relação [3.18] fazendo-se

$$\rho = \rho_k,$$

ou

$$w = (1 + \rho \cdot \cos \delta) + j \cdot (\rho \cdot \sin \delta).$$

O módulo de w , representado por $|w|$, e seu argumento, representado por $\arg w$, são dados respectivamente por:

$$|w| = \sqrt{\rho^2 + 2 \cdot \rho \cdot \cos \delta + 1},$$

e

$$\arg w = \text{Tan}^{-1} \left[\frac{\rho \cdot \sin \delta}{1 + \rho \cdot \cos \delta} \right]$$

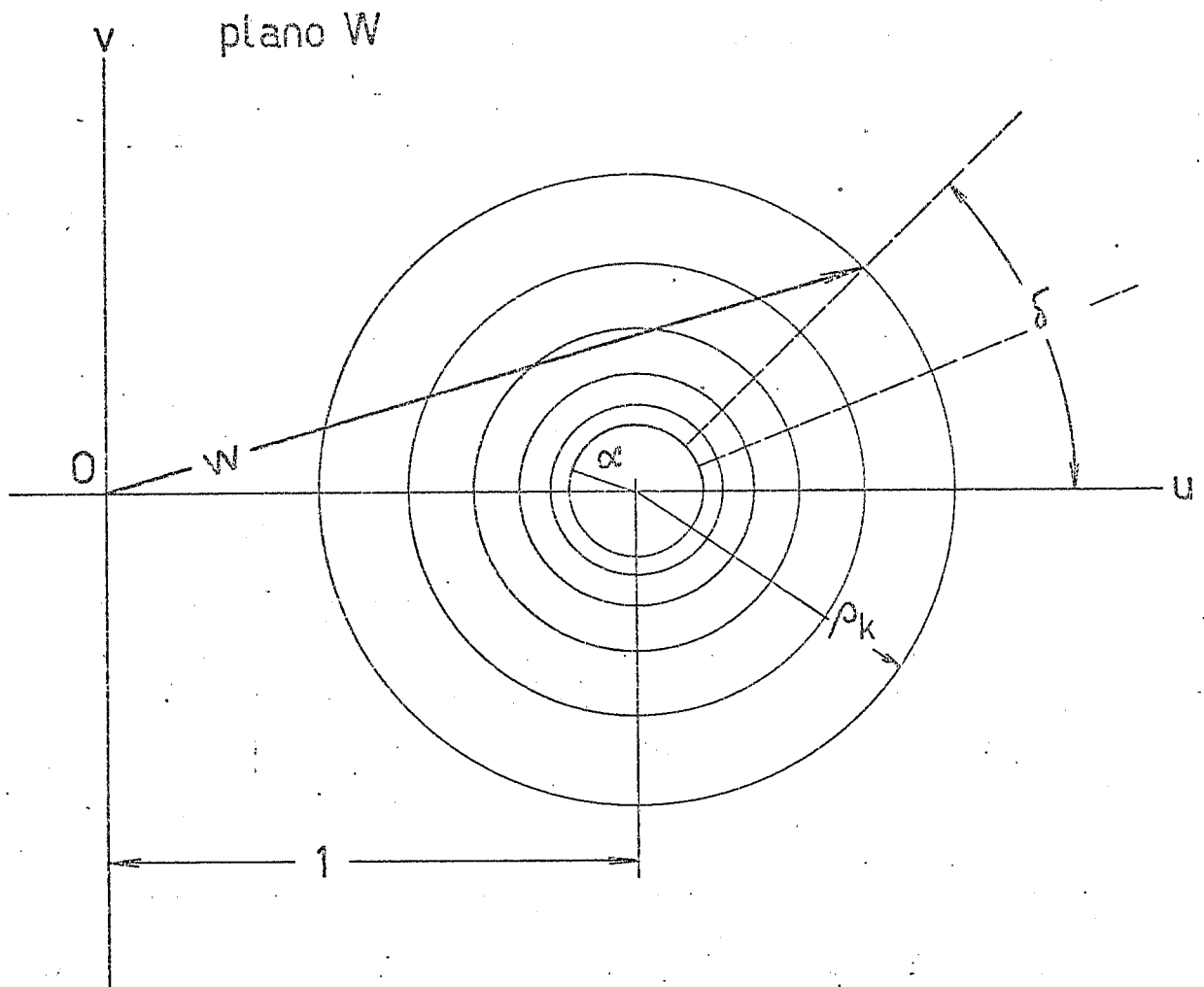


Figura B.1 - Condutor localizado no ponto $(1,0)$, no plano com plexo W .

Aplicando-se a transformação $w = z^N$, tem-se:

$$|z| = \sqrt[N]{|w|} = \sqrt[2N]{\rho^2 + 2 \cdot \rho \cdot \cos \delta + 1}$$

$$\arg z = (\arg w)/(N) = (1/N) \cdot \text{Tan}^{-1} \left[\frac{\rho \cdot \text{sen } \delta}{1 + \rho \cdot \cos \delta} \right].$$

Portanto, em termos de coordenadas retangulares no plano Z, tem-se

$$x = |z| \cdot \cos(\arg z),$$

e

$$y = |z| \cdot \text{sen}(\arg z).$$

Devido à simetria existente em tórno do eixo real u, no plano W, as interseções possíveis podem ser efetuadas somente no semi-plano superior determinado por valores de $v \geq 0$, enquanto u assume qualquer valor real.

Para a obtenção dos pontos de coordenadas x e y, que representam interseções de curvas equipotenciais com linhas de força, elaboraram-se os Programas P-1 e P-2.

Estes programas são apresentados em detalhe no ítem (B.2).

B.2 - PROGRAMAS P-1 e P-2

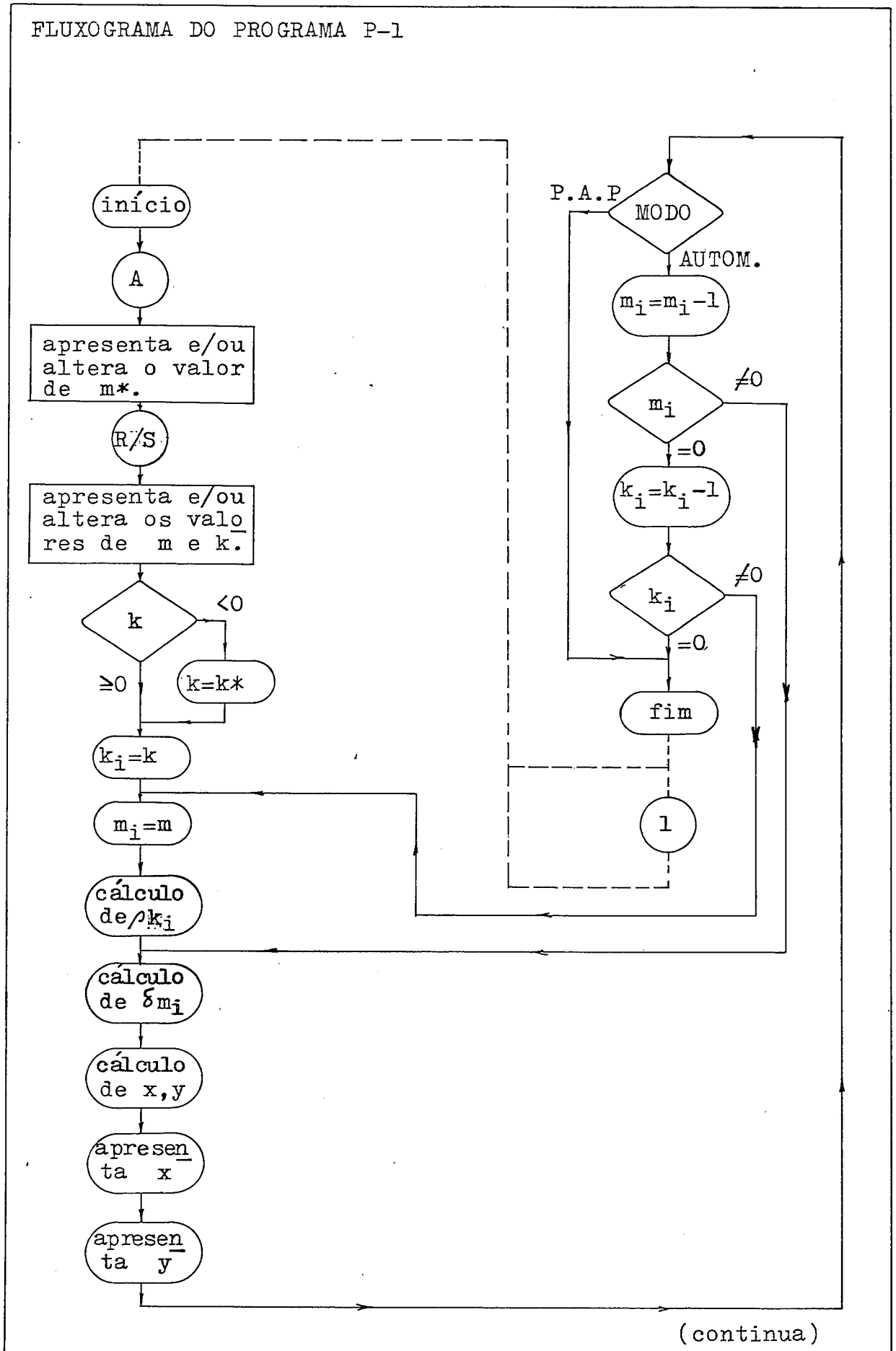
Para a determinação das interseções de linhas de força com curvas equipotenciais relativas a um feixe de N condutores de uma dada fase de uma linha de transmissão, foram elaborados os programas P-1 e P-2.

Para que os mesmos sejam utilizados convenientemente, são apresentados no ítem B.2.1 os respectivos fluxogramas, no ítem B.2.2, a descrição do modo de operação de cada um deles e, no ítem B.2.3, as respectivas listagens.

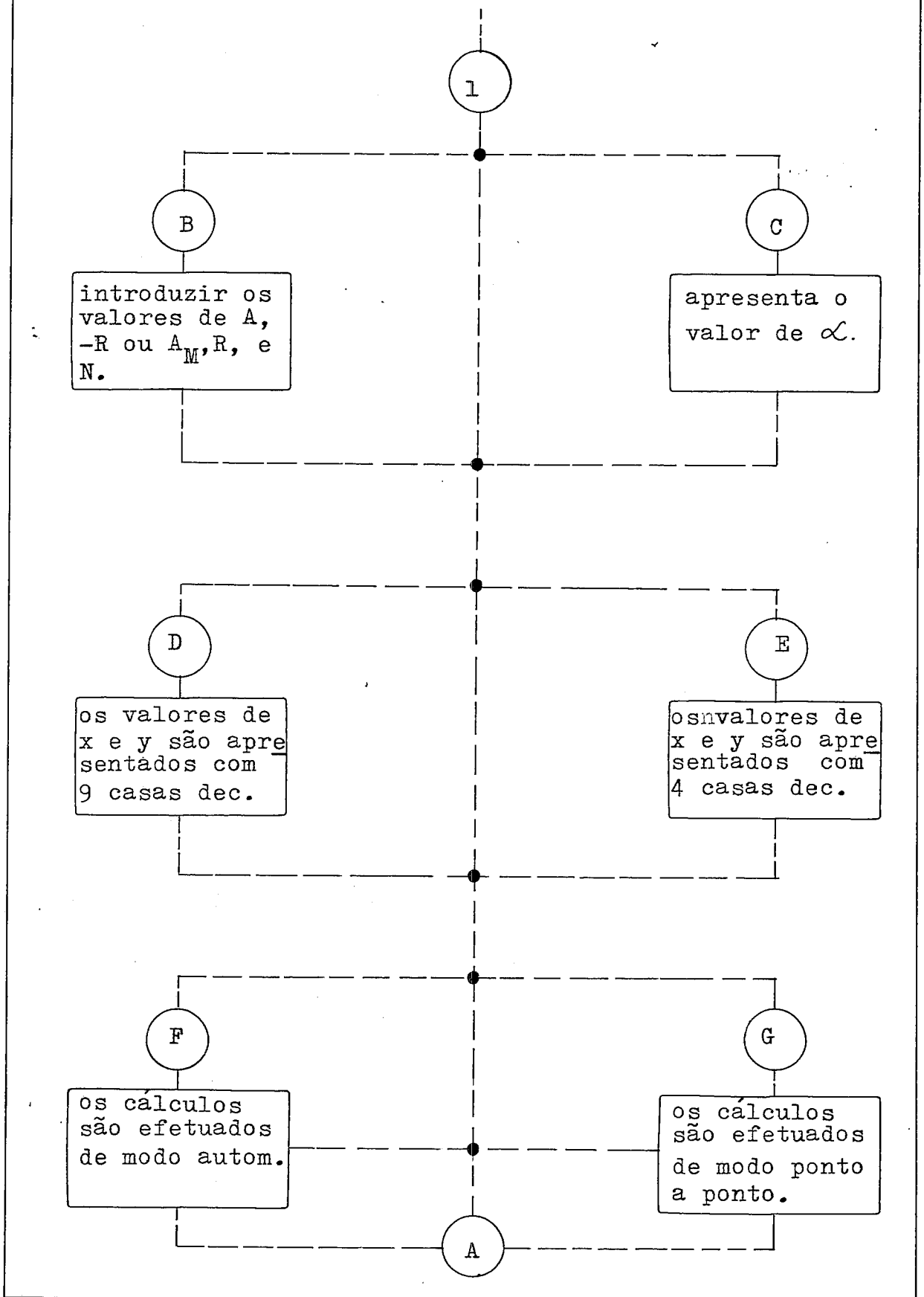
Estes dois programas foram implantados na calculadora HP-41C, empregando-se cartões magnéticos, e um caso teste para cada programa é apresentado no ítem B.2.4.

B.2.1 - Fluxogramas dos Programas P-1 e P-2

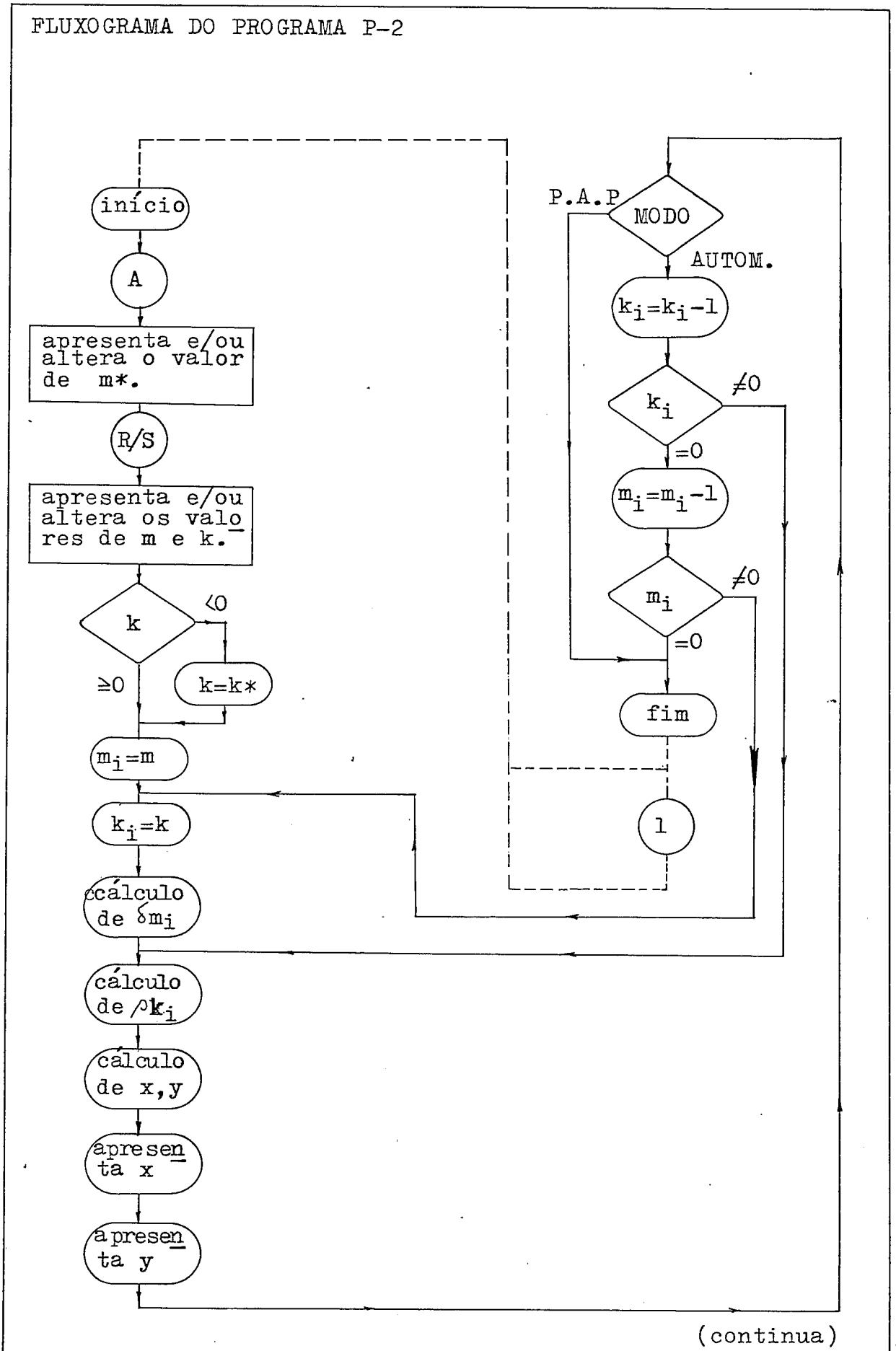
B.2.1.1 - Fluxograma do Programa P-1



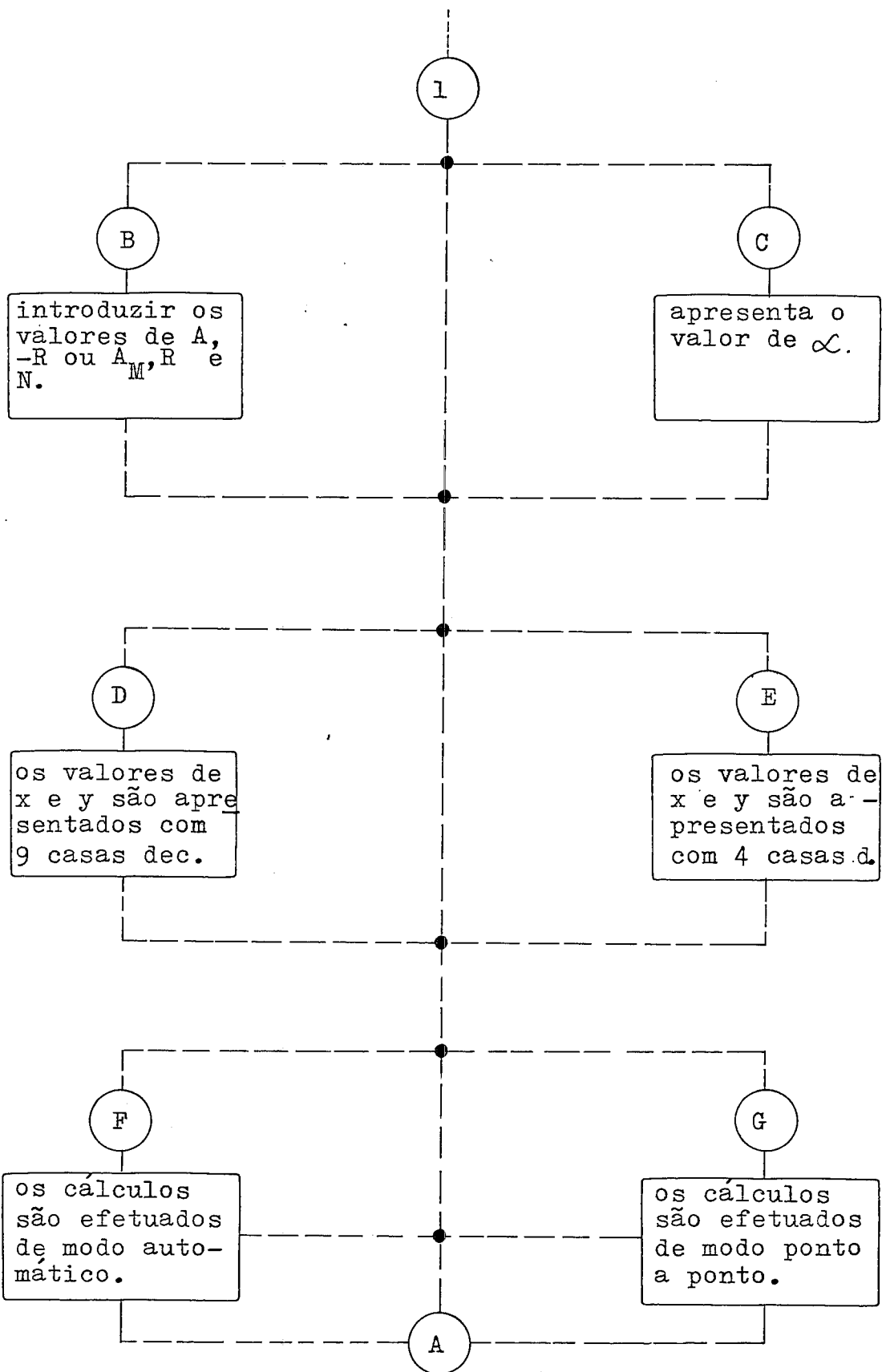
FLUXOGRAMA DO PROGRAMA P-1 (continuação)



B.2.1.2 - Fluxograma do Programa P-2



FLUXOGRAMA DO PROGRAMA P-2 (continuação)



B.2.2 - Modo de Utilização dos Programas P-1 e P-2

No ítem B.2.2.1 descreve-se o modo de utilização do programa P-1, e no ítem B.2.2.2, o modo de utilização do Programa P-2.

B.2.2.1 - Modo de Utilização do Programa P-1

Quando o programa é utilizado pela primeira vez, deve-se introduzir na calculadora, através do teclado, as instruções indicadas sequencialmente no ítem B.2.3, relativas à listagem do Programa P-1. No caso de se dispor da leitora de cartões magnéticos, estas instruções poderão ser gravadas nestes cartões, permitindo assim guardar indefinidamente o programa para utilização posterior, mesmo apagando-se as instruções a ele relativas, da memória da calculadora. Em seguida, deverão ser executadas as instruções relativas à Etapa A.

A Etapa A com as respectivas instruções, é a seguir apresentada:

ETAPA A - Nesta etapa pretende-se calcular as coordenadas x e y , relativas às interseções de curvas equipotenciais com linhas de força, ao longo de curvas equipotenciais, devidas aos N condutores de um feixe, e por fase de uma linha de transmissão, nas vizinhanças de um destes N condutores, devido à simetria existente nestas condições.

Esta etapa consiste do seguinte conjunto de procedimentos:

A.1 - Pressiona-se a tecla B, sendo apresentado no visor (A, -R/AM, R e N), devendo-se introduzir sequencialmente uma das duas seguintes combinações de dados:

(1) A, -R e N;

(2) AM, R e N.

A.1.1 - No caso dos dados serem como em (1), procede-se do seguinte modo:

- introduz-se o valor de A através do teclado, pressionando-se a seguir a tecla ENTER;
- introduz-se o valor de (-R) através do teclado, pressionando-se a seguir a tecla ENTER;
- introduz-se finalmente o valor de N, pressionando-se a seguir a tecla R/S.

O programa executa então uma série de instruções, apresentando no visor o valor de α , da seguinte forma: (ALFA = n n).

A.1.2 - No caso dos dados de entrada serem como em (2), o procedimento é análogo ao mostrado em A.1.1, introduzindo-se porém, o valor de A_M ao invés do de A , o valor de R , ao invés de $(-R)$, e finalmente, o valor de N .

Os valores de A , A_M , R e $(-R)$ devem ser introduzidos nas mesmas unidades de comprimento.

A.2 - A escolha do número de casas decimais, quatro ou nove, apresentadas para os valores de x , y e k^* , é determinada apertando-se respectivamente as teclas E ou D.

Apertando-se a tecla E, é apresentado no visor (FIX = 4) e, no caso da tecla D, (FIX = 9).

A.3 - Os cálculos das interseções poderão ser efetuados de dois modos:

- modo automático;
- modo ponto a ponto.

A.3.1 - Para que os cálculos sejam efetuados no modo automático, deverá ser pressionada a tecla F, sendo apresentado no visor (MOD. AUT.).

Neste caso, para os valores de M^* , M e K introduzidos (vide procedimento A.4), os valores de x e y , correspondentes às interseções, são calculados continuamente, até que todas as interseções de linhas de força com curvas equipotenciais tenham sido efetuadas.

A.3.2 - Para que os cálculos sejam efetuados no modo ponto a ponto, deverá ser pressionada a tecla G , sendo apresentado no visor (MOD.P A P). Neste caso, o programa calcula apenas uma interseção, função dos valores de M^* , M e K introduzidos (vide procedimento A.4).

A.4 - Pressionando-se a tecla A , é apresentado no visor ($M^* = n \dots n M K$), onde M indica o número de "quadrados curvilíneos" considerados para o cálculo das interseções no semi-plano superior do plano complexo W , devido à simetria existente nestas condições, e guardado na memória da calculadora, M refere-se ao número de linhas de força consideradas, e K relaciona-se ao número de curvas equipotenciais consideradas.

Os valores de M^* , M e K são números inteiros não negativos, havendo somente restrição ao valor de M , o qual não poderá ser maior do que aquele atribuído a M^* .

A introdução dos valores de M e K é obrigatória para os cálculos que serão efetuados, enquanto o valor de M^* poderá ou não ser alterado, conforme descrito respectivamente em A.4.1 e A.4.2.

- A.4.1 - Neste caso, o valor de M^* será alterado pela introdução de novo valor através do teclado, pressionando-se a seguir a tecla ENTER. A seguir, deverá ser introduzido o valor de M, pressionando-se após, a tecla ENTER. O valor de K deverá ser introduzido através do teclado, pressionando-se a seguir a tecla R/S.
- A.4.2 - Neste caso, o valor de M^* será mantido. Portanto, analogamente ao procedimento A.4.1, deverão ser executados somente os procedimentos relativos a M e K.
- A.4.3 - Se ao invés de se introduzir o valor de K nos procedimentos A.4.1 ou A.4.2, se introduz um valor negativo, os cálculos são efetuados considerando somente a interseção das linhas de força com a equipotencial de ordem K^* , ao longo desta equipotencial, que passa pelo centro geométrico do feixe de N condutores.

A seguir, uma vez efetuados os procedimentos A.1 a A.4.3, é apresentado no visor $(M^* = n \dots n M K)$, ou seja, o valor de M^* considerado para os cálculos a serem efetuados.

Para cada interseção operada, são apresentados no visor os valores de M e K relativos à interseção da linha de força de ordem m_i , com a equipotencial de ordem k_i .

Os valores de x e y relativos às coordenadas de cada interseção são mostrados no visor.

Desde que os cálculos são efetuados para interseções de k curvas equipotenciais com m linhas de força, ao longo de k curvas equipotenciais, para cada valor de k_i , correspondentes a uma dada equipotencial k_i , desde $k_i = k$ até $k_i = 0$, $k_i = k, k-1, \dots, 0$ para $i = k, k-1, \dots, 0$, respectivamente, serão calculadas as coordenadas de interseção com as linhas de força de ordem m_i , $m_i = m, m-1, \dots, 0$ para $i = m, m-1, \dots, 0$, onde m_i deverá variar desde $m_i = m$ até $m_i = 0$.

Após o cálculo da interseção desejada, estando o programa no modo ponto a ponto, ou de todas as interseções, estando o programa no

modo automático, o visor apresenta novamente ($M^* = n \dots n M K$), requisitando novos valores para estas variáveis.

B.2.2.2 - Modo de Utilização do Programa P-2

O modo de utilização do Programa P-2 é análogo ao descrito para o Programa P-1, no item B.2.2.1, sendo que as interseções das k curvas equipotenciais com as m linhas de força são efetuadas ao longo de cada linha de força. Neste caso, conforme descrito em A.4.2, para cada linha de força de ordem m_i são calculadas todas as interseções com as equipotenciais de ordem k_i , m_i e k_i variando respectivamente desde $m_i = m$ até $m_i = 0$, e $k_i = k$, até $k_i = 0$.

B.2.3 - Listagens dos Programas P-1 e P-2

B.2.3.1 - Listagem do Programa P-1

1	LBL 'P1	RCL 06	1/x	RCL 04	STO:03
	LBL A	:	y ^x	y ^x	163 END
	T80	CHS	P-R	XEQ 03	
	STO 01	e ^x	FIX 4	LBL C	
	CF 29	STO 09	FS?02	RCL 03	
	CF 01	RCL 03	FIX 9	1/x	
	FIX 0	STO:09	'X=	'ALFA=	
	' M K	LBL 06	XEQ 00	FIX 6	
	ASTO X	FIX 0	PSE	ARCL X	
10	'M*=	50 ' K=	90 X ≤ Y	130 AVIEW	
	RCL 02	FS?01	'Y=	RTN	
	ARCL X	' K*=	XEQ 00	LBL D	
	ARCL Y	ASTO 00	FC?00	SF 02	
	PROMPT	RCL 07	GTO A	'FIX=9	
	STO 08	CHS	ISG 07	AVIEW	
	R↓	'M=	GTO 06	RTN	
	STO 05	ARCL X	FS?01	LBL E	
	R↓	ARCL 00	GTO A	CF 02	
	STO 02	RCL 08	ISG 08	'FIX=4	
20	'M*=	60 CHS	100 GTO 05	140 AVIEW	
	ARCL X	FC?01	GTO A	RTN	
	AVIEW	GTO 01	LBL B	LBL F	
	π	FIX 4	'A,-R/AM,	SF 00	
	:	FS?02	R e N	'MOD. AUT	
	STO 06	FIX 9	PROMPT	AVIEW	
	RCL 03	LBL 01	STO 04	RTN	
	LN	ARCL X	R↓	LBL G	
	X	AVIEW	X > 0?	CF 00	
	RCL 08	X ≤ Y	GTO 08	'MOD. P_A_P	
30	X < 0?	RCL 02	ABS	150 AVIEW	
	SF 01	:	110 RCL 01	RTN	
	X < 0?	RCL 01	RCL 04	LBL 00	
	X ≤ Y	X	:	ARCL X	
	CHS	RCL 09	SIN	AVIEW	
	STO 08	P-R	2	PSE	
	LBL 05	1	X	RTN	
	RCL 05	+	X	LBL 03	
	CHS	R-P	LBL 08	STO 03	
	STO 07	RCL 04	:	1	
40	RCL 08	80 STO:Z	XEQ 03	160 STO+03	
			120 RCL 03	-	

B.2.3.2 - Listagem do Programa P-2

1	LBL 'P2	RCL 02	y^x	y^x
	LBL A	:	P-R	XEQ 03
	180	RCL 01	FIX 4	LBL C
	STO 01	X	FS?02	RCL 03
	CF 29	CHS	FIX 9	1/x
	CF 01	STO 08	'X=	'ALFA=
	FIX 0	LBL 06	XEQ 00	FIX 6
	' M K	FIX 0	PSE	ARCL X
	ASTO X	' K=	$X \geq Y$	AVIEW
10	'M*=	50 FS?01	90 'Y=	130 RTN
	RCL 02	' K*=	XEQ 00	LBL D
	ARCL X	ASTO 00	FC?00	SF 02
	ARCL Y	RCL 05	GTO A	'FIX=9
	PROMPT	CHS	FS?01	AVIEW
	STO 09	'M=	GTO A	RTN
	R↓	ARCL X	ISG 07	LBL E
	CHS	ARCL 00	GTO 06	CF 02
	STO 05	RCL 08	ISG 05	'FIX=4
	R↓	RCL 07	GTO 05	AVIEW
20	STO 02	60 CHS	100 GTO A	140 RTN
	'M*=	FC?01	LBL B	LBL F
	ARCL X	GTO 01	'A,-R/AM,	SF 00
	AVIEW	FIX 4	R e N	'MOD.AUT.
	π	FS?02	PROMPT	AVIEW
	:	FIX 9	STO 04	RTN
	STO 06	LBL 01	R↓	LBL G
	RCL 03	ARCL X	$X > 0?$	CF 00
	LN	AVIEW	GTO 08	'MOD. P A P
	X	RCL 06	ABS	AVIEW
30	RCL 09	70 :	RCL 01	150 RTN
	$X < 0?$	e ^x	110 RCL 04	LBL 00
	SF 01	RCL 03	:	ARCL X
	$X < 0?$:	SIN	AVIEW
	$X \geq Y$	P-R	2	PSE
	CHS	1	X	RTN
	STO 09	+	X	LBL 03
	LBL 05	R-P	LBL 08	STO 03
	RCL 09	RCL 04	:	1
	STO 07	STO:Z	XEQ 03	STO+03
40	RCL 05	80 1/x	RCL 03	160 -
			120 RCL 04	STO:03
				END

B.2.4 - Casos Testes para os Programas P-1 e P-2

No ítem B.2.4.1 apresenta-se um caso teste para o Programa P-1, e, em B.2.4.2, um caso teste para o Programa P-2.

B.2.4.1 - Caso Teste para o Programa P-1

Considere-se como exemplo para a Etapa A, os seguintes dados:

$$A = 10 \text{ cm};$$

$$R = 1,5 \text{ cm};$$

$$N = 2;$$

$$M^* = 10;$$

$$M = 10;$$

$$K = 1;$$

Deseja-se ainda:

- operar no modo automático;
- resultados apresentados com quatro casas decimais.

Observa-se a seguinte sequência de instruções:

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA A
B	A, -R/AM, R e N	A.1
10	10	A.1.1
ENTER	10	A.1.1
-1,5	-1,5	A.1.1
ENTER	-1,5	A.1.1
2	2	A.1.1
R/S		A.1.1
	ALFA = 0,550459	A.1.1
E	FIX = 4	A.2
F	MOD. AUT.	A.3.1
A	M* = 0 M K	A.4
10	10	A.4.1
ENTER	10	A.4.1
10	10	A.4.1
ENTER	10	A.4.1
1	1	A.4.1
R/S		A.4.1

O quadro a seguir resume os resultados conforme apresentados no visor:

$M^* = 10$	$M = 3 \quad K = 1$	$X = 0,8621$	$Y = 0,0000$
$M = 10 \quad K = 1$	$X = 1,2267$	$Y = 0,2583$	$M^* = 10 \quad M \quad K$
$X = 0,4963$	$Y = 0,2485$	$M = 6 \quad K = 0$	
$Y = 0,0000$	$M = 2 \quad K = 1$	$X = 0,9516$	
$M = 9 \quad K = 1$	$X = 1,2805$	$Y = 0,2751$	
$X = 0,5701$	$Y = 0,1730$	$M = 5 \quad K = 0$	
$Y = 0,2043$	$M = 1 \quad K = 1$	$X = 1,0348$	
$M = 8 \quad K = 1$	$X = 1,3132$	$Y = 0,2660$	
$X = 0,7002$	$Y = 0,0887$	$M = 4 \quad K = 0$	
$Y = 0,3163$	$M = 0 \quad K = 1$	$X = 1,1072$	
$M = 7 \quad K = 1$	$X = 1,3242$	$Y = 0,2364$	
$X = 0,8315$	$Y = 0,0000$	$M = 3 \quad K = 0$	
$Y = 0,3666$	$M = 10 \quad K = 0$	$X = 1,1662$	
$M = 6 \quad K = 1$	$X = 0,6705$	$Y = 0,1909$	
$X = 0,9531$	$Y = 0,0000$	$M = 2 \quad K = 0$	
$Y = 0,3760$	$M = 9 \quad K = 0$	$X = 1,2096$	
$M = 5 \quad K = 1$	$X = 0,7009$	$Y = 0,1337$	
$X = 1,0612$	$Y = 0,1214$	$M = 1 \quad K = 0$	
$Y = 0,3551$	$M = 8 \quad K = 0$	$X = 1,2362$	
$M = 4 \quad K = 1$	$X = 0,7736$	$Y = 0,0688$	
$X = 1,1530$	$Y = 0,2091$	$M = 0 \quad K = 0$	
$Y = 0,3108$	$M = 7 \quad K = 0$	$X = 1,2452$	

Desejando-se obter as interseções das linhas de força desde $m_i = 5$ até $m_i = 0$, com a equipotencial K^* , que passa pela origem, observa-se a seguinte sequência de instruções, após o final do cálculo anterior:

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA A
5	5	A.4.1
ENTER	5	A.4.1
-1	-1	A.4.3
R/S		A.4.3

O quadro a seguir resume os resultados conforme apresentados no visor:

$M^* = 10$
$M = 5 \quad K^* = 1,9003$
$X = 1,0987$
$Y = 0,4551$
$M = 4 \quad K^* = 1,9003$
$X = 1,2098$
$Y = 0,3931$
$M = 3 \quad K^* = 1,9003$
$X = 1,2980$
$Y = 0,3116$
$M = 2 \quad K^* = 1,9003$
$X = 1,3622$
$Y = 0,2157$
$M = 1 \quad K^* = 1,9003$
$X = 1,4011$
$Y = 0,1103$
$M = 0 \quad K^* = 1,9003$
$X = 1,4142$
$Y = 0,0000$
$M^* = 10 \quad M \quad K$

B.2.4.2 - Caso Teste para o Programa P-2

Considere-se como exemplo para a Etapa A, os seguintes dados:

$$A = 10 \text{ cm};$$

$$R = 1,5 \text{ cm};$$

$$N = 2;$$

$$M^* = 10;$$

$$M = 3;$$

$$K = 1.$$

Deseja-se ainda:

- operar no modo automático;
- resultados apresentados com quatro casas decimais.

