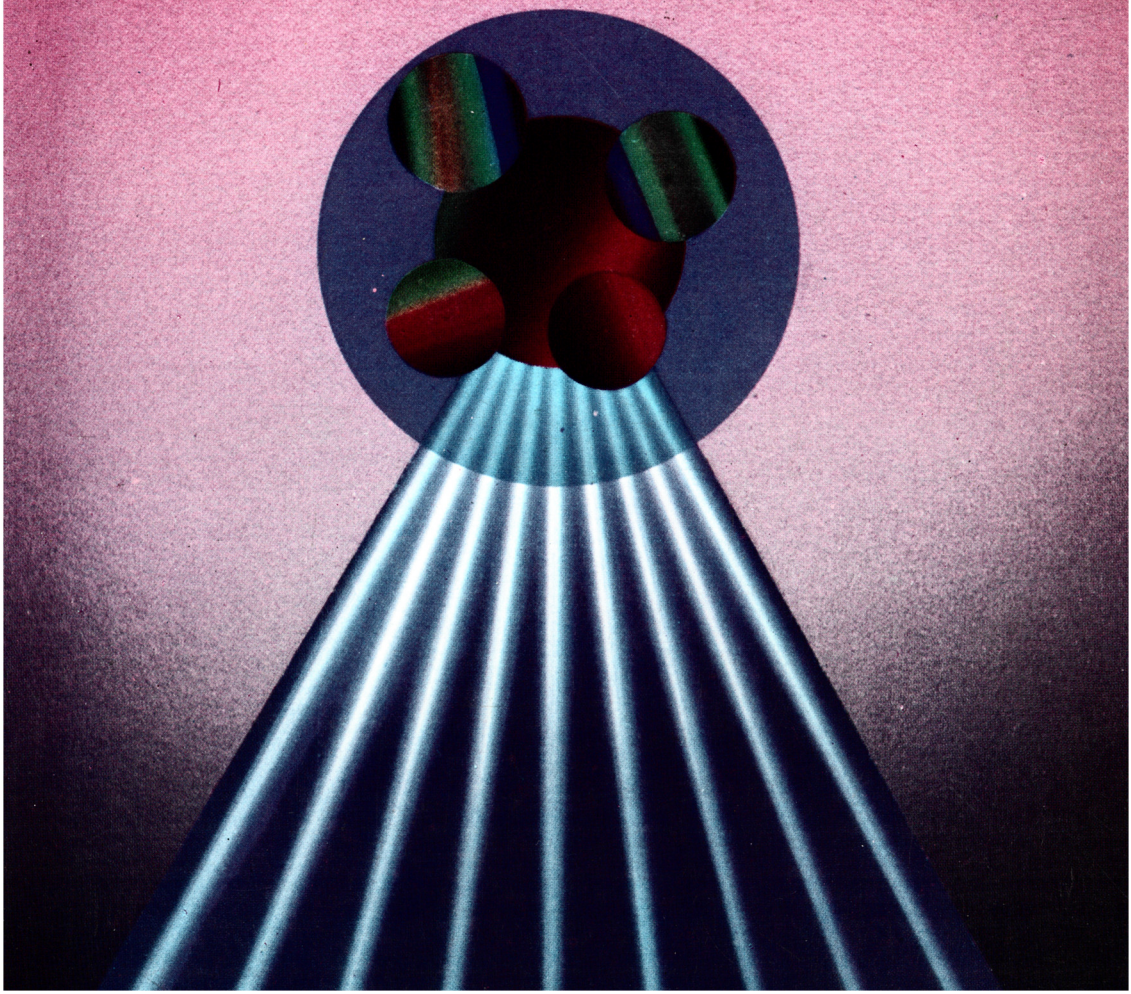
Aplicacoes de Computadores

FAUSTO ARINOS DE ALMEIDA BARBUTO

35 PROGRAMAS BASIC PARA MICROCOMPUTADORES



35 PROGRAMAS BASIC PARA MICROCOMPUTADORES

SÉRIE "APLICAÇÕES DE COMPUTADORES"

Coordenação Técnica
DONALDO DE SOUZA DIAS

BARBUTO	– 35 Programas BASIC Para Microcomputadores
BASTOS	– Programação COBOL
BORATTO	– BASIC Para Engenheiros e Cientistas
CARDOSO	– Programação Estruturada em COBOL
COUCEIRO/BARRENECHA	– Sistemas de Gerência de Banco de Dados Distribuídos
DIAS/GAZZANEO	– Projeto de Sistemas de Processamento de Dados
EADIE	– Minicomputadores – Teoria e Prática
GANE/SARSON	– Análise Estruturada de Sistemas
KUGLER/FERNANDES	– Planejamento e Controle de Sistemas de Informação
MAYNARD	– Programação Modular
PRADO	– Administração de Projetos com PERT/CPM
PRATES	– BASIC Aplicado – Um Enfoque Profissional
PUCCINI	– Introdução à Programação Linear
SÁ	– Programação PL/1
SILVA/HEIBEL	– VISICALC/CIRCALC/PROCALC
STRACK	– GPSS – Modelagem e Simulação de Sistemas
STRACK	– Sistemas de Processamento Distribuído
TAROUCO	– Redes de Comunicação de Dados
VON STAA	– Engenharia de Programas
ZIMMERMANN	– Linguagem de Programação APL

CONHEÇA TAMBÉM:

ALBRECHT	– Análise Numérica
BIANCHI/BEZERRA	– Microcomputadores – Arquitetura, Projeto e Programação
BORGES	– BASIC – Aplicações Comerciais
BOSCH	– COBOL – Fundamentos e Aplicações
DIAS/LUCENA/LIMA	– Programação FORTRAN
FORSYTHE	– Ciência dos Computadores vols. 1 e 2
FURTADO	– Teoria dos Grafos
FURTADO/PASSOS	– Introdução à Programação com PL/1
GUIMARÃES/LAGES	– Introdução à Ciência da Computação
GUIMARÃES/LAGES	– Algoritmos e Estruturas de Dados
GUSMAN/VASCONCELLOS	– Fluxogramas e Programação COBOL
KATZAN	– Segurança de Dados em Computação
PACITTI	– FORTRAN Monitor
PACITTI/ATKINSON	– Programação e Métodos Computacionais vols. 1 e 2
PEREIRA	– Computes, Grilo! Computador Para Todas as Idades
ROYO DOS SANTOS	– Processamento de Dados
RUAS	– Curso de Cálculo Numérico
SAPIRA/NESANELOVICZ	– Introdução à Linguagem APL
SUCESU	– Dicionário de Informática
VASCONCELLOS	– Computadores Eletrônicos e Processamento
VASCONCELLOS/SZERMAN	– O Centro de Processamento de Dados
WOOLDRIDGE/LONDON	– O Computador e o Executivo

35 PROGRAMAS BASIC PARA MICROCOMPUTADORES

FAUSTO ARINOS DE ALMEIDA BARBUTO

Engº químico, graduado pela UFRJ
Engº de produção, PETROBRÁS, Região de Produção do Sudeste,
Divisão Regional de Óleo e Gás

LIVROS
TÉCNICOS E
CIENTÍFICOS EDITORA S.A.
Rio de Janeiro-RJ • São Paulo-SP

FAUSTO ARINOS DE ALMEIDA BARBUTO

Revisor do texto:
Raymundo Paula de Arruda

Comissão de computação:
Professor Donaldo de Souza Dias
Professor Ysmar Vianna da Silva Filho
Professor Luís de Castro Martins
Professor Antonio Roberto Ramos Nogueira

Coordenador técnico da série:
Professor Donaldo de Souza Dias

CIP-Brasil. Catalogação na fonte
Sindicato Nacional dos Editores de Livros, RJ

Barbuto, Fausto Arinos de Almeida.
B215t 35 programas BASIC para microcomputadores / Fausto Arinos de Almeida Barbuto. — Rio de Janeiro: LTC — Livros Técnicos e Científicos Editora S.A., 1985.
(Aplicações de computadores)

1. Microcomputadores — Programação I. Título II. Série

84-0702 CDD — 001.642

ISBN: 85-216-0388-6

LIVROS TECNICOS E CIENTIFICOS EDITORA S.A.

MATRIZ	FILIAL
Av. Churchill, 94 — 4.º andar 20.220 — Rio de Janeiro — RJ Brasil — End. Telegráfico: LITECE Tels.: 240-6322 Vendas: 580-9374	Rua Vitória, 486 — 2.º andar 01.210 — São Paulo — SP Tel.: (011) 223-6823 Caixa Postal 4.817

À
Maria de Almeida

PREFÁCIO DO AUTOR

O objetivo desta obra é levar ao leitor alguns programas selecionados entre os de maior interesse e importância pertencentes às áreas de Economia e Finanças, Matemática, Estatística, Engenharia e aplicações gerais.

Dos trinta e cinco programas escolhidos que apresento, acompanho-os com algum embasamento teórico, assim como exemplos já resolvidos e sua aplicação.

Suponho que o leitor já tenha ultrapassado a fase inicial de aprendizado de BASIC, linguagem de alto nível em que foram concebidos os programas que constituem o cerne da obra. Por outro lado, não é necessário um conhecimento mais aprofundado de BASIC, a menos que queira o leitor compreender um pouco melhor a mecânica interna de cada programa, adaptá-la ou modificá-la de acordo com suas preferências e necessidades. Esta última parte é a mais interessante se encararmos o lado didático: os programas não são, em hipótese alguma, um ponto final; pelo contrário, deve deles fazer o leitor uma base para novos avanços vislumbrando dentro de sua ótica particular novas aplicações para eles.

Os programas que apresentaremos são orientados para o BASIC com que trabalham os microcomputadores compatíveis com o TRS-80 Modelo I — computador pessoal norte-americano muito difundido nos EUA e que possui vários similares nacionais — e rodam sem problemas nos seguintes aparelhos: CP 300, CP 500, D-8000, D-8001, JR Sysdata, DGT-100, DGT-101, entre outros. A configuração de hardware empregada é: CPU com 16 K Bytes de RAM e um gravador K-7 (unidade de fita). Não há comandos de impressora ou disk drive. Se, todavia, o leitor possuir um micro de outra linha que não a do TRS-80, saliente-se que os diversos dialetos BASIC não diferem muito entre si, e não há maiores problemas para adaptar os programas para outro aparelho, bastando para isso comparar as equivalências entre os comandos da máquina em questão e os do TRS-80 Mod. I.

Espero que esta obra seja de grande valia para o leitor. Se isto acontecer, por pouco que seja, sentir-me-ei plenamente recompensado.

Rio de Janeiro, 10 de setembro de 1984.

SUMÁRIO

PARTE I

ECONOMIA E FINANÇAS

PROGRAMA GERADOR DE TABELA PRICE (VERSÃO 1)

1

A Tabela Price é, sem dúvida, uma das ferramentas mais empregadas em economia e em finanças. Através dela convertemos fluxos de caixa em valores atuais e/ou futuros, fazemos previsões de rentabilidade, projetamos investimentos etc. Não poucos livros sobre economia, têm-na como apêndice. Essas tabelas não são suficientes, apesar disso, para cobrir uma extensa gama de taxas e períodos, que às vezes se fazem necessários usar.

O programa sobre o qual em breve falaremos obtém os fatores que convertem:

Valor atual (P) em montante (S): FPS
Valor futuro (S) em atual (P): FSP
Valor futuro (S) em série uniforme (R): FSR
Série uniforme (R) em valor futuro (S): FRS
Valor atual em série uniforme: FPR
Série uniforme em valor atual: FRP.

O valor atual é o capital que aplicamos ou tomamos emprestado no "momento zero" da transação financeira.

O valor futuro é o capital que recuperaremos — ou devolveremos — após "N" períodos, a uma taxa "i".

A série uniforme é constituída de parcelas fixas que aplicamos ou resgatamos no final de cada um dos "N" períodos que compõe a série.

USANDO O PROGRAMA:

É necessário o *INPUT* das variáveis:

- Taxa de juros (%), variável "I";
- Período inicial, variável "N1"
- Período final, variável "N2"

Nenhuma das variáveis é indexada.

Os dois últimos definem o intervalo da tabela que vamos obter. O período inicial e final devem ser necessariamente números inteiros.

→ Exemplo 1:

Taxa de juros: 5%
Período inicial: 1
Período final: 3

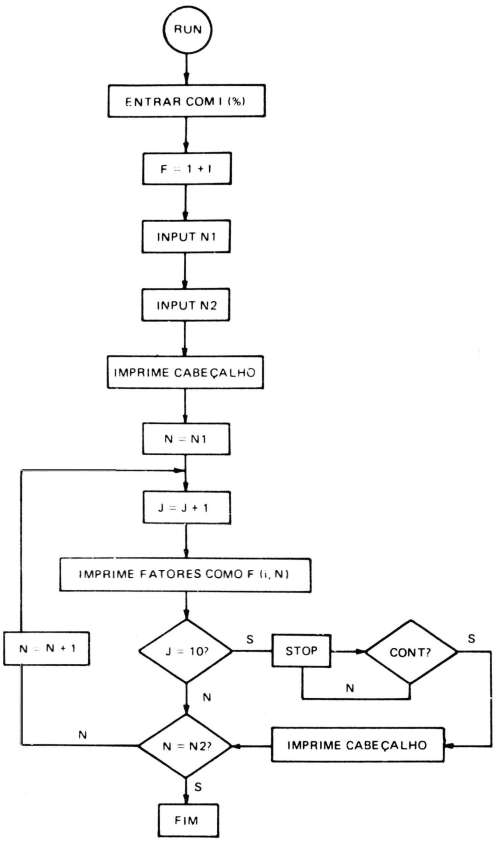
Estes dados causam o seguinte *OUTPUT*:

TAXA DE JUROS: 5%

N	FSP	FPS	FSR	FRS	FPR	FRP
1	Ø.9524	1.Ø5ØØ	1.ØØØØ	1.ØØØØ	1.Ø5ØØ	Ø.9524
2	Ø.9Ø7Ø	1.1Ø25	Ø.4878	2.Ø5ØØ	Ø.5378	1.8594
3	Ø.8638	1.1576	Ø.3172	3.1525	Ø.3672	2.7233

Se N > 10, o comando "CONT" <NEWLINE> listará os períodos restantes, tantas vezes quantas foram necessárias.

FLUXOGRAMA:



TEORIA:

Vamos representar graficamente um fluxo de caixa.

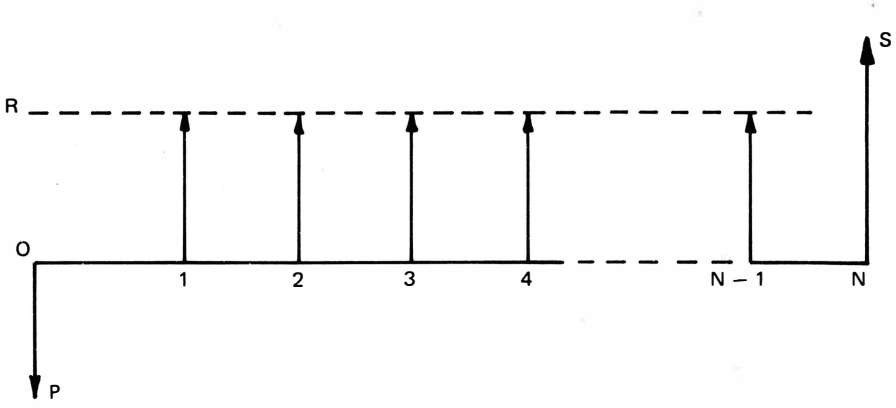


FIG. 1. Um fluxo de caixa típico.

As setas que apontam para cima e para baixo representam fluxos positivos (entrada de capital) e negativos (saída), respectivamente.

O valor futuro "S" é também chamado de *montante.*

A equação que representa a Fig. 1 é:

$$S = R * FRS\ (i, N) - P * FPS\ (i, N)$$

ou ainda:

$$-P = R * FRP\ (i, N) + S * FSP\ (i, N)$$

ou então:

$$R = S * FSR\ (i, N) - P * FPR\ (i, N)$$

Todas as três são válidas. Analisando-as, pode o leitor julgar melhor a Tabela Price como ferramenta poderosa que é.

As expressões para os fatores são:

$$FSP = \frac{1}{(1+i)^n} \qquad FPS = (1+i)^n$$

$$FSR = \frac{i}{(1+i)^n - 1} \qquad FRS = \frac{(1+i)^n - 1}{i}$$

$$FPR = \frac{i\,(1+i)^n}{(1+i)^n - 1} \qquad FRP = \frac{(1+i)^n - 1}{i\,(1+i)^n}$$

```
TAXA DE JUROS(%)? 5
PERIODO INICIAL? 1
PERIODO FINAL? 3
```

```
                    TAXA DE JUROS:  5 %
 N        FSP        FPS        FSR        FRS        FPR        FRP
 1      0.9524     1.0500     1.0000     1.0000     1.0500     0.9524
 2      0.9070     1.1025     0.4878     2.0500     0.5378     1.8594
 3      0.8638     1.1576     0.3172     3.1525     0.3672     2.7233
READY
).
```

```
10 'PROGRAMA GERADOR DE TABELA PRICE.
20 'AUTOR: FAUSTO A. DE A. BARBUTO.
30 'TELS: (021)2709765 E (021)2304576.
40 'DATA: 25/OUT/83.   VERSAO 1.0
50 ON ERROR GOTO 220
60 CLS:INPUT"TAXA DE JUROS(%)";I
70 I=.01*I
80 F=1+I
90 INPUT"PERIODO INICIAL";N1
100 INPUT"PERIODO FINAL";N2
110 GOSUB 190
120 FOR N=N1 TO N2
130 J=J+1
140 PRINT@(64*J+64),N;
150 PRINT USING"######.####";1/F[N;F[N;I/(F[N-1);(F[N-1)/I;I*F[N/(F[N-1);(F[N-1)/
I/F[N
160 IF J=10 OR J=20 OR J=30 THEN STOP
170 IF J=10 OR J=20 OR J=30 THEN GOSUB 190
180 NEXT N: END
185 'SUBROTINA PARA IMPRESSAO DE CABECALHO.
190 CLS:PRINT@20,"TAXA DE JUROS: ";100*I;"%"
200 PRINT@65,"N";"        FSP        FPS        FSR        FRS        FPR        FRP"
210 J=0: RETURN
220 END
```

PROGRAMA GERADOR DE TABELA PRICE (VERSÃO 2)

Existem outros fatores de conversão de fluxo de caixa que, por motivos de otimização da memória de vídeo foram colocados num programa à parte. Esta versão discutiremos agora.