Observa-se a seguinte sequência de instruções:

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA A
B	A, -R/AM, R e N	A.1
10	10	A.1.1
ENTER	10	A.1.1
-1,5	-1,5	A.1.1
ENTER	-1,5	A.1.1
2	2	A.1.1
R/S		A.1.1
	ALFA = 0,550459	A.1.1
E	FIX = 4	A.2
F	MOD. AUT.	A.3.1
A	M* = 0 M K	A.4
10	10	A.4.1
ENTER	10	A.4.1
3	3	A.4.1
ENTER	3	A.4.1
1	1	A.4.1
R/S		A.4.1

O quadro a seguir resume os resultados conforme apresentados no visor:

```

M* = 10
M = 3 K = 1
X = 1,2267
Y = 0,2485
M = 3 K = 0
X = 1,1662
Y = 0,1909
M = 2 K = 1
X = 1,2805
Y = 0,1730
M = 2 K = 0
X = 1,2096
Y = 0,1337
M = 1 K = 1
X = 1,3132
Y = 0,0887
M = 1 K = 0
X = 1,2362
Y = 0,0688
M = 0 K = 1
X = 1,3242
Y = 0,0000
M = 0 K = 0
X = 1,2452
Y = 0,0000
M* = 10 M K

```

Desejando-se obter a interseção da linha de força $m_i = 2$, com a equipotencial K^* , que passa pela origem, observa-se a seguinte sequência de instruções, após o final do cálculo anterior:

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA A
2	2	A.4.1
ENTER	2	A.4.1
-1	-1	A.4.3
R/S		A.4.3

O quadro seguinte resume os resultados conforme apresentados no visor:

$M^* = 10$ $M = 2 \quad K^* = 1,9003$ $X = 1,3622$ $Y = 0,2157$ $M^* = 10 \quad M \quad K$
--

O tempo de execução dos Programas P-1 e P-2 é de cerca de 3 segundos para cada cálculo de interseção, sendo os resultados apresentados no visor durante cerca de 5 segundos.

O tempo de apresentação dos resultados poderá ser alterado, bastando a introdução conveniente de instruções nestes dois programas.

APÊNDICE CRESOLUÇÃO DA EQUAÇÃO TRANSCENDENTAL

$$\underline{R_e/R = \gamma \cdot (Q_e/Q_B)}$$

Para a determinação do valor de R_e , é necessário resolver-se a equação acima. A resolução desta equação é efetuada através do Programa P-3, implantado na máquina de calcular HP-41C, utilizando o método de Newton.

No ítem C.1 deste apêndice, é apresentado o desenvolvimento teórico, relativo às equações empregadas, e o método de resolução da equação transcendental, e, no ítem C.2, é apresentado o Programa P-3, incluindo seu modo de utilização e um caso teste.

C.1 - DESENVOLVIMENTO TEÓRICO

Na equação transcendental

$$R_e/R = \gamma \cdot (Q_e/Q_B) , \quad |C.1|$$

cujas variáveis R_e , R e γ já foram definidas nos capítulos II, III e IV, as cargas Q_e e Q_B são dadas por

$$Q_i = 2\pi\epsilon \cdot \frac{\left[\ln \frac{H_1 \cdot \sqrt{1+(1/\lambda)^2} \cdot \sqrt{1+1/(2\lambda)^2}}{R_i} \right]}{\left[\ln \frac{H_1 \sqrt{1+(1/\lambda)^2}}{R_i} \cdot \ln \frac{H_1}{R_i \sqrt{1+(1/\lambda)^2}} + \ln \frac{H_1}{R_i} \cdot \ln \sqrt{1+1/(2\lambda)^2} - \ln^2 \sqrt{1+(1/\lambda)^2} \right]} \cdot V_{\phi T}, \quad |C.2|$$

onde i é o índice relativo a Q_e ou Q_B , conforme se substitua R_i por R_e ou R_c , respectivamente.

A Eq. |C.2| pode ser simplificada, utilizando-se as Eqs. |4.16|, |4.17| e |4.18|, obtendo-se assim a equação:

$$Q_i = 2\pi\epsilon \cdot \left[\frac{\ln(H_1/R_i) + \beta' + \beta''}{\ln^2(H_1/R_i) + \beta'' \cdot \ln(H_1/R_i) - 2 \cdot \beta'^2} \right] \cdot V_{\phi T}.$$

Fazendo-se

$$d = \beta' + \beta'',$$

$$a = 2 \cdot \beta'^2,$$

e

$$y = \ln(R_i/H_1), \quad |C.3|$$

obtem-se

$$Q_i = 2\pi\epsilon \cdot \left[\frac{(d-y)}{y^2 - (\beta'' \cdot y) - a} \right] \cdot V_{\phi T}.$$

Manipulando-se a Eq. |C.1|, tem-se:

$$R_e/R = (R_e/R) \cdot (H_1/H_1) = (R_e/H_1) \cdot (H_1/R) = \gamma \cdot (Q_e/Q_B),$$

ou

$$R_e/H_1 = \left(\frac{\gamma \cdot R}{H_1 \cdot Q_B} \right) \cdot Q_e . \quad |C.4|$$

Seja

$$A' = \frac{\gamma \cdot R}{H_1 \cdot Q_B / (2\pi\epsilon V \phi_T)} \quad |C.5|$$

Da relação |C.3|, tem-se:

$$R_e/H_1 = e^y .$$

A partir das relações |C.4| e |C.5|, obtêm-se

$$e^y = \left[\frac{A' \cdot (d-y)}{y^2 - \beta'' \cdot y - a} \right] ,$$

ou, ainda,

$$y = \ln \left[\frac{A' \cdot (d-y)}{y^2 - \beta'' \cdot y - a} \right] , \quad |C.6|$$

Para a resolução da Eq. |C.6|, aplica-se o método de Newton, de modo a se obter uma solução que satisfaça uma dada precisão, através de iterações sucessivas, de acordo com a relação:

$$y_{i+1} = y_i - \left[\frac{f(y_i)}{f'(y_i)} \right] = g(y_i), \quad i = 0, 1, 2, \dots,$$

onde a função $f(y_i)$ é dada por

$$f(y_i) = y_i + \ln \left[\frac{y_i^2 - \beta'' \cdot y_i - a}{A' \cdot (d - y_i)} \right]$$

e,

$$f'(y_i) = 1 + \left[\frac{(2 \cdot y_i) - \beta''}{y_i^2 - (\beta'' \cdot y_i) - a} \right] + \frac{1}{(d - y_i)}.$$

Uma estimativa do valor de y_0 pode ser obtida através dos valores de R_e e H_1 , na Eq. |C.3|. Como $H_1 \gg R_e$, para os casos estudados usualmente, o quociente $(R_e/H_1) \ll 1$, indicando que y assume valores negativos.

Supondo a relação $(R_e/H_1) \cong 10^{-3}$, então,
 $\ln(R_e/H_1) \cong -6,7$.

No Programa P-3, y_0 é escolhido igual a -9,0.

C.2 - PROGRAMA P-3

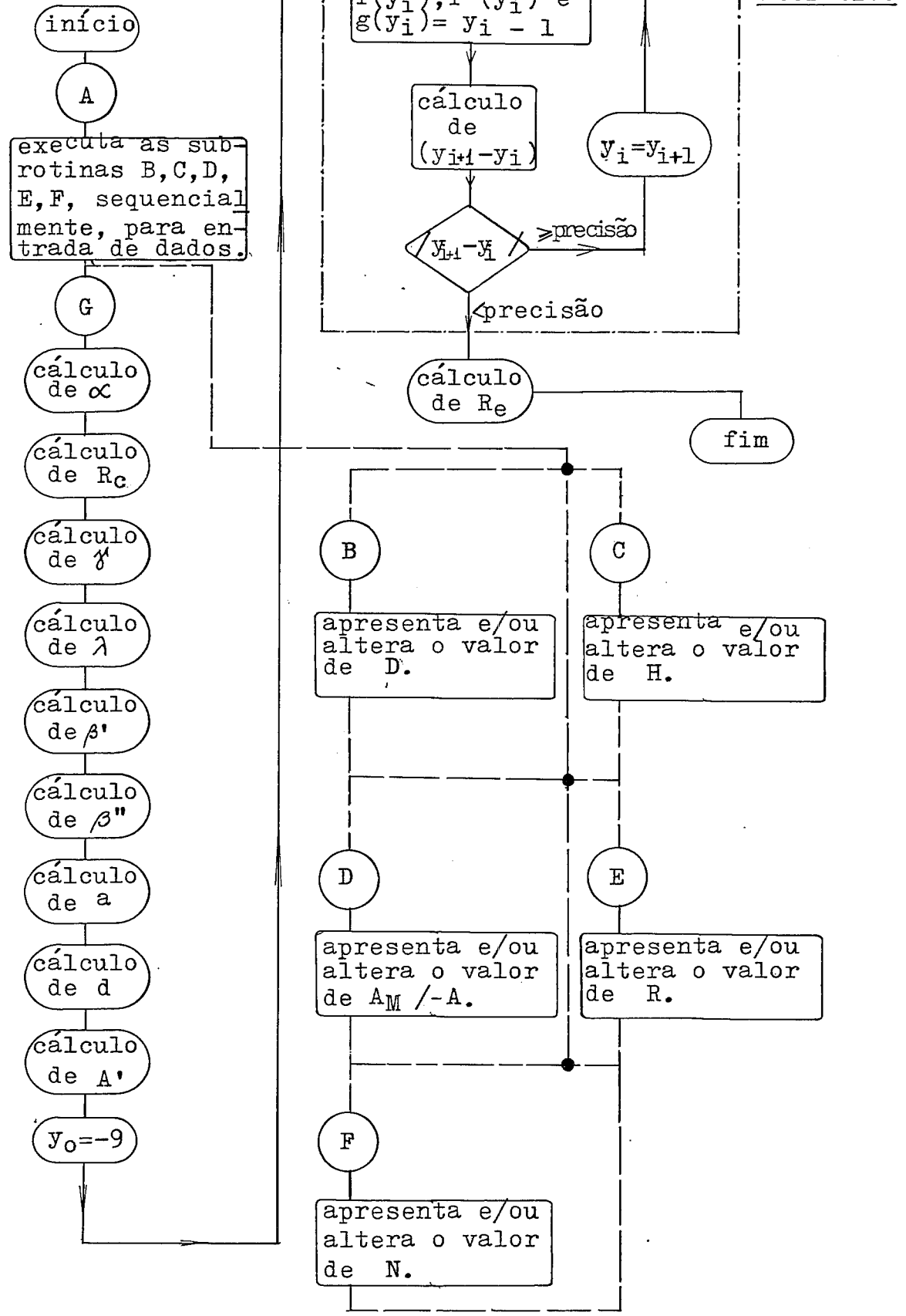
Para a resolução da Eq. [C.1], conforme o método iterativo proposto no ítem C.1, elaborou-se o Programa P-3.

Para utilizá-lo convenientemente, é apresentado no ítem C.2.1 o fluxograma, no ítem C.2.2, a descrição do modo de operação e no ítem C.2.3, a respectiva listagem.

Este programa foi implantado na calculadora HP-41C, empregando-se cartões magnéticos, e um caso teste é apresentado no ítem C.2.4.

C.2.1 - Fluxograma do Programa P-3

FLUXOGRAMA DO
PROGRAMA P-3.



C.2.2 - Modo de Utilização do Programa P-3

É apresentado a seguir, por etapas, o modo de utilização do Programa P-3.

Quando o programa é utilizado pela primeira vez, deve-se introduzir na calculadora, através do teclado, as instruções indicadas sequencialmente no item C.2.3, relativas à listagem do Programa P-3. No caso de se dispor da leitora de cartões magnéticos, estas instruções poderão ser gravadas nestes cartões, permitindo assim guardar indefinidamente o programa para utilização posterior, mesmo apagando-se as instruções a ele relativas, da memória da calculadora. Em seguida, deverão ser executadas as instruções relativas à Etapa A.

Desde que a Etapa A tenha sido utilizada uma única vez, qualquer das Etapas, em número de duas, pode ser executada.

As Etapas com as respectivas instruções, são as seguintes:

ETAPA A - Nesta etapa, pretende-se calcular pela primeira vez, o valor de R_e . Consiste do seguinte conjunto de procedimentos:

A.1 - Pressiona-se a tecla A, sendo apresentado no visor ($D =$), cujo valor deverá ser intro

duzido pelo teclado. Em seguida, a tecla R/S deverá ser pressionada.

A.2 - É apresentado no visor ($H =$), cujo valor deverá ser introduzido pelo teclado. Em seguida, a tecla R/S deverá ser pressionada.

A.3 - É apresentado no visor ($AM =$ / $-A =$) , um dos dois valores devendo ser introduzido pelo teclado. Introduzindo-se um valor positivo, o programa interpreta como sendo o valor correspondente a A_M ; no caso deste valor ser negativo, o programa interpreta-o como sendo o valor correspondente a $-A$. Em seguida, a tecla R/S deverá ser pressionada.

A.4 - É apresentado no visor ($R =$), cujo valor deverá ser introduzido pelo teclado. Em seguida, a tecla R/S deverá ser pressionada. A unidade usada no valor de R introduzido, será a do valor de R_e calculado pelo Programa P-3.

A.5 - É apresentado no visor ($N =$), cujo valor deverá ser introduzido pelo teclado. Em seguida, a tecla R/S, ou G, deverá ser pressionada, dando início ao cálculo indicado no fluxograma, para obtenção de R_e .

Nesta Etapa A, havendo um dado qualquer introduzido em posição trocada, a tecla A deverá ser apertada, repetindo-se a sequência de entrada de dados acima descrita.

Após o procedimento A.5, o programa executa uma série de instruções, indicando posteriormente valores relativos à convergência do processo iterativo, indicado no fluxograma.

Uma vez que o valor 0 (zero) tenha sido indicado no visor, ou seja, havendo convergência do processo iterativo, o valor de R_e é indicado. Caso o valor 0 (zero) não seja apresentado no visor, o valor de y_0 , indicado no fluxograma, deverá ser modificado convenientemente.

A unidade apresentada para R_e , depende de serem introduzidos os valores de D , H , A_M ou $-A$, e R , nas mesmas unidades. Por exemplo, caso estas quatro variáveis sejam dadas em metro, o valor de R_e também será calculado em metro.

ETAPA B - Neste etapa possibilita-se a apresentação de qualquer dos dados de entrada introduzidos para o cálculo de R_e , optando-se por alterá-los, ou não. Esta etapa consiste dos seguintes procedimentos, todos, ou parte deles, podendo ser executados em qualquer ordem:

- B.1 - Apertando-se a tecla B, apresenta-se no visor o valor de D existente na memória, na unidade previamente escolhida. Desejando-se alterar o valor de D, deve-se introduzir pelo teclado o novo valor, pressionando-se a seguir a tecla R/S. No caso de simples verificação do valor anterior, pode-se executar qualquer um dos procedimentos desta etapa e, em seguida, para calcular-se o novo valor de R_e , deve-se pressionar a tecla G.
- B.2 - Este procedimento é igual ao do item B.1, à exceção da variável indicada, que passa a ser H, quando se aperta a tecla C.
- B.3 - Este procedimento é igual ao descrito no item B.1, à exceção da variável indicada, que passa a ser A_M ou -A, quando se aperta a tecla D.
- B.4 - Este procedimento é igual ao descrito no item B.1, à exceção da variável indicada, que passa a ser R, quando se aperta a tecla E.
- B.5 - Este procedimento é igual ao descrito no item B.1, à exceção da variável indicada, que passa a ser N, quando se aperta a tecla F.

C.2.3.2 - Listagem do Programa P-3

1	LBL 'P3	1/x	X	$X \geq Y$	LBL F
	LBL A	RCL 04	RCL 06	STO 00	'N=
	XEQ B	1/x	-	-	4
	XEQ C	y^x	RCL 00	FIX 7	GTO 02
	XEQ D	1	RCL 07	RND	LBL D
	XEQ E	$X \geq Y$	+	VIEW X	'AM=
	XEQ F	-	:	$X \neq 0?$	5
	LBL G	R↑	STOX 09	GTO 01	LBL 02
	'	X	-9	RCL 00	FIX 4
10	AVIEW	50 STO:09	90 STO 00	130 e^x	170 ABS
	RCL 05	RCL 02	LBL 01	RCL 02	RCL IND X
	$X > 0?$	STO:09	RCL 00	2	$X < 0?$
	GTO 08	2	RCL 08	X	'A=
	ABS	STO 06	-	X	ARCL X
	180	X	RCL X	'Re=	PROMPT
	RCL 04	STO 10	RCL 00	FIX 6	STO IND L
	:	RCL 00	STO+Z	ARCLX	177 END
	SIN	:	X	AVIEW	
	2	LN	RCL 06	FIX 0	
20	X	60 STO 00	100 -	140 RTN	
	:	RCL 01	STO:Y	LBL 03	
	LBL 08	STO:10	RCL 07	STO 11	
	STO 00	RCL 10	RCL 00	1	
	RCL 03	1	-	STO+11	
	STO 09	R-P	1/x	-	
	:	LN	STO+Z	STO:11	
	XEQ 03	STO 07	X	RCL 11	
	RCL 04	X^2	RCL 09	RTN	
	y^x	STOX06	:	LBL B	
30	XEQ 03	70 RCL 10	110 LN	150 'D=	
	RCL 04	2	RCL 00	1	
	1/x	:	-	GTO 02	
	CHS	1	$X \geq Y$	LBL C	
	y^x	R-P	1	'H=	
	STOX00	LN	+	2	
	RCL 11	STO+07	:	GTO 02	
	1	STO 08	CHS	LBL E	
	+	RCL 00	RCL 00	'R=	
	RCL 11	+	+	3	
40	XEQ 03	80 RCL 00	120 RCL 00	160 GTO 02	

C.2.4 - Caso Teste para o Programa P-3

Considere-se como exemplo para a Etapa A, os seguintes dados:

$D = 1500 \text{ cm} ;$

$H = 1500 \text{ cm} ;$

$A_M = 20 \text{ cm} ;$

$R = 1,5 \text{ cm} ;$

$N = 2 .$

Observa-se a seguinte sequência de instruções:

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ETAPA CORRESPONDENTE
A		A.1
	D =	A.1
1500	1500	A.1
R/S		A.1
	H =	A.2
1500	1500	A.2
R/S		A.2
	AM = /-A =	A.3
20	20	A.3
R/S		A.3
	R =	A.4
1,5	1,5	A.4
R/S		A.4
	N =	A.5
2	2	A.5
R/S ou G		A.5

O programa apresenta a seguinte s̄erie de valores de converḡencia:

INDICAÇÃO NO VISOR
-1,77 E0
-3,32 E-2
-1,64 E-5
0,00 E0

O valor de R_e ẽ a seguir apresentado no visor:

$R_e = 2,244018$ (cm) (coerente com os dados de entrada).

Como exemplo para a Etapa B, deseja-se o seguinte:

- observar o valor de H;
- alterar o valor de D, para 3000 cm;
- alterar o valor de N, para 3.

Observa-se a seguinte seqūencia de instruções:

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ETAPA CORRESPONDENTE
C	H = 1500,0000	B.2
B	D = 1500,0000	B.1
3000	3000	B.1
R/S		B.1
F	N = 2	B.5
3	3	B.5
R/S		B.5
G		B.1

Após serem apresentados os valores de convergência, como na Etapa A, o valor de R_e é indicado no visor:

$R_e = 3,081557$ (cm) (coerente com os dados de entrada).

O tempo de execução do Programa P-3 é de cerca de 18 segundos.

APÊNDICE DOBTENÇÃO DE CURVAS EQUIGRADIENTES PARA UM FEIXE
DE N CONDUTORES POR FASE DE UMA LINHA DE TRANSMISSÃO

Para se efetuar o traçado de curvas equigradien-
tes, torna-se necessário obter as interseções destas curvas,
com linhas de campo devidas ao feixe de N condutores considera-
dos, relativos a uma dada fase de uma linha de transmissão.

A determinação destas curvas é apresentada atra-
vés do desenvolvimento teórico, constante do item D.1 deste
apêndice, bem como das observações sobre alguns valores parti-
culares de σ , constantes do item D.2. Finalmente no item D.3
é apresentada a abordagem utilizada para o cálculo propriamen-
te dito das interseções, e empregado na elaboração dos Progra-
mas P-4 e P-5, ambos implantados na calculadora HP-41C.

D.1 - DESENVOLVIMENTO TEÓRICO

Considere-se no plano complexo W um dos N condu-
tores de um feixe, e de uma fase de uma linha de transmissão,
cujo centro esteja localizado no ponto de coordenadas (1,0).
As linhas de força devidas a este condutor, contidas no plano
W, transversal a ele, são retas que passam por (1,0), e fazem
um ângulo $\delta_{i,m} = (\pi/m).i$ radianos em relação ao eixo real u,
onde m e i são números inteiros positivos, i podendo ser igual
a zero, conforme mostrado na Fig. (D.1).

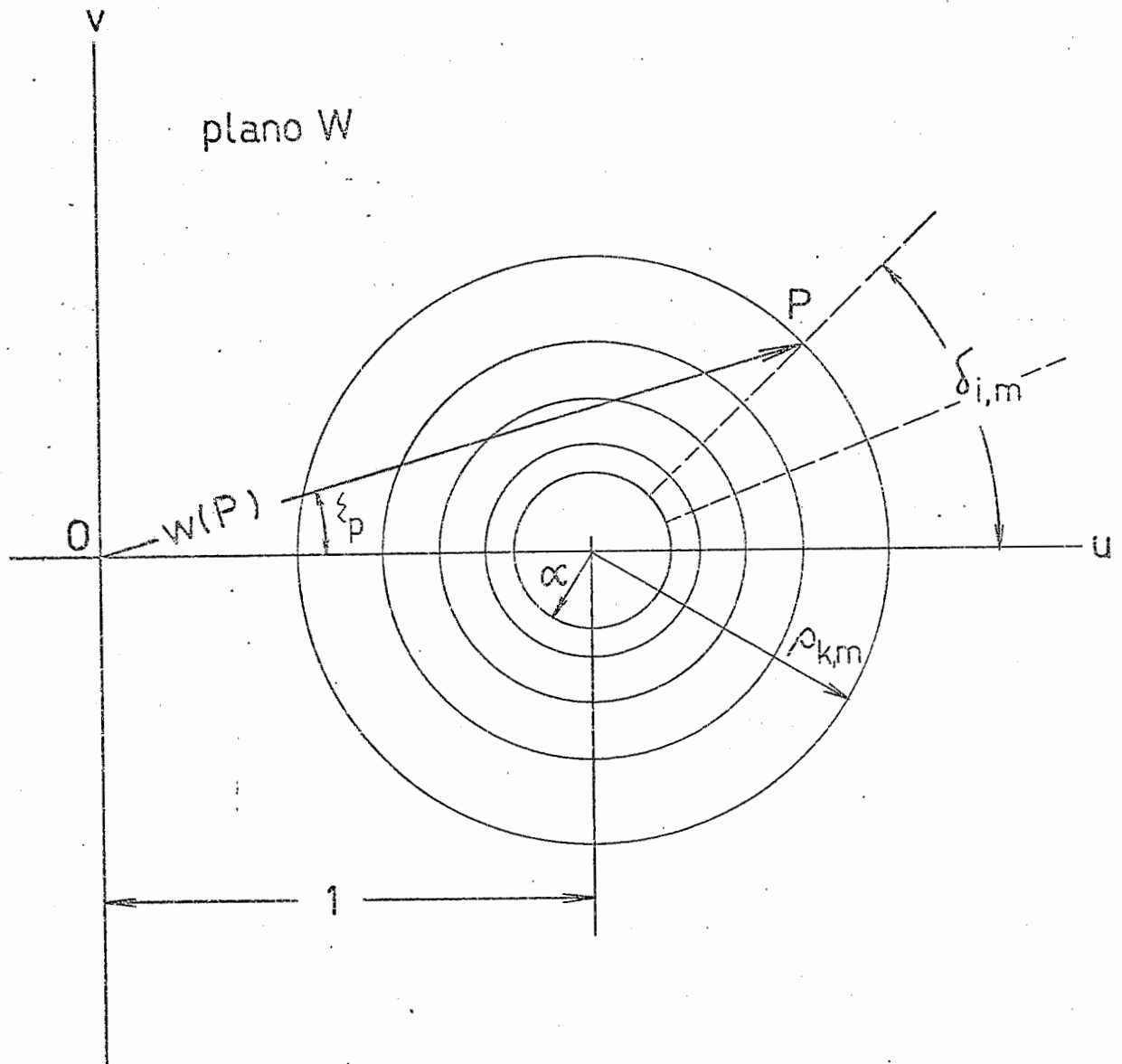


Figura D.1 - Conductor localizado no plano complexo W , com centro coincidente com o ponto $(1,0)$.

As curvas equipotenciais devidas a este condutor são dadas em função da relação $\rho_{k,m} = \alpha \cdot e^{(\pi/m) \cdot k}$, onde k é um número inteiro não negativo e α , é o raio do referido condutor. Desta relação, já referida no Apêndice A, como A.8, conclui-se que

$$\rho_{k,m} = \rho_{k-1,m} \cdot e^{(\pi/m)}.$$

A distância de um ponto P_0 , obtido pela interseção de uma curva equipotencial dada por $\rho_{k_0,m_0} = \alpha \cdot e^{(\pi/m_0) \cdot k_0}$, com uma linha de força dada por $\delta_{i_0,m_0} = (\pi/m_0) \cdot i_0$, em relação à origem (0), do referido plano W, é dada por

$$|w(P_0)| = \sqrt{1 + \rho_{k_0,m_0}^2 + 2 \cdot \rho_{k_0,m_0} \cdot \cos \delta_{i_0,m_0}}.$$

Considere-se associado ao ponto P_0 , o valor médio de campo elétrico $|\vec{E}(P_0)| = |\vec{E}_{k_0,i_0,m_0}|$, dado aproximadamente pela relação $(\Delta V/|\Delta W|)$, sendo ΔV , a diferença de potencial entre a curva equipotencial de ordem k_0 , e a equipotencial imediatamente seguinte, de ordem $(k_0 + 1)$, que intercepta a mesma linha de força δ_{i_0,m_0} , e define o ponto P, e $|\Delta W|$, da do por

$$|\Delta W| = |w(P) - w(P_0)| = \left| \sqrt{1 + \rho_{k_0+1, m_0}^2 + 2 \cdot \rho_{k_0+1, m_0} \cdot \cos(\delta_{i_0, m_0})} \right| \angle \xi_P - \\ - \left| \sqrt{1 + \rho_{k_0, m_0}^2 + 2 \cdot \rho_{k_0, m_0} \cdot \cos(\delta_{i_0, m_0})} \right| \angle \xi_{P_0} |,$$

onde

$$\xi_P = \text{Tan}^{-1} \left[\frac{\rho_{k_0+1, m_0} \cdot \text{sen}(\delta_{i_0, m_0})}{1 + \rho_{k_0+1, m_0} \cdot \cos(\delta_{i_0, m_0})} \right],$$

e

$$\xi_{P_0} = \text{Tan}^{-1} \left[\frac{\rho_{k_0, m_0} \cdot \text{sen}(\delta_{i_0, m_0})}{1 + \rho_{k_0, m_0} \cdot \cos(\delta_{i_0, m_0})} \right],$$

verificando-se que esta distância $|\Delta W|$, é tomada ao longo da linha de força δ_{i_0, m_0} .

Para obter-se o valor exato do campo elétrico em P_0 , necessita-se calcular o limite da relação $\Delta V/|\Delta W|$, quando $|\Delta W|$ tende a zero, ou seja, quando o ponto P tende ao ponto P_0 .

Assim, considere-se o ponto P_0 fixo, enquanto o número de quadrados curvilíneos é dado por $(\lambda \cdot m_0)$, sendo λ , um número inteiro positivo. Deste modo, as curvas equipotenciais são dadas pela expressão $\rho_{k, \lambda m_0} = \alpha \cdot e^{(\pi/\lambda m_0) \cdot k}$. A equipotencial ρ_{k_0, m_0} seria coincidente com a equipotencial $\rho_{\lambda \cdot k_0, \lambda m_0}$, ou seja, $\rho_{\lambda \cdot k_0, \lambda m_0} = \alpha \cdot e^{(\pi/\lambda \cdot m_0) \cdot (\lambda \cdot k_0)} = \alpha \cdot e^{(\pi/m_0) \cdot k_0}$ para qualquer valor de λ , ou seja, existe sempre uma equipoten

cial que passa pelo ponto P_0 , independente do valor de λ escolhido. A equipotencial de ordem imediatamente seguinte, é dada por

$$\rho_{\lambda k_0+1, \lambda m_0} = \alpha \cdot e^{(\frac{\pi}{\lambda \cdot m_0}) \cdot (\lambda \cdot k_0 + 1)}$$

ou

$$\rho_{\lambda k_0+1, \lambda m_0} = \alpha \cdot e^{[(\frac{\pi}{m_0}) k_0] (\frac{\pi}{\lambda \cdot m_0})}$$

ou

$$\rho_{\lambda k_0+1, \lambda m_0} = \rho_{k_0, m_0} \cdot e^{(\frac{\pi}{\lambda \cdot m_0})} = \rho_{\lambda k_0, \lambda m_0} \cdot e^{(\frac{\pi}{\lambda m_0})}$$

O ângulo associado a uma linha de força é dado por $\delta_{i, \lambda m_0} = (\pi/\lambda \cdot m_0) i$; no caso do ponto P_0 , a expressão adequada seria $\delta_{i, \lambda \cdot m_0} = \delta_{\lambda \cdot i_0, \lambda m_0} = (\pi/m_0) \cdot i_0 = \delta_{i_0, m_0}$, sendo independente do valor de λ .

Quando $\lambda \rightarrow +\infty$, o produto $(\lambda \cdot m_0) \rightarrow +\infty$, e, deste modo, $m \rightarrow +\infty$, fazendo-se ($m = \lambda \cdot m_0$). Do exposto acima, obtém-se a expressão de uma equipotencial que passa por P, ou seja, sendo imediatamente seguinte à equipotencial que passa por P_0 , quando o número de quadrados curvilíneos é igual a $(\lambda \cdot m_0)$.

Considere-se a seguinte relação a ser determinada no ponto genérico P_0 :

$$\sigma(P_0) = \frac{|\vec{E}(P_0)|}{|\vec{E}|_{\text{m\`{a}x}}}, \quad |D.1|$$

onde

$|\vec{E}(P_0)|$ é o valor exato do campo elétrico no ponto genérico P_0 , e

$|\vec{E}|_{\text{m\`{a}x}}$ é o valor exato do campo elétrico máximo para um ponto P'_0 situado na superfície de um condutor do feixe de N condutores, conforme considerado nas hipóteses básicas do texto principal.

A expressão |D.1|, observando-se que a diferença de potencial ΔV entre equipotenciais é constante, de acordo com as hipóteses do Apêndice A, pode ser expressa por:

$$\sigma(P_0) = \frac{\lim_{m \rightarrow +\infty} \frac{\Delta V}{|w(P) - w(P_0)|}}{\lim_{m \rightarrow +\infty} \frac{\Delta V}{|w(P') - w(P'_0)|}},$$

ou

$$\sigma(P_0) = \lim_{m \rightarrow +\infty} \frac{|w(P') - w(P'_0)|}{|w(P) - w(P_0)|}, \quad |D.2|$$

referida ao plano W .

Aplicando-se a transformação conforme $w = z^N$, a Eq. |D.2|, no plano Z , passa a ser:

$$\sigma_{(P_0)} = \lim_{m \rightarrow +\infty} \frac{|\Delta z'|}{|\Delta z|} = \lim_{m \rightarrow +\infty} \frac{|z(P') - z(P'_0)|}{|z(P) - z(P_0)|}, \quad |D.3|$$

onde, para um ponto genérico P_0 , e para um ponto P , conforme anteriormente exposto, tem-se

$$|\Delta z| = |z(P) - z(P_0)| = \left(\sqrt{1 + \rho_{k_0+1, m_0}^2} + 2 \cdot \rho_{k_0+1, m_0} \cdot \cos(\delta_{i_0, m_0}) \right)^{1/N} \triangle_{\xi'_P} - \left(\sqrt{1 + \rho_{k_0, m_0}^2} + 2 \cdot \rho_{k_0, m_0} \cdot \cos(\delta_{i_0, m_0}) \right)^{1/N} \triangle_{\xi'_{P_0}}, \quad |D.4|$$

onde

$$\xi'_j = \xi_j / N, \quad j = P, P_0.$$

A expressão de $|\Delta z|$, tendo-se em conta um número crescente de quadrados curvilíneos e, referindo-se à equipotencial que passa por P_0 , permite reescrever a Eq. |D.4| da seguinte forma:

$$|z(P) - z(P_0)| = \left(\sqrt{1 + \rho_{k_0, m_0}^2 \left[e^{(\pi/m)} \right]^2} + 2 \cdot (\rho_{k_0, m_0} \cdot e^{\pi/m}) \cdot \cos(\delta_{i_0, m_0}) \right)^{1/N} \triangle_{\xi'_P} - \left(\sqrt{1 + \rho_{k_0, m_0}^2} + 2 \cdot \rho_{k_0, m_0} \cdot \cos(\delta_{i_0, m_0}) \right)^{1/N} \triangle_{\xi'_{P_0}}, \quad |D.5|$$

onde

$$m = \lambda \cdot m_0 ,$$

e

$$\xi'_P = 1/N \cdot \text{Tan}^{-1} \left[\frac{\rho_{k_0, m_0} \cdot \text{sen} \delta_{i_0, m_0} \cdot e^{(\pi/m)}}{1 + \rho_{k_0, m_0} \cdot \text{cos}(\delta_{i_0, m_0}) \cdot e^{(\pi/m)}} \right] \quad |D.6|$$

A relação $|z(P') - z(P'_0)|$ é o caso particular da Eq. |D.4| quando $\delta_{i_0, m_0} = 0$, e $\rho_{k_0, m_0} = \rho_{0, m_0} = \alpha$, ou seja,

$$|z(P') - z(P'_0)| = |(1 + \alpha \cdot e^{\pi/m})^{1/N} - (1 + \alpha)^{1/N}| ,$$

ou

$$|z(P') - z(P'_0)| = [(1 + \alpha \cdot e^{\pi/m})^{1/N} - (1 + \alpha)^{1/N}] . \quad |D.7|$$

Uma vez que as expressões anteriores estão referidas aos índices k_0 , m_0 e i_0 , relativos ao ponto P_0 , para efeito de simplificação de notação, considere-se $\rho_{k_0, m_0} = \rho$, e $\delta_{i_0, m_0} = \delta$, de modo que as expressões |D.5| e |D.6|, passam a ser reescritas, respectivamente, como:

$$|z(P) - z(P'_0)| = \left| \left[1 + (\rho \cdot e^{\pi/m})^2 + 2 \cdot \rho \cdot e^{\pi/m} \cdot \text{cos} \delta \right]^{1/2N} \xi'_P - (1 + \rho^2 + 2\rho \text{cos} \delta)^{1/2N} \xi'_{P_0} \right| \quad |D.8|$$

e

$$\xi_p' = 1/N \cdot \text{Tan}^{-1} \left(\frac{\rho \cdot \text{sen} \delta \cdot e^{\pi/m}}{1 + \rho \cdot \text{cos} \delta \cdot e^{\pi/m}} \right) \quad |D.9|$$

A relação |D.3|, passa a ser referida como $\sigma = \sigma(p_0)$.

Aplicando-se as Eqs. |D.8| e |D.7|, a Eq. |D.3|, tem-se que:

$$\sigma = \lim_{m \rightarrow +\infty} \frac{(1 + \alpha \cdot e^{\pi/m})^{1/N} - (1 + \alpha)^{1/N}}{[1 + (\rho \cdot e^{\pi/m})^2 + 2 \cdot \rho \cdot e^{\pi/m} \cdot \text{cos} \delta]^{1/2N} \left| \xi_p' - (1 + \rho^2 + 2\rho \text{cos} \delta)^{1/2N} \xi_{p_0}' \right|} \quad |D.10|$$

Tanto o numerador, quanto o denominador desta equação, tendem a zero, quando o limite é aplicado, sendo portanto indeterminada nestas condições.

Esta indeterminação pode ser levantada, aplicando-se a regra de l'Hôpital a Eq. |D.10|.