Tais fatores são os de conversão da série gradiente. Esse tipo de série consiste de entradas/saídas de capital ao final de cada um dos "N" períodos que a compõe e que aumentam sucessivamente de uma quantia "X", segundo o fluxo de caixa:

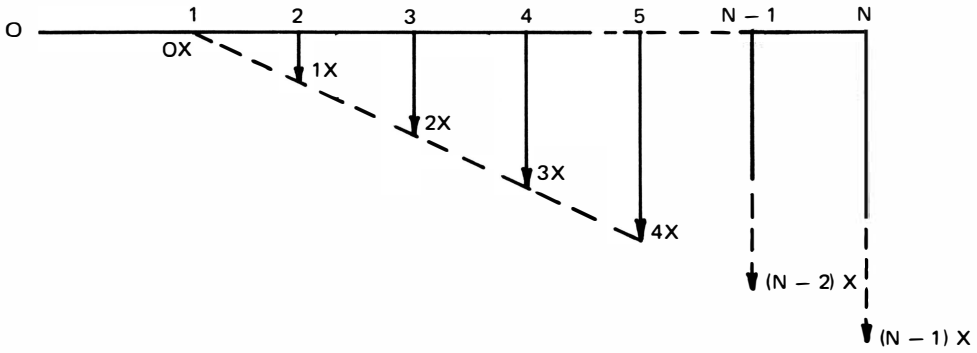


FIG. 1. Série Gradiente.

Esta versão calcula quatro fatores, que são:

FGR: Série gradiente em uniforme
FRG: Série uniforme em gradiente
FGP : Série gradiente em valor atual
FPG: Valor atual em série gradiente.

USANDO O PROGRAMA:

Como na versão 1, esse programa solicita os mesmos "*INPUT'S*"

→ Exemplo 1:

Os dados:

Taxa de juros: 2,5%
Período inicial: 5
Período final: 8

Causam o seguinte "*OUTPUT*":

N	FGR	FRG	FGP	FPG
5	1.950	0.513	9.061	0.110
6	2.428	0.412	13.372	0.075
7	2.901	0.345	18.419	0.054
8	3.370	0.297	24.164	0.041

Se N2 — N1 > 10, "CONT" <NEWLINE> listará os períodos restantes, tantas vezes quantas forem necessárias.

→ Exemplo 2: Aplicou-se o capital de Cr$ 100.000,00 a juros de 2,5% ao mês, durante sete meses. Qual a série gradiente equivalente?

RESPOSTA:

$$P = G \cdot FGP\ (2{,}5\%, 7)$$

ou:

$$G = P \cdot FPG\ (2{,}5\%, 7)$$

Executando o programa conforme o exemplo 1, temos, na 5ª coluna 3ª linha FPG = 0,054 (Vide tabela acima). Então:

$$G = 100.000 * 0{,}054 = Cr\$\ 5.400{,}00$$

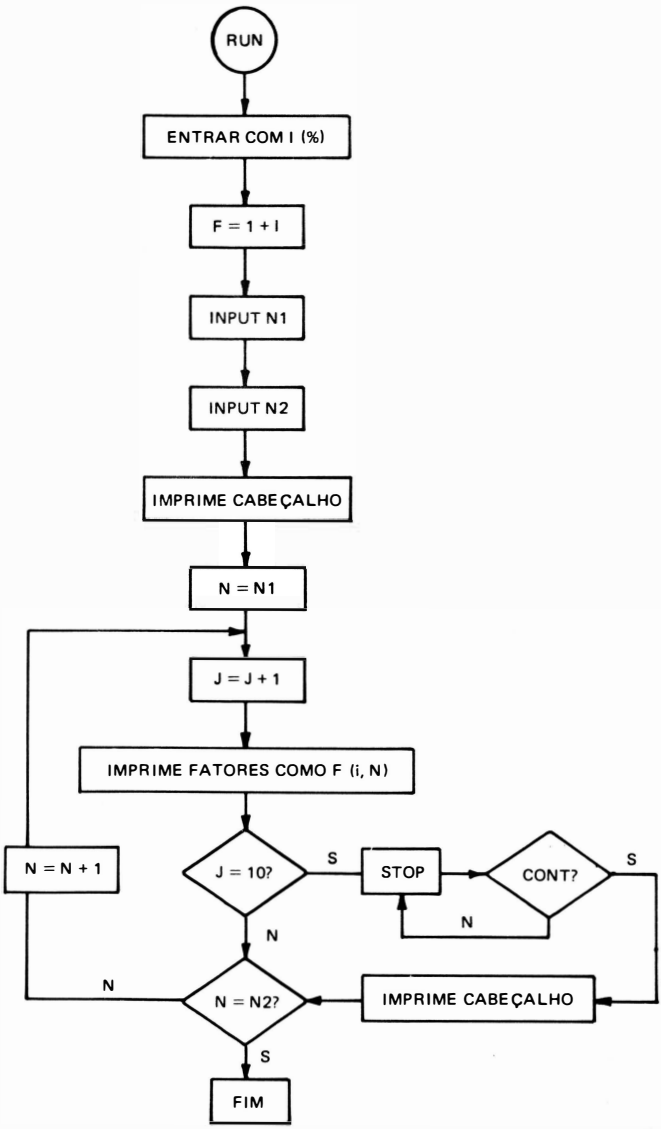
Assim, temos as receitas equivalentes à aplicação de capital:

N	RECEITAS (Cr$)
1	0
2	5.400
3	10.800

N	RECEITAS (Cr$)
4	16.200
5	21.600
6	27.000
7	32.400

Correspondentes a (N – 1) * 5.400 cruzeiros. É evidente que a soma das sete parcelas acima é superior a Cr$ 100.000,00. Saberia o leitor explicar porquê?

FLUXOGRAMA: IDÊNTICO AO DA VERSÃO 1.



```
TAXA DE JUROS(%)? 2.5
PERIODO INICIAL? 5
PERIODO FINAL? 8.
```

```
                    TAXA DE JUROS:  2.5 %
 N          FGR          FRG          FGP          FPG
 5        1.950        0.513        9.061        0.110
 6        2.428        0.412       13.372        0.075
 7        2.901        0.345       18.419        0.054
 8        3.370        0.297       24.164        0.041
READY
>.
```

```
10 'PROGRAMA GERADOR DE TABELA PRICE.
20 'AUTOR: FAUSTO A. DE A. BARBUTO.
30 'TELS: (021)2709765 E (021)2304576.
40 'DATA: 25/OUT/83.   VERSAO 1.1
60 CLS:INPUT"TAXA DE JUROS(%)";I
70 I=.01*I
80 F=1+I
90 INPUT"PERIODO INICIAL";N1
100 INPUT"PERIODO FINAL";N2
110 GOSUB 190
120 FOR N=N1 TO N2
130 J=J+1
135 FR=1/I-N/(F[N-1): FGP=((F[N-1)/I[2-N/I)/F[N
140 PRINT@(64*J+64),N;
150 PRINT USING"########.###";FR,1/FR,FGP,1/FGP
160 IF J=10 OR J=20 OR J=30 THEN STOP
170 IF J=10 OR J=20 OR J=30 THEN GOSUB 190
180 NEXT N: END
185 'SUBROTINA PARA IMPRESSAO DO CABECALHO.
190 CLS:PRINT@20,"TAXA DE JUROS: ";100*I;"%"
200 PRINT@65,"N";"          FGR          FRG          FGP          FPG  "
210 J=0: RETURN
```

PROGRAMA-CREDIÁRIO

É comum, hoje em dia, pagar-se um débito contraído para aquisição de um bem de consumo ou o recebimento de um serviço com um plano de pagamento em que parcelas fixas de capital — as prestações — são cedidas ao credor no final de cada período que compõe uma série de "N" períodos. No n-ésimo pagamento, a dívida está saldada. Este plano de pagamento é o *crediário.*

Esta prática é usual — e cada vez mais — em tempos de dinheiro difícil e inflação elevada. Como normalmente a soma das "N" parcelas supera (às vezes em muito) o preço do bem/serviço à vista, concluímos que existem juros embutidos nos crediários. Eles são aí colocados para cobrir as desvalorizações da moeda e riscos da falta de pagamento por parte do devedor.

Veremos a seguir como calcular esses juros.

USANDO O PROGRAMA:

O programa necessita do *INPUT* de cinco variáveis:

PV: Preço à vista do produto (Cr$)
E: Valor da entrada (Cr$)
R: Valor das prestações (Cr$)
N: Número das prestações
T: Prazo de carência (períodos)

Como nenhuma delas é indexada, não há dimensionamentos.

A carência é o período que o credor leva para receber a primeira parcela (prestação). É muito comum nos planos de pagamento do tipo "leve agora no Natal e só comece a pagar depois do carnaval." Para efeito de cálculo, usamos juros de carência de 10% ao período (linha 250), mas o usuário pode modificá-la se bem o quiser. Por exemplo, para 5%:

```
250 P = P * 1.05 [T]
```

(1.05 é o fator 1 + i, onde i é 5% (0,05).

Vejamos alguns exemplos práticos:

→ Exemplo 1: Compra de um microcomputador em abril/82.

PV = Cr$ 70.000,00
E = Cr$ 26.666,00
R = Cr$ 26.666,00
N = 2
T = 0 (não há carência)

Fazemos então:

COMANDO

RUN
<NEWLINE>
70.000
<NEWLINE>
26.666
<NEWLINE>
26.666
<NEWLINE>
2
<NEWLINE>
0
<NEWLINE>

Após este último "NEWLINE", o programa propriamente dito é executado. A resposta é acompanhada do "eco" dos dados introduzidos para simples conferência. A resposta é:

Taxa de juros = 15.0312% ao período.

→ Exemplo 2: Preço à vista: Cr$ 100.000,00 em dezembro ou 3 X Cr$ 50.000,00 a partir de março.

PV = Cr$ 100.000,00
E = Não há
R = Cr$ 50.000,00
N = 2
T = 3 meses/períodos (dezembro a março)

RESPOSTA: Juros de 6,22323% ao período.

→ Exemplo 3: Escolhendo crediários

1º Plano:

À vista: Cr$ 40.000,00
Entrada: Cr$ 10.000,00
Prestação: Cr$ 12.000,00
Número de prestações: 3
Carência: Não há
"Total": Cr$ 46.000,00

2º Plano:

À vista: Cr$ 40.000,00
Entrada: Cr$ Não há
Prestações: Cr$ 10.000,00
Número de prestações: 5
Carência: 2 períodos
"Total": Cr$ 50.000,00

Escolhemos o 2º plano, apesar do "total a pagar" ser Cr$ 4.000,00 mais caro que o 1º plano. Acontece que o 2º plano tem juros de 1,094% ao período, enquanto o 1º plano tem juros de 9,70108% ao período. É bom frisar que este tipo de comparação só é válido quando os preços à vista são idênticos para os planos a serem analisados. Pagamos (ilusoriamente) mais no plano 2, embora o plano 1 é o que realmente nos sai mais caro.

Esta talvez seja a aplicação mais útil para o programa que apresentamos.

TEORIA:

A equação financeira que representa um crediário é a seguinte:

$$P = R \cdot FRP(i, n)$$

onde:

P: é o valor atual ou principal (leia-se preço à vista)

R: é a prestação, ou "série uniforme", um valor constante

i: Taxa de juros

n: Número de períodos

Como num crediário "P", "R" e "n" estão fixados, existe um – e somente um – valor de "i" que satisfaz a equação acima, de modo que:

$$P - R \cdot FRP(i, n) \equiv 0$$

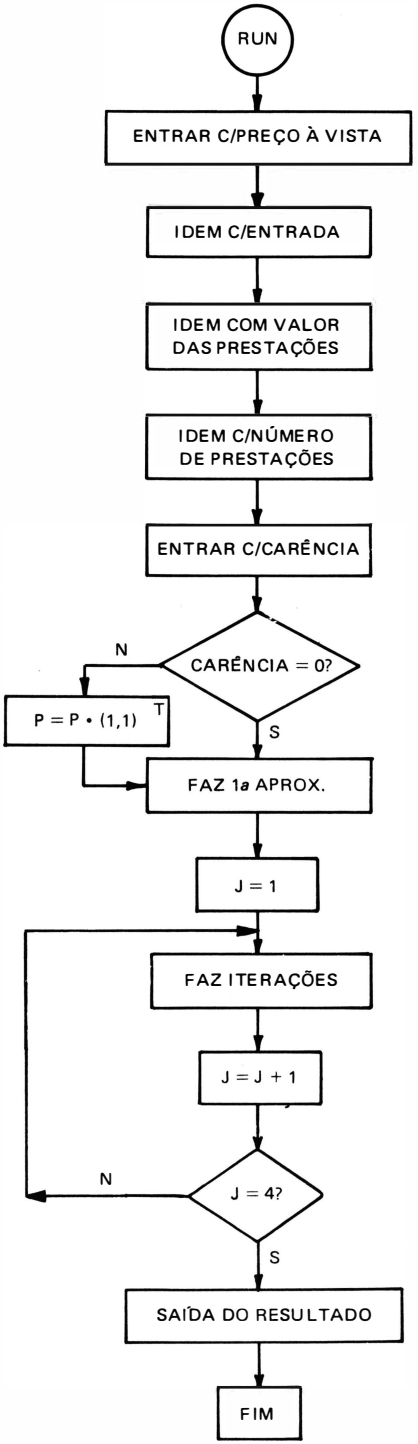


FIG. 1. Fluxograma do Programa.

onde:

$$FRP = \frac{(1 + i)^n - 1}{i\,(1 + i)^n}$$

Trata-se, obviamente, de calcular a raiz "i" que satisfaça às igualdades acima, anulando-as. Vamos usar para isso o método da raiz de Newton-Raphson, que é iterativo. Sua premissa de cálculo é:

$$X_{i+1} = X_i - \frac{f(X_i)}{f'(X_i)}\text{, onde } f'(X_i) = \left(\frac{df}{dX}\right)_{X_i}$$

Usando termos gerais, no nosso caso,

$$i_{j+1} = i_j - \frac{R \cdot FRP(i, n)}{R \cdot \dfrac{\partial}{\partial_i}\,[FRP(i, n)]}$$

ou:

$$i_{j+1} = i_j - \frac{FRP(i, n)}{\dfrac{\partial}{\partial_i}\,[FRP(i, n)]}$$

que, desenvolvendo, fornece:

$$i_{j+1} = i_j - \frac{\dfrac{(1 + i)^n - 1}{i\,(1 + i)^n} - \dfrac{P}{R}}{\left(\dfrac{1}{i^2\,(1 + 1)^{n+1}}\right)[1 + (n + 1)\,i - (1 + i)^{n+1}]}$$

Uma primeira aproximação toma a forma:

$$i_0 = \frac{R}{P} - \frac{P}{n^2\,R}$$

e esta consta na linha 100. Quatro iterações são mais do que o suficiente para conseguirmos um bom resultado.

```
PRECO 'A VISTA: CR$  70000

ENTRADA: CR$  26666

NUMERO DE PRESTACOES:  2

VALOR DAS PRESTACOES: CR$  26666

PRAZO DE CARENCIA:  0  PERIODOS


--------)TAXA DE JUROS:  15.0312 % AO PERIODO.
Break na 240
READY
).
10 'PROGRAMA PARA CALCULAR OS JUROS DE UM CREDIARIO
20 'PELA RAIZ DE NEWTON-RAPHSON.
30 'AUTOR: FAUSTO ARINOS DE ALMEIDA BARBUTO.
40 'DATA: 27/OUT/83        VERSAO 1.0
50 CLS:INPUT"PRECO 'A VISTA";PV
60 INPUT"VALOR DA ENTRADA";E
70 P=PV-E
80 INPUT"VALOR DAS PRESTACOES";R
85 INPUT"NUMERO DE PRESTACOES";N
90 INPUT"PRAZO DE CARENCIA";T
95 IF T)0 THEN  GOSUB 250
100 I0=R/P-P/(R*N[2)
110 FOR J=1 TO 4
120 F0=(1+I0)[N
130 F1=(1+I0)[(N+1)
140 I1=I0-(((F0-1)/(I0*F0)-P/R)*F1*I0[2)/(1+I0*(N+1)-F1)
150 I0=I1
160 NEXT J
170 CLS
180 PRINT@64,"PRECO 'A VISTA: CR$ ";PV
190 PRINT@192,"ENTRADA: CR$ ";E
200 PRINT@320,"NUMERO DE PRESTACOES: ";N
210 PRINT@448,"VALOR DAS PRESTACOES: CR$ ";R
220 PRINT@576,"PRAZO DE CARENCIA: ";T;" PERIODOS"
230 PRINT@768,"--------)TAXA DE JUROS: ";100*I1;"% AO PERIODO."
240 STOP:END
245 'SUBROTINA PARA CASOS DE CARENCIA.
250 P=P*1.1[T
260 RETURN
```

CONVERSÃO DE JUROS

De uma maneira geral, todos sabemos o significado da palavra "juros", ainda que de forma vaga ou imprecisa. "Sentimos" os efeitos dessa palavra através da inflação, do custo de vida, da majoração das taxas de luz e gás, do aumento do aluguel etc. Às vezes, no entanto, fica difícil calcular os efeitos repetidos ao longo do tempo por uma taxa incidente em "N" períodos. Exemplo: Qual o efeito de 1% ao mês em quatro anos? Em um ano? Veremos adiante como responder a essa pergunta.

Juros, normalmente, são compostos. Significa dizer a grosso modo: incidem sobre o montante imediatamente anterior ou, ainda, são cumulativos. Existem também os juros nominais, não discutidos aqui. Assim, a equação que calcula os juros no período composto de "N" subperíodos tem a forma:

$$i_N = (1 + i_K)^N - 1 \text{ (em fração decimal)}$$

onde i_K. É a taxa do subperíodo. Se o subperíodo é composto, a fórmula se torna:

$$i_N = (1 + i_K)^{N/K} - 1$$

Inversamente, para obtermos i_K a partir de i_N:

$$i_K = (1 + i_N)^{K/N} - 1$$

Parece complicado, à primeira vista. Com a resolução dos exemplos, as coisas ficarão mais claras.

→ Exemplo 1: Qual a taxa de inflação mensal que, repetida por doze meses, ocasiona uma inflação de 100%?

RESPOSTA: A taxa é 100%, o número de subperíodos (meses) que compõe o período (ano) é igual a 12 e, como queremos a taxa no subperíodo, temos o número de subperíodos igual a 1 (um). (Se quiséssemos a taxa bimestral, o número de subperíodos é igual a 2 (dois)).

Obtemos como resposta:

$$i_K = 5{,}94631\%$$

A princípio, seríamos levados a pensar que a taxa seria:

$$i = \frac{100}{12} = 8{,}33\%$$

Verificamos, portanto, que não é.

→ Exemplo 2: A taxa da inflação em maio/84 foi de 8,9%. Qual a inflação anual, contando com a repetição, durante 12 meses, desta taxa de 8,9%?

RESPOSTA: Subperíodos no período: 12
Subperíodos: 1
Código de operação: 2
→ Taxa: 178,186%

→ Exemplo 3: Voltando à pergunta no início do capítulo: Qual a taxa acumulada ao final de quatro anos, com 1% ao mês? E ao final do 1º ano? E do 2º ano?

RESPOSTA: Em quatro anos:
Subperíodos no período: 48 (4 X 12)
Subperíodo: 01 (mês)
Código de operação: 2
→ Taxa = 61,2217%

Ao final do 1º ano:

→ Taxa = 12,6823%

Ao final do 2º ano:

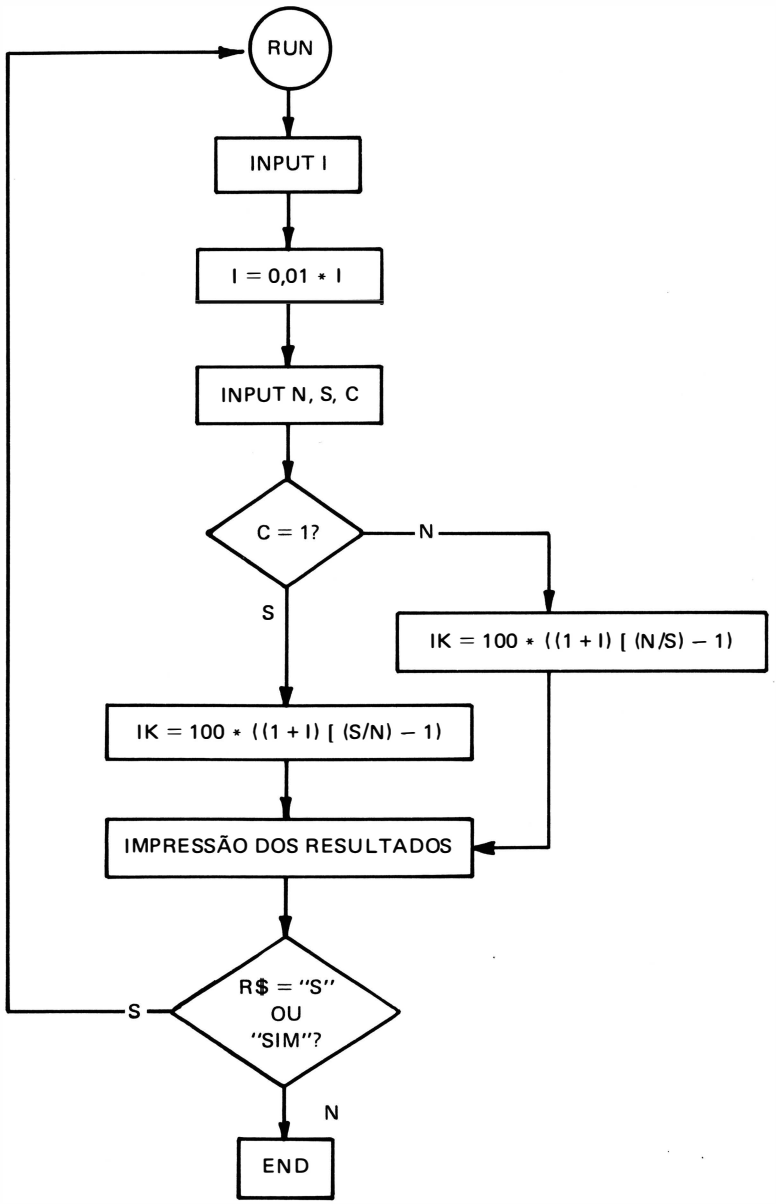
→ Taxa = 26,9731%

→ Exemplo 4: Qual a taxa trimestral que, em dois anos, acumula 30% de juros?

RESPOSTA: Subperíodos no período: 24 (meses)
Número de subperíodos: 3 (meses)
Código de operação: 1
→ Taxa: 3,33393%

OBS.: Se usássemos como subperíodo o trimestre, o número de subperíodos no período seria igual a oito, e a resposta seria a mesma. Verifique.

FLUXOGRAMA



```
TAXA DE JUROS (%)? 100

NU'MERO DE PERI'ODOS? 1

NU'MERO DE SUBPERI'ODOS? 12

PARA OBTER:
             -TAXA DO SUBPERI'ODO: CO'DIGO '1'.
             -TAXA DO PERI'ODO:    CO'DIGO '2'.

CO'DIGO? 2.

**************************************************
TAXA DE JUROS DO SUBPERI'ODO: 100 %.

NU'MERO DE SUBPERI'ODOS: 12

NU'MERO DE PERI'ODOS: 1

-----) TAXA DE JUROS DO PERI'ODO: 5.94631 %.
**************************************************

QUER EXECUTAR DE NOVO (S/N)?
```

```
10 'PROGRAMA CONVERSAO DE JUROS;
20 'AUTOR: FAUSTO A. DE A. BARBUTO.   DATA: 12/MAIO/'84.
30 CLS: INPUT"TAXA DE JUROS (%)"; I
40 I=.01*I
50 PRINT: INPUT"NU'MERO DE PERI'ODOS"; N
60 PRINT: INPUT"NU'MERO DE SUBPERI'ODOS"; S
70 PRINT: PRINT"PARA OBTER:
             -TAXA DO SUBPERI'ODO: CO'DIGO '1'.
             -TAXA DO PERI'ODO:    CO'DIGO '2'."
80 PRINT: INPUT"CO'DIGO"; C
90 ON C GOTO 100, 170
95 ' CA'LCULO DA TAXA DO SUBPERI'ODO.
100 IK=100*((1+I)[(S/N)-1)
105 CLS: PRINT@256,STRING$(50,"*")
110 PRINT@320,"TAXA DE JUROS DO PERI'ODO:";100*I;"%."
120 PRINT@448,"NU'MERO DE PERI'ODOS:";N
130 PRINT@576,"NU'MERO DE SUBPERI'ODOS:";S
140 PRINT@704,"-----) TAXA DE JUROS DO SUBPERI'ODO:";IK;"%."
150 PRINT@768,STRING$(50,"*")
160 GOTO 240
165 ' CA'LCULO DA TAXA DO PERI'ODO.
170 IK=100*((1+I)[(N/S)-1)
180 CLS: PRINT@256,STRING$(50,"*")
190 PRINT@320,"TAXA DE JUROS DO SUBPERI'ODO:";100*I;"%."
200 PRINT@448,"NU'MERO DE SUBPERI'ODOS:";S
210 PRINT@576,"NU'MERO DE PERI'ODOS:";N
220 PRINT@704,"-----) TAXA DE JUROS DO PERI'ODO:";IK;"%."
230 PRINT@768,STRING$(50,"*")
240 PRINT: INPUT"QUER EXECUTAR DE NOVO (S/N)"; R$
250 IF R$="S" OR R$="SIM" THEN RUN ELSE END
```

CONVERSÃO DE FLUXOS DE CAIXA

Já vimos anteriormente um programa gerador de Tabela Price apresentado em duas versões para séries P-R-S e G-R-P. Agora, usaremos os fatores gerados por essas tabelas mas, desta vez, faremos a conversão diretamente, sem auxílio da Price Table, utilizando diretamente os fatores.

O PROGRAMA:

Para que o usuário possa utilizá-lo, há que se introduzir os dados seguintes:

- Código da operação: de 1 a 6
- Número de períodos: N (N é um número inteiro)
- Taxa de juros (%)
- O que se deseja converter: montante, principal ou série uniforme.

Destarte, acionando o comando *RUN,* uma mensagem no vídeo nos indica o código a introduzir. O código é 3 (três), se quisermos converter uma série em montante; é 1 (um), se quisermos obter montante a partir do valor atual etc. Após o *INPUT* do código, introduzimos o número de períodos e, após, os juros que usaremos para o (s) cálculo (s). Por último, o programa necessita da introdução do valor que se deseja converter. As mensagens são bem claras, tornando este *software* fácil e auto-explicativo.

→ 1º Exemplo: Converter valor atual (VA) e valor futuro (S).
(OBS.: Valor futuro e montante são sinônimos).

Seja VA = Cr$ 100.000,00, I = 6%, N = 8 períodos

S = ?

Sigamos o roteiro:

COMANDO	*MENSAGEM*	*INPUT*
RUN	—	—
NEWLINE	Tabela de Códigos	—
—	—	1
NEWLINE	A) Número de Períodos?	8
NEWLINE	B) Taxa de Juros (%)?	6
NEWLINE	N = 8 I = 6%	—
—	C) Principal (Cr$)?	100.000
NEWLINE	Montante: Cr$ 159.385 Quer mais alguma conversão? (S/N)	

Quando desta última pergunta, respondemos "S" (Sim) ou "N" (Não) para converter outros valores ou encerrar o programa. "S" nos remete à linha 40, que lista o menu da conversão. Vamos a outros exemplos propostos:

→ Exemplo 2: Dado montante, achar valor atual

S = Cr$ 700.000,00
I = 12% N = 10

RESPOSTA: VA = Cr$ 225.382

→ Exemplo 3: Dado "R" (Série), achar "S" (montante)

R = Cr$ 35.000,00
I = 2,5% N = 48

RESPOSTA: S = Cr$ 3.18004 E + 06

→ Exemplo 4: Dado montante (S), achar série uniforme (R)

S = Cr$ 1.000.000,00
I = 5,6% N = 14

RESPOSTA: R = Cr$ 48.936,20

→ Exemplo 5: Transformar série uniforme em valor atual

R = Cr$ 40.000,00
I = 3,8% N = 10

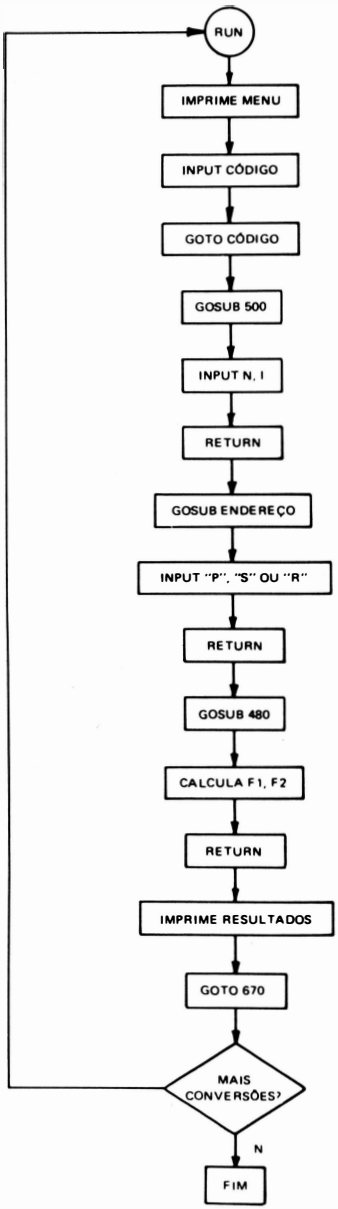
RESPOSTA: VA = Cr$ 3.276.89

→ Exemplo 6: Transformar valor atual em série uniforme

VA = Cr$ 1.500,00
I = 9,05% N = 36

RESPOSTA: R = Cr$ 1.420.28 *

FLUXOGRAMA:



```
MONTANTE EM VALOR ATUAL: TECLE '2'

SERIE UNIFORME EM MONTANTE: TECLE '3'

MONTANTE EM SERIE UNIFORME: TECLE '4'

SERIE UNIFORME EM VALOR ATUAL: TECLE '5'

VALOR ATUAL EM SERIE UNIFORME: TECLE '6'


QUAL CODIGO? 1.

A) NUMERO DE PERIODOS? 8
B) TAXA DE JUROS(%) ? 6


N=  8                              I=  6 %









C) PRINCIPAL (CR$)? 100000.









                          MONTANTE: CR$  159385


QUER MAIS ALGUMA CONVERSAO? (S/N)
? .
```

```
10 'PROGRAMA PARA CONVERSAO DE FLUXOS DE CAIXA
20 'DATA 28/OUT/83     VERSAO 1.0
30 'AUTOR: FAUSTO ARINOS DE ALMEIDA BARBUTO.
40 CLS:PRINT@15,"PARA CONVERTER:"
50 PRINT@64,"VALOR ATUAL EM MONTANTE: TECLE '1'"
60 PRINT@192,"MONTANTE EM VALOR ATUAL: TECLE '2'"
70 PRINT@320,"SERIE UNIFORME EM MONTANTE: TECLE '3'"
80 PRINT@448,"MONTANTE EM SERIE UNIFORME: TECLE '4'"
90 PRINT@576,"SERIE UNIFORME EM VALOR ATUAL: TECLE '5'"
100 PRINT@704,"VALOR ATUAL EM SERIE UNIFORME: TECLE '6'"
120 PRINT:PRINT:INPUT"QUAL CODIGO";C
130 CLS
140 ON C GOTO 150, 200, 250, 300, 350, 400
150 GOSUB 500
160 GOSUB 580
170 GOSUB 480
180 PRINT@470,S$;P*F1
190 GOTO 670
200 GOSUB 500
210 GOSUB 610
220 GOSUB 480
230 PRINT@470,P$;S/F1
```

```
240 GOTO 670
250 GOSUB 500
260 GOSUB 640
270 GOSUB 480
280 PRINT@470,S$;R*F2/I
290 GOTO 670
300 GOSUB 500
310 GOSUB 610
320 GOSUB 480
330 PRINT@470,R$;S*I/F2
340 GOTO 670
350 GOSUB 500
360 GOSUB 640
370 GOSUB 480
380 PRINT@470,P$;R*F2/I/F1
390 GOTO 670
400 GOSUB 500
410 GOSUB 580
420 GOSUB 480
430 PRINT@470,R$;P*I*F1/F2
440 GOTO 670
450 CLS
460 FOR K=1 TO 500
470 NEXT K
480 CLS:F1=(1+I)[N : F2=F1-1
490 RETURN
500 CLS:INPUT"A) NUMERO DE PERIODOS";N
510 INPUT"B) TAXA DE JUROS(%) ";I
520 PRINT@256,"N= ";N TAB(32)"I= ";I;"%"
525 FOR K=1 TO 500: NEXT K
530 I=.01*I
540 R$="SERIE UNIFORME: CR$ "
550 S$="MONTANTE: CR$ "
560 P$="PRINCIPAL: CR$"
570 RETURN
580 CLS:INPUT"C) PRINCIPAL (CR$)";P
600 RETURN
610 CLS:INPUT"C) MONTANTE (CR$)";S
630 RETURN
640 CLS:INPUT"C) SERIE UNIFORME (CR$)";R
660 RETURN
670 PRINT@704,"QUER MAIS ALGUMA CONVERSAO? (S/N)"
680 INPUT X$
690 IF X$="S" THEN GOTO 40 ELSE END
```

CÁLCULO DO TEMPO DE RETORNO DE UM INVESTIMENTO

Um dos fatores empregados para se avaliar uma aplicação de capital é o tempo de retorno. Existem outros, que discutiremos mais tarde, e que são igualmente importantes para uma análise completa.

TEORIA: Diz-se que um investimento é bom quando é pequeno o seu tempo de retorno. Este último, que também é chamado de *"PAYOUT"*, é o tempo necessário para que o investimento feito seja recuperado com as receitas líquidas (atualizadas) da aplicação. Para representar esse tipo de operação, fica válida a equação:

$$-P + \sum_{j=1}^{n} [R_j \cdot FSP(i, N_j)] \qquad (1)$$

onde:

P = Investimento
R_j = Receitas líquidas ao final de cada período

e

$$R_j = R - D$$

onde:

R = Receita bruta do período
D = Despesas do período

OBS.: "R", no presente caso, não significa série uniforme. Reescrevendo a equação (1):

$$-P + \sum_{j=1}^{n} [(R - D)_j \cdot FSP(i, N_j)]$$

Graficamente, poder-se-ia representar:

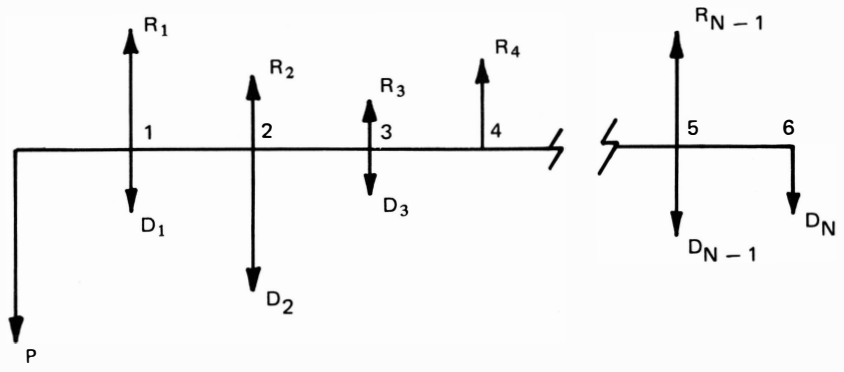


FIG. 1.