No plano Z, o comprimento $|\Delta z|$ é dado por:

$$|\Delta z| = \Delta \ell = \sqrt{\Delta x^2 + \Delta y^2},$$

onde

$$\Delta x = |\Delta z| \cdot \cos \bar{\xi} ,$$

$$\Delta y = |\Delta z| \cdot \text{sen } \bar{\xi} ,$$

$$\Delta z = |\Delta z| \angle \bar{\xi} .$$

Como $\Delta x = \Delta x(m)$, e $\Delta y = \Delta y(m)$, então $\Delta \ell = \Delta \ell(m)$. Portanto, aplicando-se a regra de l'Hôpital à Eq. |D.10|, ou Eq. |D.3|, o denominador passa a ser:

$$\frac{d}{dm} \Delta \ell(m) = \sqrt{\left(\frac{d}{dm} \cdot \Delta x\right)^2 + \left(\frac{d}{dm} \cdot \Delta y\right)^2} = \sqrt{\left[\frac{d}{dm} (|\Delta z| \cdot \cos \bar{\xi})\right]^2 + \left[\frac{d}{dm} (|\Delta z| \cdot \text{sen } \bar{\xi})\right]^2} ,$$

|D.11|

enquanto o numerador passa a ser:

$$\frac{d}{dm} \left[(1 + \alpha \cdot e^{\pi/m})^{1/N} - (1 + \alpha)^{1/N} \right] = -\left(\frac{\pi}{m^2 \cdot N}\right) \cdot (\alpha \cdot e^{\pi/m}) \cdot (1 + \alpha \cdot e^{\pi/m})^{\left(\frac{1-N}{N}\right)} .$$

|D.12|

O valor de Δx é dado por:

$$\Delta x = |z(P)| \cdot \cos \xi'_p - |z(P_0)| \cdot \cos \xi'_{p_0} .$$

Assim,

$$\frac{d}{dm} \Delta x(m) = \frac{d}{dm} [|z(P)| \cdot \cos \xi'_p] - \frac{d}{dm} [|z(P_0)| \cdot \cos \xi'_{p_0}] ;$$

porém, o segundo termo é constante, e a equação anterior reduz-se a

$$\frac{d}{dm} \Delta x(m) = \frac{d}{dm} [|z(P)| \cdot \cos \xi'_p]. \quad |D.13|$$

Das Eqs. |D.8| e |D.9|, a Eq. |D.13| pode ser escrita como:

$$\frac{d}{dm} \Delta x(m) = \frac{d}{dm} \left\{ [1 + (\rho e^{\pi/m})^2 + 2 \cdot \rho \cdot e^{\pi/m} \cdot \cos \delta]^{1/2N} \cdot \cos \left(\frac{1}{N} \cdot \text{Tan}^{-1} \frac{\rho \cdot \text{sen} \delta \cdot e^{\pi/m}}{1 + \rho \cos \delta \cdot e^{\pi/m}} \right) \right\}. \quad |D.14|$$

Analogamente,

$$\frac{d}{dm} \Delta y(m) = \frac{d}{dm} [|z(P)| \cdot \text{sen} \xi'_p],$$

e, das Eqs. |D.8| e |D.9|, obtem-se:

$$\frac{d}{dm} \Delta y(m) = \frac{d}{dm} \left\{ [1 + (\rho \cdot e^{\pi/m})^2 + 2 \cdot \rho \cdot e^{\pi/m} \cdot \cos \delta]^{1/2N} \cdot \text{sen} \left(\frac{1}{N} \cdot \text{Tan}^{-1} \frac{\rho \cdot \text{sen} \delta \cdot e^{\pi/m}}{1 + \rho \cdot \cos \delta \cdot e^{\pi/m}} \right) \right\}. \quad |D.15|$$

Efetuada-se os cálculos, as Eqs. |D.14| e |D.15| podem ser escritas como:

$$\begin{aligned} \frac{d}{dm} \Delta x(m) = & - \left(\frac{\pi}{N \cdot m^2} \right) \cdot \rho \cdot (1 + \rho^2 \cdot e^{2\pi/m} + 2 \cdot \rho \cdot e^{\pi/m} \cdot \cos \delta)^{(1/2N) - 1} \cdot \\ & \cdot [(\rho \cdot e^{2\pi/m} + \cos \delta \cdot e^{\pi/m}) \cdot \cos \xi'_p - (\text{sen} \delta \cdot e^{\pi/m}) \cdot \text{sen} \xi'_p], \quad |D.16| \end{aligned}$$

e,

$$\frac{d}{dm} \Delta y(m) = - \left(\frac{\pi}{N \cdot m^2} \right) \cdot \rho \cdot (1 + \rho^2 \cdot e^{2\pi/m} + 2 \cdot \rho \cdot e^{\pi/m} \cdot \cos \delta)^{(1/2N) - i} .$$

$$\cdot [(\rho \cdot e^{2\pi/m} + \cos \delta \cdot e^{\pi/m}) \cdot \text{sen } \xi'_p + (\text{sen } \delta \cdot e^{\pi/m}) \cdot \text{cos } \xi'_p] \quad |D.17|$$

Utilizando-se as expressões |D.16| e |D.17| , a Eq. |D.11| será igual a:

$$\frac{d}{dm} \cdot \Delta \ell(m) = - \left(\frac{\pi}{N \cdot m^2} \right) \cdot \rho \cdot (1 + \rho^2 \cdot e^{2\pi/m} + 2 \cdot \rho \cdot \cos \delta \cdot e^{\pi/m})^{(1-N)/2N} \cdot e^{\pi/m} . \quad |D.18|$$

Considerando-se as Eqs. |D.18| e |D.12| , σ será dado por:

$$\sigma = \lim_{m \rightarrow +\infty} \frac{- \left(\frac{\pi}{m^2 \cdot N} \right) \cdot \alpha \cdot e^{\pi/m} \cdot (1 + \alpha \cdot e^{\pi/m})^{(1-N)/N}}{- \left(\frac{\pi}{m^2 \cdot N} \right) \cdot \rho \cdot (1 + \rho^2 \cdot e^{2\pi/m} + 2 \cdot \rho \cdot e^{\pi/m} \cdot \cos \delta)^{(1-N)/2N} \cdot e^{\pi/m}}$$

ou,

$$\sigma = \frac{\alpha \cdot (1 + \rho^2 + 2 \cdot \rho \cdot \cos \delta)^{(N-1)/2N}}{(1 + \alpha)^{(N-1)/N} \cdot \rho} , \quad |D.19|$$

onde σ é o valor da relação entre o campo elétrico num ponto do plano Z, tal que a equipotencial de ordem k, o contêm, e o valor do campo elétrico máximo na superfície do condutor do feixe, sendo função de ρ e δ , ou seja, $\sigma = \sigma(\rho, \delta)$.

Alguns valores particulares de σ são a seguir ob-
tidos:

- a) Valor do campo elétrico mínimo na superfície do condutor do feixe em relação ao valor de campo elétrico máximo.

Neste caso, $\rho = \alpha$, e a Eq. |D.19|, passa a ser:

$$\sigma = \frac{(1 + \alpha^2 + 2 \cdot \alpha \cdot \cos \delta)^{(N-1)/2N}}{(1 + \alpha)^{(N-1)/N}} \quad |D.20|$$

Para obtermos o valor de δ correspondente ao valor de campo elétrico mínimo na superfície do condutor, deve-se satisfazer a relação $(d\sigma/d\delta) = 0$, ou seja:

$$\frac{d\sigma}{d\delta} = \frac{1}{(1 + \alpha)^{(N-1)/N}} \cdot \frac{(N-1)}{2 \cdot N} \cdot \frac{-2 \cdot \alpha \cdot \sin \delta}{(1 + \alpha^2 + 2 \cdot \alpha \cdot \cos \delta)^{(N+1)/2N}} = 0,$$

impondo-se que $\sin \delta = 0$. Como o valor procurado é mínimo, $\delta = \pi$; substituindo-se este valor na Eq. |D.20|, obtem-se:

$$\sigma = \sigma_m = \frac{(1 - \alpha)^{(N-1)/N}}{1 + \alpha} \quad |D.21|$$

- b) Valor do campo elétrico ao longo da direção $\delta = \pi$, em relação ao valor do campo elétrico máximo.

Neste caso, a Eq. |D.19| reduz-se a:

$$\sigma = \frac{\alpha \cdot (\rho - 1)^{(N-1)/N}}{(1 + \alpha)^{(N-1)/N} \cdot \rho} \quad |D.22|$$

Devido à simetria dos condutores do feixe, quando $\rho = 1$, a expressão anterior indica que o valor do campo elétrico no centro do feixe é nulo. Efetivamente, o valor do campo elétrico será dependente da superposição de campos elétricos dos feixes de condutores das duas outras fases, dos demais condutores além destes, bem como do solo.

Ao longo desta direção, $\delta = \pi$, o valor máximo de σ é obtido, fazendo-se $d\sigma/d\rho = 0$, ou

$$\frac{d\sigma}{d\rho} = \frac{\alpha}{(1+\alpha)^{(N-1)/N}} \cdot \frac{[(N-1)/(N)] \cdot (\rho-1)^{[(N-1)/(N)]-1}}{\rho} - \frac{(\rho-1)^{(N-1)/N}}{\rho^2} = 0,$$

o que implica em $\rho = N$. Impondo-se esta condição à Eq. |D.22|, tem-se:

$$\sigma = \sigma_D = \frac{\alpha}{N} \cdot \left(\frac{N-1}{1+\alpha}\right)^{\frac{N-1}{N}}. \quad |D.23|$$

D.2 - OBSERVAÇÕES SOBRE ALGUNS VALORES PARTICULARES DE σ

No traçado dos lugares geométricos dos pontos de mesmo valor de campo elétrico, deve-se observar a diminuição de seu valor em duas direções: uma, quando os pontos considerados se afastam progressivamente do feixe de condutores, ao longo do plano transversal aos condutores, e outra, quando os pontos considerados se aproximam do centro geométrico do feixe, em virtude dos campos elétricos tenderem a ser simétricos neste ponto.

É de se esperar portanto, que, a partir de um certo valor de σ , existam dois lugares geométricos de pontos de mesmo valor de campo elétrico, correspondentes a valores de campo elétrico menores, tanto ao se aproximarem do centro geométrico do feixe, quanto ao se afastarem dele.

O ponto tal que $\rho = N$ e $\delta = \pi$, pertence ao lugar geométrico dos pontos tais que $\sigma = \sigma_D$. Conforme deduzido anteriormente, para obtenção de σ_D , impôs-se a condição expressa pela Eq. |D.23|, que, entretanto, não conduz ao valor máximo nem mínimo de $\sigma(\rho, \alpha)$, caracterizando portanto, um ponto de sela desta função. Portanto, pontos tais que $\delta = \pi$, e $\rho = N + \bar{\epsilon}$, e $\rho = N - \bar{\epsilon}$, sendo $\bar{\epsilon}$, um valor real tão pequeno quanto desejado, pertencerão respectivamente, aos lugares geométricos dos pontos de valor de campo elétrico constante menores que $\sigma_D \cdot |\vec{E}|_{\max}$, distintos, uma vez que estes lugares geométricos, para $\rho = N - \bar{\epsilon}$, se aproximam do centro geométrico do feixe, e para $\rho = N + \bar{\epsilon}$, se afastam do feixe. O valor de $\sigma = \sigma_D$, vem caracterizar o menor valor de σ ao qual está associado um só lugar geométrico de pontos de mesmo valor de campo elétrico.

Para $\sigma < \sigma_D$, deverão existir dois lugares geométricos de pontos associados a este valor escolhido. No plano Z, um destes lugares geométricos envolve a origem 0, aproximando-se dela à medida que σ diminui, enquanto o outro se afasta, envolvendo as curvas cujos valores de σ são maiores. Neste último caso, os valores de ρ são bastante maiores que a unidade.

As curvas para os valores de σ devem estar mais próximas entre si, na região compreendida entre o centro geométrico do feixe, e cada condutor, do que na região externa ao feixe, pois o valor de campo elétrico tende a zero numa região limitada, no primeiro caso, e, infinita, no segundo. Esta descrição é referente ao plano Z.

O valor de $\sigma = \sigma_m$, conforme a Eq. |D.21|, indica o primeiro lugar geométrico de pontos que envolve completamente cada condutor do feixe; valores de σ , tais que $\sigma_m < \sigma \leq 1$, referem-se a lugares geométricos de pontos que se originam sobre o condutor, para um certo valor máximo de δ , sem entretanto chegarem a envolvê-lo. Isto pode ser observado pelo fato dos valores de σ sobre a superfície do condutor irem variando desde $\sigma = \sigma_{m\max} = 1$, para $\delta = 0$, até $\sigma = \sigma_m$, para $\delta = \pi$, na região do condutor voltada para o centro do feixe. Quando atinge seu valor máximo, $\sigma = 1$, o lugar geométrico de pontos resume-se a um único ponto sobre o condutor, assim como no caso de $\sigma = 0$, no centro geométrico do feixe.

Quando o número N. de condutores do feixe é igual a 2, pode-se verificar que o lugar geométrico dos pontos tais que a relação $\sigma = \sigma_D$ é satisfeita, é constituído de segmentos de círculos; cujos centros são coincidentes com os centros destes condutores no plano Z, conforme mostrado na Figura (D.1).

Neste caso, tem-se que $\sigma = \sigma_D$, ou seja:

$$\frac{\alpha \cdot (1 + 2 \cdot \rho \cdot \cos \delta + \rho^2)^{(N-1)/2N}}{(1+\alpha)^{(N-1)/N} \cdot \rho} = \frac{\alpha \cdot (N-1)^{(N-1)/N}}{(1+\alpha)^{(N-1)/N} \cdot N} .$$

Para $N = 2$, a equação acima simplifica-se, obtendo-se:

$$\frac{(1 + 2 \cdot \rho \cdot \cos \delta + \rho^2)^{1/4}}{\rho} = \frac{1}{2} . \quad |D.24|$$

No plano Z , deseja-se verificar a seguinte igualdade, caso os trechos do lugar geométrico para o qual $\sigma = \sigma_D$ pertençam a um círculo de centro no condutor, e raio igual a R :

$$(|z| \cdot \cos \bar{\xi} - 1)^2 + (|z| \cdot \sin \bar{\xi})^2 = R^2 = \text{constante},$$

ou

$$|z|^2 - 2 \cdot |z| \cdot \cos \bar{\xi} + 1 = \text{constante}. \quad |D.25|$$

De acordo com a função $w = z^N$, para o mapeamento conforme,

$$|z| = \sqrt[N]{|w|} = \sqrt{|w|} , \quad |D.26|$$

e

$$\bar{\xi} = \frac{\delta'}{2}, \quad |D.27|$$

onde

$$w = |w| \angle \delta' .$$

Portanto, substituindo-se as Eqs. |D.26| e |D.27|, em |D.25|, tem-se

$$|w| - 2 \sqrt{|w|} \cdot \cos(\delta'/2) + 1 = \text{constante.} \quad |D.28|$$

Porém, no plano W, de acordo com a Fig. (D.2), tem-se

$$|w| = \sqrt{1 + \rho^2 + 2 \cdot \rho \cdot \cos \delta} \quad |D.29|$$

e,

$$\cos \delta' = \frac{1 + \rho \cdot \cos \delta}{\sqrt{1 + \rho^2 + 2 \cdot \rho \cdot \cos \delta}} . \quad |D.30|$$

Aplicando-se a igualdade trigonométrica

$$\cos^2 x = (1 + \cos 2x)/2,$$

tem-se que

$$\cos^2(\delta'/2) = \frac{1 + \cos \delta'}{2} . \quad |D.31|$$

Substituindo-se a Eq. |D.30| em |D.31|, obtem-se

$$\cos^2(\delta'/2) = \frac{1 + \rho \cdot \cos \delta + \sqrt{1 + \rho^2 + 2 \cdot \rho \cdot \cos \delta}}{2 \cdot \sqrt{1 + \rho^2 + 2 \cdot \rho \cdot \cos \delta}},$$

ou

$$\cos(\delta'/2) = \frac{\sqrt{1 + \rho \cdot \cos \delta + \sqrt{1 + \rho^2 + 2 \cdot \rho \cdot \cos \delta}}}{\sqrt{2 + \sqrt{1 + \rho^2 + 2 \cdot \rho \cdot \cos \delta}}}. \quad |D.32|$$

Substituindo-se as Eqs. |D.29| e |D.32|, em |D.28|, tem-se

$$\sqrt{1 + 2 \cdot \rho \cdot \cos \delta + \rho^2} - \sqrt{2 \cdot \sqrt{1 + \rho \cdot \cos \delta + \sqrt{1 + \rho^2 + 2 \cdot \rho \cdot \cos \delta}} + 1} = A. \quad |D.33|$$

Aplicando-se a condição da Eq. |D.24| ser satisfeita para os pontos tais que $\sigma = \sigma_D$, na Eq. |D.33|, tem-se

$$A = \frac{\rho^2}{4} - \sqrt{2 \cdot \sqrt{1 + \rho \cdot \cos \delta + \rho^2/4}} + 1 = \frac{\rho^2}{4} - \sqrt{2 + 2 \cdot \rho \cdot \cos \delta + \frac{\rho^2}{2}} + 1 \quad |D.34|$$

Porém, da Eq. |D.24|, tem-se

$$\frac{\rho^4}{16} = 1 + 2 \cdot \rho \cdot \cos \delta + \rho^2,$$

ou

$$\frac{\rho^2}{2} + 2 \cdot \rho \cdot \cos \delta + 2 = \frac{(\rho^2 - 4)^2}{4} . \quad |D.35|$$

Levando-se a Eq. |D.35| em |D.34|, obtem-se

$$A = \frac{\rho^2}{4} - \frac{\rho^2 - 4}{4} + 1 = 2 .$$

Porém, da Eq. |D.25|,

$$2 = R^2$$

ou

$$R = \sqrt{2} ,$$

o que demonstra o fato do lugar geométrico ser composto de trechos de círculo de raio $R = \sqrt{2} = \text{cte}$, no plano Z.

D.3 - TRAÇADO DE CURVAS DE MESMO VALOR DE CAMPO ELÉTRICO

Para a determinação de curvas de mesmo valor de campo elétrico, torna-se necessária a resolução da equação transcendental |D.19|, a qual é efetuada através dos Programas P-4 e P-5, implantados na calculadora HP-41C, utilizando processo iterativo baseado no método de Newton.

No ítem D.3.1 é apresentado o desenvolvimento teórico relativo às equações empregadas e o método de resolução

da equação transcendental |D.19|, e, no ítem D.3.2, são apresentados os Programas P-4 e P-5, incluindo os modos de operação respectivos, listagens e casos teste.

D.3.1 - Desenvolvimento Teórico

Para que se torne possível o traçado de curvas de módulo de campo elétrico constante no plano Z , é necessário resolver a equação |D.19|, aplicando-se posteriormente, a transformação conforme $w = z^N$.

Conforme salientado no ítem D.1.2, para cada valor de σ menor que σ_D , surgem dois lugares geométricos a ele associados: um deles, é limitado pelo trecho da curva relativa a $\sigma = \sigma_D$, que envolve a origem (0) do plano complexo W ; o outro, afasta-se progressivamente do condutor, de modo que os valores de ρ , se tornam cada vez maiores, para qualquer valor do ângulo δ .

Na Eq. |D.19|, ρ mede a distância entre o condutor localizado no ponto (1,0) do eixo real u , no plano complexo W , e pontos de uma dada curva associada a um valor de σ , enquanto δ é o ângulo determinado pelo eixo real u , e a linha de força oriunda do condutor considerado localizado neste mesmo eixo, medido no sentido trigonométrico.

Para que se torne possível determinar para um dado valor de δ , um único valor de ρ , correspondente a um ponto

pertencente à curva associada a σ , basta-se-ia realizar uma mudança adequada de coordenadas no plano W.

Considerando-se as linhas de força oriundas do condutor localizado em (1,0), no plano W, em geral sãõ hã interseção destas com as curvas equigradientes em um único ponto, para valores de σ reais, e menores que a unidade, a menos dos casos em que o ângulo δ seja próximo de 180° , e σ próximo de σ_D . Neste caso, quando $\sigma < \sigma_D$, incluem-se em geral os trechos de curvas equigradientes que se afastam progressivamente do condutor, ou seja, $\rho \rightarrow \infty$, caracterizados como sendo os ramos "externos" de equigradientes associadas a um mesmo valor de σ .

Os pontos pertencentes a estas equigradientes sãõ obtidos a partir da resolução da equação transcendental

$$\sigma = \frac{\alpha \cdot (1 + \rho^2 + 2 \cdot \rho \cdot \cos \delta)^{(N-1)/2 \cdot N}}{(1 + \alpha)^{(N-1)/N} \cdot \rho} \quad |D.19|$$

Através do método de Newton, pretende-se obter uma solução desta equação, satisfazendo uma dada precisão, através de iterações sucessivas, e de acordo com a relação

$$\rho_{i+1} = \rho_i - \frac{f(\rho_i)}{f'(\rho_i)} = g(\rho_i), \text{ para } i = 0, 1, 2, \dots, \quad |D.36|$$

onde ρ_0 deve ser uma aproximação inicial para um raiz de $f(\rho)$.

Fazendo-se na Eq. |D.19|,

$$C = \sigma \cdot \left[\frac{(1+\alpha)^{(N-1)/N}}{\alpha} \right],$$

a mesma pode ser escrita como

$$\rho = (1/C) \cdot (1 + 2 \cdot \rho \cdot \cos \delta + \rho^2)^{(N-1)/2N} \quad |D.37|$$

Dentre várias hipóteses quanto à escolha da função $f(\rho)$, verificou-se ser a seguinte a mais conveniente

$$f(\rho) = \frac{C \cdot \rho - (1 + 2 \cdot \rho \cdot \cos \delta + \rho^2)^{(N-1)/2N}}{C}.$$

A partir da equação acima, obtém-se

$$f'(\rho) = \frac{C - [(N-1)/(2N)] \cdot (1 + 2 \cdot \rho \cdot \cos \delta + \rho^2)^{[(N-1)/(2N)]-1} \cdot 2 \cdot (\rho + \cos \delta)}{C}.$$

De acordo com a Eq. |D.36|, o valor de $g(\rho)$ é da do por

$$g(\rho) = \frac{(1 + 2 \cdot \rho \cdot \cos \delta + \rho^2) - \rho \cdot [(N-1)/N] \cdot (\cos \delta + \rho)}{C \cdot (1 + 2 \cdot \rho \cdot \cos \delta + \rho^2)^{(N+1)/2N} - [(N-1)/N] \cdot (\cos \delta + \rho)} \quad |D.38|$$

A expressão acima pode ser reescrita como

$$g(\rho) = \frac{N + N \cdot \rho \cdot \cos \delta + \rho^2 + \rho \cdot \cos \delta}{N \cdot C \cdot (1 + 2 \cdot \rho \cdot \cos \delta + \rho^2)^{(N+1)/2N} - (N-1) \cdot (\rho + \cos \delta)}.$$

Para valores de $\rho \gg 1$, a equação anterior pode ser aproximada por

$$g(\rho) \cong \frac{\rho^2}{N \cdot C \cdot \rho^{(N+1)/N} - N \cdot \rho} = \frac{\rho}{N \cdot C \cdot \rho^{1/N} - N},$$

obtendo-se para $g'(\rho)$, nestas condições,

$$g'(\rho) \cong \frac{[(C/N) - (C/N^2)] \cdot \rho^{1/N} - 1/N}{(C \cdot \rho^{1/N} - 1)^2},$$

que se reduz a

$$g'(\rho) \cong \frac{1}{N(C \cdot \rho^{1/N} - 1)},$$

para valores de $N > 2$.

Para que a iteração expressa pela Eq. [D.36] seja convergente, é necessário que $|g'(\rho)| < 1$, ou seja, $|N(C \cdot \rho^{1/N} - 1)| > 1$, obtendo-se então a seguinte condição:

$$\rho > (1 + 1/N)^N \cdot \frac{1}{C}.$$

Desde que $(1 + 1/N) \leq 2$, a desigualdade anterior, é satisfeita para $\rho = \rho_0 = (2/C)^N$, para a aproximação inicial da raiz de $f(\rho)$.

Para valores de $\sigma > \sigma_D$, verifica-se que para $\rho_0 = 0,9$, obtem-se convergência da função $g(\rho)$.

A expressão de $g(\rho)$, obtida em [D.38], não é apropriada para a determinação dos lugares geométricos para os quais $\sigma < \sigma_D$, que se aproximam cada vez mais da origem.

Para a determinação dos pontos pertencentes aos trechos de curvas equigradientes tais que $\sigma < \sigma_D$, que se aproximam cada vez mais da origem (0), no plano W, é conveniente obter-se as interseções a partir de um condutor de "teste", localizado na origem (0), ou seja, para uma translação de t , conforme mostrado na Fig. (D.2), de forma que a interseção de linhas de força oriundas deste ponto interceptam estes trechos "internos" das equigradientes em apenas um único ponto, desde que o valor inicial para a convergência de ρ , no processo iterativo empregando o método de Newton, seja adequado.

Para a determinação de σ , para valores de σ próximos de σ_D sendo o ângulo δ próximo de 180° , também seria conveniente a translação de t , do condutor de "teste", ao longo do eixo real u , e à esquerda da origem (0), conforme mostrado na Fig. (D.2). Portanto nestas condições, tem-se as seguintes relações a serem determinadas:

$$\rho = f(|w|, t, \delta')$$

e

$$\delta = g(|w|, t, \delta') ,$$

de forma a serem substituídas na equação transcendental |D.19| escrita do seguinte modo:

$$\rho \cdot C = (1 + 2 \cdot \rho \cdot \cos \delta + \rho^2)^{(N-1)/2 \cdot N} \quad |D.37|$$

onde

$$C = \frac{\sigma}{[\alpha / (1 + \alpha)^{(N-1)/N}]}$$

Da Fig. (D.2), as seguintes relações são válidas onde, para efeito de simplificação de notação, $w = |w|$:

$$w \cdot \cos \delta' = -t + 1 + \rho \cdot \cos \delta$$

$$w \cdot \sin \delta' = \rho \cdot \sin \delta$$

$$\rho \cdot \cos \delta = w \cdot \cos \delta' + t - 1$$

$$\rho \cdot \sin \delta = w \cdot \sin \delta' .$$

Destas relações obtem-se portanto:

$$\rho = \sqrt{w^2 + 2 \cdot w \cdot \cos \delta' \cdot (t-1) + (t-1)^2}$$

$$2 \cdot \rho \cdot \cos \delta = 2(w \cdot \cos \delta' + t - 1).$$

Substituindo-se estas relações na Equação |D.37|, tem-se que

$$w = \sqrt{k \cdot [w^2 + (t-1) \cdot 2 \cdot w \cdot \cos \delta' + (t-1)^2]^{N/(N-1)} - 2 \cdot w \cdot t \cdot \cos \delta' - t^2}.$$

Considerando-se a Equação |D.36|, a mesma pode ser reescrita da seguinte forma

$$w_{i-1} = \frac{h(w_i) - w_i \cdot [h'(w_i)/2]}{\sqrt{h(w_i)} - [h'(w_i)/2]}, \quad |D.39|$$

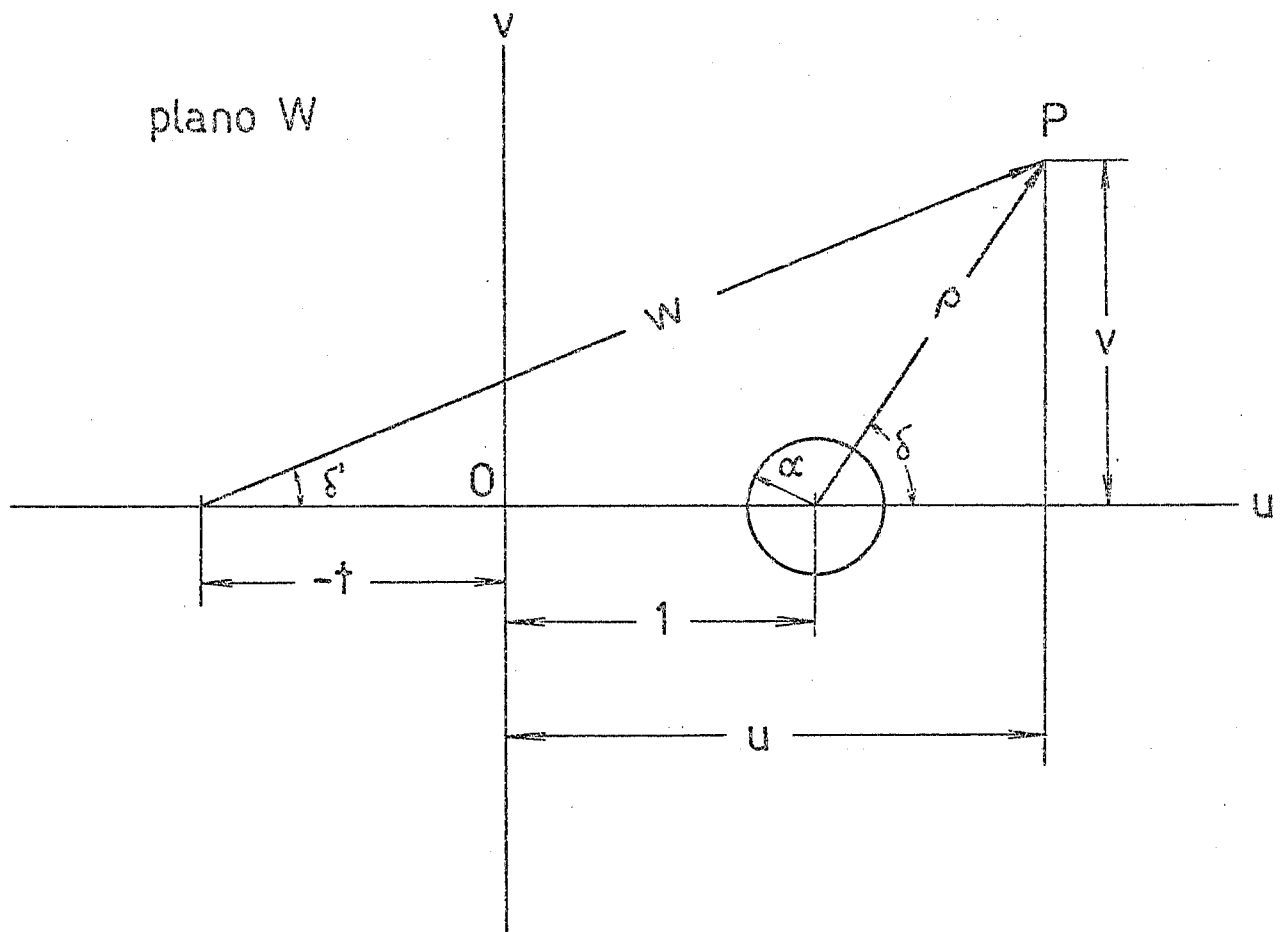


Figura D.2 - Disposição de condutor no plano complexo W , considerando-se as variáveis para mudança de coordenadas.

onde

$$h(w) = k \cdot [w^2 + 2 \cdot w \cdot \cos \delta' \cdot (-1+t) + (-1+t)^2]^{N/(N-1)} - 2 \cdot w \cdot t \cdot \cos \delta' - t^2$$

e

$$[h'(w)/2] = k_1 \cdot [w^2 + (t-1) \cdot (2 \cdot w \cdot \cos \delta' + t-1)]^{1/(N-1)} \cdot [w + \cos \delta' \cdot (t-1)] - t \cdot \cos \delta' ,$$

sendo

$$k = (\sigma/\alpha)^{2N/(N-1)} \cdot (1+\alpha)^2$$

e

$$k_1 = k \cdot [N/(N-1)] .$$

As formulações aqui apresentadas foram empregadas nos Programas P-4 e P-5, respectivamente destinados ao cálculo de equigradientes tais que $\sigma > \sigma_D$ e ramos "externos", quando $\sigma < \sigma_D$, e equigradientes tais que $\sigma = \sigma_D$ e ramos "internos", quando $\sigma < \sigma_D$.

Aplicando-se a transformação conforme $w = z^N$, obtém-se para coordenadas x e y , no plano Z , as seguintes expressões, utilizadas nos Programas P-4 e P-5, conforme se deduz da Figura (D.2):

$$w \cdot \cos \delta' + t = u,$$

$$w \cdot \sin \delta' = v,$$

$$|W| = \sqrt{u^2 + v^2},$$

$$|W| = \sqrt{w^2 + 2 \cdot w \cdot t \cdot \cos \delta' + t^2},$$

$$\psi = \cos^{-1}(u/|W|),$$

e

$$\psi = \cos^{-1} \left(\frac{w \cdot \cos \delta' + t}{\sqrt{w^2 + 2 \cdot w \cdot t \cdot \cos \delta' + t^2}} \right).$$

Portanto,

$$x = \frac{N}{\sqrt{|W|} \cdot \cos(\psi/N)},$$

e

$$y = \frac{N}{\sqrt{|W|} \cdot \sin(\psi/N)}.$$

D.3.2 - Programas P-4 e P-5

Para a resolução da equação transcendental [D.19], foram elaborados os Programas P-4 e P-5, que utilizam o método iterativo proposto pela relação [D.36], baseando-se na resolução das Eqs. [D.38] e [D.39], respectivamente.

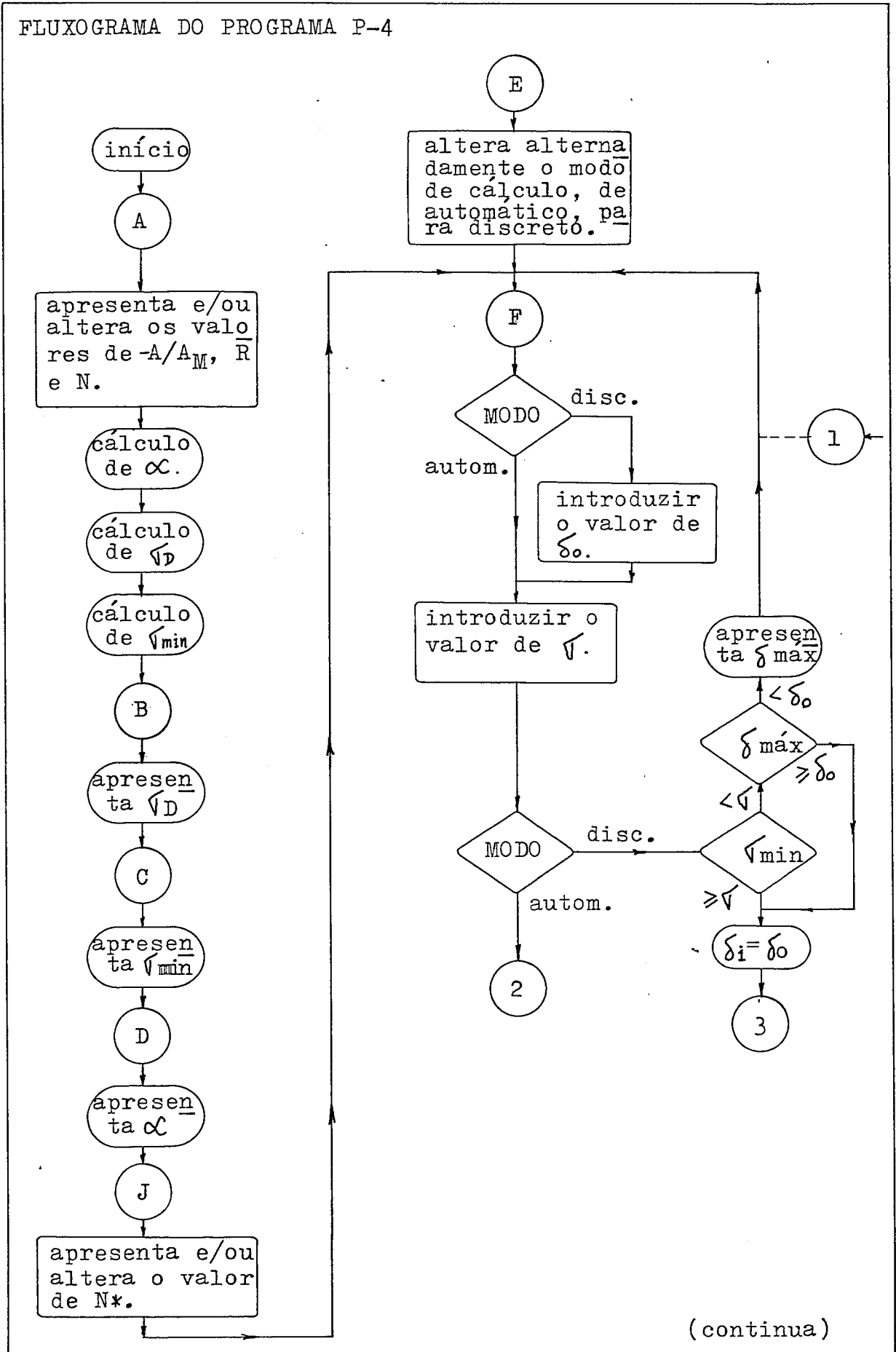
Para a utilização conveniente destes dois programas, apresentam-se no ítem (D.3.2.1) seus fluxogramas, no ítem (D.3.2.2), a descrição dos respectivos modos de operação e, no ítem (D.3.2.3), as respectivas listagens.