Em coordenadas cartesianas, plotando (1) contra N (número de períodos)

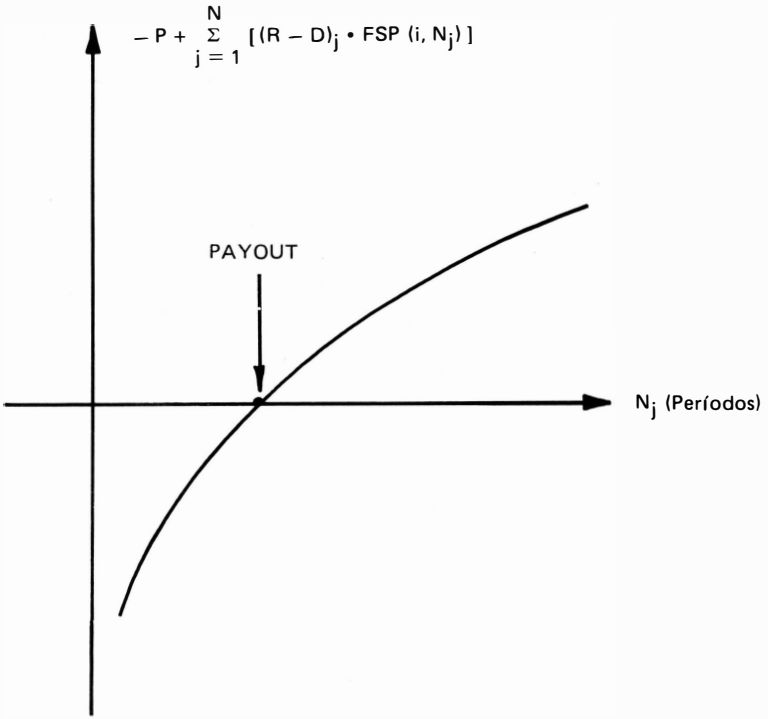


FIG. 2.

O investimento na forma Gráfica.

O tempo de retorno — saliente-se — não é um índice de rentabilidade e sim de liquidez, que reflete apenas o tempo em que o capital aplicado está em risco. Um investimento com *PAYOUT* pequeno pode não ser rentável, e necessitamos de outros subsídios para o analisarmos definitivamente.

O PROGRAMA:

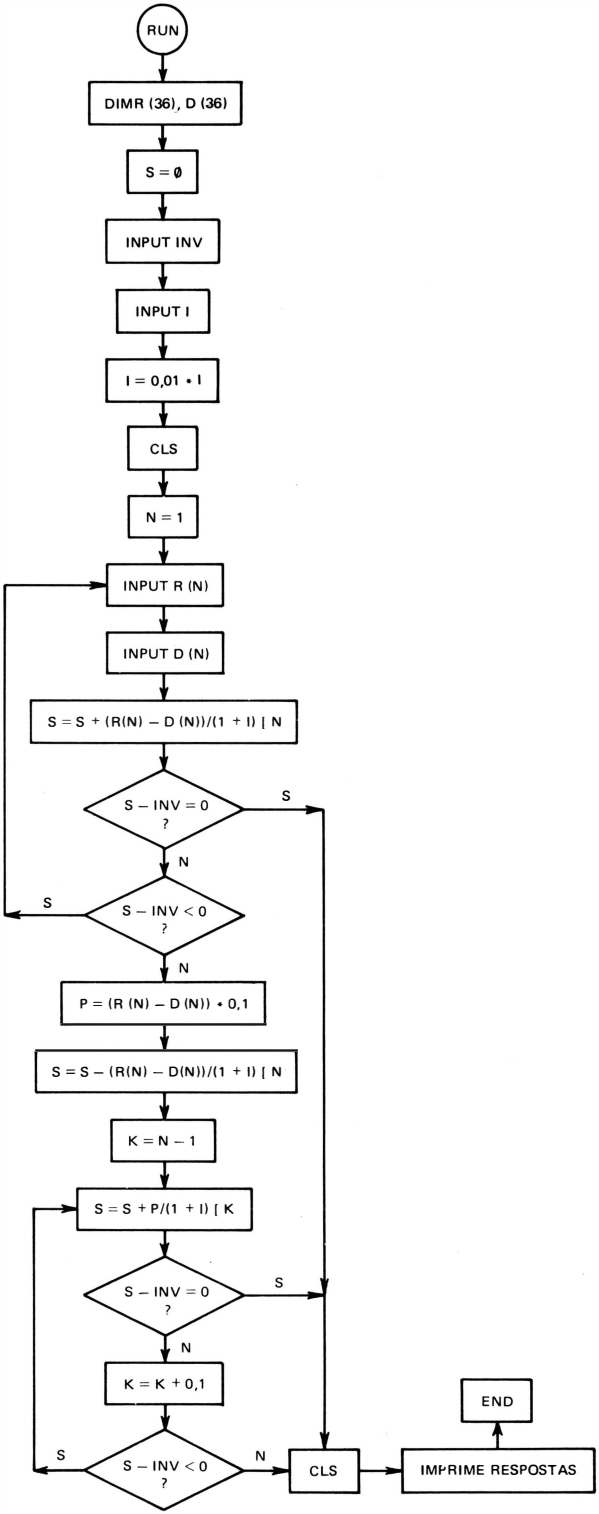
O programa solicita os seguintes dados, nesta ordem:

- O investimento inicial, P.
- A taxa de atratividade mínima, %.
- A receita bruta do período.
- A despesa do período.

Por taxa de atratividade mínima, entenda-se aquela abaixo da qual não é compensador aplicar-se o capital. Se, digamos, a inflação mensal é de 10%, não se deve investir em qualquer tipo de aplicação que não nos remunere em, pelo menos, 10% A. M. Perderá dinheiro quem o fizer.

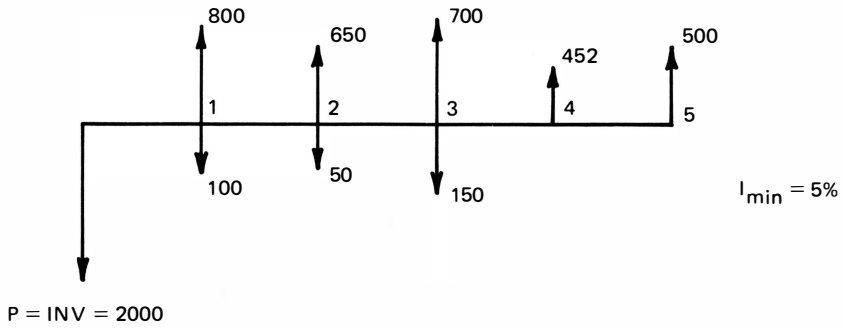
Após introduzirmos o investimento inicial e a taxa de atratividade mínima, vamos necessitar entrar com os valores de R (1), D (1), R (2), D (2) etc., até que em determinado instante a tela se escurece e volte logo após com a resposta, emoldurada com duas linhas contínuas de asteriscos. Este momento ocorre quando a equação (1) se torna idêntica ou maior que zero. Neste ponto, o computador "assume o controle" da situação e executa um laço *FOR-NEXT* que nos fornece a subdivisão do período em que ocorre a anulação do fluxo de caixa. Assim, como exemplo, a primeira parte do programa "descobre" que o *PAYOUT* está entre o décimo e o décimo-primeiro período, suponhamos. A segunda parte faz um "ajuste fino" e retorna com a resposta aproximada até a 1ª casa decimal: 10,7 períodos, por hipótese.

O programa tem o seguinte fluxograma:


RUN
DIMR (36), D (36)
S = Ø
INPUT INV
INPUT I
I = 0,01 • I
CLS
N = 1
INPUT R (N)
INPUT D (N)
S = S + (R(N) – D (N))/(1 + I) [N
S – INV = 0 ?
S
N
S – INV < 0 ?
S
N
P = (R (N) – D (N)) • 0,1
S = S – (R(N) – D(N))/(1 + I) [N
K = N – 1
S = S + P/(1 + I) [K
S – INV = 0 ?
S
N
K = K + 0,1
S – INV < 0 ?
S
N
CLS
IMPRIME RESPOSTAS
END

USANDO O PROGRAMA:

→ Exemplo: Sejam as seguintes receitas e despesas relativas à aplicação do capital de Cr$ 2.000,00.



Assim, (usando uma taxa de atratividade de 5%)

R (1) = 800	D (1) = 100
R (2) = 650	D (2) = 50
R (3) = 700	D (3) = 150
R (4) = 452	D (4) = 0
R (5) = 500	D (5) = 0

Entrando com estes dados, obtemos como resposta:

Tempo de retorno: 3.8 períodos

Tente novamente com I = 2.5% e compare os resultados obtidos.

```
INVESTIMENTO INICIAL? 2000

INV= CR$  2000

TAXA DE JUROS(%)? 5

TAXA=  5 %

*********************************************************************

TEMPO DE RETORNO:  3.8  PERI'ODOS.

*********************************************************************

READY
)
```

```
10 'CA'LCULO DO TEMPO DE RETORNO DE UM INVESTIMENTO.
20 'POR: FAUSTO A. DE A. BARBUTO.
30 'DATA: 23/NOV/83.
40 '
50 DIM D(36), R(36)
60 S=0
70 CLS: INPUT"INVESTIMENTO INICIAL";INV
80 PRINT@128,"INV= CR$ ";INV
90 PRINT: INPUT"TAXA DE JUROS(%)";I
100 PRINT:PRINT"TAXA= ";I;"%"
110 FOR T=1TO500:NEXT T
120 I=.01*I
130 CLS
140 FOR N=1TO36
150 PRINT@384,"RECEITA R(";N;")= "
160 INPUT R(N)
170 PRINT@512,"DESPESA D(";N;")= "
180 INPUT D(N)
190 S=S+(R(N)-D(N))/(1+I)[N
200 IF S-INV=0 THEN 280
210 IF S-INV<0 THEN NEXT N
220 P=(R(N)-D(N))*.1
230 S=S-(R(N)-D(N))/(1+I)[N
240 FOR K=N-1 TO N STEP .1
250 S=S+P/(1+I)[K
260 IF S-INV=0 THEN 280
270 IF S-INV<0 THEN NEXT K
280 CLS
290 FOR X=0TO63
300 PRINT@(X+320),"*";
310 PRINT@(X+576),"*";
320 NEXT X
325 IF S-INV=0 THEN 350
330 PRINT@448,"TEMPO DE RETORNO: ";K;" PERI'ODOS."
335 PRINT@640," "
340 END
350 PRINT@448,"TEMPO DE RETORNO: ";N;" PERI'ODOS."
360 PRINT@640," "
370 END
```

BENEFÍCIO

7

Entre os critérios de avaliação de um investimento, o do benefício figura como um dos mais usados, ao lado dos critérios do tempo de retorno e da taxa interna de juros. Uma definição de benefício: É o valor atual do investimento acrescido do somatório das receitas líquidas geradas pela aplicação deste capital investido, transportadas para o período em que foi feita a aplicação, com base na taxa mínima de atratividade, esta última determinada pelas opções de mercado para aplicações de capital. Em outras palavras, é o "saldo atual" de um investimento, calculado para o momento em que foi aplicado. Em um dado número de períodos, "n", o benefício pode ser assim representado:

$$B_n = - P + \sum_{t=0}^{n} [R \cdot FSP\ (i, t)]$$

ou então:

$$B_n = - P + \sum_{t=0}^{n} [(R_t - D_t) \cdot FSP\ (i, t)]$$

onde:

B_n = benefício em "n" períodos
P = investimento principal
$\overline{R}$ = receita líquida no t-ésimo período
FSP = fator de conversão montante → principal
R_t = receita bruta no t-ésimo período
D_t = encargos do t-ésimo período
i = taxa mínima de atratividade.

O investimento aparece na fórmula acima com sinal trocado (negativo) porque as saídas de capital assim são representadas. As receitas líquidas ou brutas, por outro lado, aparecem como fluxo positivo. Estas últimas, por sinal, são nulas quando t = 0, ocorrendo o mesmo para as despesas líquidas e brutas. Este fato constitui uma regra usualmente geral.

Um investimento que tem um tempo de retorno pequeno nem sempre representa benefícios elevados após o *"PAY-OFF"*; enquanto outros com retorno a mais longo prazo podem exibir receitas líquidas após atingir o tempo necessário para recuperar o capital investido, segundo a Fig. 1:

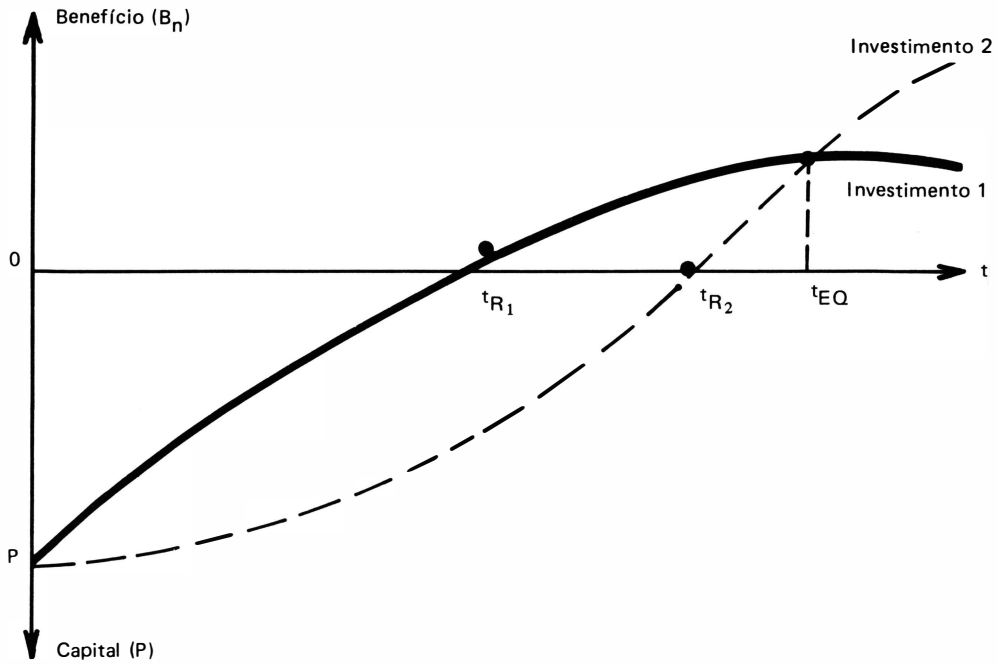


FIG. 1.
Dois investimentos iguais com fluxos de caixa diferentes,
a uma mesma taxa de atratividade.

Em se tratando do investimento 1, representado pela linha contínua, observamos que o seu tempo de retorno é menor que o do investimento 2, linha pontilhada. Entretanto, a partir de t_{EQ}, é mais vantajoso o investimento 2, ou seja, de t_{EQ} em diante $(B_n)_2 > (B_n)_1$; e dependendo da diferença $(t_{R_2} - t_{R_1})$ pode ser melhor optar pelo investimento 2. Claro está que a decisão em torno da escolha deste ou daquele tipo de aplicação deve levar em conta outros fatores que não só o tempo de retorno e o benefício, mas também aqueles inerentes às necessidades e possibilidades de cada investidor. Dentro deste contexto, porém, o método do benefício só vem a somar.

No que tange ao programa em si, há a possibilidade de analisar os benefícios de um fluxo de caixa com receitas líquidas constantes, como também receitas diferentes. Neste último caso, os valores do investimento, taxa de atratividade, número de períodos, períodos inicial e final (para efeitos de listagem em vídeo) são colocados, nesta ordem, em uma declaração "DATA" na linha 280, e as receitas líquidas em outra "DATA" na linha 290, do primeiro ao último perío-

do, ordenadamente. Quando as receitas são constantes, a execução se torna mais simples. É difícil se conseguir isso, mesmo quando o número de períodos a analisar é pequeno.

→ Exemplo 1: Calcular o benefício a cada período sobre um investimento de Cr$ 5.000.000,00 a 6% ao período, com receitas líquidas iguais a Cr$ 900.000,00 durante 10 períodos. Em que período o benefício se torna positivo? Qual o valor deste benefício?

Introduzimos, nesta ordem, o principal, a taxa de atratividade, as receitas líquidas, o número de períodos e os períodos inicial e final. O investimento e as receitas são introduzidos por milhares de cruzeiros. O código de operação é "1"(UM), receitas iguais.

```
?
5000 <ENTER>
?
6 <ENTER>
?
900 <ENTER>
?
10<ENTER>
?
1 <ENTER>
??
10<ENTER>
```

Pela saída dos resultados, notamos que o benefício se torna positivo ao sétimo período, com B = Cr$ 24.143 (no *DISPLAY*, 24.143).

→ Exemplo 2: Idem para um investimento de Cr$ 7.000.000,00 a 3,5% ao período, durante 21 períodos com as seguintes receitas:

```
R (1) = 100;  R (2) = 200;  R (3) = 300;  R (4) = 400;  R (5) = 500
R (6) = 500;  R (7) = 600;  R (8) = 650;  R (9) = 600;  R (10) = 650
R (11) = 700; R (12) = 690; R (13) = 700; R (14) = 715; R (15) = 705
R (16) = 735; R (17) = 680; R (18) = 685; R (19) = 0;   R (20) = 693
R (21) = 711
```

A linha 280 toma a forma:

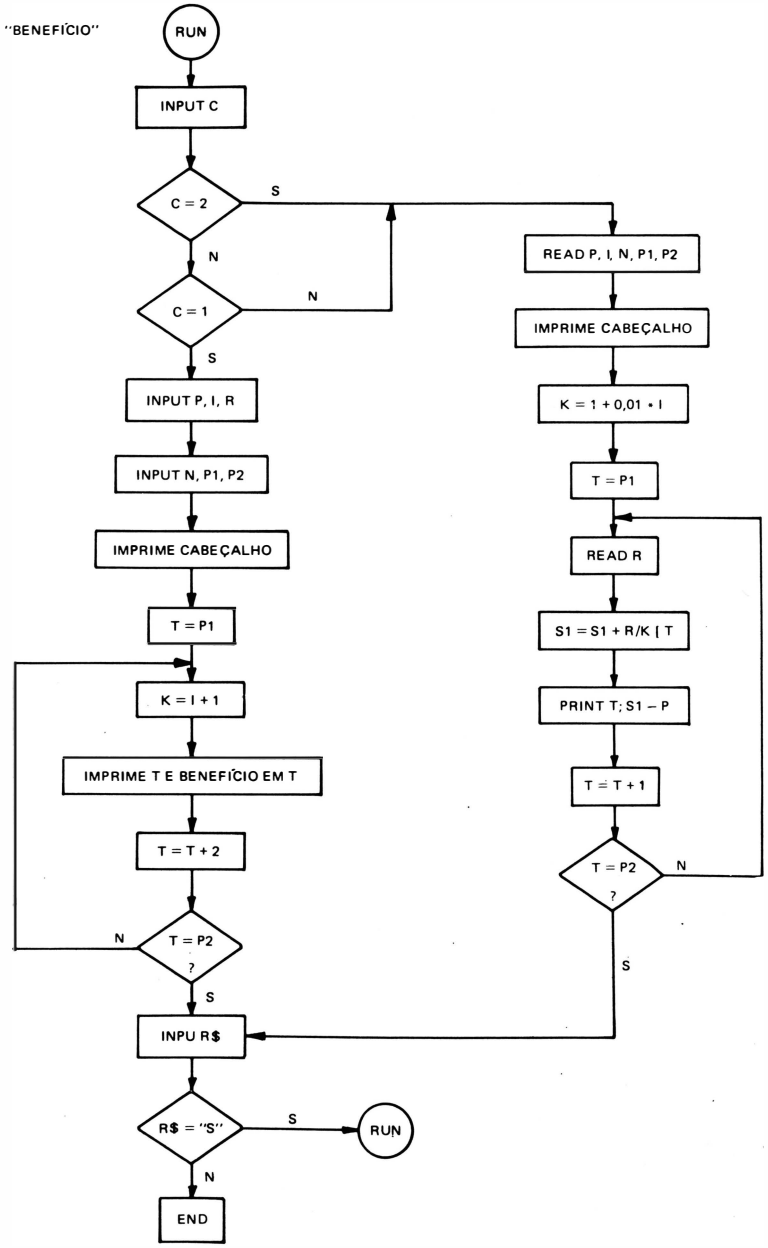
```
280 DATA 7000, 3.5, 21, 1, 21
```

e a linha 290, também uma DATA, com os valores das receitas, de R (1) a R (21). Tendo tudo corrido bem, após *RUN* e <ENTER>, selecionamos o código 2, receitas diferentes. O benefício se torna positivo no 18º período, com B_{18} = Cr$ 14.221,00. Note que $B_{18} = B_{19}$, porque R (19) = 0. O benefício no 20º período é Cr$ 362.499,00. Se o nº de períodos for muito grande, os valores irão "rolando" na tela. Para parar momentaneamente este efeito, teclar

<SHIFT> @ simultaneamente. Para reiniciar a saída dos resultados, tecle qualquer outra tecla, <ENTER> por exemplo.

OBS.: Alguns autores só consideram o benefício como sendo os valores positivos do mesmo; explicando melhor, os valores calculados após o *"PAY-OFF"*.

FLUXOGRAMA:



```
PRINCIPAL (EM CR$1000)? 5000

TAXA DE JUROS(%)? 6

FLUXO LI'QUIDO CONSTANTE (EM CR$1000)? 900

NU'MERO DE PERI'ODOS? 10

PERI'ODO INICIAL E FINAL? 1
?? 10




          VALOR DO BENEFICIO (EM CR$1000).
   PERIODO     BENEFICIO        PERIODO    BENEFICIO
      1        -4150.940            2       -3349.950
      3        -2594.290            4       -1881.410
      5        -1208.870            6        -574.409
      7          +24.143            8        +588.813
      9        +1121.520           10       +1624.080
QUER REINICIAR (S/N)?
```

```
10 'PROGRAMA BENEFI'CIO.
20 'ENTRADAS/SAI'DAS IGUAIS OU DIFERENTES.
30 'POR: FAUSTO A. DE A. BARBUTO.   DATA: 05/JUL/84.
40 CLS:INPUT"FLUXOS IGUAIS=1 ; DIFERENTES=2 "; C
50 ON C GOTO 60, 200
60 'FLUXOS IGUAIS.
70 CLS:INPUT"PRINCIPAL (EM CR$1000)"; P
80 PRINT:INPUT"TAXA DE JUROS(%)"; I
90 I=.01*I
100 PRINT:INPUT"FLUXO LI'QUIDO CONSTANTE (EM CR$1000)"; R
110 PRINT:INPUT"NU'MERO DE PERI'ODOS"; N
120 PRINT:INPUT"PERI'ODO INICIAL E FINAL"; P1, P2
125 GOSUB 310
130 FOR T=P1 TO P2 STEP 2
140 K=I+1
150 PRINT USING"  #####        +######.###         ######       +######.###";T;-P+R*(K
[T-1)/I/K[T;T+1;-P+R*(K[(T+1)-1)/I/K[(T+1)
160 NEXT T
170 INPUT"QUER REINICIAR (S/N)"; R$
180 IF R$="S" THEN RUN ELSE END
190 'FLUXOS DIFERENTES.
200 READ P,I,N,P1,P2
210 GOSUB 340
220 I=.01*I : K=1+I
230 FOR T=P1 TO P2
235 READ R
240 S1=S1+R/K[T
250 PRINT USING"               #####           +######.###";T;S1-P
260 NEXT T
270 GOTO 170
280 DATA 3790.79,10,8,1,8
290 DATA 1000,1000,1000,1000,1000,1000,1000,1000
310 CLS:PRINT@10,"VALOR DO BENEFICIO (EM CR$1000)."
320 PRINT@67,"PERIODO":PRINT@79,"BENEFICIO":PRINT@96,"PERIODO":PRINT@107,"BENEFI
CIO"
330 RETURN
340 CLS:PRINT@10,"VALOR DO BENEFICIO (EM CR$1000)."
350 PRINT@79,"PERIODO":PRINT@95,"BENEFICIO"
360 RETURN
```

DEPRECIAÇÃO

A depreciação é a perda de valor de um equipamento devido ao uso, ação do tempo, obsolescência ou falta de conservação do mesmo. Observamos isso quando, por exemplo, comparamos o preço de um carro "zero km" com um já usado do mesmo modelo. A diminuição do valor é gradativa com o tempo. Quando se esgota o tempo de vida útil de uma máquina qualquer, esta apresenta um valor chamado *residual* pelo qual ainda pode ser vendida. Se este valor é nulo — ou quase — podemos afirmar com certeza que a máquina se transformou em "sucata" e só mesmo um "ferro-velho" se interessaria em comprá-la . . .

Numa economia caracterizada por fortes índices de inflação, pode ocorrer que o valor residual de um equipamento seja maior que o seu preço de compra! Para que se possa fazer qualquer comparação, é preciso, primeiro, corrigir-se o preço de compra deste equipamento para valores atuais, ou tomar-se como referência o preço atual do mesmo equipamento, novo.

Podemos analisar a depreciação sob três pontos de vista: o linear, o da taxa constante e o da capitalização. Veremos no final do capítulo as diferenças entre esses métodos.

USANDO O PROGRAMA:

Os seguintes dados são necessários para se usar o *SOFT:*

VARIÁVEIS

NOME	*TIPO*	*OBSERVAÇÕES*
CD	INTEIRA	CÓDIGO DE ENDEREÇAMENTO
T	INTEIRA	VIDA ÚTIL (PERÍODOS)
VR	REAL	VALOR RESIDUAL, CR$
VA	REAL	VALOR ATUAL, CR$
T1	INTEIRA	PERÍODO INICIAL } P/AS TABELAS
T2	INTEIRA	PERÍODO FINAL } P/AS TABELAS
I	REAL	TAXA DE JUROS (%) — MÉTODO DA CAPITALIZAÇÃO

Com esses dados, outras variáveis são calculadas e usadas no decorrer da execução do programa.

→ Exemplo 1:

Calcular o valor intermediário de um equipamento no 17º período de uso, sabendo:

Valor atual (novo): Cr$ 1.500.000,00

Valor residual: Cr$ 200.000,00

Vida útil (períodos): 24

Juros: 10% ao período.

A) Linear:

RUN <NEWLINE>

Vida útil (períodos)/valor residual (Cr$)?

Fazemos:

24 <NEWLINE>

200.000 <NEWLINE>

Valor atual (Cr$)?

1.500.000 <NEWLINE>

(ou 15E5 <NEWLINE>)

Períodos inicial e final?

1 <NEWLINE>
24 <NEWLINE>

Na linha correspondente ao 17º período, notamos que VR = Cr$ 579.167.

B) Taxa Constante:

Nesse método, além do valor residual obtemos também a desvalorização acumulada, "DA". Procedemos do mesmo modo que no exemplo anterior para executar o subprograma da taxa constante. Respondemos antes, porém, à mensagem:

PARA CONTINUAR APERTE QUALQUER TECLA

apertando uma tecla qualquer. (Só não valem "*BREAK*" e "*SHIFT*".)

Os "*INPUTS*" são idênticos.

Nesse subprograma, durante a execução, a tela "rola" até o período final, T2. O período inicial é sempre 1 (um). Para fazer a tela parar de rolar continuamente, pressione "*BREAK*". Para prosseguir, use "*CONT*" e <NEWLINE>.

Neste caso, obtemos como resposta:

VR = Cr$ 359.962

DA = Cr$ 1,14004 * 10^6

C) Capitalização:

A única diferença em relação aos dados solicitados nos subprogramas anteriores é que neste há um a mais a ser fornecido: A taxa de juros do fundo de depreciação, que normalmente é igual à inflação no período, ou melhor, entre cada período. Se no exemplo atual o equipamento tem seu uso fixado em 24 meses, então o período é o mês, e a taxa de juros passa a ser a inflação mensal. Usaremos 10% para o exemplo que estamos analisando.

Assim, para o 17º período,

VR = Cr$ 904.410

FC = Cr$ 595.590

"FC" é o fundo de capitalização, que é constituído de parcelas fixas e iguais e que, depositadas ao final de cada período, vão, no fim da vida útil do equipamento, compor um fundo que equivale ao montante da depreciação.

TEORIA:

Pode-se analisar a depreciação sob três pontos de vista:

A) O linear
B) O da taxa constante
C) O da capitalização

O primeiro assume que o equipamento deprecia-se linearmente com o tempo, de acordo com o gráfico da Fig. 1:

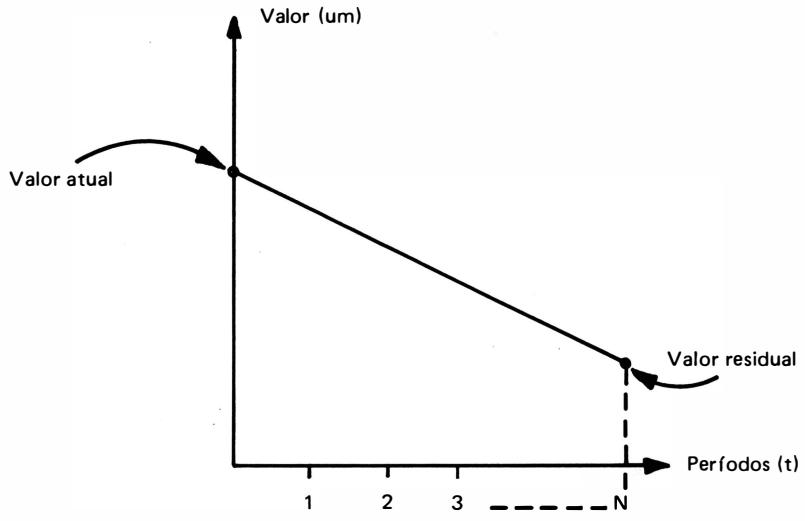


FIG. 1. Depreciação linear.

Assim,

$$\text{Valor (t)} = \frac{(VR - VA)}{N} (t - N) + VR$$

Onde "VR" é o valor residual e "VA" é o valor atual do equipamento (seu preço como novo). "N" é a vida útil em períodos e "t" são os períodos, genericamente falando. A fórmula acima calcula o valor do equipamento a qualquer período "t". Se o valor residual é nulo (VR = = 0), simplifica-se a equação para:

$$\text{Valor (t)} = \frac{VA}{N} (N - t)$$

O segundo método — taxa constante — assume que o equipamento se deprecia segundo a taxa

$$i = 1 - \left(\frac{VR}{VA}\right)^{1/N}$$

onde "N" é o número de períodos ao fim dos quais esgotou-se a vida útil do equipamento. O valor deste, a qualquer momento, é dado por

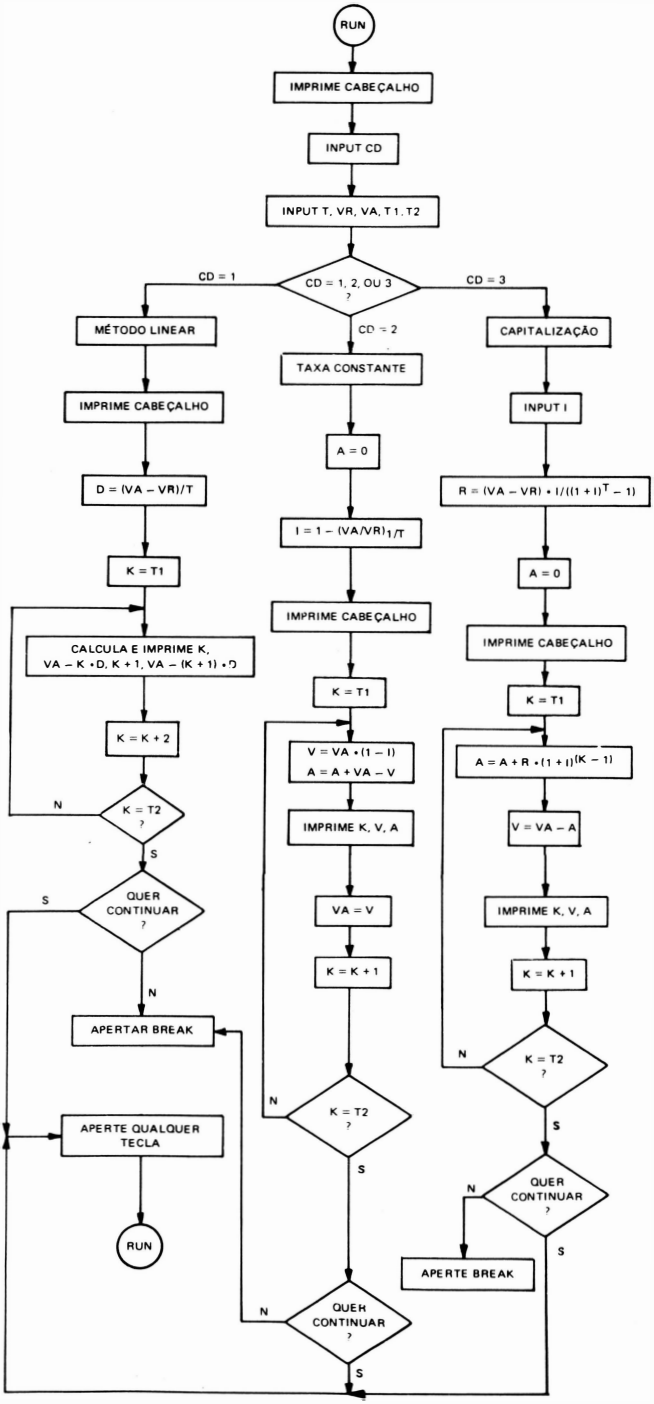
$$V (t) = V (t - 1) (1 - i)$$

Já a capitalização faz com que coloquemos quotas fixas ao final de cada período, de modo que ao se atingir o n-ésimo período, teremos acumulado o montante da depreciação, conforme já havíamos dito.

Assim,

$$R = (VA - VR) \text{ FSR } (i, N)$$

FLUXOGRAMA:



OBS.: O programa só é interrompido por "BREAK".

```
ESCOLHA O ME'TODO QUE VOCE VAI UTILIZAR:
                    *1) LINEAR: CO'DIGO 1.
                    *2) TAXA CONSTANTE: CO'DIGO 2.
                    *3) CAPITALIZACAO: CO'DIGO 3.
CO'DIGO? 2



VIDA U'TIL (PERIODOS)/VALOR RESIDUAL (CR$)? 12
?? 50000

VALOR ATUAL (CR$)? 400000

PERI'ODO INICIAL E FINAL? 1
?? 12



PERIODO          VALOR RESIDUAL      DEPRECIACAO ACUMULADA
 1                  336359                63641.4
 2                  282843                117157
 3                  237841                162159
 4                  200000                200000
 5                  168179                231821
 6                  141421                258579
 7                  118921                281079
 8                  100000                300000
 9                  84089.7               315910
 10                 70710.7               329289
 11                 59460.4               340540
 12                 50000                 350000
PARA CONTINUAR APERTE QUALQUER TECLA
```

```
10 'PROGRAMA DEPRECIACAO.
20 'TRES ME'TODOS SAO EMPREGADOS:
30 '     *1) ME'TODO LINEAR
40 '     *2) ME'TODO DA TAXA CONSTANTE
50 '     *3) ME'TODO DA CAPIALIZACAO
60 'AUTOR: FAUSTO A. DE A. BARBUTO.    DATA: 10/FEV/84
70 CLS:PRINT@128,"ESCOLHA O ME'TODO QUE VOCE VAI UTILIZAR:
                    *1) LINEAR: CO'DIGO 1.
                    *2) TAXA CONSTANTE: CO'DIGO 2.
                    *3) CAPITALIZACAO: CO'DIGO 3."
        80 INPUT"CO'DIGO";CD
        100 'ENTRADA DOS PARAMETROS
        110 CLS:INPUT"VIDA U'TIL (PERIODOS)/VALOR RESIDUAL (CR$)";T,VR
        120 PRINT:INPUT"VALOR ATUAL (CR$)";VA
        130 PRINT:INPUT"PERI'ODO INICIAL E FINAL";T1,T2
        135 ON CD GOTO 145, 200, 320
        140 'ME'TODO LINEAR
        145 CLS:PRINT"PERI'ODO","VALOR RESIDUAL","PERI'ODO","VALOR RESIDUAL"
        150 D=(VA-VR)/T
        155 FOR K=T1 TO T2 STEP 2
        160 PRINTK,VA-K*D,K+1,VA-(K+1)*D
        170 NEXT K
```

```
 180 PRINT"PARA CONTINUAR APERTE QUALQUER TECLA."
 190 IF INKEY$="" THEN 190 ELSE RUN
 200 'ME'TODO DA TAXA CONSTANTE.
 205 A=0
 210 I=1-(VR/VA)[(1/T)
 220 CLS:PRINT"PERIODO","VALOR RESIDUAL","   DEPRECIACAO ACUMULADA"
 230 FOR K=T1 TO T2
 240 V=VA*(1-I)
 250 A=A+VA-V
 260 PRINT K,"  ";V,"           ";A
 270 VA=V
 280 NEXT K
 290 PRINT"PARA CONTINUAR APERTE QUALQUER TECLA"
 300 IF INKEY$="" THEN 300 ELSE RUN
 310 'METODO DA CAPITALIZACAO
320 CLS:PRINT@448,"METODO DA CAPITALIZACAO: ESCOLHA A TAXA DE JUROS (%).
330 INPUT I
340 I=.01*I
350 R=(VA-VR)*I/((1+I)[T-1)
360 A=0
370 CLS:PRINT"PERIODO","VALOR RESIDUAL","    FUNDO DE DEPRECIACAO"
380 FOR K=T1 TO T2
390 A=A+R*(1+I)[(K-1)
400 V=VA-A
410 PRINT K,"   ";V,"           ";A
420 NEXT K
430 PRINT"PARA CONTINUAR APERTE QUALQUER TECLA"
440 IF INKEY$="" THEN 440 ELSE RUN
```

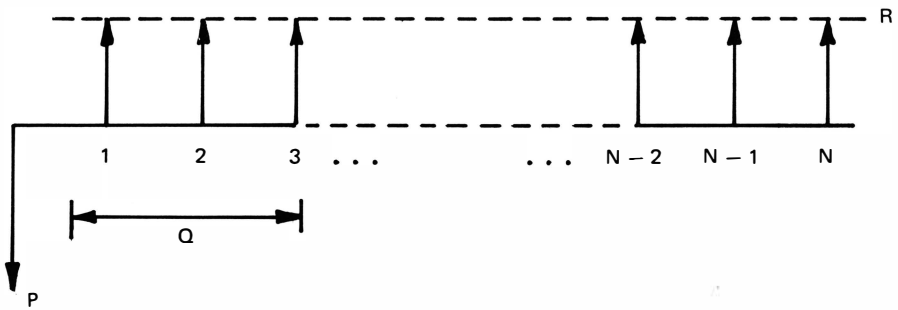
AMORTIZAÇÃO DE UM DÉBITO

A liquidação de uma dívida é composta de pagamentos periódicos – as prestações – sendo que todos esses pagamentos são compostos de uma quota de amortização e uma quota de juros. A quota de juros num determinado período incide sempre sobre o saldo devedor, isto é, o estado da dívida no período anterior. A amortização propriamente dita é a quota que restitui parte da dívida contraída. Assim fica válida a igualdade:

PRESTAÇÃO = AMORTIZAÇÃO + JUROS

Há vários sistemas para a amortização de uma dívida, e vamos agora nos deter sobre dois deles: o sistema francês e o sistema de amortização constante (SAC).