Ambos os programas foram implantados na calculadora HP-41C, empregando-se cartões magnéticos, e um caso teste para cada programa é apresentado no ítem (D.3.2.4).

D.3.2.1 - Fluxogramas dos Programas P-4 e P-5

D.3.2.1.1 - Fluxograma do Programa P-4

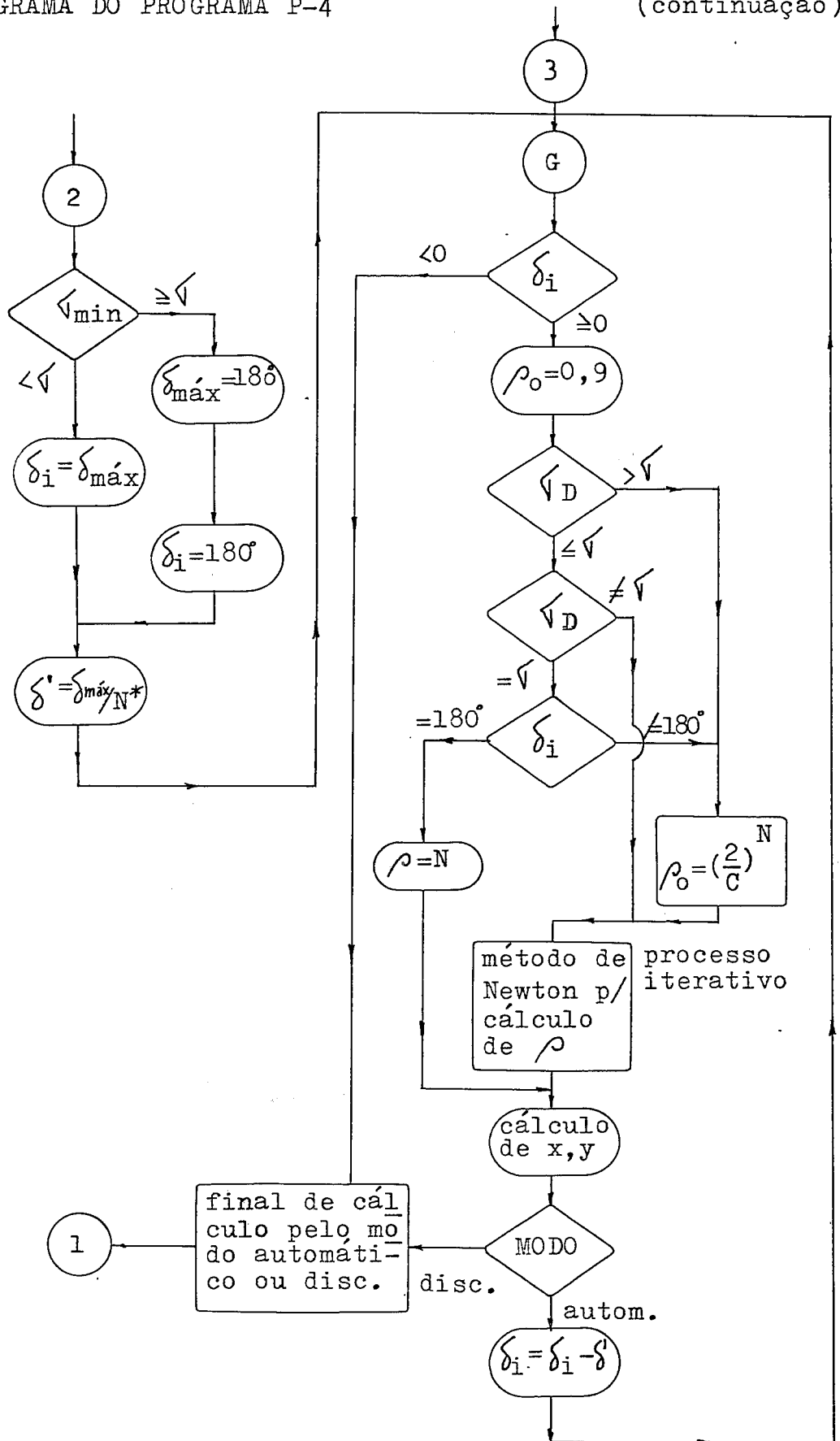
FLUXOGRAMA DO PROGRAMA P-4



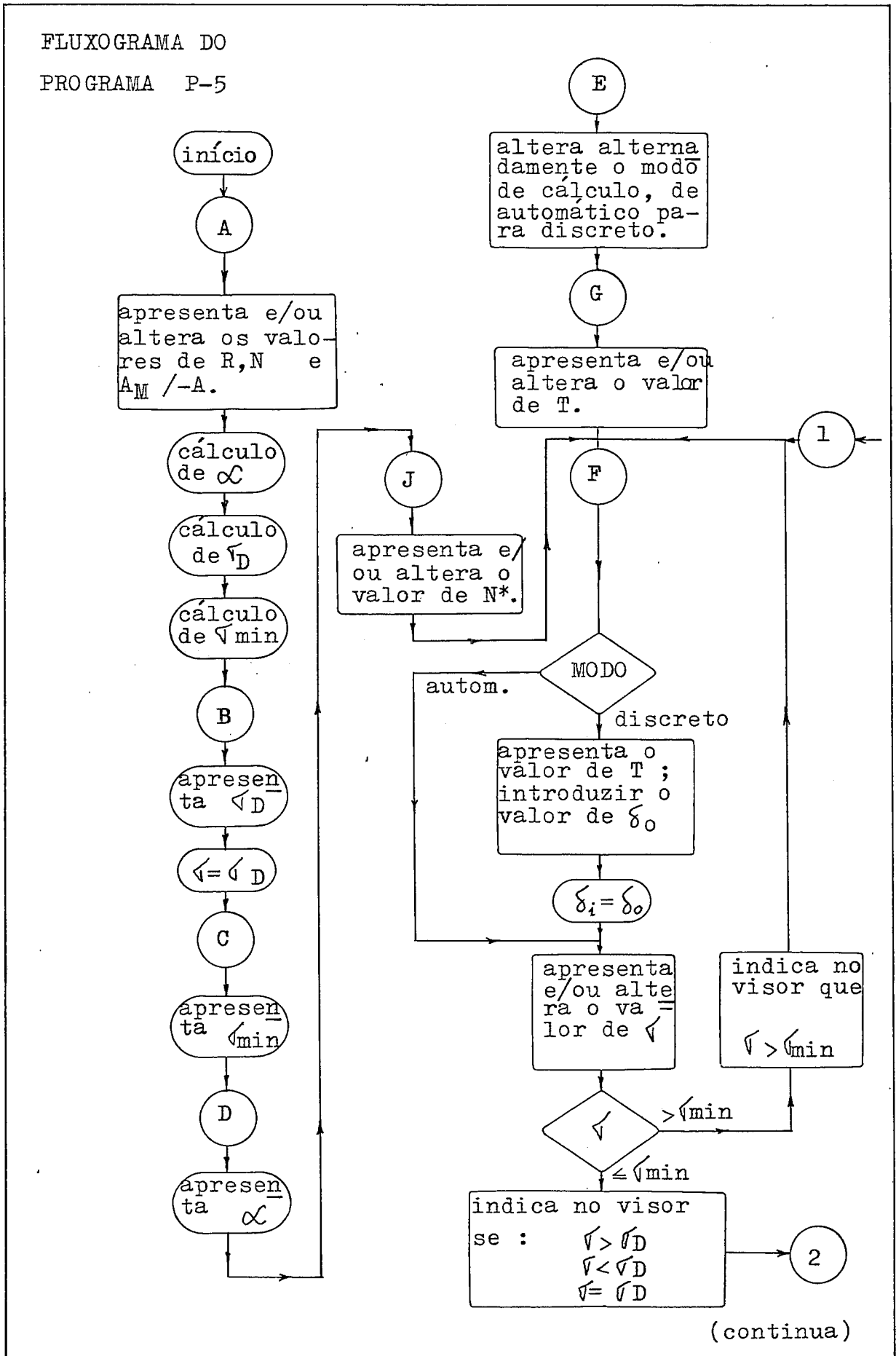
(continua)

FLUXOGRAMA DO PROGRAMA P-4

(continuação)

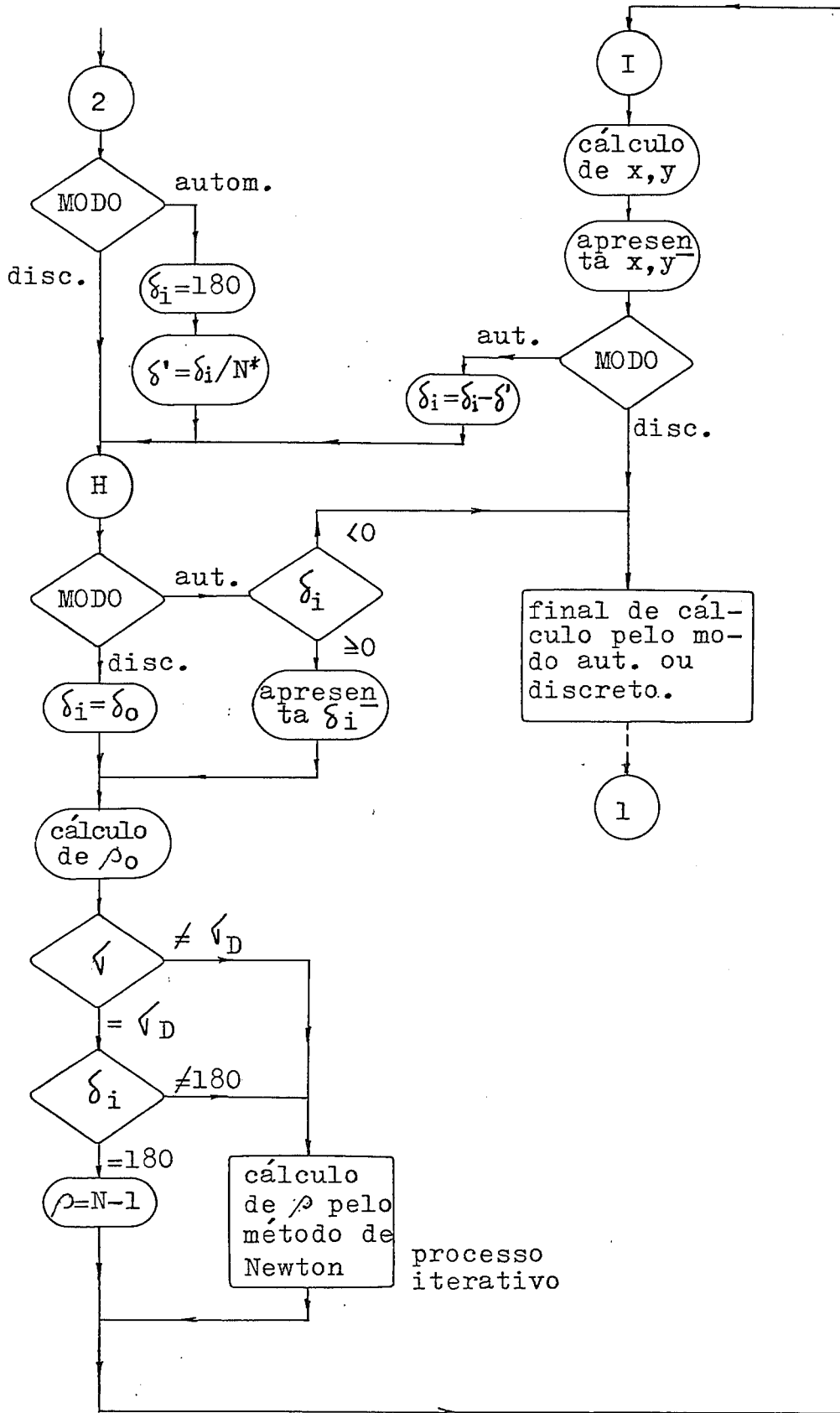


D.3.2.1.2 - Fluxograma do Programa P-5



FLUXOGRAMA DO PROGRAMA P-5

(continuação)



D.3.2.2 - Modo de Operação dos Programas P-4 e P-5

No ítem (D.3.2.1) descreve-se o modo de operação do Programa P-4, e no ítem (D.3.2.2), o modo de operação do Programa P-5.

D.3.2.2.1 - Modo de Operação do Programa P-4

Quando o programa é utilizado pela primeira vez, deve-se introduzir na calculadora, através do teclado, as instruções indicadas sequencialmente no ítem D.3.2.3.1, relativas à listagem do Programa P-4. No caso de se dispor da leitora de cartões magnéticos, estas instruções poderão ser gravadas nestes cartões, permitindo assim guardar indefinidamente o programa para utilização posterior, mesmo apagando-se as instruções a ele relativas, da memória da calculadora. Em seguida, deverão ser executadas as instruções relativas à Etapa A.

Desde que a Etapa A tenha sido utilizada uma única vez, qualquer das etapas, em número de duas, poderá ser executada.

As etapas, com os respectivos procedimentos, são as seguintes:

ETAPA A - Nesta etapa, pretende-se calcular pela primeira vez, as coordenadas x e y no plano Z , relativas às curvas equigradientes obtidas pela interseção destas com

linhas de força oriundas de um dos N condutores do feixe, e por fase de uma linha de transmissão, localizado no ponto $(1,0)$ do plano complexo W , fazendo-se variar o ângulo δ (computado no sentido trigonométrico), que estas linhas de força fazem com o eixo real u , conforme apresentado no desenvolvimento teórico, no item (D.3.1).

Devido à simetria existente nestas condições, as interseções serão calculadas apenas para o semi-plano superior do plano complexo W .

Esta etapa consiste do seguinte conjunto de procedimentos:

A.1 - Deve ser escolhido inicialmente, o modo de operação do programa quanto aos cálculos das interseções, que poderão ser efetuadas no modo automático ou discreto. Para tanto deve-se pressionar a tecla E , que alterna os mesmos cada vez que é pressionada.

A.2 - Pressiona-se a tecla A , sendo apresentado no visor ($R = n \dots n$), com o valor existente na memória anteriormente. Introduce-se o valor de R desejado através do teclado da calculadora, apertando-se em seguida a tecla R/S . O programa deverá executar algumas instruções, apre-

sentando então no visor ($N = n...n$), com o valor existente na memória anteriormente. Introduce-se o valor de N desejado através do teclado, apertando-se em seguida a tecla R/S. O programa deverá executar algumas instruções, apresentando no visor ($AM = n...$) ou ($-A = n...n$), respectivamente conforme o valor existente na memória anteriormente seja positivo ou negativo. Introduce-se o valor de A_M ou $-A$ desejado através do teclado, apertando-se em seguida a tecla R/S.

Os valores de A_M , $-A$ e R introduzidos devem estar nas mesmas unidades de comprimento.

- A.3 - Após os cálculos referidos em A.2, o programa deverá executar uma série de instruções, sendo apresentado a seguir no visor ($\Sigma D = n...n$), ou seja, o valor de σ_D .
- A.4 - Apertando-se a tecla R/S a seguir, é apresentado no visor ($\Sigma M = n...n$), ou seja, o valor de σ_{min} .
- A.5 - Apertando-se a tecla R/S a seguir, é apresentado no visor ($ALFA = n...n$), ou seja, o valor de α .

A.6 - Apertando-se a tecla R/S a seguir, é apresentado no visor ($N^* = n...n$), ou seja, o número anteriormente existente na memória, relativo ao número de divisões do ângulo de 180° ou do ângulo $\delta_{\text{máx}}$ calculado para o cálculo das interseções de equigradientes com linhas de força. Introduce-se portanto o valor desejado de N^* através do teclado, apertando-se a seguir a tecla R/S.

A.7 - Conforme o modo de operação, o programa requer diferentes entradas de dados, dependendo de estar operando.

(1) no modo automático

(2) no modo discreto.

A.7.1 - Estando operando no modo automático (após o procedimento A.6) é apresentado no visor ($\Sigma =$), devendo ser introduzido o valor de σ desejado para os cálculos dos pontos pertencentes a esta equigradiante, apertando-se a seguir a tecla R/S.

A.7.1.1 - O programa apresenta a seguir no visor ($\delta = n...n$), indicando o ângulo máximo ($\delta_{\text{máx}}$) para o cálculo das interseções no plano complexo W , o qual será dividido por

N^* , sendo calculadas as interseções desde $\delta_i = \delta_{\text{m}\bar{\text{a}}\text{x}}$ até $\delta_i = 0^0$, onde $\delta_i = \delta_{\text{m}\bar{\text{a}}\text{x}} - (\delta_{\text{m}\bar{\text{a}}\text{x}}/N^*) \cdot i$, para $i = 0, 1, 2, \dots, N^*$.

A.7.1.2 - O programa apresenta a seguir, durante o processo iterativo de convergência, os valores de ρ no visor, ou seja, ($R0 = n \dots n$), indicando posteriormente no visor, os valores das coordenadas x e y de cada ponto de interseção com a equigradiente relativa ao valor de σ introduzido.

O processo é repetido conforme descrito em A.7.1.1 e A.7.1.2, até que $\delta_i = 0^0$, quando o programa finaliza os cálculos e volta ao procedimento A.7.1, à espera de serem introduzidos novos dados.

A.7.2 - Estando no modo discreto, e após o procedimento A.6, é indicado no visor ($\angle = n \dots n$), devendo ser introduzido o ângulo δ_0 , segundo o qual se deseja fazer a interseção com a equigradiente σ , cujo valor deverá ser introduzido logo a seguir, pressionando-se após a tecla R/S. É então apresentado no visor ($\Sigma =$), devendo-se introduzir o valor de σ desejado através do teclado, apertando-se a seguir a tecla R/S.

- A.7.2.1 - O programa executa uma s̄erie de instruções e verifica se o valor de δ_0 , segundo o qual a interseção deve ser efetuada, é compatível com o valor de σ introduzido.
- A.7.2.2 - No caso do ângulo δ_0 não ser compatível, é apresentado no visor (δ MAX = n...), indicando assim o maior valor de δ_0 compatível com o valor de σ introduzido. Para voltar ao procedimento A.7.2, basta ser pressionada a tecla R/S ou F.
- A.7.2.3 - No caso do ângulo δ_0 introduzido ser compatível com o valor de σ , o programa apresenta no visor (δ = n...n), ou seja, o valor do ângulo δ_0 introduzido.
- A.7.2.4 - A seguir o programa apresenta no visor, durante o processo iterativo de convergência, os valores de ρ , ou seja, (R0 = n...n), indicando posteriormente no visor, os valores das coordenadas x e y do ponto de interseção desejado, e finaliza o cálculo voltando então ao procedimento A.7.2, ã espera de serem introduzidos novos dados.

ETAPA B - Neste etapa possibilita-se a apresentação de qualquer dos dados de entrada introduzidos para o cálculo das equigradientes (R , N e A_M , ou $-A$), optando-se por alterá-los, ou não, bem como verificar alguns valores particulares utilizados nos cálculos. Esta etapa consiste no seguinte conjunto de procedimentos, todos, ou parte deles podendo ser executados em qualquer ordem:

B.1 - Para se verificar o valor de R existente na memória, deve ser pressionada a tecla A , sendo apresentado no visor ($R = n...n$), ou seja, o valor previamente escolhido em A.2, na unidade escolhida. Desejando-se alterar o valor de R , deve-se introduzir pelo teclado o novo valor, pressionando-se a seguir a tecla R/S . No caso de simples verificação do valor anterior, pode-se executar qualquer um dos procedimentos desta etapa e, em seguida, para o cálculo das equigradientes, deve-se pressionar a tecla F .

B.2 - Este procedimento é semelhante ao do item (B.1), à exceção da variável indicada, que passa a ser N , quando se aperta a tecla A e, a seguir, a tecla R/S , após ser indicada a variável R no visor.

- B.3 - Este procedimento é semelhante ao descrito em B.1, à exceção da variável indicada, que passa a ser A_M ou $-A$, quando se aperta a tecla A, a seguir a tecla R/S, após ser indicada no visor a variável R, e novamente a tecla R/S, quando indicada no visor a variável N.
- B.4 - Para ser apresentado no visor o valor de N^* , ($N^* = n...n$), existente na memória, deve ser pressionada a tecla J. No caso de se desejar alterá-lo, basta introduzir pelo teclado o novo valor, pressionando-se a seguir a tecla R/S.
- B.5 - Para ser apresentado o valor de σ_D a qualquer tempo, basta ser pressionada a tecla B, sendo apresentado no visor ($\Sigma D = n...n$).
- B.6 - Para ser apresentado o valor de σ_{min} a qualquer tempo, basta ser pressionada a tecla C, sendo apresentado no visor ($\Sigma M = n...n$).
- B.7 - Para ser apresentado o valor de α , a qualquer tempo, basta ser pressionada a tecla D, sendo indicado no visor (ALFA = $n...n$).
- B.8 - Para ser trocado o modo de cálculo, de automático para discreto, e vice-versa, basta ser

pressionada a tecla E, até que o modo desejado seja indicado no visor, (MOD DISCRETO) ou (MOD AUTOM).

- B.9 - No caso de se desejar interromper a execução do programa durante os cálculos, deve-se pressionar a tecla R/S e, a seguir, a tecla F, sendo indicado no visor, conforme o modo de cálculo, os novos dados a serem introduzidos para a determinação de equigradientes. A tecla F poderá ser pressionada para introdução de novos dados a qualquer tempo, desde que o programa não esteja executando nenhum cálculo.
- B.10 - No caso de não haver convergência durante algum cálculo, o valor de ρ_0 deverá ser alterado convenientemente (vide listagem do Programa P-4, no ítem D.3.2.3.1).
- B.11 - Não havendo convergência no modo de operação automático, pressiona-se a tecla R/S, de modo a interromper a execução do programa, pressionando-se a seguir a tecla amarela e a tecla A; daí em diante, o cálculo se processa normalmente.

D.3.2.2.2 - Modo de Operação do Programa P-5

Quando o programa é utilizado pela primeira vez, deve-se introduzir na calculadora, através do teclado, as instruções indicadas sequencialmente no item (D.3.2.3.2), relativas à listagem do Programa P-5. No caso de se dispor da leitora de cartões magnéticos, estas instruções poderão ser gravadas nestes cartões, permitindo assim guardar indefinidamente o programa para utilização a qualquer tempo, mesmo apagando-se as instruções a ele relativas, da memória da calculadora.

Desde que a Etapa A tenha sido utilizada uma única vez, qualquer das etapas, em número de duas, poderá ser executada.

As etapas com os respectivos procedimentos são as seguintes:

ETAPA A - Nesta etapa, pretende-se calcular pela primeira vez, as coordenadas x e y no plano Z , relativas às curvas equigradientes obtidas pela interseção destas com linhas de força oriundas de um dos N condutores do feixe, e por fase de uma linha de transmissão, localizado no ponto $(1,0)$ do plano complexo W , efetuando-se entretanto, uma adequada mudança de variáveis, de modo a possibilitar a determinação destas interseções, através de translações dada por t , ao longo do eixo real u , conforme apresentado no desenvolvimento

teórico, no ítem (D.3.1). No programa apresentado, o ângulo δ (computado no sentido trigonométrico), é relativo às linhas de força devidas a um condutor de teste localizado no ponto $(t,0)$, ao longo do eixo u , no plano W .

Devido à simetria existente nestas condições, as interseções serão calculadas apenas para o semi-plano superior do plano W .

Esta etapa consiste do seguinte conjunto de procedimentos:

A.1 - Deve ser escolhido inicialmente, o modo de operação do programa quanto aos cálculos das interseções, que poderão ser efetuadas no modo automático, ou discreto. Para tanto, deve-se pressionar a tecla E , que alterna os mesmos, cada vez que é pressionada.

A.2 - Pressiona-se a tecla A , sendo apresentado no visor ($R = n...n$), com um valor existente nesta memória, anteriormente. Introduce-se o valor de R desejado através do teclado da calculadora, apertando-se a seguir a tecla R/S . O programa deverá executar algumas instruções, apresentando então no visor ($N = n...n$), com um valor existente na memória anteriormente. Intro-

duz-se o valor de N desejado através do teclado, apertando-se em seguida a tecla R/S. O programa deverá executar algumas instruções, apresentando no visor ($AM = n...n$), ou ($-A = n...n$), respectivamente conforme o valor existente na memória anteriormente seja positivo ou negativo. Introduce-se o valor de A_M ou $-A$ desejado, através do teclado, apertando-se em seguida a tecla R/S. Os valores de A_M , $-A$ e R introduzidos devem estar nas mesmas unidades de comprimento.

- A.3 - Após os cálculos referidos em A.2, o programa deverá executar uma série de instruções, sendo apresentado a seguir no visor ($\Sigma D = n...n$), ou seja, o valor de σ_D .
- A.4 - Apertando-se a tecla R/S a seguir, é apresentado no visor ($\Sigma M = n...n$), ou seja, o valor de σ_{min} .
- A.5 - Apertando-se a tecla R/S a seguir, é apresentado no visor ($ALFA = n...n$), ou seja, o valor de α .
- A.6 - Apertando-se a tecla R/S a seguir, é apresentado no visor ($N^* = n...n$), ou seja, o número anteriormente existente na memória, relativo ao

número de divisões do ângulo de 180^0 , ou do ângulo δ_{\max} calculado para as interseções de equigradientes com linhas de força. Introduce-se portanto o valor desejado de N^* através do teclado, apertando-se a seguir a tecla R/S.

A.7 - Apertando-se a tecla G, é apresentado no visor o valor de t anteriormente existente na memória da calculadora, ou seja, $(T = n...n)$. Introduce-se então pelo teclado, o valor de t desejado, pressionando-se a seguir a tecla R/S.

A.8 - Conforme o modo de operação, o programa requisita diferentes entradas de dados, dependendo de estar operando

(1) no modo automático

ou

(2) no modo discreto.

A.8.1 - Estando operando no modo automático (após o procedimento A.7) é apresentado no visor $(\Sigma = \quad)$, devendo ser introduzido o valor de σ desejado para os cálculos dos pontos pertencentes a esta equigradiante, apertando-se a seguir a tecla R/S.

A.8.1.1 - O programa compara o valor de σ introduzi-

do com os valores calculados de σ_{\min} e σ_D , apresentando no visor o resultado desta comparação, da seguinte forma: ($\Sigma > \Sigma \text{MIN}$), quando σ é maior do que σ_{\min} , e neste caso o programa para sua execução, devendo-se apertar a tecla F para voltar ao item A.8.1; ($\Sigma = \Sigma D$), quando σ é igual a σ_D ; ($\Sigma < \Sigma D$), quando σ é menor que σ_D , e ($\Sigma > \Sigma D$), quando σ é maior que σ_D , e menor ou igual a σ_{\min} . Desde que este programa foi elaborado para a determinação de curvas equigradientes ditas "internas", e devido à escolha do valor inicial de ρ_0 para o processo iterativo de convergência, não é conveniente introduzir-se valores de σ que não muito próximos de σ_D , no caso de serem maiores, pois a convergência poderá ser lenta ou não se processar. Nestas condições, deverá ser empregado o Programa P-4. Daí portanto a utilidade de se efetuar esta comparação de valores de σ , antes dos cálculos de interseções propriamente ditos.

- A.8.1.2 - O programa apresenta a seguir no visor ($\delta = n \dots n$), indicando o valor do ângulo δ_i para o cálculo das interseções no plano complexo W, o qual será dividido por N^* ,

sendo calculadas as interseções desde $\delta_i = 180^\circ$, até $\delta_i = 0^\circ$, onde $\delta_i = 180^\circ - (180^\circ/N^*).i$, para $i = 0, 1, 2, \dots, N^*$.

A.8.1.3 - O programa apresenta durante o processo iterativo de convergência, os valores de $(\rho_{i+1} - \rho_i)$, no visor, ou seja, (n...n), indicando posteriormente no visor os valores das coordenadas x e y de cada ponto de interseção com a equigradiente relativa ao valor de σ introduzido, ou seja, (X = n...n) e (Y = n...n). O processo é repetido conforme descrito em A.8.1.1 a A.8.1.3, até que $\delta_i = 0^\circ$, quando o programa finaliza os cálculos e volta ao procedimento A.8.1, à espera de serem introduzidos novos dados.

A.8.2 - Estando no modo discreto, e após o procedimento A.7, é indicado no visor ($T = n...n$ = ...), devendo ser introduzido o ângulo δ_0 , segundo o qual se deseja fazer a interseção com a equigradiente σ , cujo valor deverá ser introduzido logo a seguir, pressionando-se após a tecla R/S. É então apresentado no visor ($\Sigma = n...n$), devendo-se introduzir o valor de σ desejado através do teclado, apertando-se a seguir a tecla R/S. Desde que o programa não verifica a compatibilidade des-

te valor de ângulo com o valor de σ introduzido, e levando-se ainda em consideração a translação t , pode acontecer de não ocorrer convergência durante o processo iterativo por duas razões: não existe realmente a interseção considerada ou o valor de ρ_0 não é adequado, devendo-se neste caso alterá-lo convenientemente conforme o valor de σ .

A.8.2.1 - O programa apresenta no visor ($\underline{\quad} = n\dots n$), ou seja, o valor de δ_0 introduzido.

A.8.2.2 - A seguir, o programa apresenta no visor, durante o processo iterativo de convergência, os valores de $(\rho_{i+1} - \rho_i)$, ou seja, ($n\dots n$), indicando posteriormente ainda no visor os valores das coordenadas x e y de cada ponto de interseção com a equigradiante relativa ao valor de σ introduzido, ou seja, ($X = n\dots n$) e ($Y = n\dots n$). O processo de cálculo é finalizado, voltando o programa ao procedimento A.8.2, à espera de novos dados.

ETAPA B - Nesta etapa possibilita-se a apresentação de qualquer dos dados de entrada introduzidos para o cálculo das equigradiantes, (R, N, A_M ou $-A$), optando-se por alterá-los, ou não, bem como verificar alguns

valores particulares utilizados nos cálculos. Os itens B.1 a B.9, constantes do modo de operação do Programa P-4, relativos à Etapa B, são válidos para a Etapa B do Programa P-5, incluindo-se ainda os seguintes procedimentos:

- B.10 - No caso de não haver convergência durante algum cálculo em que se constate não ser possível a existência de interseção com equigradiente segundo um dado ângulo, o valor de δ_0 deverá ser alterado convenientemente (vide listagem do Programa P-5, no item D.3.2.3.2).
- B.11 - Para ser apresentado no visor o valor de t , ($T = n...n$), existente na memória, deve ser pressionada a tecla G. No caso de desejar alterá-lo, bastará introduzir através do teclado o valor desejado, sendo pressionada em seguida a tecla R/S.
- B.12 - Para que durante os cálculos de equigradientes o valor de σ_D seja exatamente igual ao calculado, basta ser pressionada a tecla B, sendo apresentado no visor ($\Sigma D = n...n$) e, a seguir, a tecla F, para continuação dos cálculos em qualquer dos modos. Após se introduzir o valor de δ_0 , se for o caso, é apresentado no visor o valor de σ , ($\Sigma = n...n$). No caso

do cálculo com $\sigma = \sigma_D$, ou quando o valor de σ apresentado não precisar ser alterado, deverá ser pressionada a tecla R/S.

B.14 - Tanto no modo automático quanto discreto, o valor de σ apresentado no visor é sempre o valor utilizado no cálculo imediatamente anterior. Não sendo necessário alterá-lo em cálculos posteriores, e para dar prosseguimento da execução do programa, basta ser pressionada a tecla R/S, após sua indicação no visor.

B.15 - A tecla I, quando pressionada, permite que sejam reapresentados no visor os valores das coordenadas x e y, quando o programa estiver sendo executado no modo discreto.

B.16 - Não havendo convergência no modo de operação automático, pressiona-se a tecla R/S, para interromper o programa executado, pressionando-se a seguir a tecla amarela e a tecla A; daí em diante o cálculo se processa normalmente.

D.3.2.3 - Listagens dos Programas P-4 e P-5D.3.2.3.1 - Listagem do Programa P-4

1	LBL 'P4	+	'N*=	y ^x
	LBL A	RCL 04	CF 29	RCL 03
	'R=	y ^x	FIX 0	1/x
	1	RCL 03	RCL 09	1
	XEQ 05	X	ARCL X	+
	'N=	STO 06	SF 29	X
	2	1/x	PROMPT	x ²
	XEQ 05	RCL 02	STO 09	RCL 03
	'AM=	1	GTO F	x ²
10	0	50 -	90 LBL E	130 1/x
	XEQ 05	RCL 04	'MOD-DISC.	-
	RCL 00	y ^x	FS?C00	1
	X > 0?	RCL 02	GTO 01	-
	GTO 00	:	SF 00	RCL 03
	ABS	X	'MOD AUTO	X
	180	STO 07	LBL 01	2
	RCL 02	RCL 03	PROMPT	:
	:	XEQ 09	LBL F	ACOS
	SIN	1/x	RCL 06	FC?00
20	2	60 RCL 04	100 STO 12	140 GTO 04
	X	y ^x	'∠ =	STO 10
	:	STO 08	FC?00	GTO 03
	LBL 00	LBL B	PROMPT	LBL 04
	RCL 01	'ΣD=	STO 10	X ≤ Y
	:	FIX 6	'Σ=	X ≤ Y?
	XEQ 09	ARCL 07	PROMPT	GTO 03
	RCL 02	PROMPT	STO 15	FIX 4
	y ^x	LBL C	STOX12	'∠ MAX=
	XEQ 09	'ΣM=	RCL 08	ARCL Y
30	STO 03	70 FIX 6	110 X < Y?	150 PROMPT
	1	ARCL 08	GTO 02	GTO F
	STO 04	PROMPT	180	LBL 03
	RCL 02	LBL D	FS?00	RCL 10
	1/x	'ALFA=	STO 10	RCL 09
	STO-04	FIX 6	GTO 03	:
	+	RCL 03	LBL 02	STO 11
	STO 05	1/x	RCL 10	LBL G
	RCL 03	ARCL X	RCL 15	RCL 10
	1/x	PROMPT	RCL 04	FIX 6
40	1	80 LBL J	120 1/x	160 RND

continua...