No *sistema francês* – que é considerado o método "clássico" de amortização de débitos no mercado financeiro – o débito P é liquidado através de uma série de "N" prestações "R" iguais, efetuadas no final de cada período, a uma taxa de juros "i", de acordo com a Fig. 1:



onde:

P = débito
R = prestação
Q = nº de prestações pagas num intervalo de tempo

O cálculo de R é efetuado do seguinte modo:

$$R = P \times FPR\ (i, N)$$

A amortização numa prestação "t" qualquer é calculada da seguinte forma:

$$A_t = R \times FSP\ (i, N - t + 1)$$

A amortização acumulada até a t-ésima prestação é assim expressa:

$$\overline{A}_t = \frac{A_1\ (1 + i)^t - 1}{1}$$

como:

$$\frac{(1 + i)^t - 1}{i} = FPS\ (i, t)$$

então:

$$\overline{A}_t = A_1 \cdot FPS\ (i, t)$$

Os juros correspondentes à t-ésima prestação são:

$$J_t = R - A_t$$

ou

$$J_t = R - A_1 \cdot FPS\ (i, t)$$

Os juros acumulados são:

$$\overline{J}_t = t \cdot R - \overline{A}_t = t \cdot R - A_1 \cdot FPS\ (i, t)$$

Os juros pagos correspondentes a "Q" prestações:

$$\overline{J_Q} = R\,[Q + FRP\,(i, N - t) - FRP\,(i, N - t + Q)]$$

e, finalmente, o saldo devedor após o pagamento da t-ésima prestação:

$$D_t = R \cdot FRP\,(i, N - t)$$

O *SAC* — sistema de amortização constante — propõe que a quota de amortização seja igual durante todo o financiamento da dívida. O saldo devedor, portanto, decresce de uma quantia igual à quota de amortização. Conseqüentemente, os juros e as prestações são decrescentes.

A quota de amortização dos períodos (constante) é igual a:

$$A_t = \frac{P}{N}$$

Onde "P" é o débito e "N" o número de períodos.

A amortização acumulada até o t-ésimo período é:

$$\overline{A_t} = t \cdot A_t$$

os juros correspondentes à t-ésima prestação:

$$J_t = i \cdot D_{t-1}$$

ou:

$$J_t = i \cdot \frac{P}{N}\,(N - t + 1)$$

O valor da t-ésima prestação:

$$R_t = A_t + J_t$$

$$R_t = \frac{P}{N}\,[1 + i\,(N - t + 1)]$$

e o saldo devedor após a t-ésima prestação:

$$D_t = D_{t-1} - A_t \text{ ou } D_t = P\left(1 - \frac{t}{N}\right)$$

Os exemplos a seguir permitirão ao leitor absorver através dos números gerados pelas respostas a filosofia dos métodos SAC e francês e se posicionar melhor diante da (possível) aridez das equações acima. À guisa de auxílio, as Figs. 1 e 2 representam graficamente os planos de pagamento descritos.

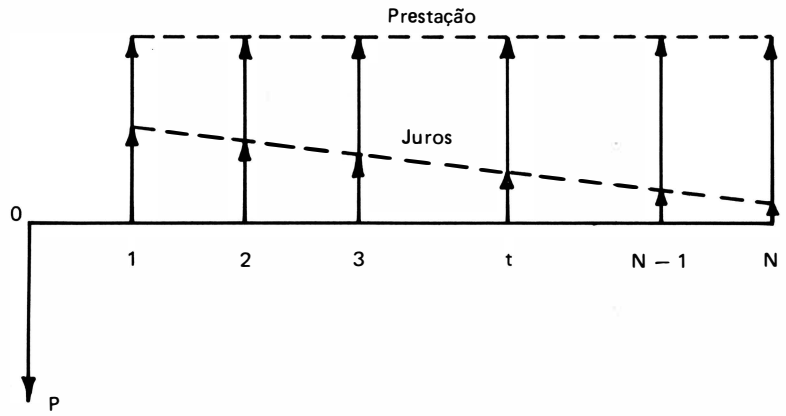


FIG. 1. Sistema Francês.

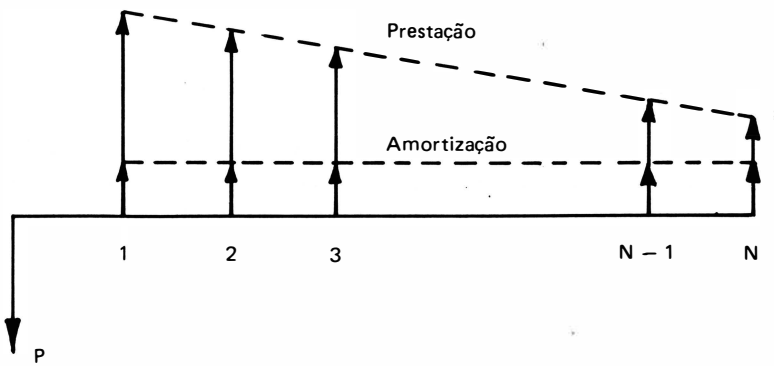


FIG. 2. Sistema de Amortização Constante (SAC).

→ Exemplo 1: Analisar pelo método francês um plano de pagamento em 12 períodos, a 8,5% ao período, de Cr$ 2.500.000,00. Quais as quotas do 12º período? E os acumulados?

Executando:

RUN <ENTER>

O código do sistema francês é 2.

```
?
2 <ENTER>
?
2500 <ENTER>  (dívida em Cr$ 1000)
?
12 <ENTER>    (número de períodos)
?
8,5 <ENTER>   (taxa de juros, %)
?
1 <ENTER>     (período inicial da listagem)
??
12 <ENTER>    (período final da listagem)
```

Desta forma, aparece a tabela do método francês. Como N > 10, a listagem para com a mensagem "BREAK IN 250". Para listar os períodos restantes, executar "CONT". No 12º período, A = Cr$ 313.716, J = Cr$ 26.666, A = Cr$ 2.500.000, J = Cr$ 1.584.580, saldo = = Cr$ 0. (Dívida quitada). A prestação é Cr$ 340.382.

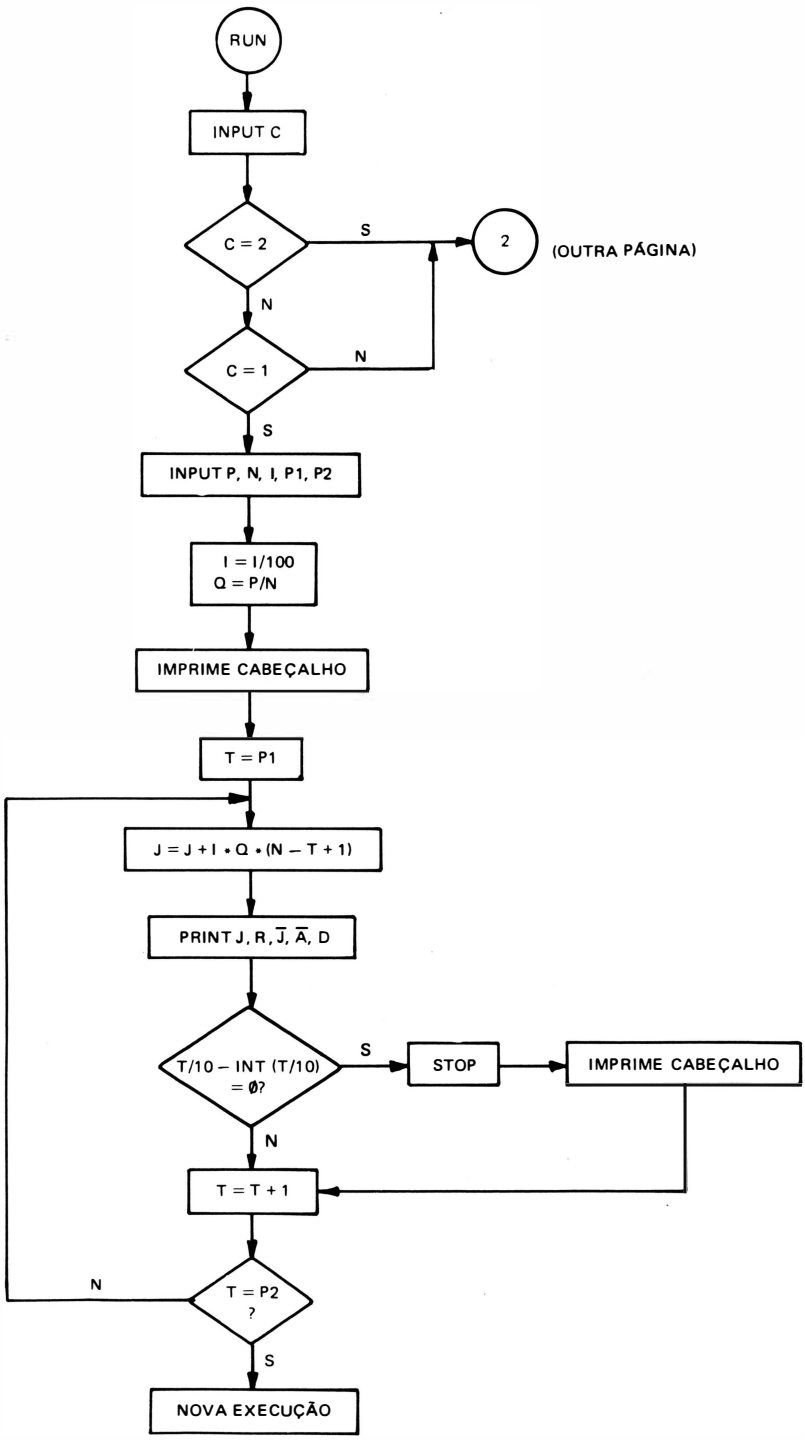
→ Exemplo 2: Idem para o sistema de amortização constante.

Executamos apertando qualquer tecla, após o final das respostas do sistema francês. O código de operação é 1.

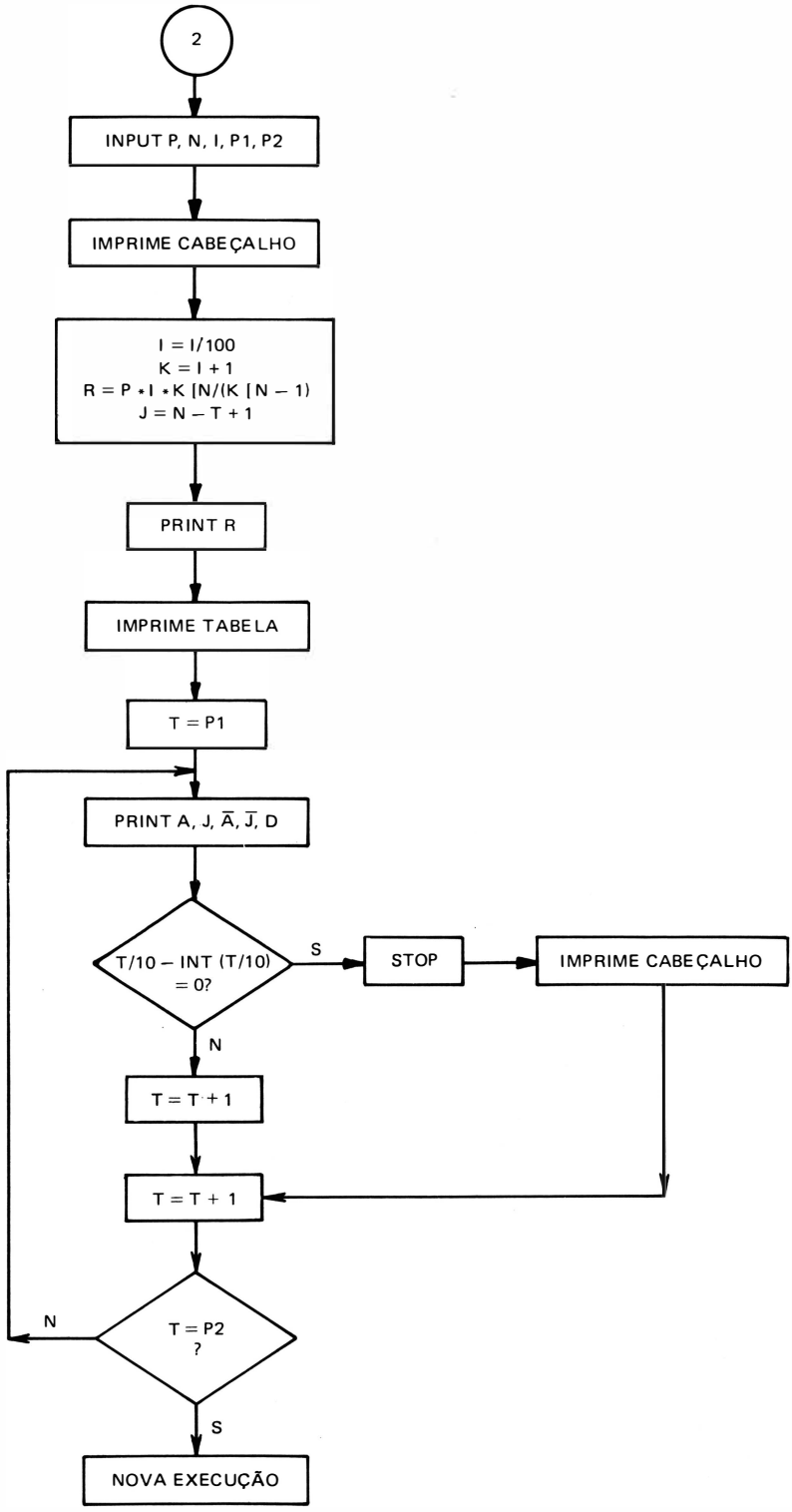
Após a introdução dos dados, obtemos como resposta a tabela de dados.

Para o 12º período, A = Cr$ 208.333; J = Cr$ 17.708; R = Cr$ 226.042; $\overline{J}$ = = Cr$ 1.381.250; $\overline{A}$ = Cr$ 2.500.000; saldo zero. As respostas saem em Cr$ 1.000; e se o número de períodos listados no vídeo é maior do que dez, o programa pára com "BREAK IN 150". Executar "CONT" para reiniciar listagem.