D.3.2.3.1 - Listagem do Programa P-4 - Continuação

161	X < 0?	STOX13	+	STO:14
	GTO F	STOX14	R-P	RCL 14
	X=0?	LAST X	RCL 02 284	END
	STO 10	+	1/x	
	FIX 4	X	y ^x	
	'< =	1	X ≥ Y	
	ARCL 10	+	RCL 02	
	AVIEW	STO-13	:	
	,9	RCL 05	X ≥ Y	
170	STO 13	210 y ^x	250 P-R	
	RCL 15	SQRT	'X=	
	RCL 07	RCL 12	ARCL X	
	X > Y?	X	AVIEW	
	GTO 06	STO-14	PSE	
	X≠Y?	RCL 13	PSE	
	GTO 07	RCL 14	'Y=	
	RCL 10	STO:13	ARCL Y	
	180	:	AVIEW	
	X ≠ Y?	R↑	PSE	
180	GTO 06	220 -	260 PSE	
	RCL 02	ABS	FC?00	
	STO 13	FIX 6	GTO F	
	GTO 08	RND	LBL a	
	LBL 06	'R=	RCL 11	
	2	ARCL 13	STO-10	
	RCL 12	AVIEW	GTO G	
	:	X=0?	RTN	
	RCL 02	GTO 08	LBL 05	
	y ^x	RCL 16	ABS	
190	STO 13	230 X=Y?	270 RCL IND X	
	LBL 07	GTO 08	X < 0?	
	RCL 04	X ≥ Y	'-A=	
	STO 14	STO 16	ARCL X	
	X ≥ 13	GTO 07	PROMPT	
	STOX13	LBL 08	STO IND L	
	RCL X	FIX 4	RTN	
	RCL X	RCL 10	LBL 09	
	RCL 10	RCL 13	STO 14	
	COS	P-R	1	
200	+	240 1	280 STO+14	
			-	

D.3.2.3.2 - Listagem do Programa P-5

1	LBL 'P5	:	ARCL 03	RCL 14
	LBL A	STO 04	PROMPT	ARCL X
	'R=	1/x	LBL J	PROMPT
	19	y ^x	'N*=	STO 14
	XEQ 05	RCL 03	CF 29	STO 12
	'N=	:	FIX 0	RCL 17
	2	1/x	RCL 16	X > Y?
	XEQ 05	RCL 02	ARCL X	GTO 02
	'AM=	1	SF 29	'Σ>Σ MIN
10	20	50 -	90 PROMPT	130 PROMPT
	XEQ 05	RCL 04	STO 16	GTO F
	RCL 20	1/x	GTO F	LBL 02
	X > 0?	y ^x	LBL E	RDN
	GTO 00	RCL 02	'MOD DIS.	RCL 07
	ABS	:	FS?C00	'Σ>ΣD
	180	X	GTO 01	X > Y?
	RCL 02	STO 07	SF 00	'Σ<ΣD
	:	RCL 03	'MOD AUTOM	X=Y?
	SIN	1/x	LBL 01	'Σ=ΣD
20	2	60 XEQ 09	100 PROMPT	140 AVIEW
	X	1/x	LBL G	RCL 06
	:	RCL 04	RCL 06	x ²
	LBL 00	1/x	CHS	STO 18
	RCL 19	y ^x	'T=	RCL 06
	:	STO 17	FIX 3	1
	XEQ 09	LBL B	ARCL X	+
	RCL 02	'ΣD=	PROMPT	x ²
	y ^x	FIX 6	CHS	STO 09
	XEQ 09	RCL 07	STO 06	RCL 14
30	1/x	70 STO 14	110 LBL F	150 RCL 03
	STO 03	ARCL X	'T=	:
	1	PROMPT	FIX 3	RCL 04
	+	LBL C	RCL 06	STO 13
	RCL 02	'ΣM=	CHS	2
	1	FIX 6	ARCL X	X
	-	ARCL 17	'-_/_ =	y ^x
	1/x	PROMPT	FC?00	RCL 03
	RCL 02	LBL D	PROMPT	1
	1/x	'ALFA =	STO 10	+
40	STO 05	80 FIX 6	120 'Σ=	160 x ²

continua

D.3.2.3.2 - Listagem do Programa P-5 - Continuação

161	X		RCL 06	-		RCL 10		STO 14
	STO 12		ABS	X		COS		1
	STOX13		LAST X	RCL 08		RCL 01		STO+14
	180		-	2		X		-
	FC?00		X > 0?	:		RCL 06		STO:14
	GTO H		STO 01	+		-		RCL 14
	STO 10		RCL 14	RCL X		X	328	END
	RCL 16		RCL 07	RCL 01		ACOS		
	:		X≠Y?	STOXZ		RCL 00		
170	STO 11	210	GTO 07	250	RCL 08	290	RCL 02	
	LBL H		RCL 10		X		1/x	
	RCL 10		180		RCL 00		STOXZ	
	FIX 6		X≠Y?	+			y ^x	
	RND		GTO 07		RCL 18		P-R	
	X < 0?		RCL 02		-		'X=	
	GTO F		1		STO-Z		ARCL X	
	X=0?		-		SQRT		AVIEW	
	STO 10		STO 01		-		PSE	
	FIX 4		GTO I		:		PSE	
180	'< =	220	LBL 07	260	RCL 01	300	'Y=	
	ARCL 10		RCL 12		X ≥ Y		ARCL Y	
	AVIEW		STO 00		STO 01		AVIEW	
	RCL 10		RCL 13		-		PSE	
	COS		RCL 01		FIX 5		PSE	
	2		RCL 15		RND		FC?00	
	X		-		VIEW X		GTO F	
	STO 08		RCL 01		X≠0?		LBL a	
	RCL 06		X		GTO 07		RCL 11	
	STOX08		RCL 09		LBL I		STO-10	
190	1	230	+	270	FIX 4	310	GTO H	
	+		RCL 04		RCL 01		RTN	
	X		y ^x		RCL 08		LBL 05	
	STO 15		STOX00		-		ABS	
	2		RCL 05		RCL 01		RCL IND X	
	1/x		y ^x		X		X < 0?	
	RCL 02		X		RCL 18		'-A=	
	1		RCL 01		+		ARCL X	
	-		RCL 15		SQRT		PROMPT	
	X		2		STO 00		STO IND L	
200	STO 01	240	:	280	1/x	320	RTN	
							LBL 09	

D.3.2.4 - Casos Teste para os Programa P-4 e P-5

No ítem (D.3.2.4.1) apresenta-se um caso teste para o Programa P-4, e, em (D.3.2.4.2), um caso teste para o Programa P-5.

D.3.2.4.1 - Caso Teste para o Programa P-4

Considere-se como exemplo para a Etapa A, os seguintes dados, desejando-se executar o programa no modo automático:

$$A = 10 \text{ cm};$$

$$R = 1,5 \text{ cm};$$

$$N = 2;$$

$$N^* = 4;$$

$$\sigma = 0,6.$$

Observa-se a seguinte sequência de instruções:

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA A
E	MOD DISCRETO	A.1
E	MOD. AUTOM.	A.1
A	R = n...n	A.2
1,5	1,5	A.2
R/S		A.2
	N = n...n	A.2
2	2	A.2
R/S		A.2
	AM=n...n(se o valor existente na memória for positivo)	A.2
-10	-10	A.2
R/S		A.2
	$\Sigma D=0,221037$	A.3
R/S		A.4
	$\Sigma M=0,538462$	A.4
		A.5
	ALFA=0,550459	A.5
R/S		A.6
	N* = n...n	A.6
4	4	A.6
R/S		A.6
	$\Sigma =$	A.7.1
0,6	0,6	A.7.1
R/S		A.7.1

O quadro a seguir resume as indicações e resultados, conforme apresentados no visor:

INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA A
$\angle = 154,2335$	A.7.1.1
R0 = 0,486 086	A.7.1.2
R0 = 0,550 040	A.7.1.2
R0 = 0,550 459	A.7.1.2
R0 = 0,550 459	A.7.1.2
X = 0,7288	A.7.1.2
Y = 0,1642	A.7.1.2
$\angle = 115,6751$	A.7.1.1
R0 = 0,710 391	A.7.1.2
R0 = 0,716 083	A.7.1.2
R0 = 0,716 089	A.7.1.2
R0 = 0,716 089	A.7.1.2
X = 0,9040	A.7.1.2
Y = 0,3570	A.7.1.2
$\angle = 77,1167$	A.7.1.1
R0 = 0,898 027	A.7.1.2
R0 = 0,898 027	A.7.1.2
X = 1,1588	A.7.1.2
Y = 0,3777	A.7.1.2
$\angle = 38,5584$	A.7.1.1
R0 = 1,016 956	A.7.1.2
R0 = 1,016 512	A.7.1.2
R0 = 1,016 512	A.7.1.2
X = 1,3598	A.7.1.2
Y = 0,2330	A.7.1.2
$\angle = 0,0000$	A.7.1.1
R0 = 1,057 756	A.7.1.2
R0 = 1,056 625	A.7.1.2
R0 = 1,056 625	A.7.1.2
X = 1,4341	A.7.1.2
Y = 0,0000	A.7.1.2
$\Sigma =$	A.7.1

Como exemplo para a Etapa B, deseja-se o seguinte:

- executar o programa no modo discreto
- alterar o valor de A, para 12 cm
- verificar o valor de σ_{\min}
- verificar o valor de α
- introduzir o ângulo δ_0 igual a 75°
- introduzir σ igual a 0,95.

Observa-se a seguinte sequência de instruções:

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA B(A)
E	MOD DISCRETO	B.8
A	R = 1,5000	B.3
R/S		B.3
	N = 2,0000	B.3
R/S		B.3
	-A = -10,0000	B.3
-12	-12	B.3
R/S		B.3
	$\Sigma D = 0,194\ 029$	B.3
C	$\Sigma M = 0,600\ 000$	B.6
D	ALFA = 0,470 588	B.7
F	$\delta =$	B.9
75	75	B.9
R/S	$\Sigma =$	B.9 (A.7.2)
0,95	0,95	B.9 (A.7.2)

continua...

continuação

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA B(A)
R/S	δ MAX=54,9862	B.9 (A.7.2)
R/S ou F	δ =	B.9 (A.7.2.1 e 2)
54	54	B.9 (A.7.2)
R/S	Σ =	B.9 (A.7.2)
0,95	0,95	B.9 (A.7.2)
R/S		B.9 (A.7.2.1)

Como se pode verificar pelo quadro anterior, o ângulo δ_0 de 75° não era compatível com o valor de σ igual a 0,95. Neste caso, o programa apresentou no visor o valor máximo de δ_0 , igual a $54,9862^\circ$. Assim, introduziu-se o valor de δ_0 de 54° , passando-se a observar os seguintes resultados, conforme apresentados no visor da calculadora:

INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA A
δ = 54,0000	A.7.2.3
RO = 0,472 657	A.7.2.4
RO = 0,471 610	A.7.2.4
RO = 0,471 610	A.7.2.4
X = 1,1424	A.7.2.4
Y = 0,1670	A.7.2.4
δ =	A.7.2

D.3.2.4.2 - Caso Teste para o Programa P-5

Considere-se como exemplo para a Etapa A, os seguintes dados, desejando-se executar o programa no modo automático:

A = 10 cm ;

R = 1,5 cm;

N = 2;

N* = 4;

σ = 0,2;

t = 0,0.

Observa-se a seguinte seqüência de instruções:

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA A
E	MOD DISCRETO	A.1
E	MOD AUTOM	A.1
A	R = n...n	A.2
1,5	1,5	A.2
R/S		A.2
	N = n...n	A.2
2	2	A.2
R/S		A.2
	AM=n...n (se o valor existente na memória for positivo)	A.2
-10	-10	A.2
R/S		A.2

continua...

continuação

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA A
R/S	$\Sigma D = 0,221\ 037$	A.3 A.4
R/S	$\Sigma M = 0,538\ 462$	A.4 A.5
R/S	$ALFA = 0,550\ 459$	A.6 A.6
4	$N^* = n \dots n$	A.6
R/S	4	A.6
G	$\Sigma = 0,221$	A.8.1
0,0	$T = n \dots n$	A.7
R/S	0,0	A.7
0,2	$\Sigma = 0,221$	A.8.1
R/S	0,2	A.8.1 A.8.1

O quadro seguinte resume as indicações e resultados, conforme apresentados no visor:

INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA A
$\Sigma < \Sigma D$	A.8.1.1
$\angle = 180,0000$	A.8.1.2
0,10228	A.8.1.3
-0,00500	A.8.1.3
-0,00001	A.8.1.3
0,00000	A.8.1.3
$X = 0,0000$	A.8.1.3
$Y = 0,6346$	A.8.1.3
$\angle = 135,0000$	A.8.1.2

continua...

continuação

INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA A
0,19654	A.8.1.3
-0,01348	A.8.1.3
-0,00006	A.8.1.3
0,00000	A.8.1.3
X = 0,2155	A.8.1.3
Y = 0,5202	A.8.1.3
$\angle = 90,0000$	A.8.1.2
0,30699	A.8.1.3
-0,02094	A.8.1.3
-0,00010	A.8.1.3
X = 0,3272	A.8.1.3
Y = 0,3272	A.8.1.3
$\angle = 45,0000$	A.8.1.2
0,35849	A.8.1.3
-0,02136	A.8.1.3
-0,00008	A.8.1.3
0,00000	A.8.1.3
X = 0,3729	A.8.1.3
Y = 0,1545	A.8.1.3
$\angle = 0,0000$	A.8.1.2
0,37257	A.8.1.3
-0,02093	A.8.1.3
-0,00007	A.8.1.3
0,0000	A.8.1.3
X = 0,3853	A.8.1.3
Y = 0,0000	A.8.1.3
$\Sigma = 0,200$	A.8.1

Como exemplo para a Etapa B, deseja-se o seguinte:

- executar o programa no modo discreto

- alterar o valor de A, para 12 cm
- verificar o valor de σ_{\min}
- verificar o valor de α
- alterar o valor de t, para - 3,0
- introduzir o ângulo δ_0 igual a 110°
- Introduzir o valor de σ igual a σ_D .

Observa-se a seguinte sequência de instruções:

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA B(A)
E	MOD DISCRETO	B.8
A	R = 1,5000	B.3
R/S		B.3
	N = 2,0000	B.3
R/S		B.3
	-A = -10,0000	B.3
-12	-12	B.3
R/S		B.3
	$\Sigma D = 0,194\ 029$	B.3
C	$\Sigma M = 0,600\ 000$	B.6
D	ALFA = 0,470 588	B.7
G	T = 0,000	B.11
-3,0	-3,0	B.11
R/S		B.11
	T = -3,000 δ =	B.11
B	$\Sigma D = 0,194\ 029$	B.12
F	T = -3,000 δ =	B.12 (A.8.2)
110	110	(A.8.2)
R/S	$\Sigma = 0,194$	(A.8.2)
R/S		(A.8.2)

O quadro a seguir resume as indicações e resultados, conforme apresentados no visor:

INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA B(A)
$\Sigma = \Sigma D$	(A.8.1.1)
$\angle = 110,0000$	(A.8.2.1)
298,40391	(A.8.2.2)
-149,29418	(A.8.2.2)
-74,67364	(A.8.2.2)
-37,39077	(A.8.2.2)
-18,80678	(A.8.2.2)
-9,63930	(A.8.2.2)
-5,30352	(A.8.2.2)
-3,07974	(A.8.2.2)
-18,20846	(A.8.2.2)
R/S	B.9
F	B.9

Como se pode observar, o processo iterativo não estava convergindo, o que obrigou a uma interrupção na execução do programa. Neste caso, não há realmente interseção com a equigradiente associada a σ igual a σ_D segundo o ângulo de 110° . Considerando-se então o ângulo de 2° , observa-se os seguintes resultados, conforme apresentados no visor da calculadora:

INDICAÇÃO NO VISOR	ITEM CORRESPONDENTE DA ETAPA B(A)
$\Sigma = \Sigma D$	(A.8.1.1)
$\Delta = 2,0000$	(A.8.2.1)
-0,58645	(A.8.2.2)
-0,31567	(A.8.2.2)
-0,21842	(A.8.2.2)
-0,14989	(A.8.2.2)
-0,09423	(A.8.2.2)
-0,05058	(A.8.2.2)
-0,01927	(A.8.2.2)
-0,00319	(A.8.2.2)
-0,00009	(A.8.2.2)
0,00000	(A.8.2.2)
X = 0,0328	(A.8.2.2)
Y = 1,0317	(A.8.2.2)
T = -3,000 $\Delta =$	(A.8.2)

O tempo de execução dos Programas P-4 e P-5, é da ordem de 13 segundos para a apresentação de cada ponto de interseção no modo automático, e de 17 segundos, no modo discreto.

APÊNDICE E

CÁLCULO DA MÉDIA DOS VALORES DE CAMPO ELÉTRICO MÁXIMO
NA SUPERFÍCIE DE N CONDUTORES DE UM FEIXE DA FASE
CENTRAL DE UMA LINHA DE TRANSMISSÃO COM
DISPOSIÇÃO HORIZONTAL

Para a determinação do valor de campo elétrico máximo, nas condições acima mencionadas, elaborou-se o Programa P-6, implantado na calculadora HP-41C.

Neste cálculo, uma série de grandezas auxiliares devem ser obtidas em passos intermediários, incluindo-se o valor de R_e , calculado através do Programa P-3, apresentado no Apêndice C, boa parte do qual é aproveitado no Programa P-6.

No item (E.1) deste apêndice, são apresentadas as grandezas auxiliares envolvidas no cálculo do campo elétrico \bar{E} , para a elaboração do Programa P-6, e, no item (E.2), é apresentado o Programa P-6, incluindo seu modo de utilização e um caso teste.

E.1 - GRANDEZAS AUXILIARES

Reproduzem-se a seguir, as fórmulas empregadas no cálculo de grandezas auxiliares necessárias para a determinação do valor de campo elétrico máximo, em alguns casos com pe-

quenas modificações. Essas fórmulas foram retiradas dos Capítulos III e IV e do Apêndice C, e por ordem de cálculo no Programa P-6, são:

$$A_M = \left[\frac{A}{2 \cdot \text{sen}(180^\circ/N)} \right] , \quad |3.23|$$

$$\alpha^{-1} = \frac{\left[\frac{(A_M/R) + 1}{(A_M/R) - 1} \right]^N + 1}{\left[\frac{(A_M/R) + 1}{(A_M/R) - 1} \right]^N - 1} , \quad |3.29|$$

$$R_C = A_M \cdot \sqrt[N]{\alpha} \quad |4.8|$$

$$\beta = \ln(H_1/R_C) , \quad |4.16|$$

onde

$$H_1 = 2H ,$$

$$\beta' = \ln \sqrt{1 + (1/\lambda)^2} , \quad |4.17|$$

$$\beta'' = \ln \sqrt{1 + 1/(2\lambda)^2} , \quad |4.18|$$

onde

$$\lambda = (D/H_1),$$

$$\gamma = \left[\frac{\frac{2 \cdot \alpha}{(1 + \alpha)}}{1 - \sqrt{\frac{1 - \alpha}{1 + \alpha}}} \right] . \quad |4.54|$$

Das equações |C.2| e |C.5| obtem-se

$$A' = \left[\frac{\gamma \cdot R}{H_1 \cdot \left(\frac{\beta + d}{\beta^2 - a + \beta'' \cdot \beta} \right)} \right] ,$$

$$Q_i = 2 \cdot \pi \cdot \varepsilon \cdot V \cdot \left(\frac{d - y}{y^2 - \beta'' \cdot y - a} \right) ,$$

onde

$$a = 2 \cdot \beta'^2$$

$$d = \beta' - \beta''^2 ;$$

$$R_e/R = \gamma \cdot (Q_e/Q_B) , \quad |4.53|$$

$$y = \ln(R_i/H_1) .$$

Das equações |4.26| a |4.29| e |4.35|, obtem-se

$$|x_1| = \left| \frac{\beta+2 \cdot \beta'}{2 \cdot (\beta + d)} \right| ,$$

e, finalmente da Eq. |4.56|, o valor médio de campo elétrico máximo na superfície de N condutores de um feixe da fase central de uma linha de transmissão com disposição horizontal, \bar{E} , é dado por

$$\bar{E} = \frac{\sqrt{3/2} \cdot V}{R_e \cdot \{ |x_1| \cdot [(\ln H_1/R_e - \beta') + \beta'' - \beta'] + \ln H_1/R_e - \beta' \}}$$

onde

V é o valor eficaz da tensão fase-fase.

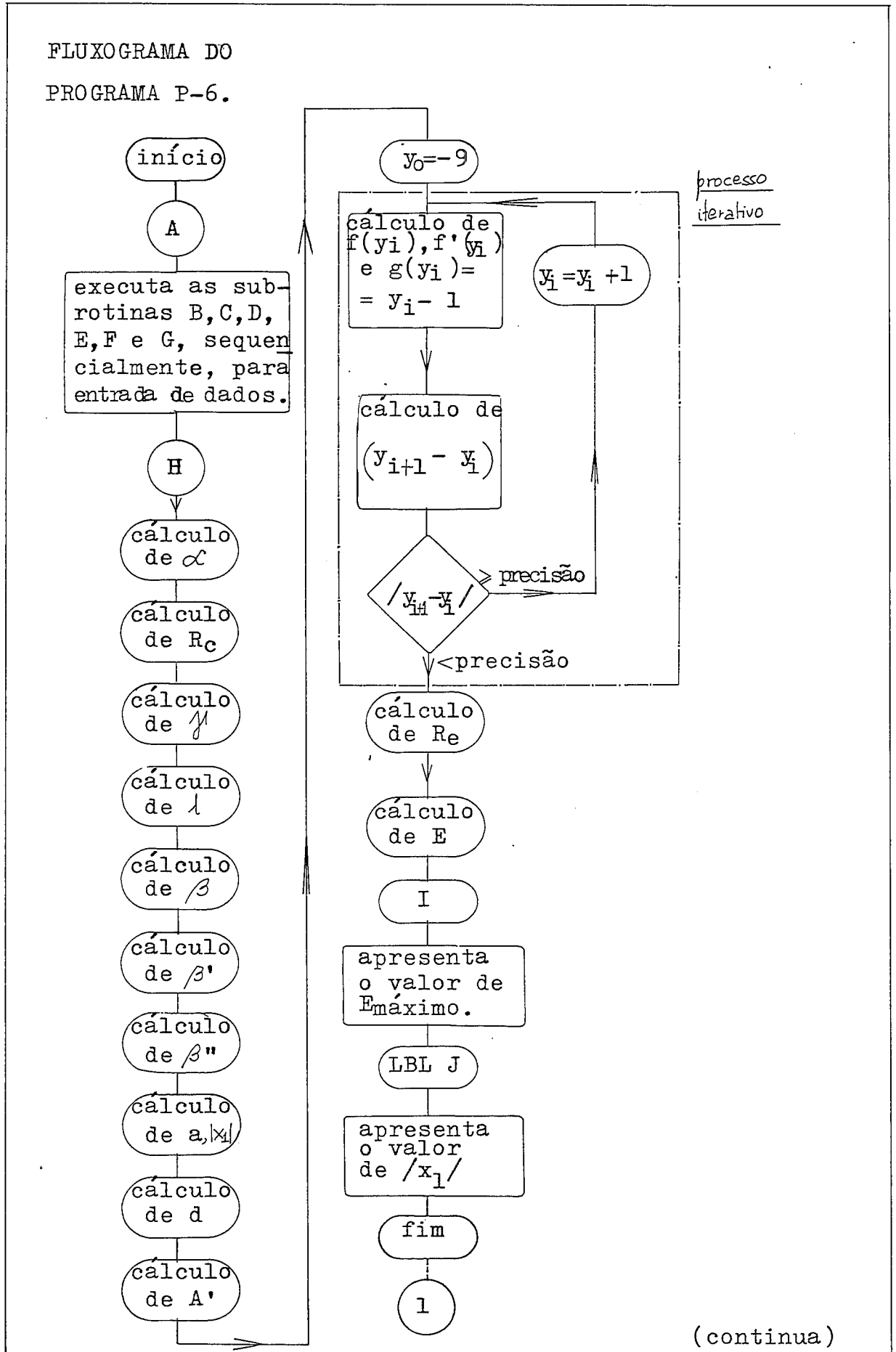
E.2 - PROGRAMA P-6

Para a obtenção do valor de \bar{E} , utilizando-se as fórmulas apresentadas no item (E.1), elaborou-se o Programa P-6.

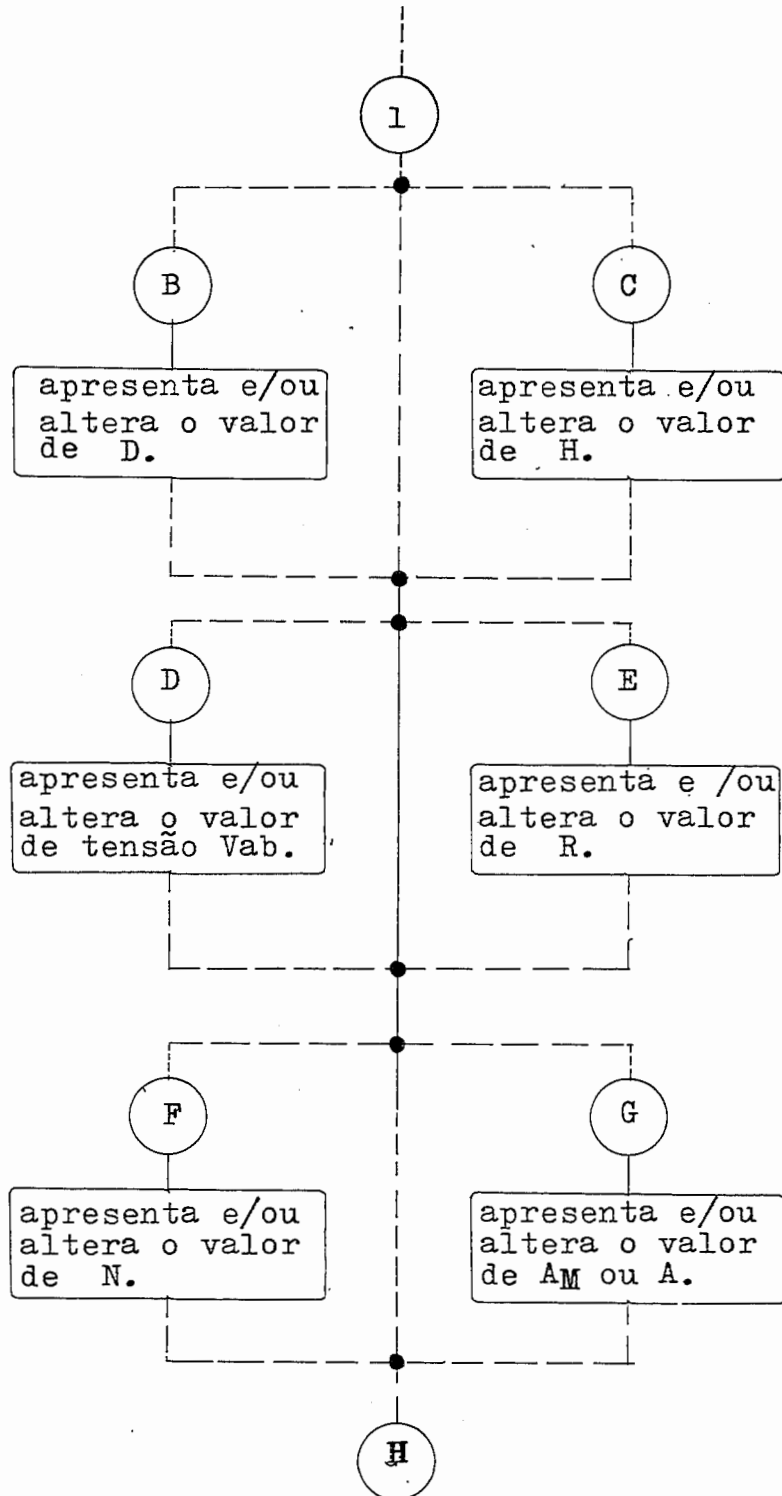
Para utilizá-lo convenientemente, é apresentado no item (E.2.1) o fluxograma, no item (E.2.2), a descrição do modo de operação, e no item (E.2.3), a respectiva listagem.

Este programa foi implantado na calculadora HP-41C, empregando-se cartões magnéticos, e um caso teste é apresentado no item (E.2.4).

E.2.1 - Fluxograma do Programa P-6



FLUXOGRAMA DO PROGRAMA P-6. (continuação)



E.2.2 - Modo de Utilização do Programa P-6

É apresentado a seguir, por etapas, o modo de utilização do Programa P-6.

Quando o programa é utilizado pela primeira vez, deve-se introduzir na calculadora, através do teclado, as instruções indicadas sequencialmente no item E.2.3 relativas à listagem do programa P-6. No caso de se dispor da leitora de cartões magnéticos, estas instruções poderão ser gravadas nestes cartões, permitindo assim guardar indefinidamente o programa para utilização posterior, mesmo apagando-se as instruções a ele relativas, da memória da calculadora. Em seguida, deverão ser executadas as instruções relativas à Etapa A.

Desde que a Etapa A tenha sido utilizada uma única vez, qualquer das Etapas, em número de 2, pode ser executada.

As etapas com as respectivas instruções, são as seguintes:

ETAPA A - Nesta etapa, pretende-se calcular pela primeira vez, o valor de \bar{E} . Consiste do seguinte conjunto de procedimentos:

A.1 - Pressiona-se a tecla A, sendo apresentado no visor ($D =$), cujo valor deverá ser

introduzido pelo teclado. Em seguida, a tecla R/S deverá ser pressionada.

- A.2 - É apresentado no visor ($H = \quad$), cujo valor deverá ser introduzido pelo teclado. Em seguida, a tecla R/S deverá ser pressionada.
- A.3 - É apresentado no visor ($AM = \quad / -A = \quad$), um dos 2 valores devendo ser introduzido pelo teclado. Introduzindo-se um valor positivo, o programa interpreta-o como sendo o valor correspondente a A_M ; no caso deste valor ser negativo, o programa interpreta-o como sendo o valor correspondente a $-A$. Em, seguida, a tecla R/S deverá ser pressionada.
- A.4 - É apresentado no visor ($R = \quad$), cujo valor deverá ser introduzido pelo teclado. Em seguida a tecla R/S deverá ser pressionada. A unidade relativa ao valor de R introduzido, será a do valor de R_e calculado pelo Programa P-6, em passos intermediários.
- A.5 - É apresentado no visor ($N = \quad$), cujo valor deverá ser introduzido pelo teclado. Em seguida a tecla R/S deverá ser pressionada.

A.6 - \bar{E} é apresentado no visor ($V_{ab} = \quad$), cujo valor, em kV, deverá ser introduzido pelo teclado. Em seguida, a tecla R/S deverá ser pressionada, dando início ao cálculo indicado no fluxograma, para obtenção de \bar{E} , $|x_1|$. Neste caso, a tecla R/S poderá ser substituída pela tecla H, para início dos cálculos.

Neste Etapa A, havendo um dado qualquer introduzido em posição trocada, a tecla A deverá ser apertada, repetindo-se a sequência de entrada de dados acima descrita.

Após o procedimento A.6, o programa executa uma série de instruções, indicando posteriormente valores relativos à convergência do processo iterativo de cálculo de R_e , indicado no fluxograma.

Uma vez que o valor 0 (zero) tenha sido indicado no visor, ou seja, havendo convergência do processo iterativo, o valor de R_e é calculado e utilizado no cálculo de \bar{E} , que é então apresentado no visor, em kV por unidade de comprimento utilizada para os valores de D, H, A_M ou -A, e R. Por exemplo, sendo estes valores fornecidos em cm, \bar{E} é dado em kV/cm. Apertando-se em seguida a tecla R/S, ou J, o valor de $|x_1|$ é apresentado no visor. Os valores de \bar{E} e $|x_1|$ podem ser apresentados novamente, bastando apertar, respectivamente, as teclas I e J.

ETAPA B - Neste etapa possibilita-se a apresentação de qualquer dos dados de entrada introduzidos para o cálculo de \bar{E} , e $|x_1|$, optando-se por alterá-los, ou não. Esta etapa consiste dos seguintes procedimentos, todos, ou parte deles, podendo ser executados em qualquer ordem:

- B.1 - Apertando-se a tecla B, apresenta-se no visor o valor de D existente na memória, na unidade previamente escolhida. Desejando-se alterar o valor de D, deve-se introduzir pelo teclado o novo valor, pressionando-se a seguir, a tecla R/S. No caso de simples verificação do valor anterior, pode-se executar qualquer um dos procedimentos desta etapa e, em seguida, para calcular-se o novo valor de \bar{E} , e de $|x_1|$, deve-se pressionar a tecla H.
- B.2 - Este procedimento é igual ao do item B.1, à exceção da variável indicada, que passa a ser H, quando se aperta a tecla C.
- B.3 - Este procedimento é igual ao descrito no item B.1, à exceção da variável indicada, que passa a ser A_M ou -A, quando se aperta a tecla D.
- B.4 - Este procedimento é igual ao descrito no item B.1, à exceção da variável indicada, que passa a ser R, quando se aperta a tecla E.

B.6 - Este procedimento é igual ao descrito no item B.1, à exceção da variável indicada, que passa a ser N, quando se aperta a tecla F.

B.7 - Este procedimento é igual ao descrito no item B.1, à exceção da variável indicada, que passa a ser Vab, quando se aperta a tecla G.

E.2.3 - Listagem do Programa P-6

1	LBL 'P6	y^x	-	$X \geq Y$	1,5
	LBL A	$1/x$	RCL 00	$1 \geq Y$	SQRT
	XEQ B	1	RCL 07	+	X
	XEQ C	$X \geq Y$	+	:	STO 10
	XEQ E	-	:	CHS	LBL I
	XEQ F	X	STOX09	RCL 10	FIX 2
	XEQ D	STO:09	LAST X	+	RCL 10
	XEQ G	RCL 02	$1/x$	RCL 10	RCL 06
	LBL H	STO:09	RCL 11	$X \geq Y$	X
10	'	50 2	90 RCL 00	130 STO 10	170 'E=
	AVIEW	STO 12	2	-	ARCL X
	RCL 05	X	:	FIX 7	AVIEW
	$X > 0?$	STO 10	+	RND	RTN
	GTO 08	RCL 00	X	SCI 2	LBL J
	ABS	:	ABS	CLA	/X1/=
	180	LN	STO 00	ARCL X	FIX 4
	RCL 04	STO 00	-9	AVIEW	ARCL 00
	:	RCL 01	STO 10	$X \neq 0?$	AVIEW
	SIN	STO:10	LBL 01	GTO 01	RTN
20	2	60 RCL 10	100 RCL 10	140 AVIEW	180 LBL B
	X	1	RCL 08	RCL 10	'D=
	:	R-P	-	RCL 11	1
	LBL 08	LN	RCL X	+	GTO 02
	STO 00	STO 11	RCL 10	RCL X	LBL C
	RCL 03	STO 07	STO+Z	RCL 08	'H=
	STO 09	x^2	X	-	2
	:	STOX12	RCL 12	RCL 11	GTO 02
	XEQ 03	RCL 10	-	+	LBL E
	y^x	2	STO:Y	RCL 00	'R=
30	XEQ 03	70 :	110 RCL 07	150 X	190 3
	$1/x$	1	RCL 10	+	GTO 02
	CHS	R-P	-	ABS	LBL F
	y^x	LN	$1/x$	RCL 10	'N=
	STOX00	STO+07	STO+Z	e^x	4
	RCL 11	STO 08	X	X	GTO 02
	1	RCL 00	RCL 09	RCL 02	LBL D
	+	+	:	2	'Vab=
	RCL 11	RCL 00	LN	X	6
	XEQ 03	X	RCL 10	X	GTO 02
40	$1/x$	80 RCL 12	120+	160 $1/x$	200 LBL G

continua

E.2.3.2 - Listagem do Programa P-6 - Continuação

```

201 'AM=
    5
    LBL 02
    FIX 4
    ABS
    RCL IND X
    X < 0?
    '-A=
    ARCL X
210 PROMPT
    STO IND L
    RTN
    LBL 03
    RCL X
    1
    STO+Z
    -
    :
    STO 11
220 RCL 04
221 END

```

E.2.4 - Caso Teste para o Programa P-6

Considere-se como exemplo para a Etapa A, os seguintes dados:

$$D = 1500 \text{ cm};$$

$$H = 1500 \text{ cm};$$

$$A_M = 20 \text{ cm};$$

$$R = 1,5 \text{ cm};$$

$$N = 2;$$

$$V_{ab} = 345 \text{ kV}.$$

Observa-se a seguinte sequência de instruções:

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ETAPA CORRESPONDENTE
A		A.1
	D =	A.1
1500	1500	A.1
R/S		A.1
	H =	A.2
1500	1500	A.2
R/S		A.2
	AM = /-A=	A.3
20	20	A.3
R/S		A.3
	R =	A.4
1,5	1,5	A.4
R/S		A.4
	N =	A.5
2	2	A.5
R/S		A.5
	Vab =	A.6
345	345	A.6
R/S ou H		A.6

O programa apresenta a seguinte série de valores de convergência:

INDICAÇÃO NO VISOR
-1,77 E0
-3,32 E-2
-1,64 E-5
0,00 E0

Em seguida \bar{E} é apresentado o valor de \bar{E} :

$\bar{E} = 19,71$ (kV/cm) (coerente com os dados de entrada, em cm).

Apertando-se a tecla R/S, ou J, \bar{E} é apresentado o valor de $|x_1|$:

$$|x_1| = 0,5322.$$

Como exemplo para a Etapa B, deseja-se o seguinte:

- observar o valor de H;
- alterar o valor de D, para 3000 cm;
- alterar o valor de N, para 3.

Observa-se a seguinte sequência de instruções:

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ETAPA CORRESPONDENTE
C		B.2
B	H = 1500,0000	B.2
		B.1
	D = 1500,0000	B.1
3000	3000	B.1
R/S		B.1
F		B.5
	N = 2	B.5
3	3	B.5
R/S		B.5
H		B.1

Após serem apresentados os valores de convergência, o valor de \bar{E} é indicado no visor:

$$\bar{E} = 13,98 \text{ (kV/cm)}.$$

Apertando-se a tecla R/S ou J, é apresentado o valor de $|x_1|$:

$$|x_1| = 0,5197.$$

O tempo de execução do Programa P-6 é de cerca de 18 segundos para apresentação dos valores de \bar{E} e $|x_1|$.

APÊNDICE F

CÁLCULO DA MÉDIA DOS VALORES DE CAMPO ELÉTRICO MÁXIMO
NA SUPERFÍCIE DE N CONDUTORES DE UM FEIXE DA FASE
CENTRAL DE UMA LINHA DE TRANSMISSÃO COM DISPOSIÇÃO
HORIZONTAL, CONSIDERANDO-SE CABOS PÁRA-RAIOS

Para a determinação do valor de campo elétrico nas condições acima mencionadas, elaborou-se o Programa P-7, implantado na calculadora HP-41C.

Neste cálculo, uma série de grandezas auxiliares devem ser obtidas em passos intermediários, incluindo-se o valor de R_e , calculado através do Programa P-3, apresentado no Apêndice C, sendo aqui aproveitado com algumas alterações convenientes.

No ítem (F.1) deste Apêndice, são apresentadas as grandezas auxiliares envolvidas no cálculo do campo elétrico \bar{E} , para a elaboração do Programa P-7, e, no ítem (7.2), é apresentado o Programa P-7, incluindo seu modo de utilização e um caso teste.

F.1 - GRANDEZAS AUXILIARES

Reproduzem-se a seguir, as fórmulas empregadas no cálculo do valor de campo elétrico \bar{E} , em alguns casos com pequenas modificações. Essas fórmulas foram retiradas dos Capí

tulos III. IV e VII, e do Apêndice C, e são apresentadas por ordem de cálculo no Programa P-7.