FLUXOGRAMA:


RUN
INPUT C
C = 2
S
2
(OUTRA PÁGINA)
N
C = 1
N
S
INPUT P, N, I, P1, P2
I = I/100
Q = P/N
IMPRIME CABEÇALHO
T = P1
J = J + I * Q * (N – T + 1)
PRINT J, R, J̄, Ā, D
T/10 – INT (T/10) = Ø?
S
STOP
IMPRIME CABEÇALHO
N
T = T + 1
T = P2 ?
N
S
NOVA EXECUÇÃO

AMORTIZAÇÕES: CONT.



```
PRINCIPAL (EM CR$ 1000)? 2500

NUMERO DE PAGAMENTOS? 12

TAXA DE JUROS(%)? 8.5

PERIODO INICIAL E FINAL? 1
?? 12


               SISTEMA FRANCES. (EM CR$ 1000)
                PRESTACAO= 340.382
   N      AMORT.      JUROS     AMORT. AC   JUROS AC.     SALDO
   1     127.882     212.500     127.882     212.500   2372.120
   2     138.752     201.630     266.634     414.130   2233.370
   3     150.546     189.836     417.180     603.966   2082.820
   4     163.342     177.040     580.523     781.005   1919.480
   5     177.226     163.156     757.749     944.160   1742.250
   6     192.291     148.091     950.040    1092.250   1549.960
   7     208.635     131.747    1158.680    1224.000   1341.320
   8     226.369     114.013    1385.050    1338.010   1114.960
   9     245.611      94.771    1630.660    1432.780    869.344
  10     266.488      73.894    1897.140    1506.670    602.856
Break na 250
READY
)
```

```
10 'PROGRAMA AMORTIZACOES;
20 'ME'TODOS "SAC" E FRANCES.
30 'POR: FAUSTO A. DE A. BARBUTO.     DATA: 03/JUL/84.
40 CLS:INPUT"ENTRE COM O CODIGO. (1=SAC , 2=FRANCES)"; C
50 ON C GOTO 60, 180
60 'ME'TODO SAC (SISTEMA DE AMORTIZACAO CONSTANTE).
70 GOSUB 350
80 I=.01*I
90 Q=P/N
110 CLS:PRINT@9,"SISTEMA DE AMORTIZACAO CONSTANTE (EM CR$ 1000)"
115 PRINT@88,"AMORT=";Q
120 GOSUB 400
130 FOR T=P1 TO P2
135 J=J+I*Q*(N-T+1)
140 PRINT USING"####    ####.###    ####.###    ####.###    ####.###    ####.###";T;I
*Q*(N-T+1);Q*(1+I*(N-T+1));J;T*Q;P*(1-T/N)
150 IF (.1*T-INT(.1*T))=0 THEN STOP
155 IF (.1*T-INT(.1*T))=0 THEN GOSUB 400
160 NEXT T
170 GOTO 440
180 'SISTEMA FRANCES.
190 GOSUB 350
200 CLS:PRINT@15,"SISTEMA FRANCES. (EM CR$ 1000)"
210 I=.01*I : K=I+1 : R=P*I*K[N/(K[N-1) : J=N-T+1
215 PRINT@80,"PRESTACAO=";R
220 GOSUB 420
230 FOR T=P1 TO P2
240 PRINT USING"####    ####.###    ####.###    ####.###    ####.###    ####.###";T;R
/K[(N-T+1);R-R/K[(N-T+1);R*(K[T-1)/I/(K[N);T*R-R*(K[T-1)/I/(K[N);R*(K[(N-T)-1)/I
/(K[(N-T))
250 IF(.1*T-INT(.1*T))=0 THEN STOP
260 IF(.1*T-INT(.1*T))=0 THEN GOSUB 420
270 NEXT T
280 GOTO 440
350 CLS: INPUT"PRINCIPAL (EM CR$ 1000)"; P
360 PRINT:INPUT"NUMERO DE PAGAMENTOS"; N
370 PRINT:INPUT"TAXA DE JUROS(%)"; I
380 PRINT:INPUT"PERIODO INICIAL E FINAL"; P1, P2
390 RETURN
400 PRINT"   N      JUROS      PREST.    JUROS AC.   AMORT. AC.   SALDO"
410 RETURN
420 PRINT"   N      AMORT.      JUROS     AMORT. AC   JUROS AC.     SALDO"
430 RETURN
440 PRINT:PRINT"PARA EXECUTAR DE NOVO: APERTE QUALQUER TECLA."
450 IF INKEY$="" THEN 450 ELSE RUN
```

MATEMÁTICA E ESTATÍSTICA

PARTE II

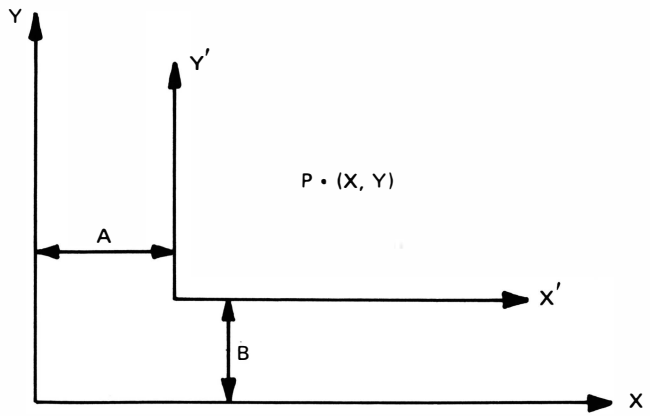
ROTAÇÃO E TRANSLAÇÃO DE EIXOS

10

Examinaremos a partir de agora um programa que reajusta os valores de um "SET" de pontos cartesianos modificando os eixos das abscissas e ordenadas transladando-os e/ou imprimindo-lhes uma notação tendo como centro a origem. As vantagens disso a princípio não são muito claras mas imaginemos por exemplo como seria interessante utilizar *todos* os pontos de um conjunto de dados na execução de um programa do tipo *"CURVE FITTING"*. (Lembre-se que isso nem sempre é possível). Com um programa como este, poderíamos fazê-lo sem problemas.

UM POUCO DE TEORIA:

Por translação define-se um novo sistema de eixos de tal modo que este permaneça em paralelismo com os antigos eixos. Graficamente, representamos:



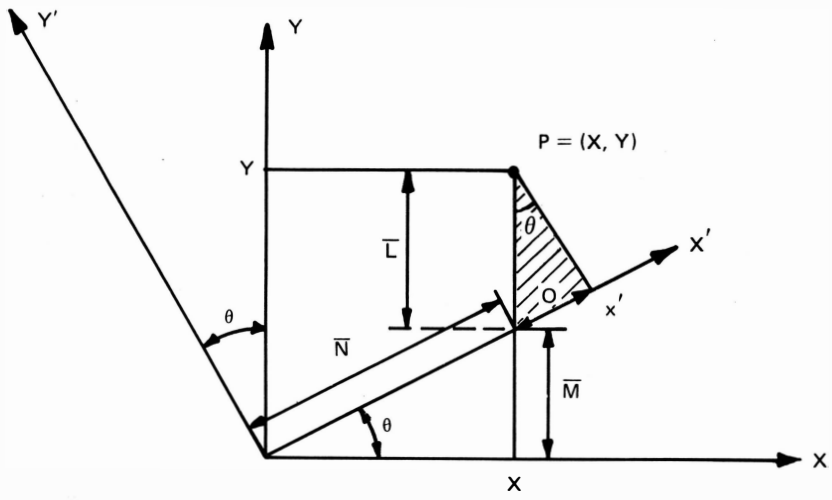
Como vemos, o novo sistema de eixos X'Y' foi deslocado de forma que o eixo X' permaneceu paralelo e com a mesma orientação do eixo X, bem como os eixos Y' e Y. Um ponto (X, Y) genérico no novo sistema fica:

$$P = (X - A, Y - B) = (X', Y')$$

uma vez que:

$$X' = X - A$$
$$Y' = Y - B$$

Vejamos agora uma rotação:



As expressões para os segmentos $\overline{L}$, $\overline{M}$ e $\overline{N}$ são, respectivamente:

$$\overline{N} = \frac{X}{\cos \theta} \qquad \overline{M} = X \operatorname{tg} \theta \qquad \overline{L} = Y - \overline{M} = Y - X \operatorname{tg} \theta$$

O segmento $\overline{O}$, correspondente ao menor cateto do triângulo hachurado é igual a:

$$\overline{O} = \overline{L} \operatorname{sen} \theta = (Y - X \operatorname{tg} \theta) \operatorname{sen} \theta$$

$$\overline{O} = Y \operatorname{sen} \theta - X \frac{\operatorname{sen} \theta}{\cos \theta} \cdot \operatorname{sen} \theta$$

$$\overline{O} = Y \operatorname{sen} \theta - X \frac{\operatorname{sen}^2 \theta}{\cos \theta}$$

e assim, X′,

$$X' = \overline{O} + \overline{N} = \frac{X}{\cos\theta} + Y \operatorname{sen}\theta - X\,\frac{\operatorname{sen}^2\theta}{\cos\theta}$$

$$X' = \frac{X}{\cos\theta}\,(1 - \operatorname{sen}^2\theta) + Y \operatorname{sen}\theta$$

$$X' = \frac{X}{\cos\theta} \cdot \cos^2\theta + Y \operatorname{sen}\theta$$

e, finalmente,

$$X' = X \cos\theta + Y \operatorname{sen}\theta$$

Similarmente para Y′, temos:

$$Y' = Y \cos\theta - X \operatorname{sen}\theta$$

Combinando rotação com translação, temos:

$$Y'' = (Y - B)\cos\theta - (X - A)\operatorname{sen}\theta$$
$$X'' = (X - A)\cos\theta + (Y - B)\operatorname{sen}\theta$$

que são as fórmulas que usamos no programa.

VARIÁVEIS DO PROGRAMA:

NOME	*TIPO*	*OBSERVAÇÃO*
X	REAL, SIMPLES PRECISÃO	ABCISSA "X"
Y	REAL, SIMPLES PRECISÃO	ORDENADA "Y"
ANG	REAL, SIMPLES PRECISÃO	ÂNGULO DE ROTAÇÃO
N	INTEIRA	NÚMERO DE PARES (X, Y)
A	REAL, SIMPLES PRECISÃO	DESLOCAMENTO DO EIXO X′
B	REAL, SIMPLES PRECISÃO	DESLOCAMENTO DO EIXO Y′

USANDO O PROGRAMA:

As variáveis X, Y acima são colocadas numa declaração 'DATA' na linha de número 230. Assim, se $P_1 = (10, 15)$, $P_2 = (20, 25)$ e $P_3 = (30, 35)$ teremos:

230 DATA 10, 15, 20, 25, 30, 35

As variáveis "ANG", "N", "A" e "B" são introduzidas via declaração "*INPUT*".

→ Exemplo 1: Transportar os pontos $P_1 = (5, 9)$ e $P_2 = (13, 25)$ para outro eixo com deslocamentos (translações) iguais a $A = 3$ e $B = 2$ e rotação de 30°.

RESPOSTA: Após tornarmos a linha 230 em:

230 DATA 5, 9, 13, 25

e executamos *RUN,* colocamos as variáveis "A", "B" e "ANG", conforme a solicitação interfacial expressa no vídeo. Acompanhados do "ECO" dos dados introduzidos, temos como resposta:

X' (1) = 5,2305 Y' (1) = 5,06218
X' (2) = 20,1602 Y' (2) = 14,9186

→ Exemplo 2: Com os mesmos pares, executar *apenas* a translação (ANG = 0).

RESPOSTAS:

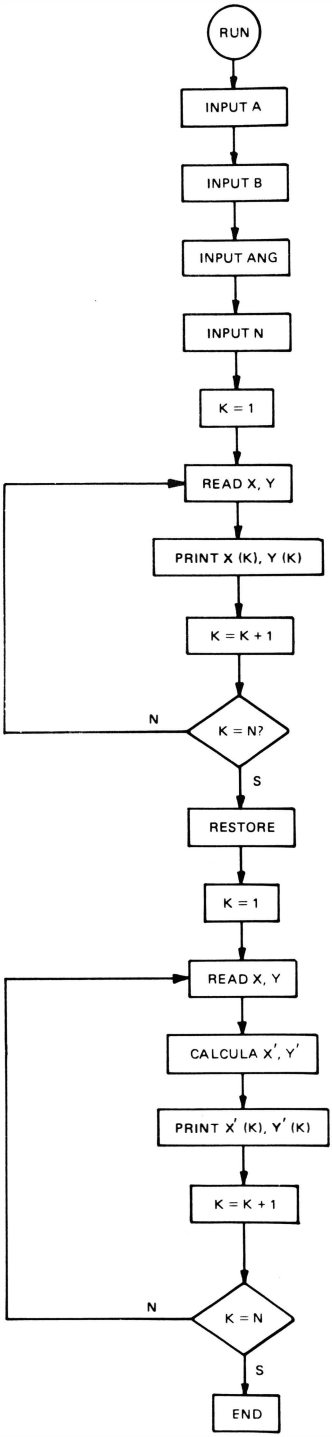
X' (1) = 2 Y' (1) = 7
X' (2) = 10 Y' (2) = 23

→ Exemplo 3: Com os mesmos pares executar *apenas* a rotação (A = 0, B = 0, ANG = 30°).

RESPOSTAS:

X' (1) = 8,83013 Y' (1) = 5,29423
X' (2) = 23,7583 Y' (2) = 15,1506

FLUXOGRAMA:


RUN
INPUT A
INPUT B
INPUT ANG
INPUT N
K = 1
READ X, Y
PRINT X (K), Y (K)
K = K + 1
K = N?
N
S
RESTORE
K = 1
READ X, Y
CALCULA X', Y'
PRINT X' (K), Y' (K)
K = K + 1
K = N
N
S
END

```
DESLOCAMENTO  DO EIXO X? 3

DESLOCAMENTO DO EIXO Y? 2

ANGULO DA ROTACAO (EM GRAUS)? 30

NU'MERO DE PARES DE PONTOS A CONVERTER? 2.



PONTOS A CONVERTER:
X( 1 )= 5    Y( 1 )= 9    X( 2 )= 13    Y( 2 )= 25
PRESSIONE QUALQUER TECLA PARA CONTINUAR



PONTOS CONVERTIDOS:
X'( 1 )= 5.23205    Y'( 1 )= 5.06218    X'( 2 )= 20.1602    Y'(
 2 )= 14.9186
READY
).
```

```
10 ' PROGRAMA ROTACAO/TRANSLACAO DE EIXOS.
20 ' POR: FAUSTO A. DE A. BARBUTO.    DATA: 17/ABR/84.
30 CLS:INPUT"DESLOCAMENTO  DO EIXO X";A
40 PRINT:PRINT:INPUT"DESLOCAMENTO DO EIXO Y";B
50 PRINT:PRINT:INPUT"ANGULO DA ROTACAO (EM GRAUS)";ANG
60 ANG=ANG*3.14159/180
70 PRINT:PRINT:INPUT"NU'MERO DE PARES DE PONTOS A CONVERTER";N
80 CLS:PRINT"PONTOS A CONVERTER:"
90 FOR K=1 TO N
100 READ X,Y
110 PRINT"X(";K;")=";X;"    ";"Y(";K;")=";Y;"    ";
120 NEXT K
130 PRINT:PRINT"PRESSIONE QUALQUER TECLA PARA CONTINUAR"
140 IF INKEY$="" THEN 140 ELSE 150
150 CLS:PRINT"PONTOS CONVERTIDOS:"
160 RESTORE
170 FOR K=1 TO N
180 READ X,Y
190 X1=(X-A)*COS(ANG)+(Y-B)*SIN(ANG)
200 Y1=(Y-B)*COS(ANG)-(X-A)*SIN(ANG)
210 PRINT"X'(";K;")=";X1;"    ";"Y'(";K;")=";Y1;"    ";
220 NEXT K
230 DATA 5,9,13,25
240 END
```

CONVERSÃO POLAR/CARTESIANO/POLAR

11

Seja o par cartesiano (X, Y) representado na Fig. 1.

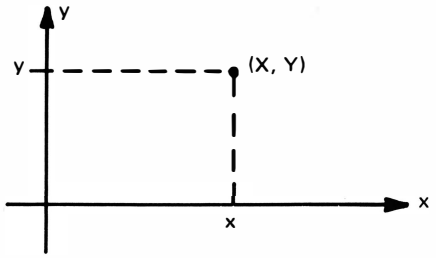


FIG. 1. Ponto (X, Y) no plano cartesiano.

Poderíamos também representar pontos (X, Y) nos quatro quadrantes:

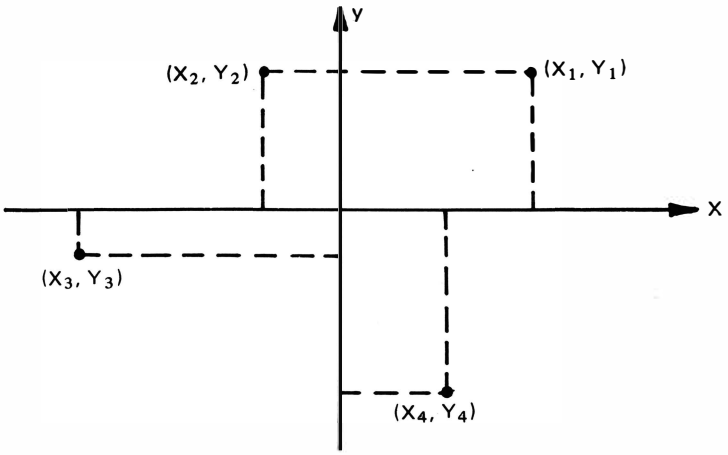


FIG. 2. Os quatro quadrantes.

Não importando o quadrante, cada e qualquer um desses pontos pode ser representado por um vetor que tem sua origem na interseção dos eixos X, Y e cuja extremidade está no ponto em questão. Esse vetor é definido por um módulo — que é seu próprio comprimento — e um ângulo. O ponto cartesiano está, assim, representado na forma *polar.*

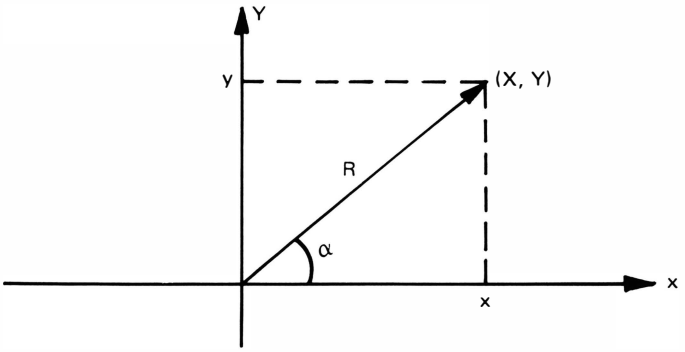


FIG. 3. Ponto (X, Y) e sua forma polar.

"R" é o módulo do vetor, e "α" é seu argumento, isto é, o ângulo que o vetor faz com o eixo dos x. "R" é sempre positivo, não importando o quadrante em que esteja, pois é a distância da origem à extremidade do vetor. Este módulo pode ser calculado geometricamente assim:

$$R = \sqrt{X^2 + Y^2}$$

e "α" trigonometricamente assim:

$$\text{tg}\ \alpha = \frac{Y}{X} \quad \therefore \quad \alpha = \text{arc tg} \left(\frac{Y}{X} \right)$$

Podemos fazer o caminho contrário, também: a partir do vetor, obter o par cartesiano (X, Y).

São essas as duas conversões feitas pelo programa polar/cartesiano/polar. Vamos aos exemplos de resolução:

→ Exemplo 1: Converter (2, 2) para a forma polar.

Comando: <RUN>
Comando: <NEWLINE>
Mensagem: Programa polar/cartesiano
Para converter polar para cartesiano: tecla '1'.
Para converter cartesiano para polar: tecla '2'.

Introduzir: 2
Comando: <NEWLINE>
Mensagem: Abscissa?
Introduzir: 2
Comando: <NEWLINE>
Mensagem: Ordenada?
Introduzir: 2
Comando: <NEWLINE>
Mensagem: Abscissa: 2 Ordenada: 2
Módulo do vetor: 2.82843 Ângulo: 45 graus
Mais alguma conversão (S/N)?