Sejam definidas as seguintes relações, obtidas a partir das Eqs. (7.8) e (7.12) a (7.14):

$$\alpha_1 = \left(\frac{\alpha_{14}}{\delta} \right) = \ln \sqrt{\frac{(H + H')^2 + D''^2}{(H' - H)^2 + D''^2}},$$

$$\alpha_2 = \left(\frac{\alpha_{15}}{\delta} \right) = \ln \sqrt{\frac{(H' + H)^2 + (D + D')^2}{(H' - H)^2 + (D + D')^2}},$$

$$\alpha_3 = \left(\frac{\alpha_{44}}{\delta} \right) = \ln(2H'/R'),$$

$$\alpha_4 = \left(\frac{\alpha_{45}}{\delta} \right) = \ln \sqrt{\frac{[(2H')^2 + (2D')^2]}{(2D')^2}},$$

$$\alpha_5 = \left(\frac{\alpha_{24}}{\delta} \right) = \ln \sqrt{\frac{(H + H')^2 + D'^2}{(H' - H)^2 + D'^2}}.$$

onde $D'' = D - D'$ e $\bar{\delta} = 1/(2\pi\epsilon_0)$;

As Eqs. (7.25) a (7.28) podem ser reescritas da seguinte forma, respectivamente:

$$g_1 = \frac{\alpha_3 \cdot (\alpha_1^2 + \alpha_2^2) - 2 \cdot \alpha_1 \cdot \alpha_2 \cdot \alpha_4}{\alpha_3^2 - \alpha_4^2},$$

$$g_2 = \frac{\alpha_5(\alpha_1 + \alpha_2)}{\alpha_3 + \alpha_4},$$

$$g_3 = \frac{2 \cdot \alpha_1 \cdot \alpha_2 \cdot \alpha_3 - \alpha_4(\alpha_1^2 + \alpha_2^2)}{\alpha_3^2 - \alpha_4^2},$$

e

$$g_4 = \frac{2 \cdot \alpha_5^2}{\alpha_3 + \alpha_4},$$

As demais equações utilizadas são as seguintes:

$$A_M = \left[\frac{A}{2 \cdot \text{sen}(180^\circ/N)} \right], \quad (3.23)$$

$$\alpha^{-1} = \left\{ \frac{\left[\frac{(A_M/R)+1}{(A_M/R)-1} \right]^N + 1}{\left[\frac{(A_M/R)+1}{(A_M/R)-1} \right]^N - 1} \right\}, \quad (3.29)$$

$$R_C = A_M \cdot \sqrt[N]{\alpha}, \quad (4.8)$$

$$\gamma = \frac{(2\alpha)/(1+\alpha)}{1 - \frac{\sqrt[N]{1-\alpha}}{1+\alpha}}, \quad (4.54)$$

$$A' = \left[\begin{array}{c} \gamma \cdot R \\ H_1 \cdot \left(\frac{Q_B}{2\pi\epsilon V_{\phi T}} \right) \end{array} \right],$$

onde

$$\frac{Q_B}{2\pi\epsilon V_{\phi T}} = \frac{\bar{\beta} + \bar{\beta}' + \bar{\beta}''}{\bar{\beta}(\bar{\beta} + \bar{\beta}'') - 2 \cdot \bar{\beta}'^2}. \quad (7.40)$$

$$\beta = \ln(H_1/R_C) \quad (4.16)$$

onde

$$H_1 = 2 \cdot H,$$

$$\beta' = \ln \sqrt{1 + (1/\lambda)^2}, \quad (4.17)$$

$$\beta'' = \ln \sqrt{1 + 1/(2\lambda)^2}, \quad (4.18)$$

onde

$$\lambda = (D/H_1).$$

$$\bar{\beta} = \beta - g_1, \quad (7.33)$$

$$\bar{\beta}' = \beta' - g_2, \quad (7.35)$$

$$\bar{\beta}'' = \beta'' - g_3, \quad (7.36)$$

$$\bar{\beta} = \beta - g_4, \quad (7.34)$$

$$|x_1| = \left| \frac{\bar{\beta} + 2\bar{\beta}'}{2(\bar{\beta} + \bar{\beta}' + \bar{\beta}'')} \right| \quad (7.42)$$

$$B'' = \bar{\beta}'' - g_1 - g_4,$$

$$d = \bar{\beta}' + \bar{\beta}'' - g_1,$$

e

$$-a = g_4(g_1 - \bar{\beta}'') - 2\bar{\beta}'^2.$$

$$\bar{E} = \frac{\sqrt{3/2} \cdot V}{R_e \{ |x_1| [\ln(H_1/R_e) - \beta' + \beta'' - \beta'] + [\ln(H_1/R_e) - \beta'] \}}, \text{ sendo}$$

esta última expressão retirada do Apêndice E.

F.2 - PROGRAMA P-7

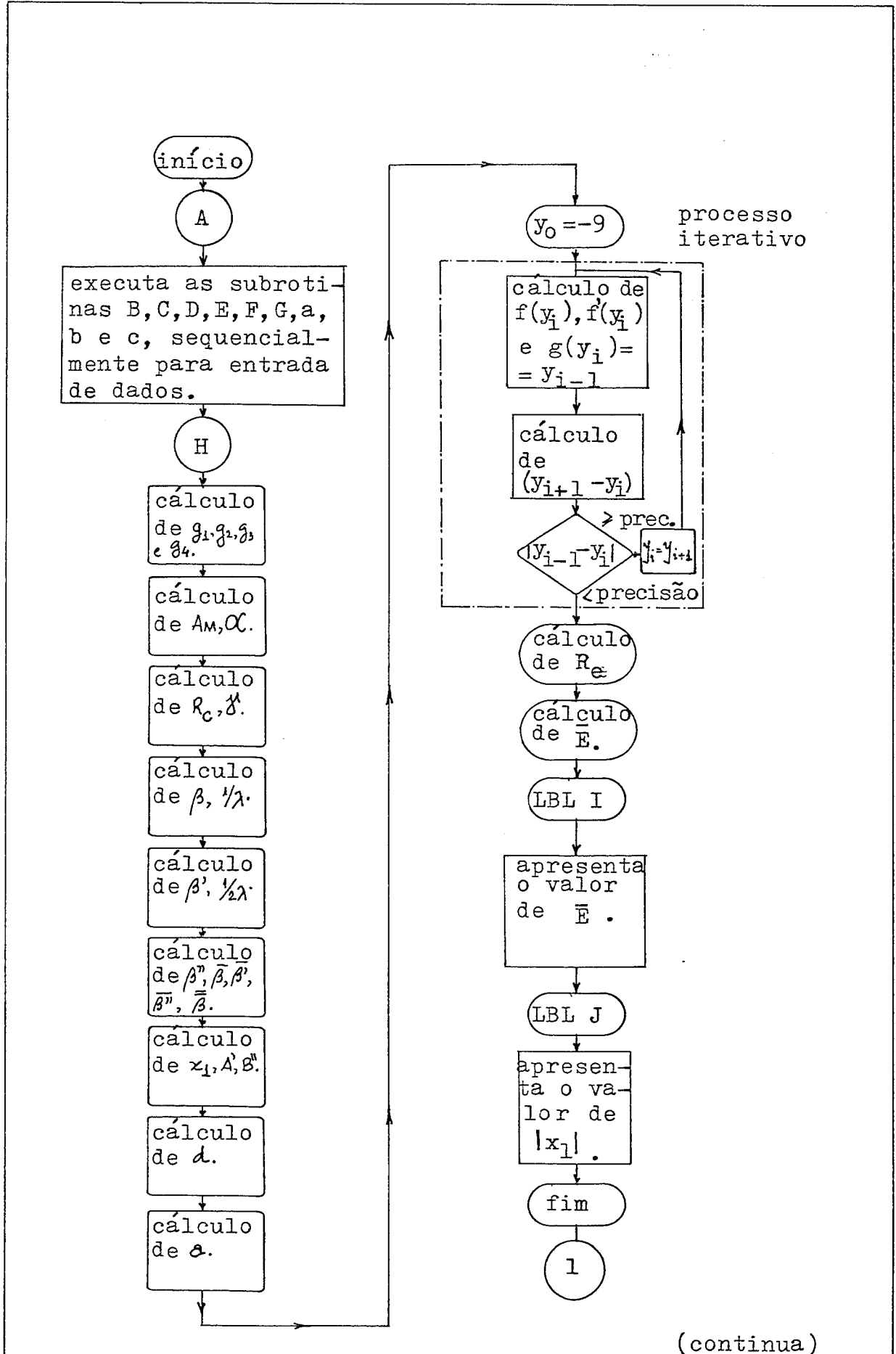
Para a obtenção do valor de \bar{E} , considerando-se a influência devida aos cabos para-raios, e utilizando-se as fórmulas apresentadas no ítem (F.1), elaborou-se o Programa P-7.

Para utilizá-lo convenientemente, \bar{E} é apresentado no ítem (F.2.1) o fluxograma, no ítem (F.2.2), a descrição do modo de operação, e, no ítem (F.2.3), a respectiva listagem.

Este programa foi implantado na calculadora HP-41C, empregando-se cartões magnéticos, e um caso teste é apresentado no item (F.2.4).

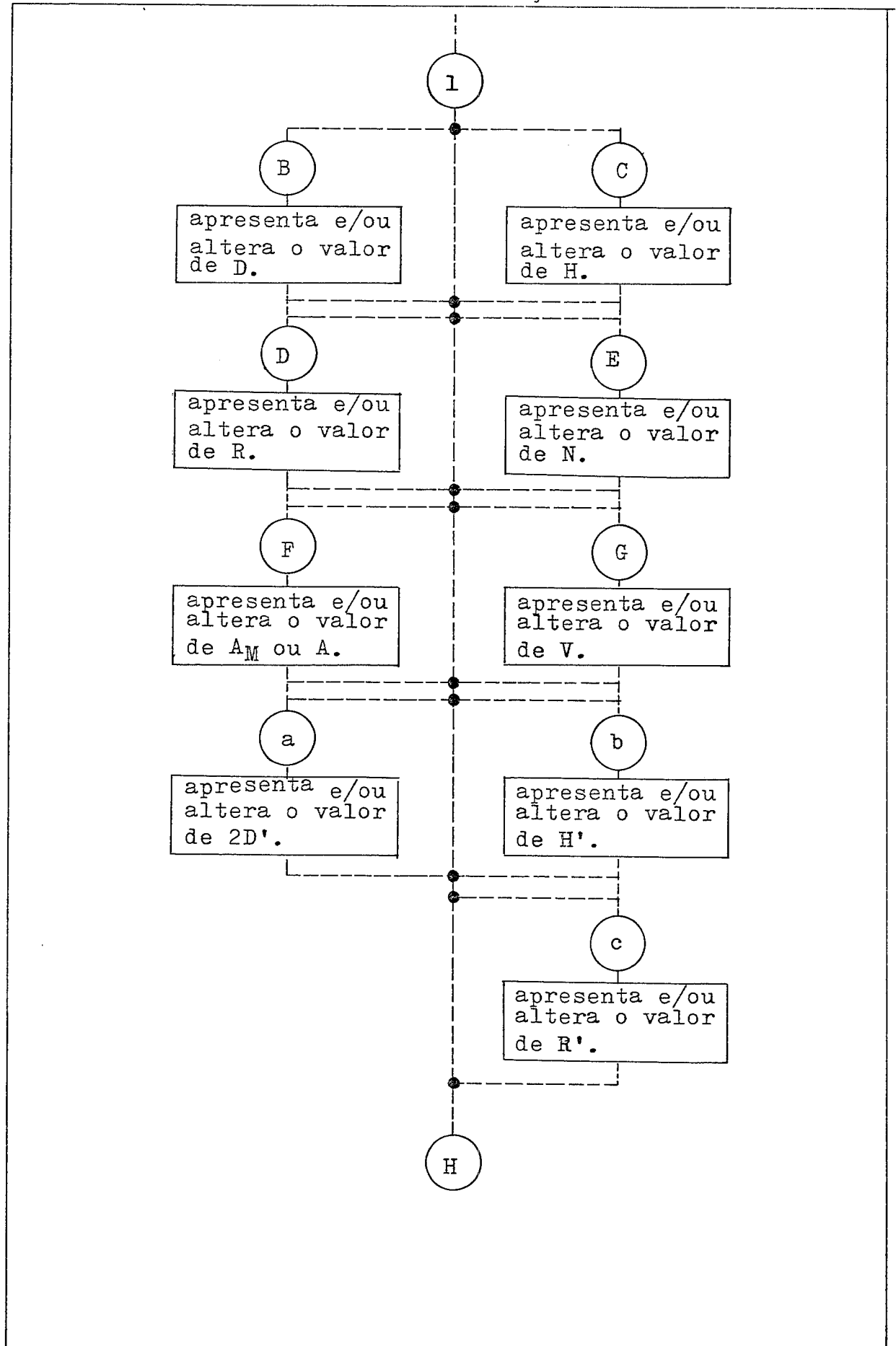
F.2.1 - Fluxograma do Programa P-7

FLUXOGRAMA DO PROGRAMA P-7



(continua)

FLUXOGRAMA DO PROGRAMA P-7 - Continuação



F.2.2 - Modo de Utilização do Programa P-7

É apresentado a seguir, por etapas, o modo de utilização do Programa P-7.

Quando o programa é utilizado pela primeira vez, deve-se introduzir na calculadora, através do teclado, as instruções indicadas sequencialmente no ítem F.2.3, relativas à listagem do Programa P-7. No caso de se dispor da leitora de cartões magnéticos, estas instruções poderão ser gravadas nestes cartões, permitindo assim guardar indefinidamente o programa para utilização posterior, mesmo apagando-se as instruções a ele relativas, da memória da calculadora. Em seguida, deverão ser executadas as instruções relativas à Etapa A.

Desde que a Etapa A tenha sido utilizada uma única vez, qualquer das Etapas, em número de duas, poderá ser executada.

As Etapas com as respectivas instruções, são as seguintes:

ETAPA A - Nesta etapa, pretende-se calcular pela primeira vez, o valor de \bar{E} , considerando-se a influência devida aos cabos para-raios dispostos simetricamente em relação ao condutor da fase central. Consiste do seguinte conjunto de procedimentos:

- A.1 - Pressiona-se a tecla A, sendo apresentado no visor ($D = nn \dots n$), cujo valor deverá ser introduzido pelo teclado. Em seguida, a tecla R/S deverá ser pressionada.
- A.2 - É apresentado no visor ($H = nn \dots n$), cujo valor deverá ser introduzido pelo teclado. Em seguida, a tecla R/S deverá ser pressionada.
- A.3 - É apresentado no visor ($R = nn \dots n$), cujo valor deverá ser introduzido pelo teclado. Em seguida, a tecla R/S deverá ser pressionada. A unidade relativa ao valor de R introduzido será a do valor de R_e calculado pelo Programa P-7, em passos intermediários.
- A.4 - É apresentado no visor ($N = nn \dots n$), cujo valor deverá ser introduzido pelo teclado. Em seguida, a tecla R/S deverá ser pressionada.
- A.5 - É apresentado no visor ($AM = nn \dots n$) ou ($-A = -nn \dots n$), um dos dois valores devendo ser introduzido pelo teclado. Introduzindo-se um valor positivo, o programa interpreta-o como sendo correspondente a A_M ; no caso deste valor ser negativo, o programa interpreta-o como sendo correspondente a $-A$. Em seguida, a tecla R/S deverá ser pressionada.

- A.6 - \bar{E} apresentado no visor ($V = nn \dots n$), cujo valor, em kV, deverá ser introduzido pelo teclado. Em seguida, a tecla R/S deverá ser pressionada.
- A.7 - \bar{E} apresentado no visor ($2D, = nn \dots n$), que corresponde ao dobro da distância de cada cabo para-raio à reta vertical que passa pelo centro geométrico do feixe da fase central, cujo valor deverá ser introduzido pelo teclado. Em seguida, a tecla R/S deverá ser pressionada.
- A.8 - \bar{E} apresentado no visor ($H, = nn \dots n$), que corresponde à distância dos cabos para-raios ao solo, cujo valor deverá ser introduzido pelo teclado. Em seguida, a tecla R/S deverá ser pressionada.
- A.9 - \bar{E} apresentado no visor ($R, = nn \dots n$), que corresponde ao raio dos condutores para-raios, cujo valor deverá ser introduzido pelo teclado. Em seguida, a tecla R/S deverá ser pressionada, dando início ao cálculo indicado no fluxograma, para obtenção de \bar{E} e $|x_1|$. Neste caso, a tecla R/S poderá ser substituída pela tecla H, para início dos cálculos.

Na Etapa A, havendo um dado qualquer introduzido em posição trocada, a tecla A deverá ser apertada, repetindo-se a sequência de entrada de dados anteriormente descrita.

Após o procedimento A.9, o programa executa uma série de instruções, indicando posteriormente valores relativos à convergência do processo iterativo de cálculo de R_e , indicado no fluxograma.

Uma vez que o valor zero tenha sido indicado no visor, ou seja, havendo convergência do processo iterativo, o valor de R_e é calculado e utilizado no cálculo de \bar{E} , que é então apresentado no visor, em kV por unidade de comprimento utilizado para os valores de D, H, A_M ou -A, R, $2D'$, H' e R' . Por exemplo, sendo estes valores fornecidos em cm, \bar{E} é calculado em kV/cm. Apertando-se em seguida a apresentação do valor de \bar{E} a tecla R/S, ou J, o valor de $|x_1|$ é apresentado no visor. Os valores de \bar{E} e $|x_1|$ podem ser apresentados novamente, bastando pressionar respectivamente as teclas I e J.

ETAPA B - Neste Etapa possibilita-se a apresentação de qualquer dos dados de entrada introduzidos para o cálculo de \bar{E} e $|x_1|$, optando-se por alterá-los ou não. Esta Etapa consiste dos seguintes procedimentos, todos, ou parte deles podendo ser executado em qualquer ordem:

- B.1 - Apertando-se a tecla B, apresenta-se no visor o valor de D existente na memória, na unidade previamente escolhida. Desejando-se alterar o valor de D, deve-se introduzir pelo teclado o novo valor, pressionando-se a seguir, a tecla R/S. No caso de simples verificação do valor anterior, pode-se executar qualquer um dos procedimentos desta Etapa e, em seguida, para calcular-se o novo valor de \bar{E} , e de $|x_1|$, deve-se pressionar a tecla H.
- B.2 - Este procedimento é igual ao do item B.1, à exceção da variável indicada, que passa a ser H, quando se aperta a tecla C.
- B.3 - Este procedimento é igual ao do item B.1, à exceção da variável indicada que passa a ser R, quando se aperta a tecla D.
- B.4 - Este procedimento é igual ao do item B.1, à exceção da variável indicada, que passa a ser N, quando se aperta a tecla E.
- B.5 - Este procedimento é igual ao do item B.1, à exceção da variável indicada, que passa a ser A_M , ou $-A$, quando se aperta a tecla F.

- B.6 - Este procedimento é igual ao do ítem B.1, à exceção da variável indicada, que passa a ser V, quando se aperta a tecla G.
- B.7 - Este procedimento é igual ao do ítem B.1, à exceção da variável indicada, que passa a ser 2D', quando se apertam as teclas amarela e A, nesta ordem.
- B.8 - Este procedimento é igual ao do ítem B.1, à exceção da variável indicada, que passa a ser H', quando se apertam as teclas amarela e B, nesta ordem.
- B.9 - Este procedimento é igual ao do ítem B.1, à exceção da variável indicada, que passa a ser R', quando se apertam as teclas amarela e C, nesta ordem.

F.2.3 - Listagem do Programa P-7

1	LBL 'P7	48	R↓	95	STO 08	142	:	180	-	236	X
	LBL A	:			RCL 06		XEQ 03	190	STO 03		ST-12
	XEQ B	50	ln		X		y ^x		RCL 01		-9
	XEQ C		STO 04		CHS		XEQ 03		RCL 09		STO 08
	XEQ E		RCL 22		STO 10		1/x		-	240	LBL 01
	XEQ F		ST+X	100	RCL 03		CHS		STO 04		RCL 08
	XEQ D		RCL 23		RCL 04		y ^x		STO 07		RCL 09
	XEQ G		:		X		STX 00		+		-
	XEQ a		ln		ST+X	150	RCL 12		RCL 02		RCL X
10	XEQ b		STO 05		RCL 01		1		RCL 10		RCL 08
	XEQ c		RCL 22		:		+		-		ST+Z
	LBL H		ST+X		RCL 05		RCL 12	200	STO 05		X
	'	60	RCL 21		STX 08		XEQ 03		+		RCL 12
	AVIEW		→ P		X ≤ Y		1/x		RCL 00		+
	RCL 22		RCL 21		X		y ^x		RCL 11	350	ST:Y
	STO 01		:	110	ST+10		1/x		-		RCL 10
	RCL 16		ln		LASTX		1		STO 06		RCL 08
	ST-01		STO 06		RCL 06		X ≤ Y		2		-
	+		RCL 00		X	160	-		:		1/x
20	STO 00		RCL 21		ST-08		X		ST+07		ST+Z
	RCL 15		2		RCL 07		ST: 13		X ≤ Y		X
	RCL 21		:		RCL 00		RCL 16	210	ST:07		RCL 13
	2	70	STO 02		:		ST:13		RCL 03		:
	:		→ P		STO 09		2		RCL 05		ln
	-		RCL 01		ST+X		X		+	260	RCL 08
	STO 02		RCL 02	120	RCL 07		STO 14		RCL 06		+
	→ P		→ P		X		RCL 00		X		X ≤ Y
	RCL 01		X ≤ Y		STO 11		:		RCL 04		1
	RCL 02		R↓		RCL 03	170	ln		x ²		+
30	→ P		:		RCL 04		STO 00		ST+X		:
	X ≤ Y		ln		+		RCL 14		CHS		CHS
	R↓		STO 07		STX 09		RCL 15	220	STO 12		RCL 08
	:	80	RCL 05		RCL 19		:		+		+
	ln		STO 00		X > 0?		STO 14		:		RCL 08
	STO 03		RCL 06		GTO 08		1		ST:13	270	X ≤ Y
	RCL 00		ST+00	130	ABS		→ P		RCL 05		STO 08
	RCL 15		-		180		ln		RCL 08		-
	RCL 21		RCL 00		RCL 18		STO 01		-		FIX 7
	2		X		:	180	RCL 14		STO 10		RND
40	:		STO 01		sin		2		RCL 11		SCI 2
	+		1/x		2		:		-		CLA
	STO 02		RCL 03		X		1	230	STO 09		ARCLX
	→ P	90	x ²		:		→ P		RCL 10		AVIEW
	RCL 01		RCL 04		LBL 08		ln		RCL 04		X ≠ 0?
	RCL 02		x ²		STO 00		STO 02		ST+10	280	GTO 01
	→ P		+	140	RCL 17		RCL 00		X ≤ Y		RCL 08
47	X ≤ Y	94	X	141	STO 13	188	RCL 08	235	RCL 11	282	RCL 01

continua...

F.2.3 - Listagem do Programa P-7 (Continuação)

```

283 +      334 LBL F
      RCL X      'N=
      RCL 02     18
      -         GTO 02
      RCL 01     LBL D
      +         'y=
290 RCL 07 340 20
      ABS       GTO 02
      STO 07     LBL a
      X         2D,=
      +         21
      ABS       GTO 02
      RCL 08     LBL b
      eX        'H,=
      X         22
      RCL 16     GTO 02
300 2         350 LBL c
      X         'R,=
      X         23
      1/x       GTO 02
      1,5       LBL G
      SQRT      'AM=
      X         19
      STO 14     LBL 02
      LBL I     FIX 4
      FIX 2     ABS
      RCL 14 360 RCL(i)X
310 RCL 20     X < 0?
      X         '-A=
      'E=       ARCLX
      ARCL X    PROMPT
      AVIEW     STO(i)L
      RTN      RTN
      LBL J     LBL 03
      '/x1/=   RCL X
      FIX 4     1
      ARCL07 370 STO+Z
320 AVIEW     -
      RTN      :
      LBL B     STO 12
      'D=       RCL 18
      15        375 END
      GTO 02
      LBL C
      'H=
      16
      GTO 02
330 LBL E
      'R=
      17
333 GTO 02

```


F.2.4 - Caso Teste para o Programa P-7

Considere-se como exemplo para a Etapa A, os seguintes dados:

$$D = 1440 \text{ cm}$$

$$H = 1660 \text{ cm}$$

$$V = 800 \text{ kV}$$

$$R = 1,75260 \text{ cm}$$

$$-A = -45 \text{ cm}$$

$$N = 4$$

$$H' = 3300 \text{ cm}$$

$$2D' = 2680 \text{ cm}$$

$$R' = 0,555625 \text{ cm.}$$

Observa-se a seguinte sequência de instruções:

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ETAPA CORRESPONDENTE
A		A.1
1440	D =	A.1
R/S	1440	A.1
		A.1
1600	H =	A.2
R/S	1600	A.2
		A.2
1,7526	R =	A.3
R/S	1,7526	A.3
		A.3
4	N =	A.4
R/S	4	A.4
		A.4
	V =	A.5

continua...

continuação

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ETAPA CORRESPONDENTE
800	800	A.5
R/S		A.5
- 45	-A =	A.6
R/S	- 45	A.6
		A.6
2680	2D' =	A.7
R/S	2680	A.7
		A.7
3300	H' =	A.8
R/S	3300	A.8
		A.8
0,555625	R' =	A.9
R/S ou H	0,555625	A.9
		A.9

O programa apresenta os seguintes valores de convergência:

INDICAÇÃO NO VISOR
- 2,31E0
- 6,31E-2
- 7,57E-5
0,00E0

Em seguida \bar{E} é apresentado o valor de \bar{E} :

$\bar{E} = 26,91$ (kV/cm) (coerente com os dados de entrada, em cm).

Apertando-se a tecla R/S, ou J, \bar{E} é apresentado no visor o valor de $|x_1|$:

$|x_1| = 0,5393$.

Como exemplo para a Etapa B, deseja-se o seguinte:

- observar o valor de H;
- alterar o valor de D, para 1500 cm;
- observar o valor de R'.

Observa-se a seguinte sequência de instruções:

INSTRUÇÃO POR TECLA	INDICAÇÃO NO VISOR	ETAPA CORRESPONDENTE
C	H = 1600,0000	B.2
B	D = 1440,0000	B.1
1500	1500	B.1
R/S		B.1
amarela, C		B.9
H	R' = 0,555625	B.9
		B.1

Após serem apresentados os valores de convergência, o valor de \bar{E} é indicado no visor:

$$\bar{E} = 26,67 \text{ (kV/cm)}.$$

Apertando-se a tecla R/S, ou J, é apresentado no visor o valor de $|x_1|$:

$$|x_1| = 0,5384.$$

O tempo de execução do programa P-7 é de cerca de 27 s para apresentação dos valores de \bar{E} e de $|x_1|$.

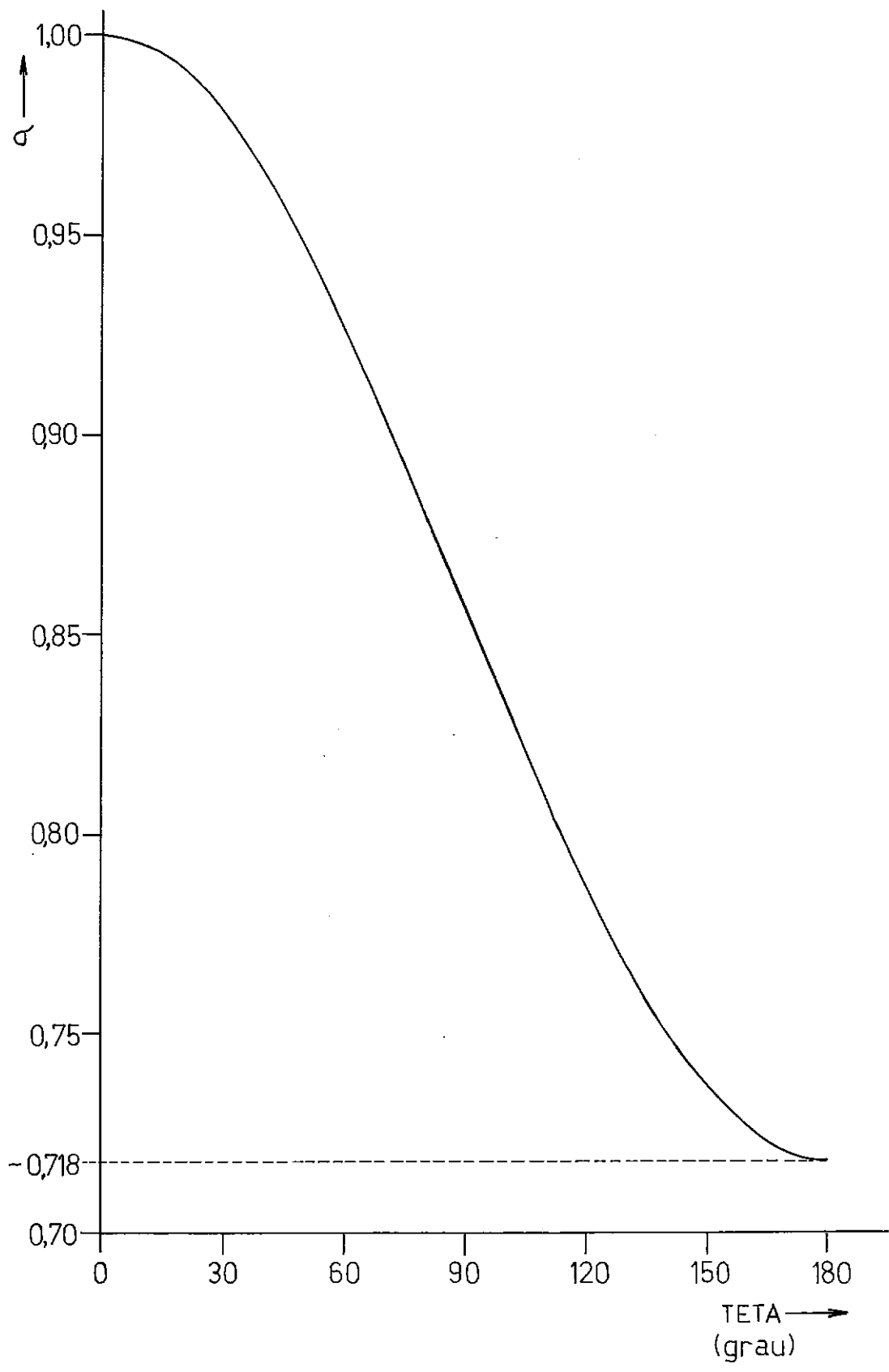
BIBLIOGRAFIA

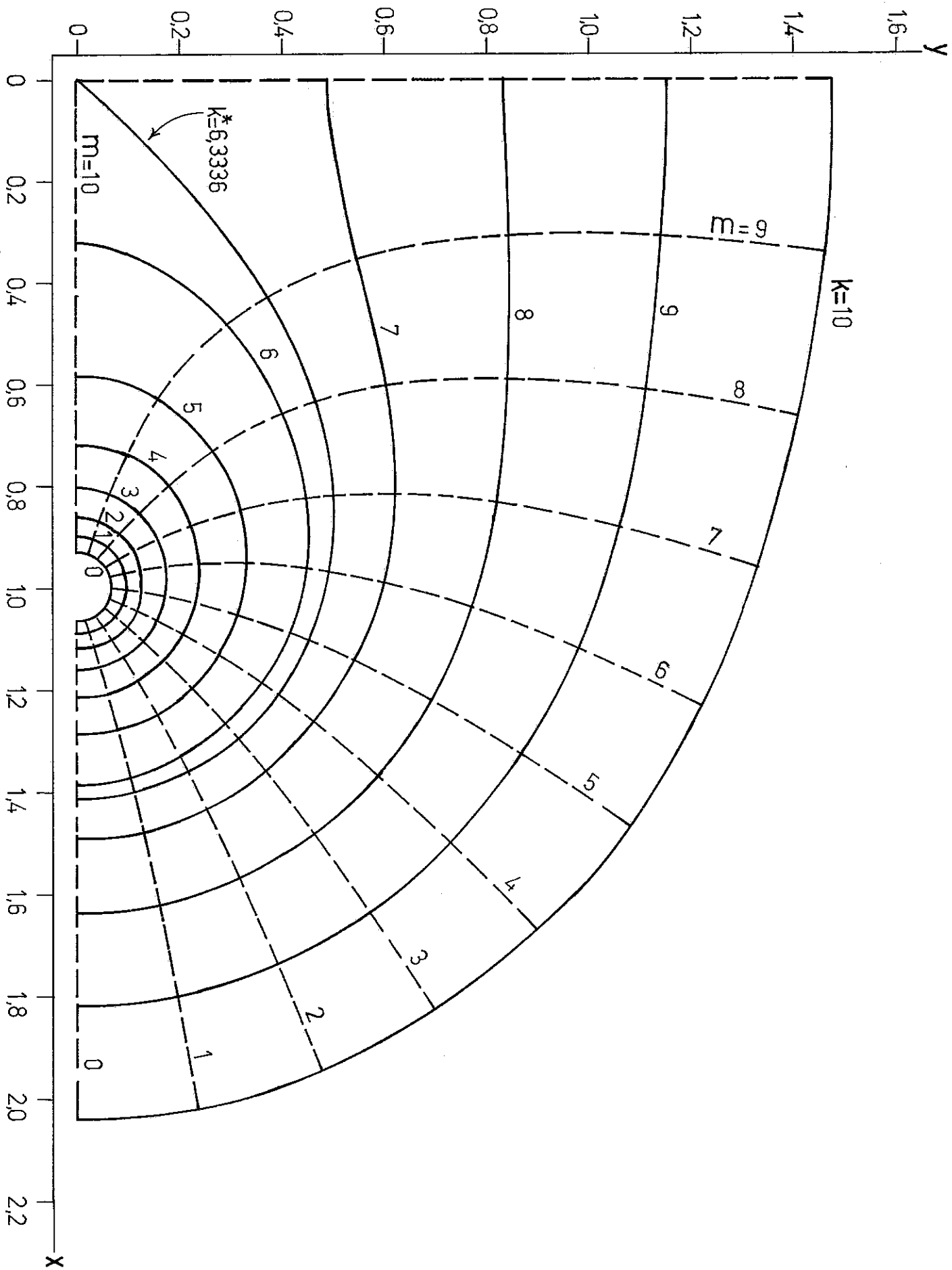
1. RUEL V. CHURCHILL - "Complex Variables and Applications" , McGraw-Hill Book Company, Inc., Second Edition, 1960.
2. L. V. BEWLEY - "Two-Dimensional Fields in Electrical Engineering", Dover Publications, Inc., 1963.
3. A. S. TIMASCHEFF - "Fast Calculation of Gradients for the Center Phase of a Three-Phase Bundle Conductor Line With any Number of Subconductors", IEEE Transactions, Vol. PAS-90, n^o 1, Jan./Feb. 1971, pp. 157-164.
4. A. S. TIMASCHEFF - "Fast Calculation of Gradients for the Center Phase of a Three-Phase Bundle Conductor Line With any Number of Subconductors - Part II. Gradients Calculation for the Side-Phase", IEEE Transactions, Vol. 94, Jan./Feb. 1975, pp. 104-107.
5. IEEE Corona and Field Effects Subcommittee Report (Radio Noise Working Group) - "A Survey of Methods for Calculating Transmission Line Conductor Surface Voltage Gradients", IEEE Transactions, Vol. PAS-98, n^o 6, Nov./Dec. 1979, pp. 2135-2141.

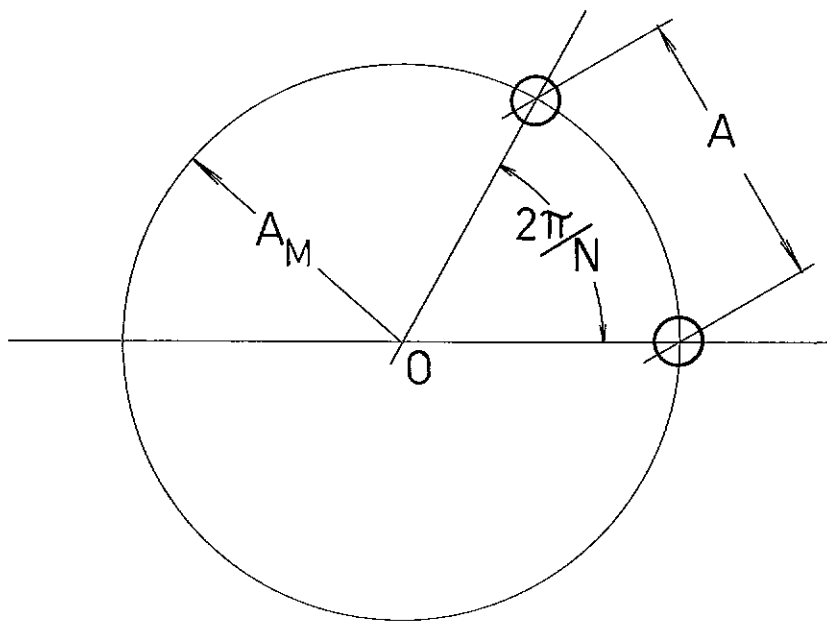
6. CHIGANER, L., "Campo Elétrico em Linhas de Transmissão - Aplicação do Método das Sucessivas Imagens", M. Sc. Thesis, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, Janeiro de 1981.

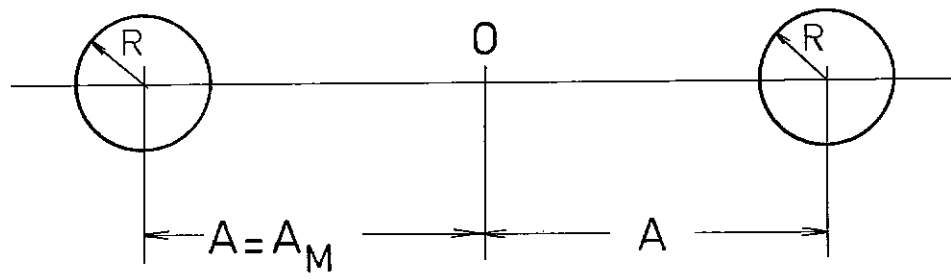
BIBLIOGRAFIA AUXILIAR

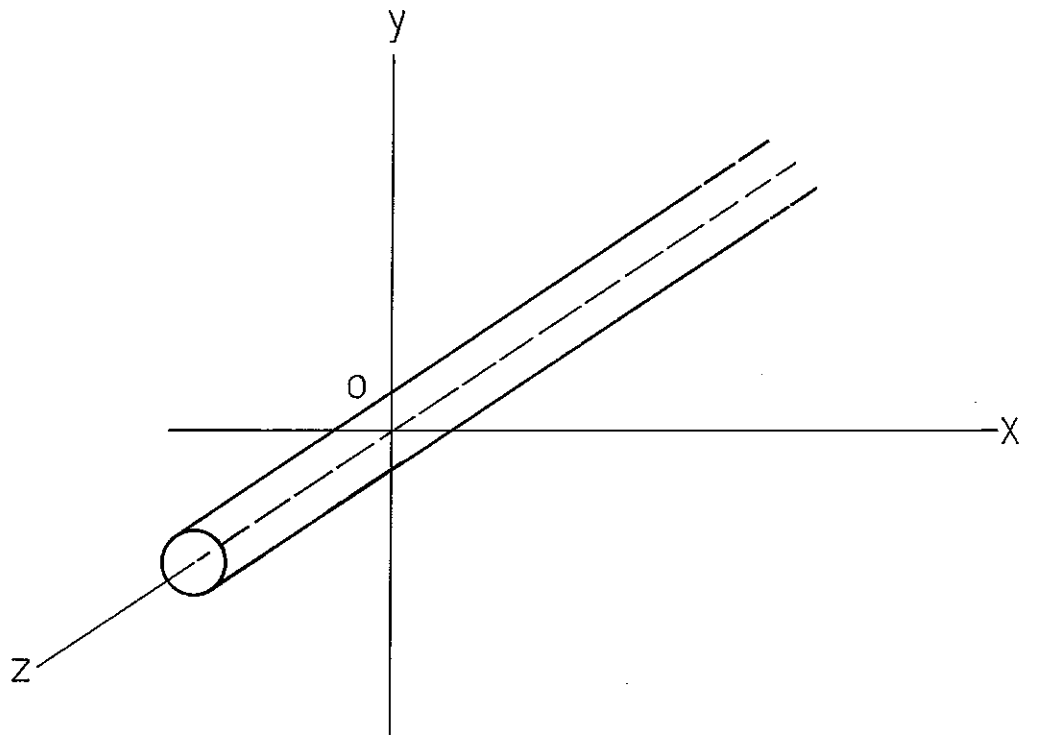
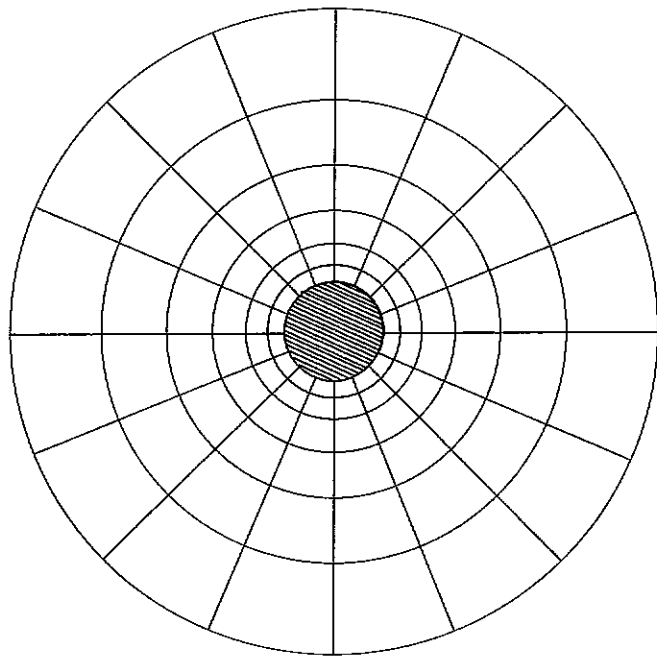
7. A. S. TIMASCHEFF - "Field Patterns of Bundle Conductors and Their Electrostatic Properties", AIEE Transactions, Vol. 80, Part III, 1961, pp. 590-597.
8. A. S. TIMASCHEFF - "Equigradient Lines in the Vicinity of Bundle Conductors", IEEE Transactions, Vol. PAS-82, April 1963, pp. 104-110.
9. "Transmission Line Reference Book 345 kV and Above", Copyright by Electric Power Research Institute (EPRI), 1975, Chapter 3.
10. SIMON RAMO, JOHN R. WHINNERY, THEODORE VAN DUZER - "Fields and Waves in Communication Electronics", Wiley I. E., 1965.
11. KRAUS and CARVER - "Electromagnetics", McGraw-Hill, Sec.Ed., 1973.

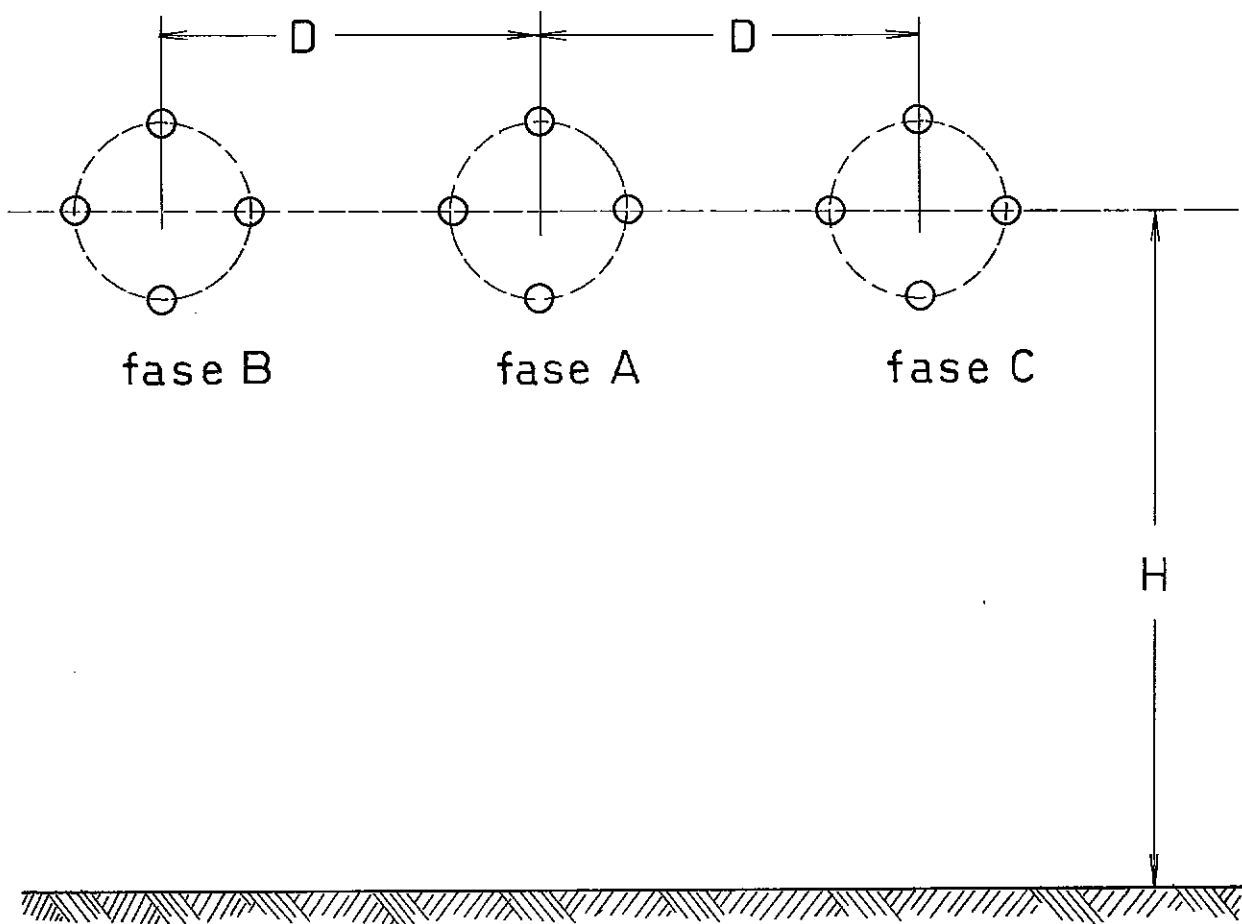
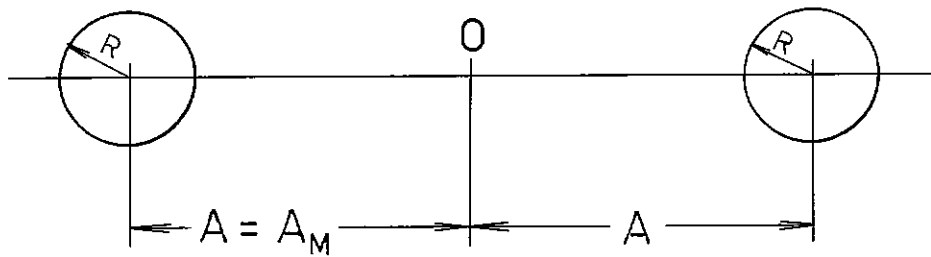


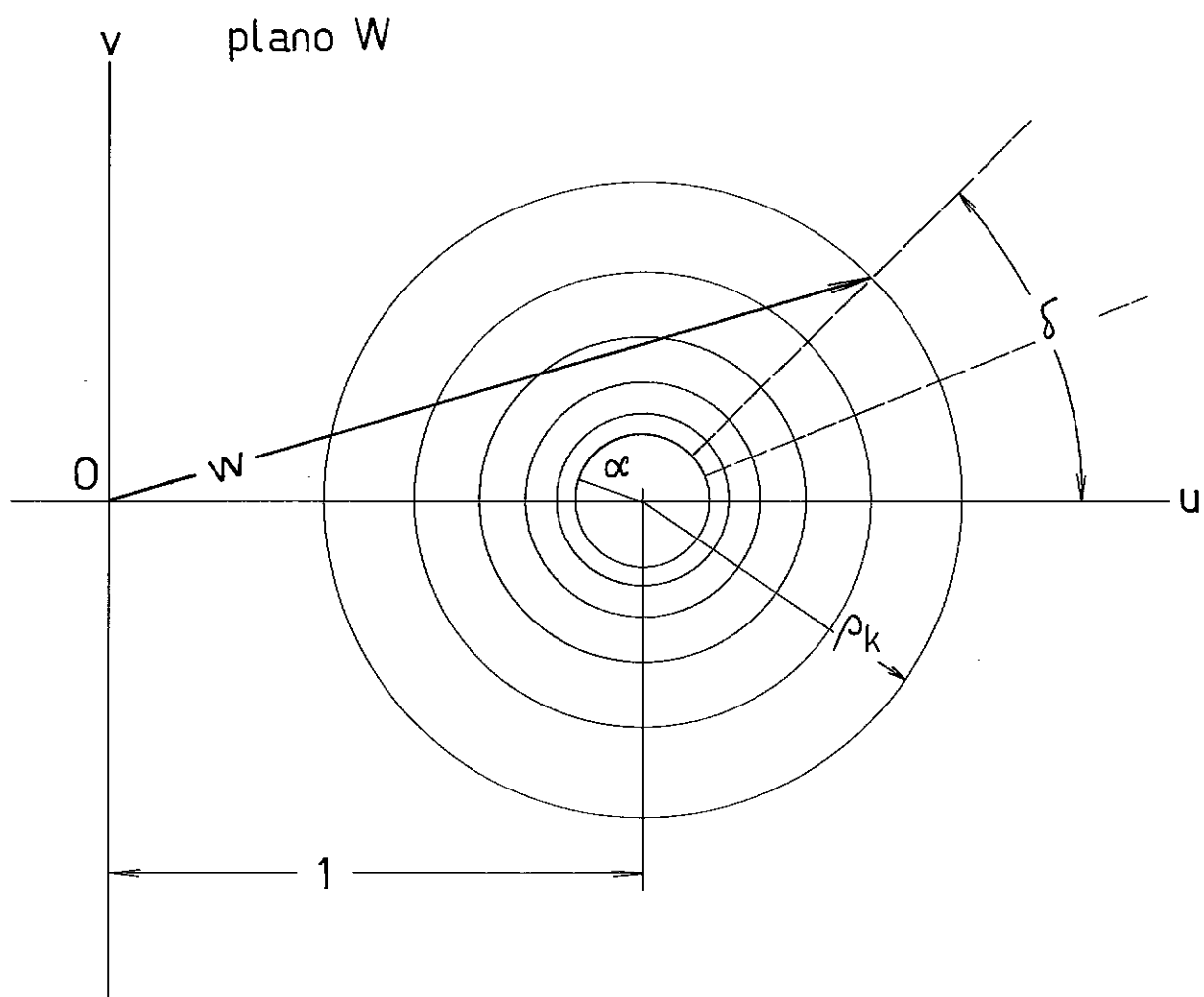


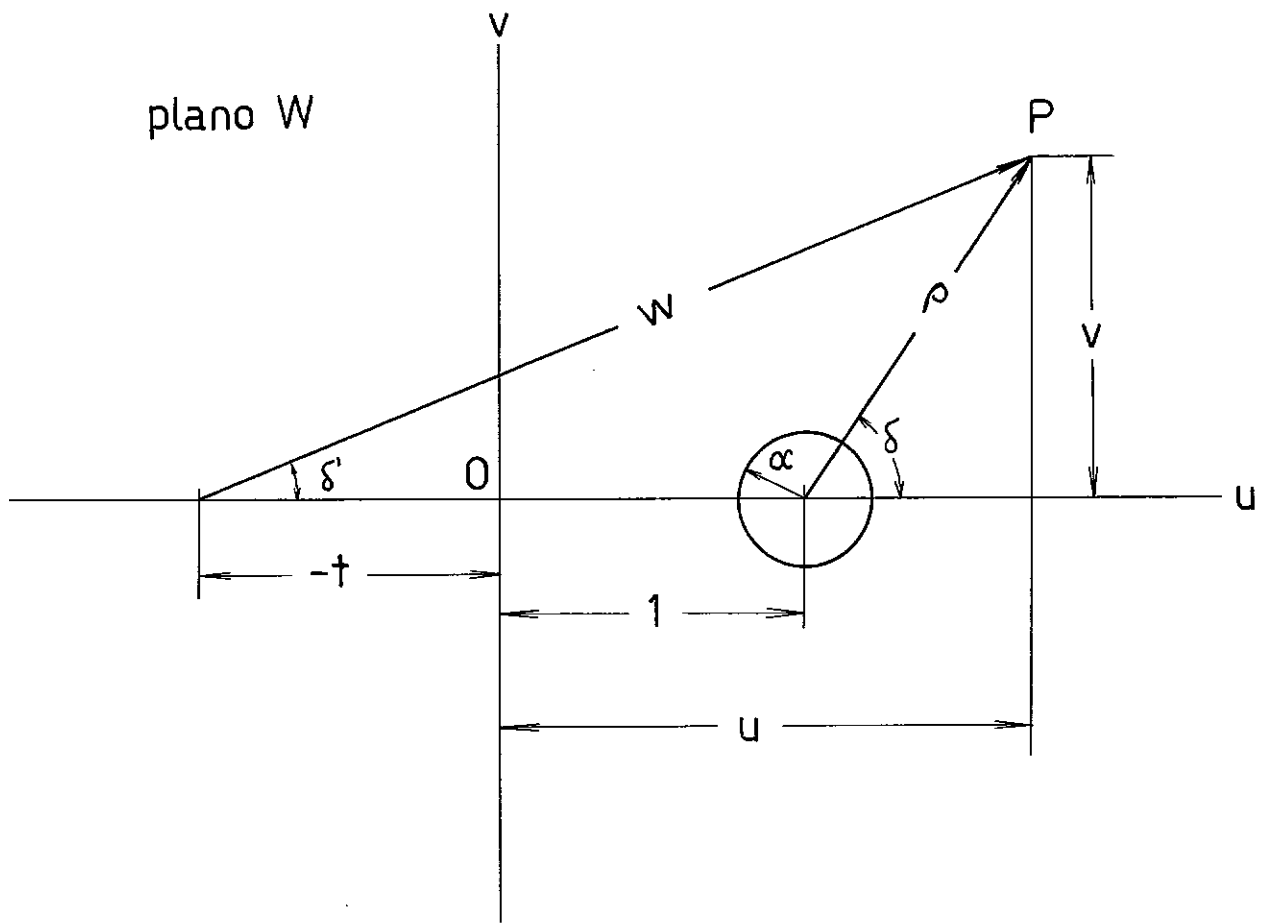


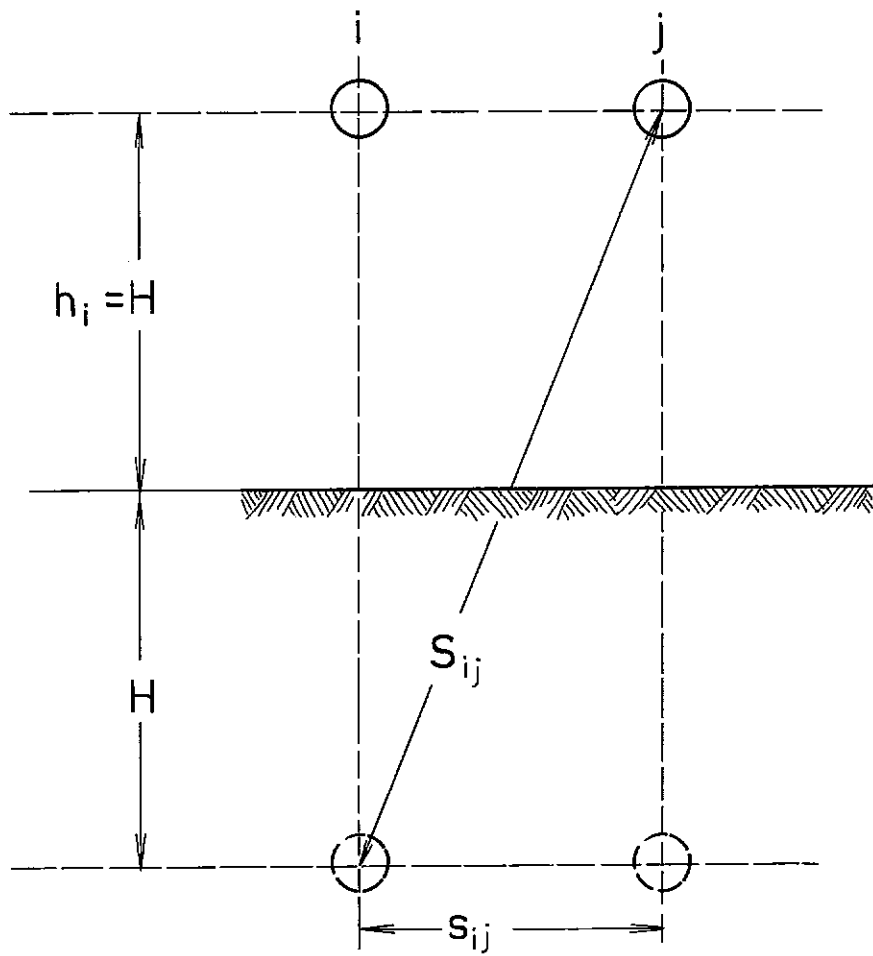


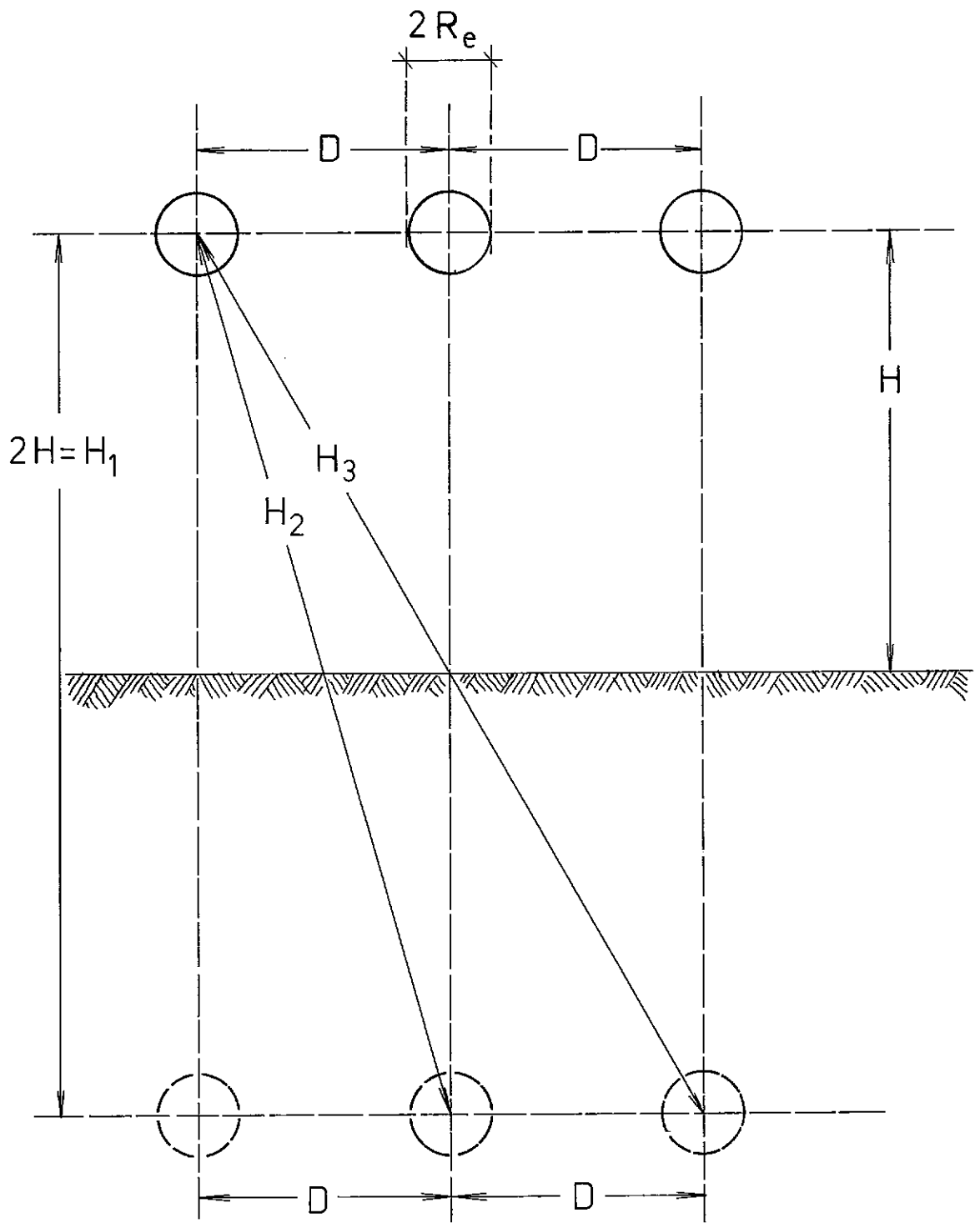


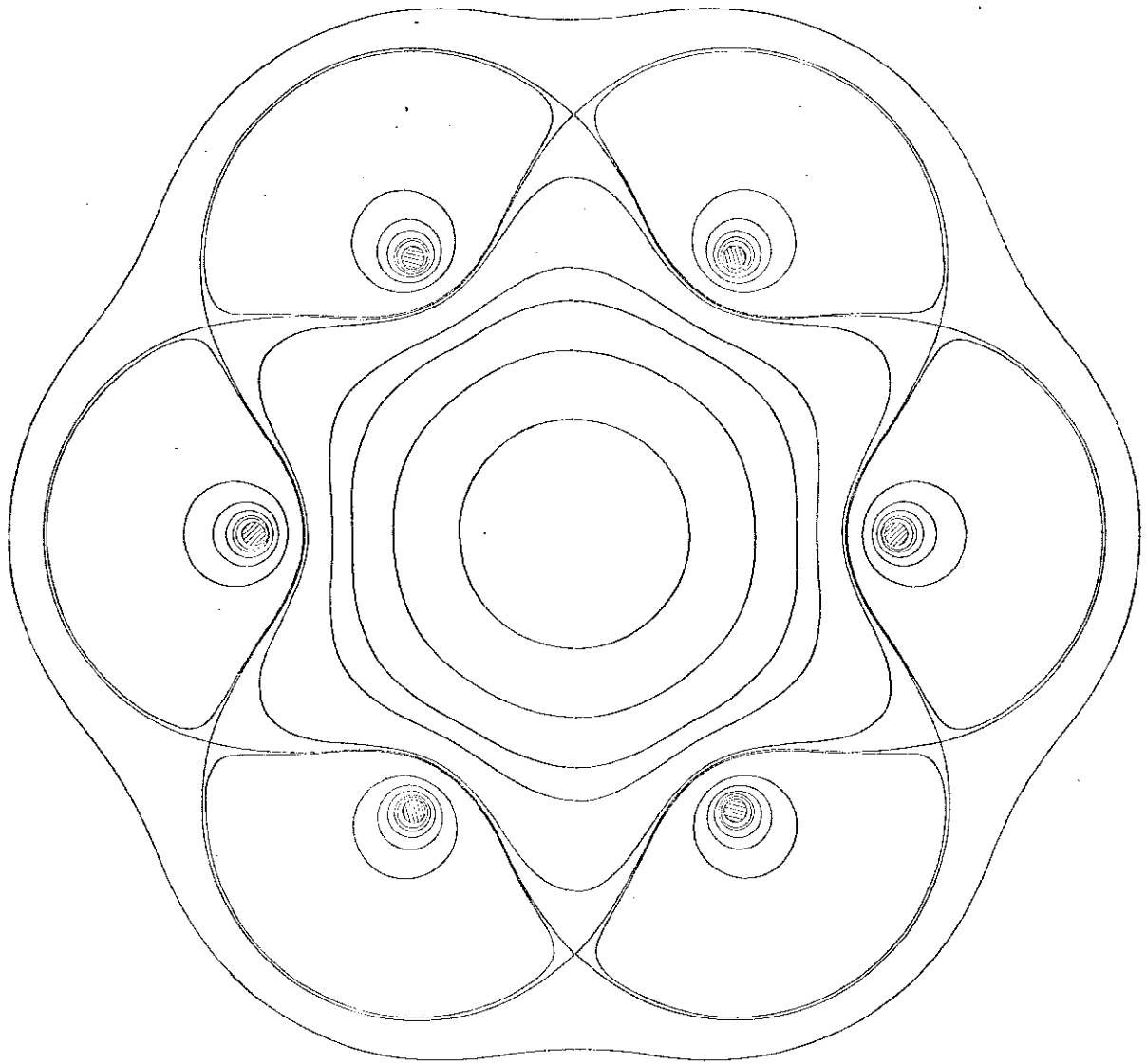


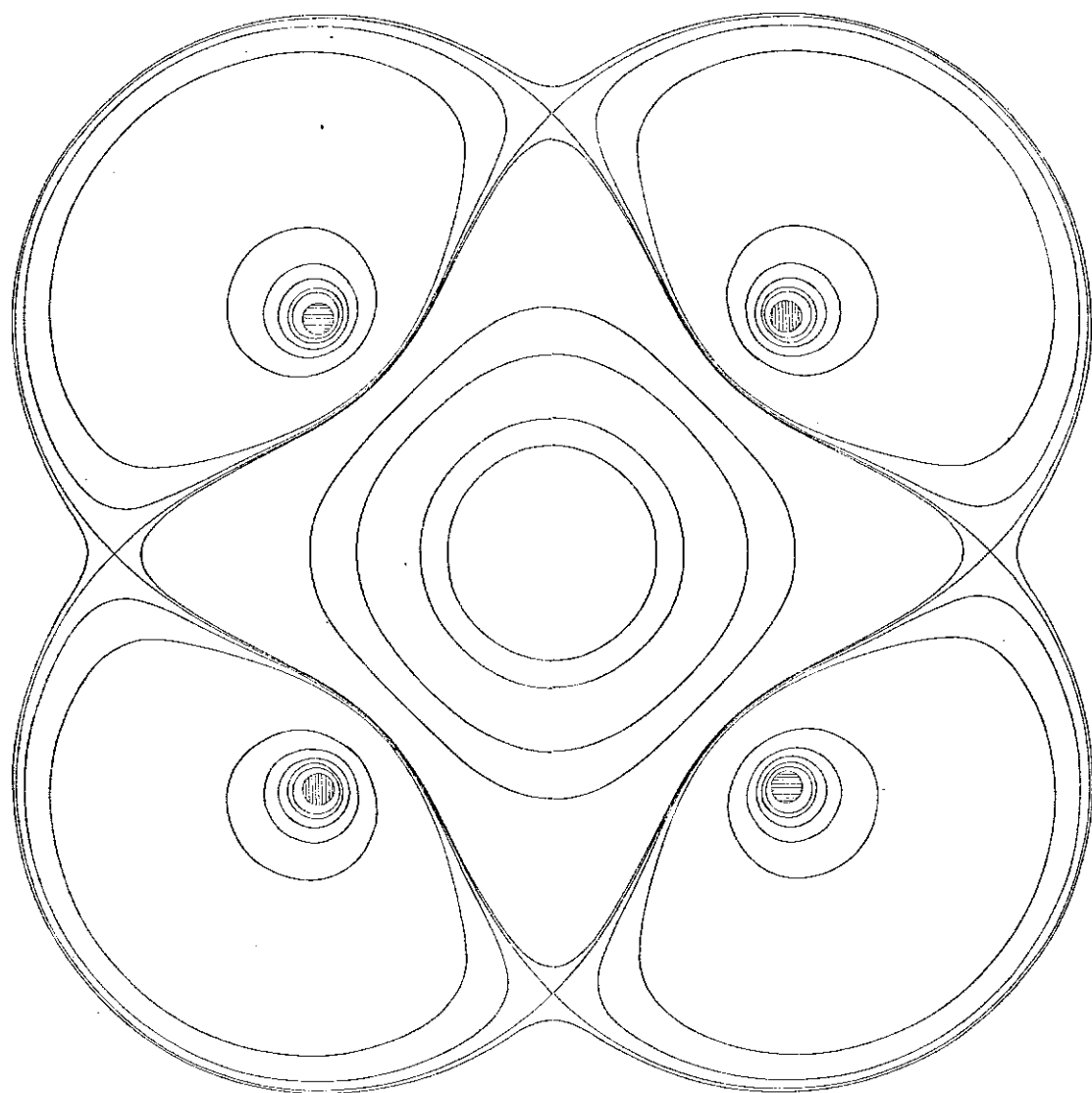


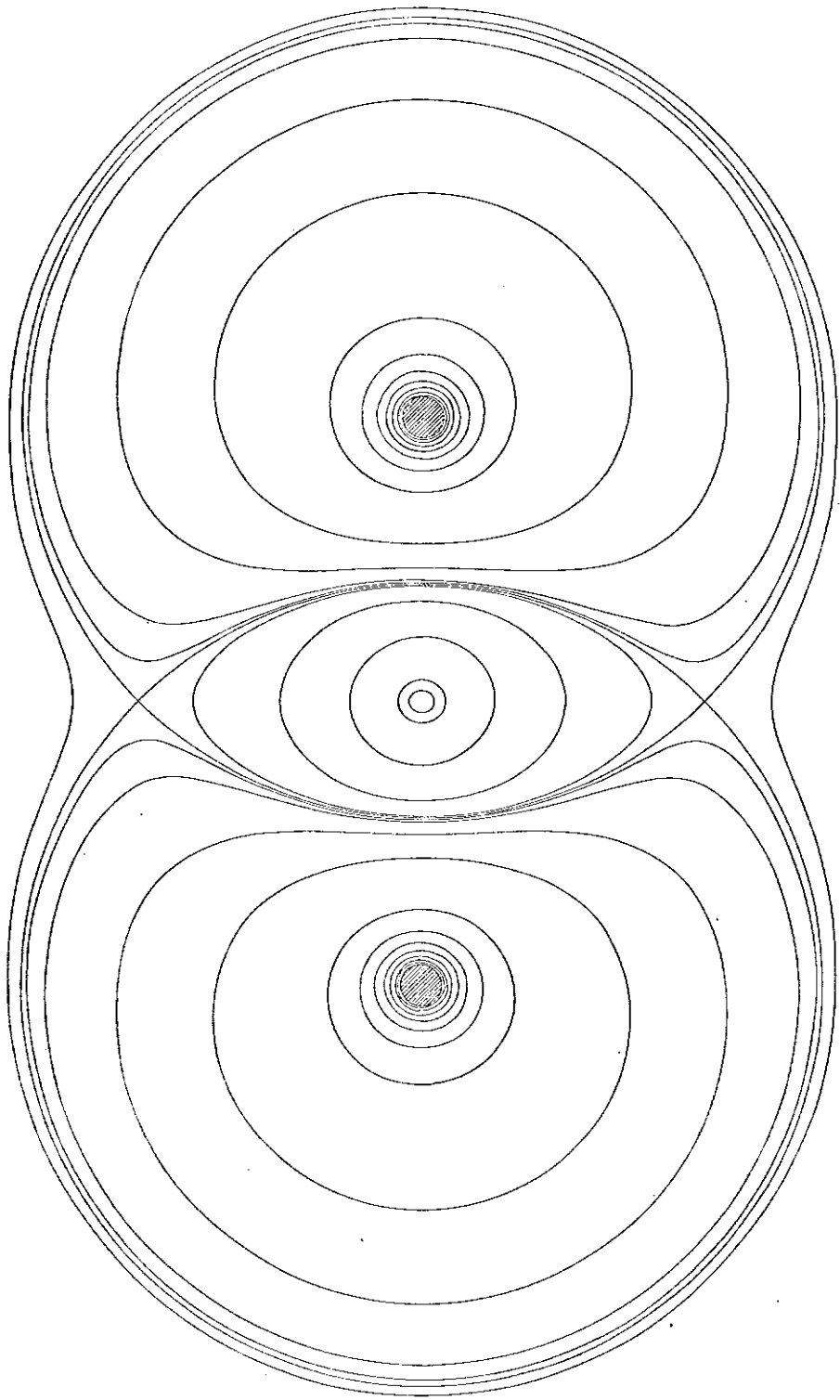


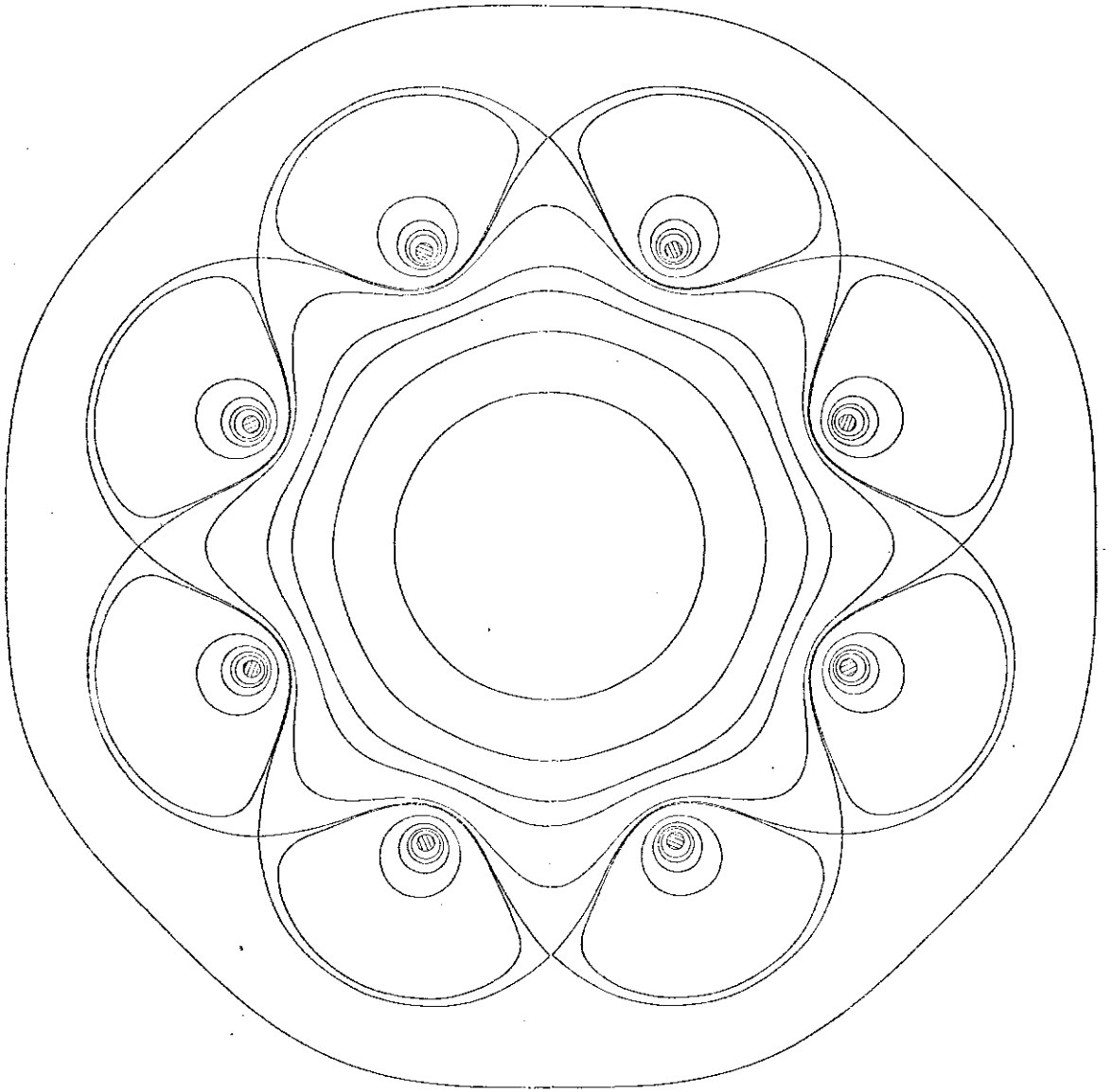


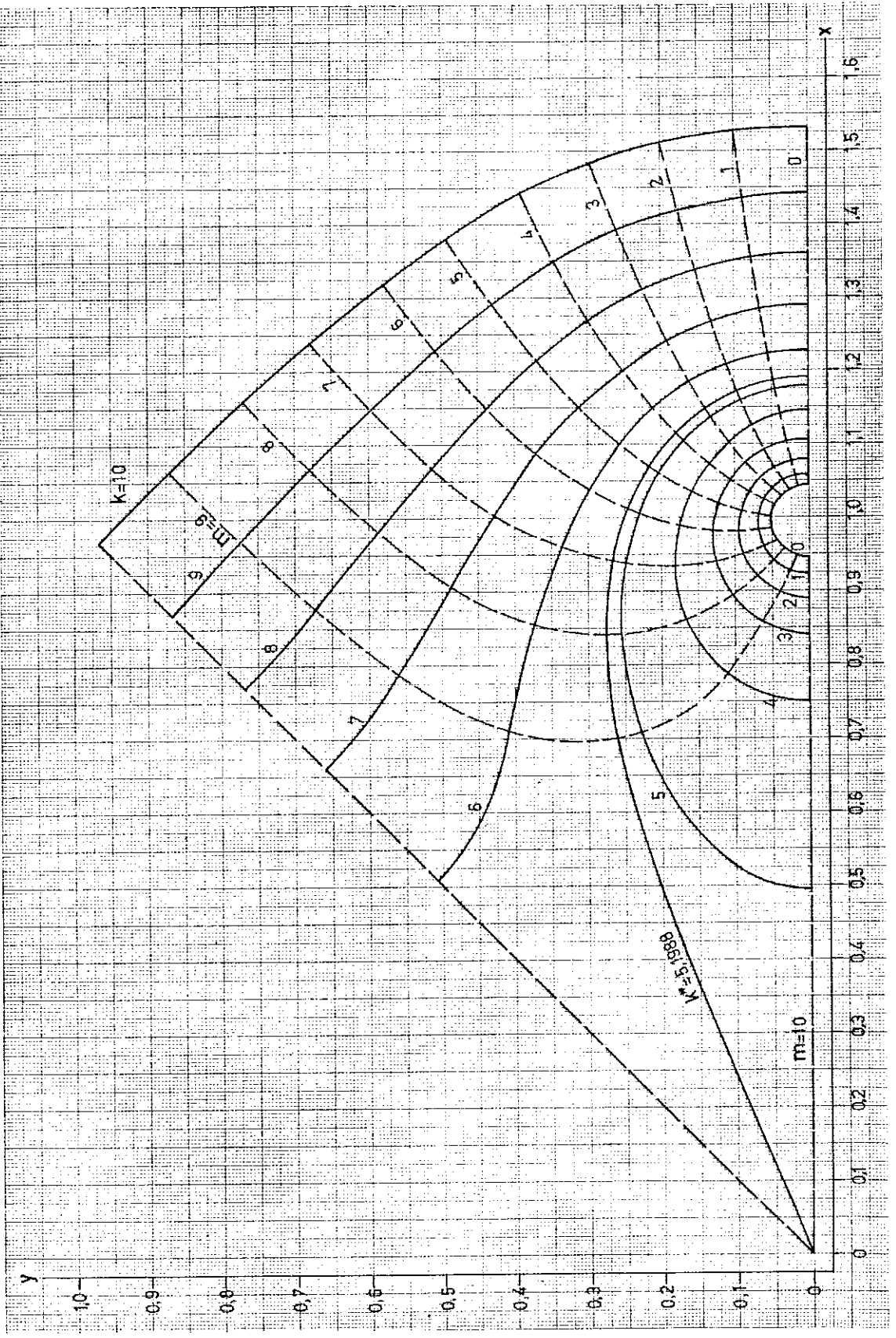


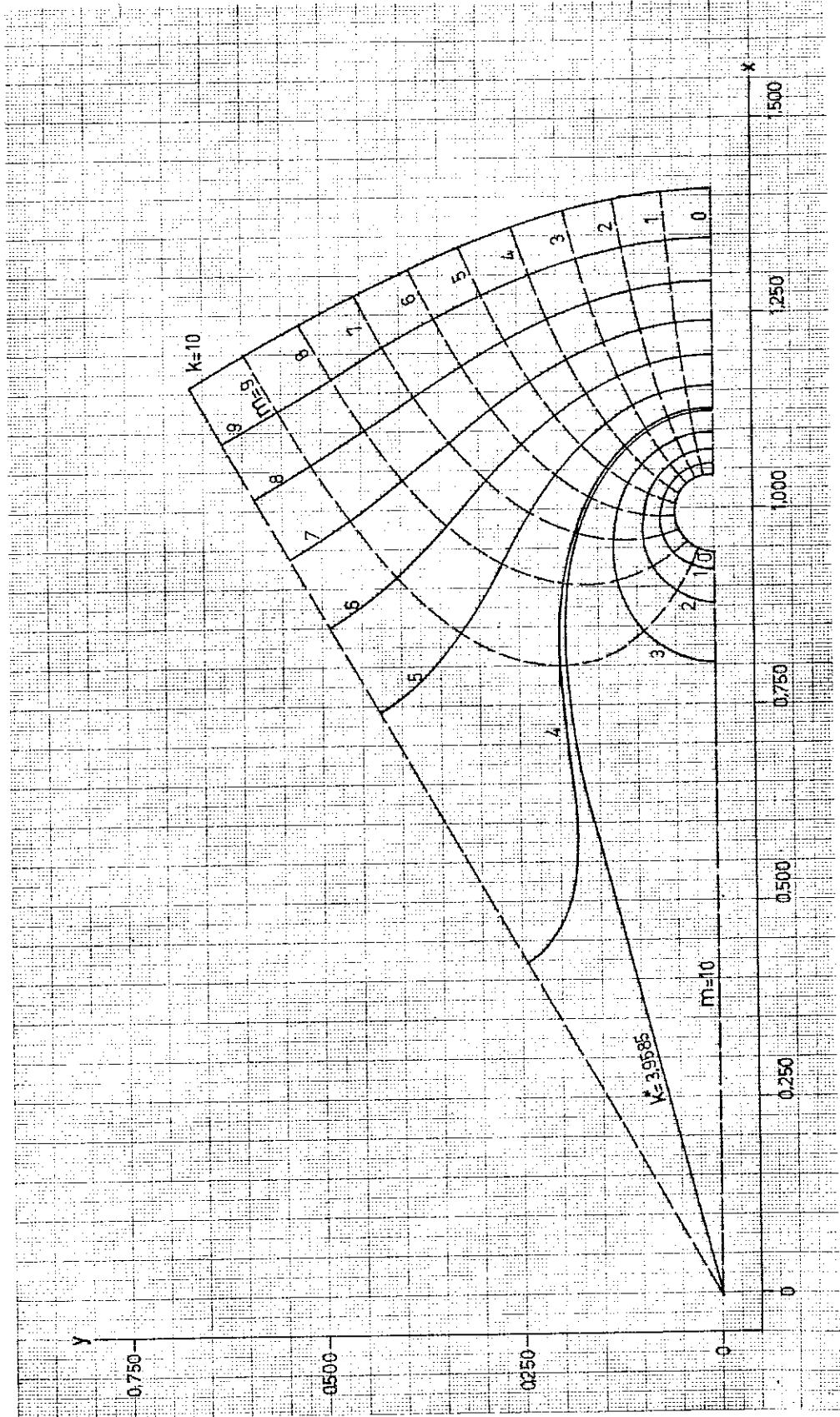


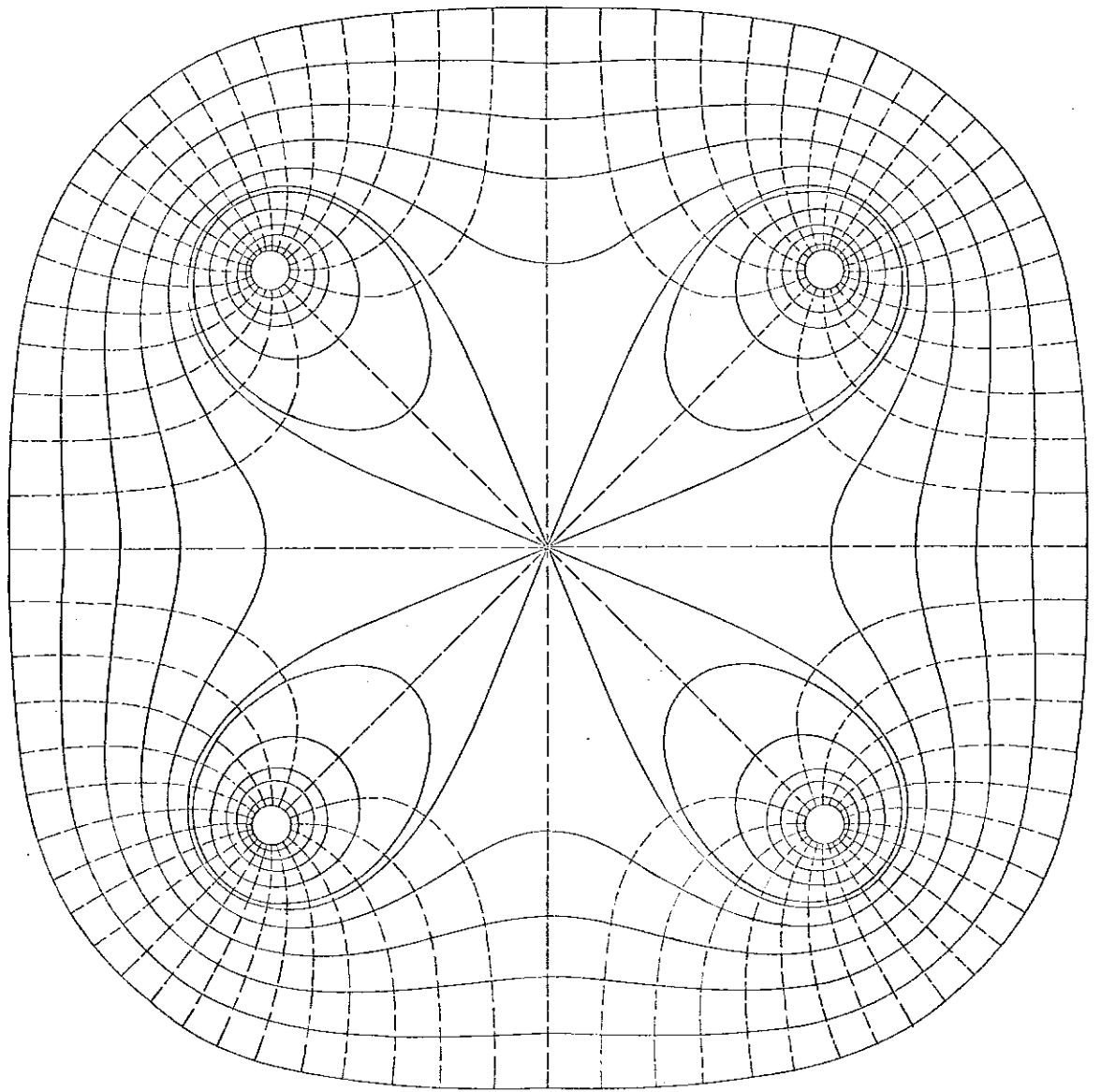


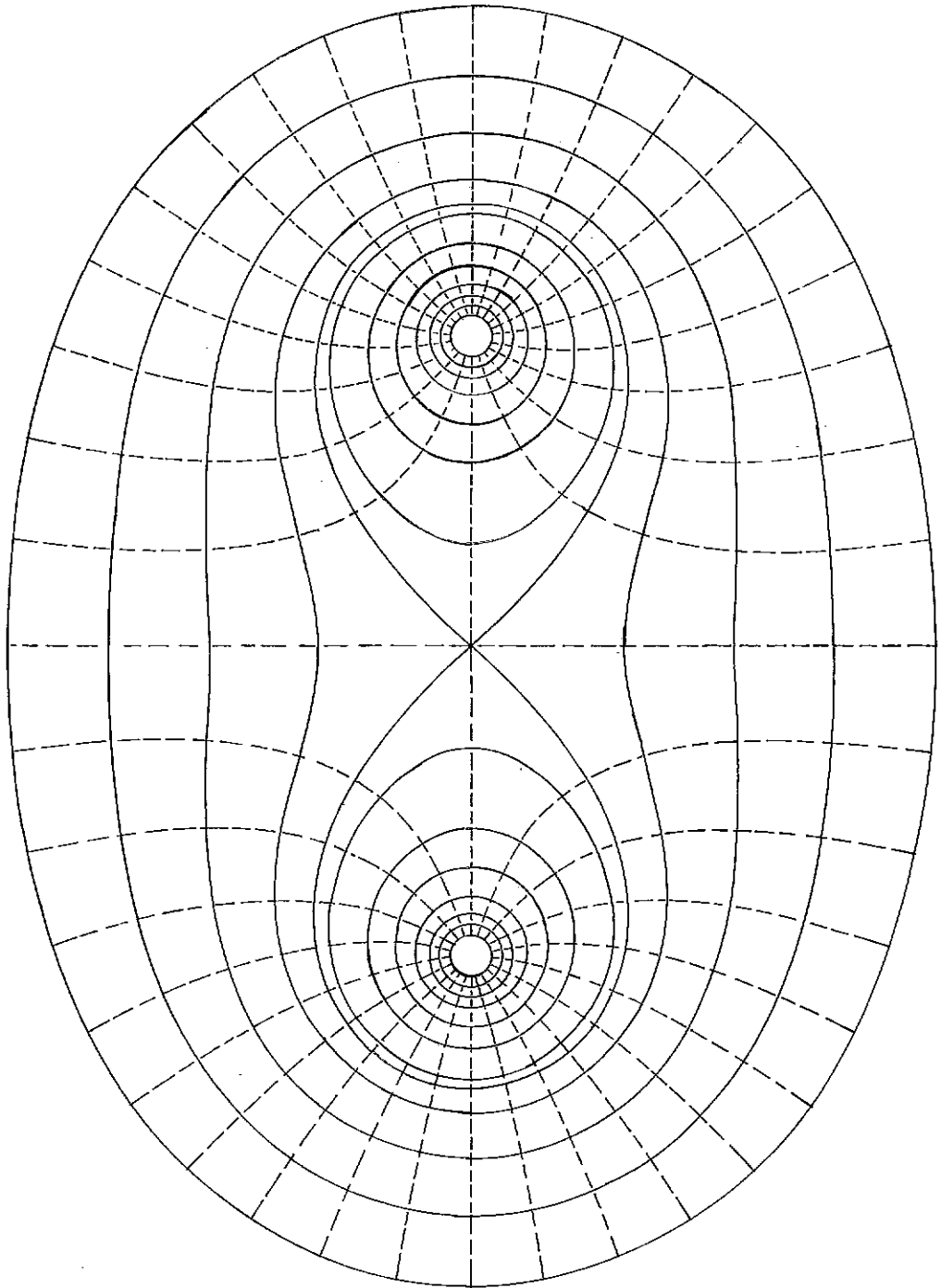


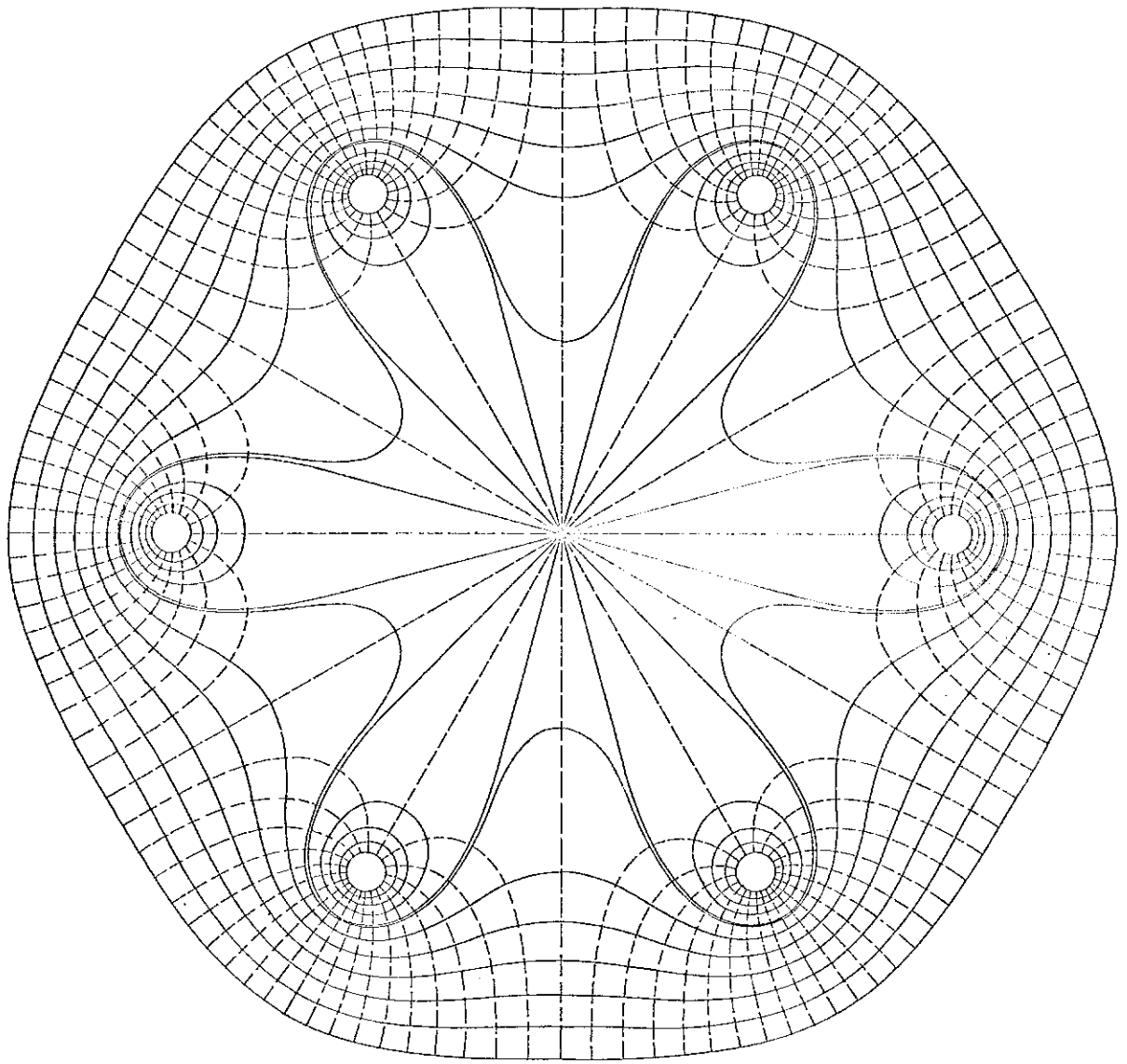


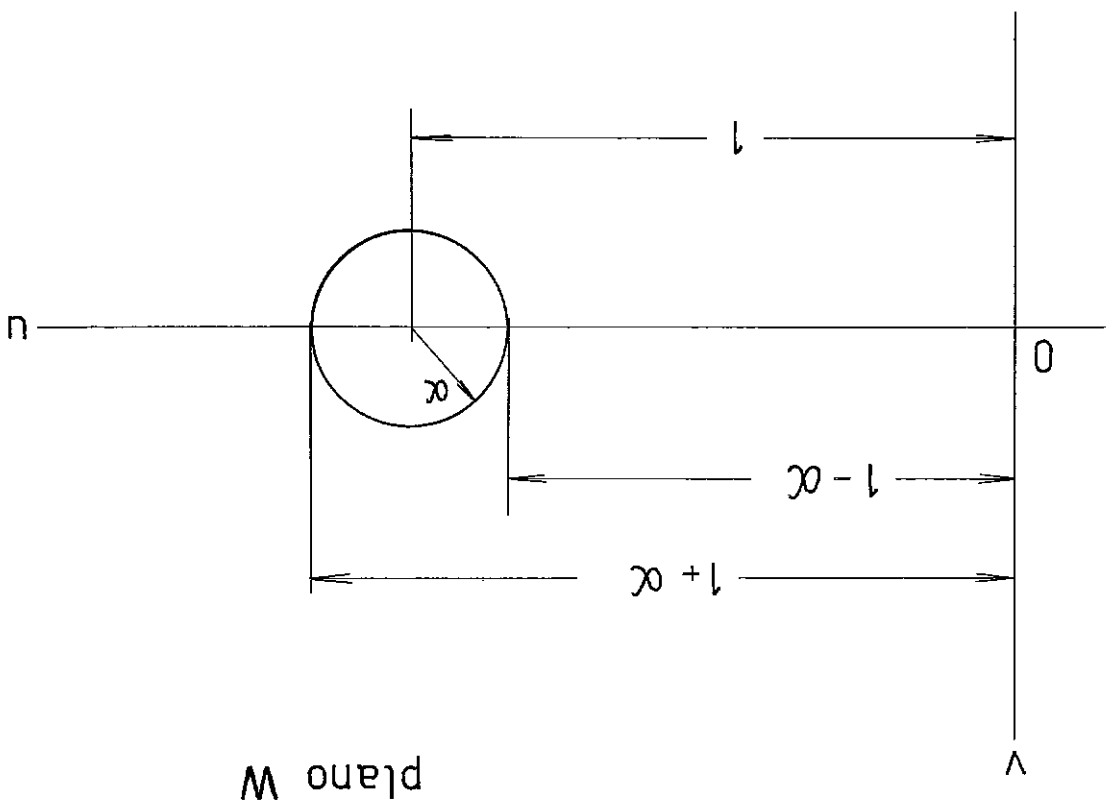
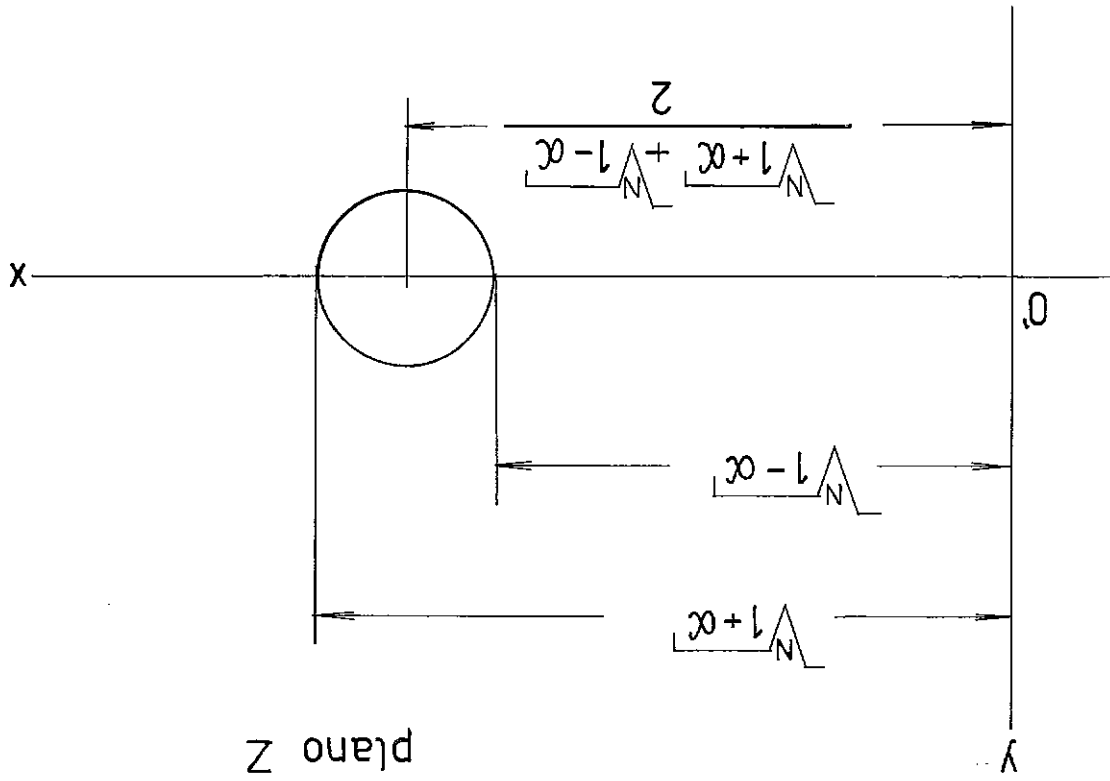


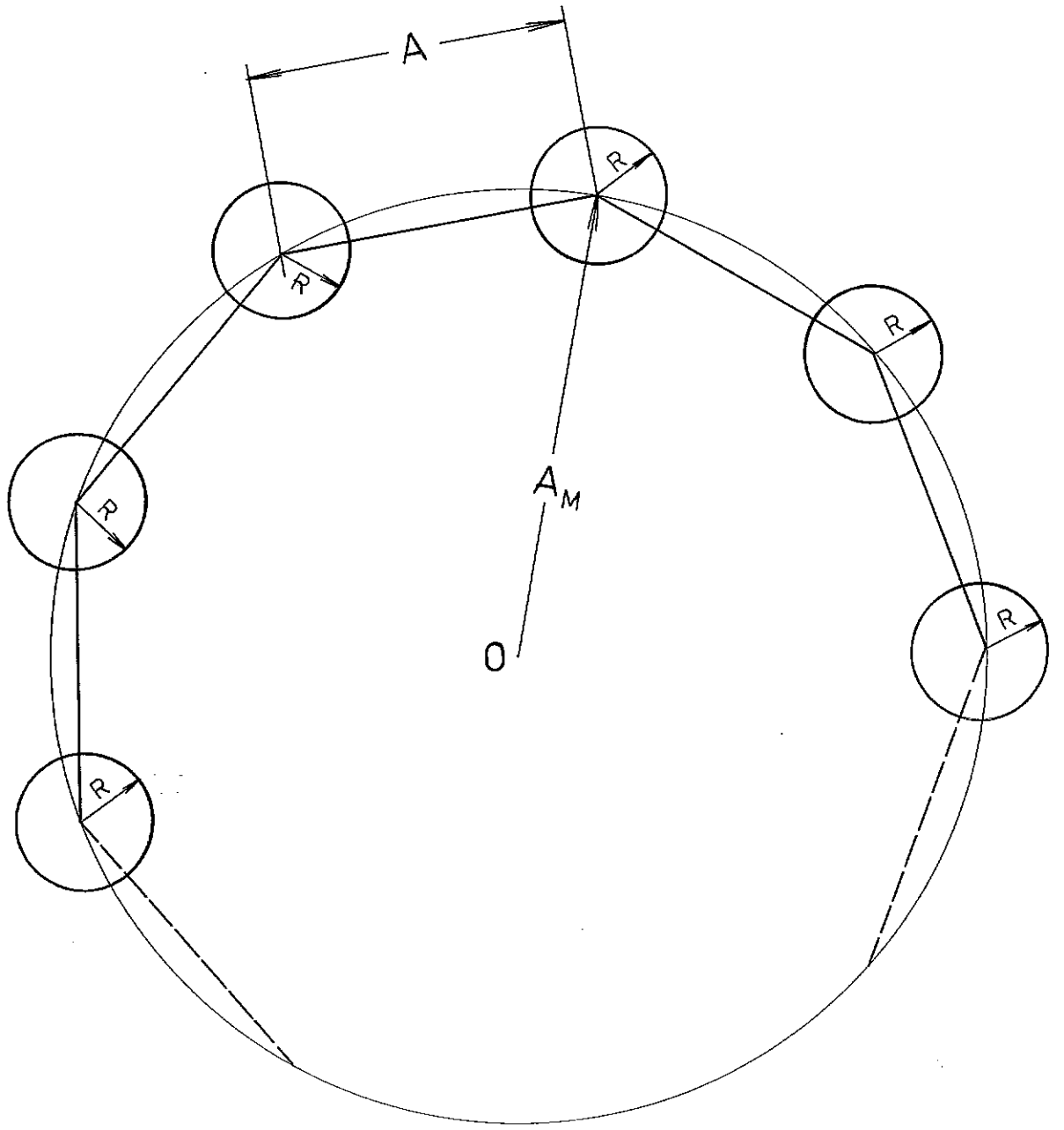


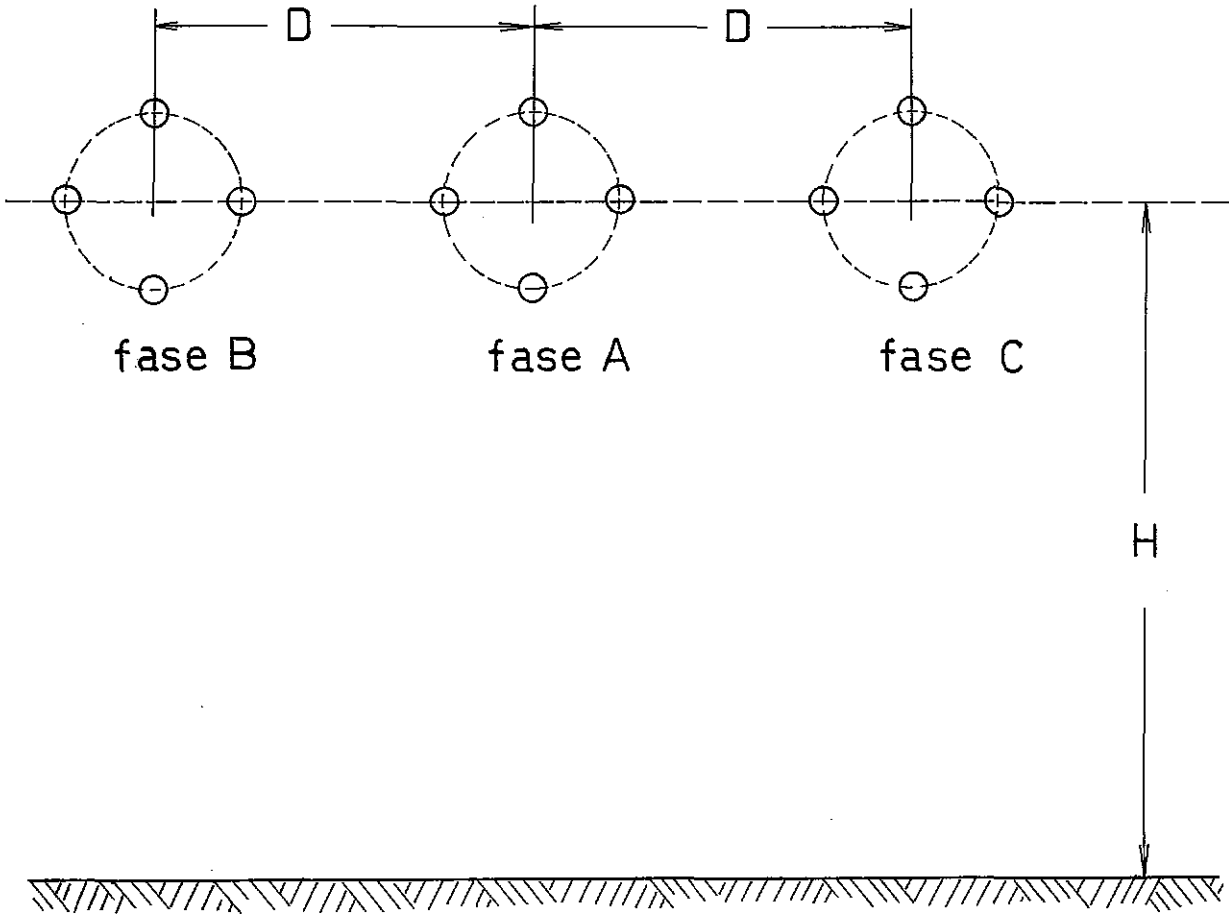


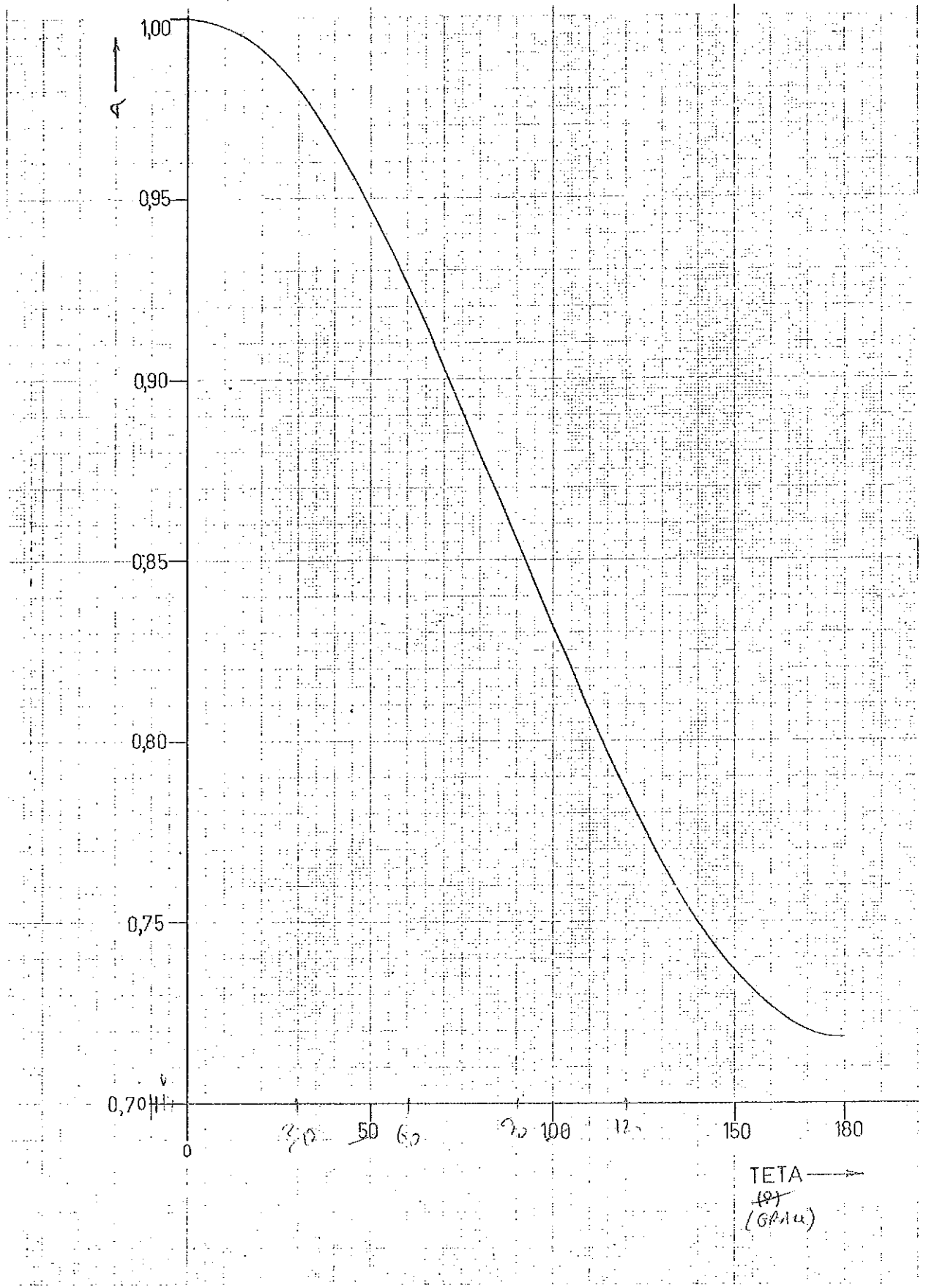












7.2.1 - Gráfico do campo elétrico relativo, σ , em função de θ , para os condutores do feixe.