Conforme vimos, R = 2,82843 e o ângulo alfa é 45° (1º quadrante).

→ Exemplo 2: Converter (6, 8) para polar.

Respondendo "S" (SIM) à última pergunta do exemplo 1, e então:

Escolha de cálculo: 2
Abscissa: 6
Ordenada: 8

RESPOSTA:

Módulo do vetor: 10 Ângulo: 53,1301 graus

→ Exemplo 3: Converter 5 $\angle$ 126,8699° para cartesiano.

Escolha de cálculo: 1
Módulo do vetor: 5
Ângulo (em graus): 126.8699

RESPOSTA:

Abscissa: – 3 Ordenada: 4 (2º Quadrante)

→ Exemplo 4: Converter 100 $\angle$ 45°

Escolha de cálculo: 1
Módulo do vetor: 100
Ângulo (em graus): 45°

RESPOSTA:

Abscissa: 70.7107 Ordenada: 70.7107

```
P R O G R A M A   P O L A R / C A R T E S I A N O

PARA CONVERTER POLAR PARA CARTESIANO: TECLE '1'.

PARA CONVERTER CARTESIANO PARA POLAR: TECLE '2'.
? 2



ABCISSA:  2            ORDENADA:  2

MODULO DO VETOR:  2.82843          ANGULO:  270  GRAUS
MAIS ALGUMA CONVERSAO (S/N)?
```

```
 10 'PROGRAMA CONVERSAO POLAR/CARTESIANO/POLAR
 20 'POR: FAUSTO A. A. BARBUTO.
 30 'DATA: 19/JAN/84
 40 CLS:PRINT@320,"P R O G R A M A   P O L A R / C A R T E S I A N O"
 50 PRINT:PRINT"PARA CONVERTER POLAR PARA CARTESIANO: TECLE '1'."
 60 PRINT:PRINT"PARA CONVERTER CARTESIANO PARA POLAR: TECLE '2'."
 70 INPUT T
 80 ON T GOTO 90, 200
 90 CLS
100 'CONVERSAO POLAR/CARTESIANO
110 INPUT"MODULO DO VETOR";R
120 INPUT"ANGULO (EM GRAUS)";ALF
130 ALF1=ALF*3.141592/180
140 X=R*COS(ALF1)
150 Y=R*SIN(ALF1)
160 CLS:PRINT@384,"VETOR: ";R;"          ";"ANGULO: ";ALF1*180/3.141592;" GRAUS"
170 PRINT@512,"ABCISSA X: ";X;"        ";"ORDENADA Y: ";Y
180 INPUT"MAIS ALGUMA CONVERSAO ? (S/N)"; R$
190 IF R$="S" THEN 40 ELSE END
200 'CONVERSAO CARTESIANO/POLAR
210 CLS
220 INPUT"ABCISSA";X
230 INPUT"ORDENADA";Y
235 IF X=0 AND Y=0 THEN 380
240 R=SQR(X[2+Y[2)
245 IF X=0 AND Y)0 THEN 340 ELSE 360
250 ALFA=ATN(Y/X)
260 ALFA=ALFA*180/3.141592
270 IF X(0 AND (Y)0 OR Y(0) THEN ALFA=ALFA+180
290 IF X)0 AND Y(0 THEN ALFA=ALFA+360
300 CLS:PRINT@384,"ABCISSA: ";X;"          ";"ORDENADA: ";Y
310 PRINT@512,"MODULO DO VETOR: ";R;"        ";"ANGULO: ";ALFA;" GRAUS"
320 INPUT"MAIS ALGUMA CONVERSAO (S/N)"; R$
330 IF R$="S" THEN 40 ELSE END
340 ALFA=90
350 GOTO 300
360 ALFA=270
370 GOTO 300
380 PRINT@512,"VETOR DE MO'DULO NULO! (X=0,Y=0)"
390 GOTO 320
```

CONVERSÃO BINÁRIO/DECIMAL

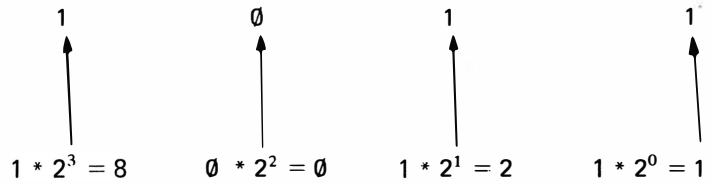
12

Estamos acostumados, desde os primeiros anos de escola, a raciocinar em termos de contagem decimal. Parece-nos difícil raciocinar com soma, multiplicação etc., em outros sistemas, emboras eles existam.

Com o microcomputador (assim como todos os outros), o que é difícil fazer é justamente o contrário — compreender outros números que não sejam representados no sistema binário, onde os únicos algarismos são o zero (0) e o um (1). Mesmo os programas em *BASIC* que nele implementamos são posteriormente decodificados como números binários para que o computador os entenda e execute. Poderíamos fornecer ao micro um programa com uma seqüência de "zeros" e "uns" — é o que comumente se chama linguagem *ASSEMBLY*, ou de máquina.

Como converter um binário em decimal? Tomemos como exemplo o número 1Ø11 (mil e onze, em decimal).

O primeiro dígito da direita é a potência zero de dois, ou seja, $2^0 = 1$, o segundo é a potência um de dois, $2^1 = 2$, e assim sucessivamente. O n-ésimo dígito corresponde à potência N — 1 de dois, 2^{N-1}. Destarte:



Somando 8 + Ø + 2 + 1 obtemos onze (11).

A maioria dos leitores já não tem essa transformação como novidade. O diagrama de blocos do programa é:

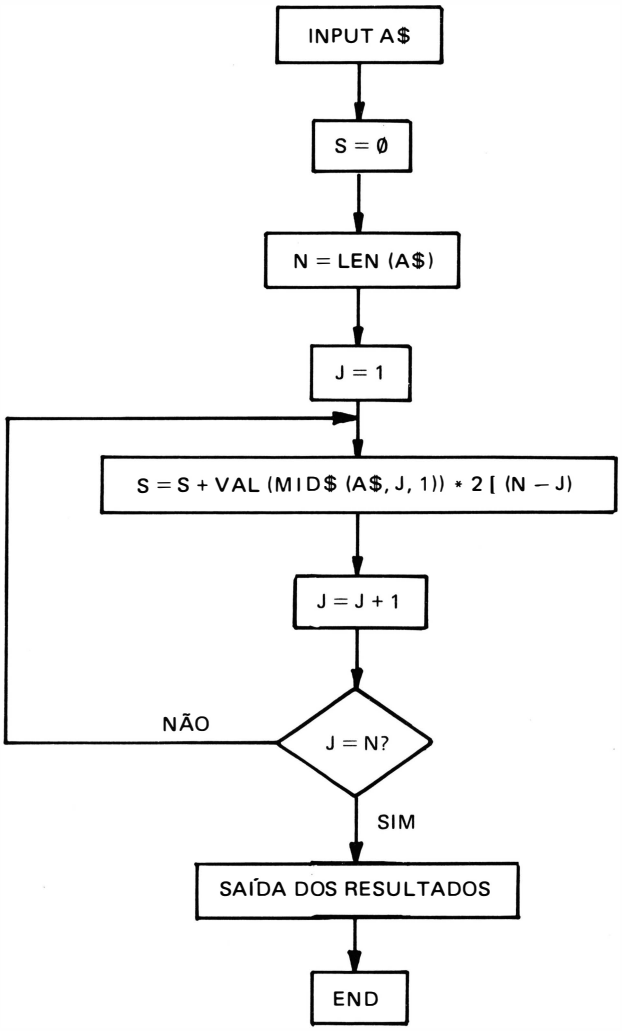


FIG. 1. Fluxograma do programa binário/decimal.

VARIÁVEIS DO PROGRAMA:

VARIÁVEL	*TIPO*	*COMENTÁRIOS*
A$	ALFANUMÉRICA	STRING BINÁRIA A CONVERTER
S	SIMPLES PRECISÃO	SOMATÓRIO DAS POTÊNCIAS DE 2
N	SIMPLES PRECISÃO	COMPRIMENTO DA STRING A$
J	SIMPLES PRECISÃO	CONTROLADORA DO LOOP
R$	ALFANUMÉRICA	CONTROLE PARAR/CONTINUAR

Só é necessário o *INPUT* de um dado: a *STRING* (número binário) A$, que é o número na base 2 que será convertido para a base 10.

→ Exemplo 1: Converter 101010111 para base decimal.

Comando: RUN, NEWLINE
Mensagem: Número binário a converter?
Introduzir: 101010111
Comando: NEWLINE

A resposta, no vídeo:

Número Binário: 101010111
----------→ Número convertido para base 10: 343
Mais alguns números a converter (S/N)?
S <NEWLINE>

→ Exemplo 2: Converter 11101 para decimal.

RESPOSTA: $(11101)_2 = (29)_{10}$

→ Exemplo 3: Converter 11110101 para decimal.

RESPOSTA: $(11110101)_2 = (245)_{10}$.

```
NUMERO BINARIO: 11111111

----------) NU'MERO CONVERTIDO PARA BASE 10:  255
MAIS ALGUM NU'MERO A CONVERTER (S/N)?
```

```
10 'PROGRAMA CONVERSAO BINARIO/DECIMAL.
20 'POR: FAUSTO A. DE A. BARBUTO        DATA: 23/JAN/1984
30 CLEAR500
60 CLS: INPUT"NUMERO BINARIO A CONVERTER";A$
70 S=0
80 N=LEN(A$)
90 FOR J=1 TO N
100 S=S+VAL(MID$(A$,J,1))*2[(N-J)
110 NEXT J
120 CLS: PRINT@448,"NUMERO BINARIO: ";A$
130 PRINT: PRINT "----------) NU'MERO CONVERTIDO PARA BASE 10: ";S
140 INPUT"MAIS ALGUM NU'MERO A CONVERTER (S/N)";R$
150 IF R$="S" THEN 60 ELSE END
```

PROGRAMA PARA CONVERSÃO DE BASE 16 PARA BASE 10

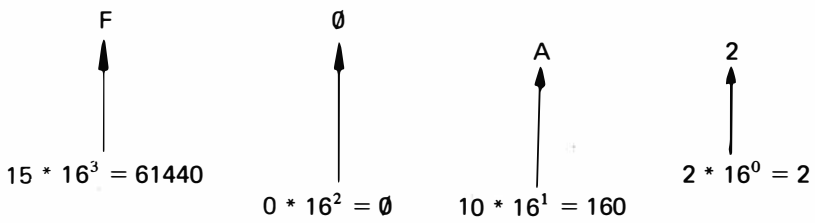
13

Já havíamos anteriormente falado sobre a importância dos números binários quanto ao seu uso em computadores e em linguagem *ASSEMBLY*. Não obstante, é comum também usar-se números hexadecimais para representar números e instruções e assim introduzi-los num programa em linguagem de máquina. A utilidade desta prática é fácil de explicar: como 16 é igual a 2^4, os números representados em hexa têm seu tamanho reduzido em quatro vezes dificultando erros na introdução de um número ou comando em linguagem de máquina e facilitando a tarefa de digitação. (Mais tarde veremos um programa *ASSEMBLY*, que permite introduzir um programa em linguagem de máquina em código hexadecimal). Vamos representar um número (176) em base 10, base 2 e base 16

$$(176)_{10} = (10110000)_2 = (B0)_{16}$$

Fica claro que é muito mais fácil introduzir "B0" através de um programa *ASSEMBLY* do que "10110000". Além disso, as chances de errar são menores.

Outro fato digno de nota é que como só temos dez algarismos, temos que lançar mão de letras para representar os seis algarismos faltantes. Assim A = 10, B = 11, C = 12 etc., até F, que é igual a 15. Representamos os números hexadecimais como uma seqüência alfanumérica, em que o primeiro dígito à direita é a potência zero de 16, e o último — na extrema esquerda — é a potência (N — 1), onde "N" é o comprimento da cadeia. Façamos um exemplo, o número hexa:

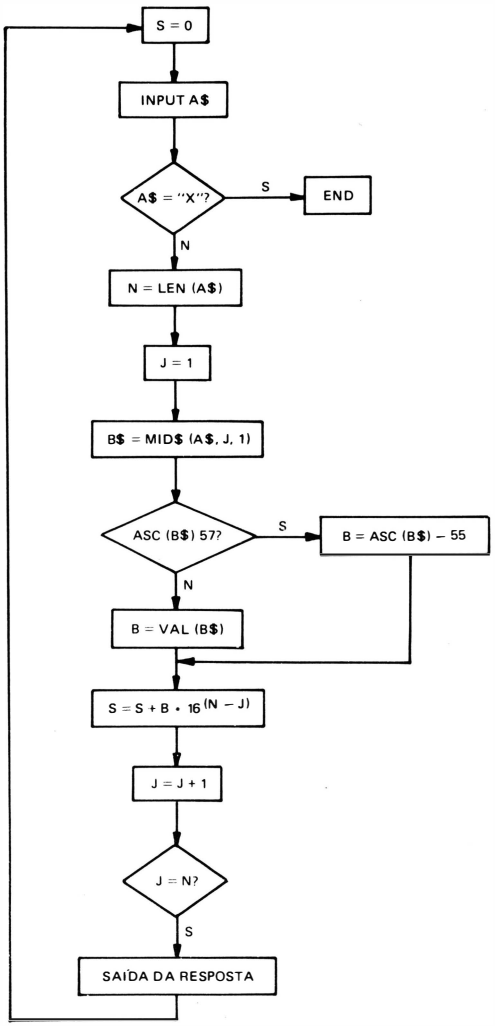


Pode ser decomposto em $2 * 16^0 + 10 * 16^1 + 0 * 16^2 + 15 * 16^3 = 2 + 160 + 0 + 61440 =$ $= 61602$.

Quando desenvolvemos um programa em *ASSEMBLY* para um certo tipo de microprocessador (no caso do TRS-80, o Z-80), normalmente esquematizamos sua estrutura no papel, com cuidado, antes de introduzi-lo na memória do computador. (A linguagem *ASSEMBLY,* neste ponto, é mais exigente que o *BASIC:* qualquer erro pode por tudo a perder). Quando o decodificamos, isto é, fazemos o caminho inverso transformando os códigos de máquina em decimal e, depois, relacionando-os com os mnemônicos.

Espero que esta explanação tenha sido válida o suficiente para que o leitor perceba a importância e a utilidade de se usar hexadecimais em programação *ASSEMBLY.*

A estrutura do programa é simples. Na linha Ø7Ø foi utilizado um artifício para associar a um caracter alfabético um valor numérico. (Saberia o leitor explicar o que foi feito?). Esse artifício é calcado no fato de que os caracteres 65 a 70 são as letras "A" a "F". Vejamos o fluxograma do programa:



VARIÁVEIS DO PROGRAMA:

NOME	*TIPO/FUNÇÃO*
S	SIMPLES; SOMATÓRIO DAS POTÊNCIAS DE 16
A$	ALFANUMÉRICA; NÚMERO HEXA A SER TRANSFORMADO
N	COMPRIMENTO DA CADEIA A$ (NÚMERO DE CARACTERES)
J	CONTROLE DO LOOP; CONTROLE DE B$; INTEIRA
B$	N-ÉSIMO CARACTER A CONTAR DA ESQUERDA; ALFAN.
B	SIMPLES; VALOR DO CARACTER.

OBS.: Se A$ = "X", o programa é encerrado (linha 2Ø).

Passemos agora aos exemplos:

→ Exemplo 1: Converter "B" para decimal.

RUN e *NEWLINE* iniciam o programa.

Aparece no alto do vídeo:

A$?

Apertamos a tecla "B" e *NEWLINE* e obtemos:

NÚMERO HEXA A CONVERTER: B

Número decimal correspondente: 11.

Para continuar aperte qualquer tecla.

→ Exemplo 2: Converter $(AC5)_{16}$ para base decimal.

RESPOSTA: $(AC5)_{16} = (2757)_{10}$.

→ Exemplo 3: Converter $(CDØ2AF)_{16}$ para base dez:

RESPOSTA: $(CDØ2AF)_{16} = (1{,}34356 \times 10^7)_{10}$.

```
A$ ? AC5


NU'MERO HEXA A CONVERTER: AC5

NU'MERO DECIMAL CORRESPONDENTE:  2757


PARA CONTINUAR APERTE QUALQUER TECLA
```

```
1 'PROGRAMA CONVERSOR DE BASE 16 PARA BASE 10
5 'POR: FAUSTO ARINOS DE ALMEIDA BARBUTO.     DATA: 02/JAN/84
10 S=0: CLS
15 INPUT"A$ ";A$
20 IF A$="X" THEN END
25 N=LEN(A$)
30 FOR J=1 TO N
50 B$=MID$(A$,J,1)
70 IF ASC(B$))57 THEN B=ASC(B$)-55 ELSE B=VAL(B$)
85 S=S+B*16[(N-J)
100 NEXT J
110 PRINT@384,"NU'MERO HEXA A CONVERTER: ";A$
120 PRINT@512,"NU'MERO DECIMAL CORRESPONDENTE: ";S
130 PRINT@768,"PARA CONTINUAR APERTE QUALQUER TECLA"
140 IF INKEY$="" THEN 140 ELSE 10
```

PROGRAMA CONVERSOR BASE 10/BASE N
(N até 20)

14

Que tal seria se dispuséssemos de um programa genérico de conversão que, para um certo número na base decimal e uma base N qualquer – N escolhido pelo usuário conforme este bem quisesse – nos fornecesse o valor deste número na base N? Pois bem, esse é o assunto a partir de agora.

As vantagens de um programa genérico são óbvias; pense no trabalho que daria para converter um número qualquer para três bases diferentes como binária, octal e hexadecimal; teríamos de descarregar e executar três programas diferentes de conversão na memória do computador, um de cada vez; ou então construir um só programa subdividido em rotinas de conversão binário – octal – hexadecimal. Ficaríamos com programa "daquele tamanho" como se diz comumente. Se a este programa acrescentássemos outras rotinas de conversão, aí então seríamos os felizes usuários de um programa tipo "elefante branco", grande, ocupando muita memória, de baixa eficiência. Agora imagine o contrário – que fosse possível, com um programa pequeno, obter qualquer conversão de um número decimal para qualquer base até 20: seria excelente, principalmente para aqueles que começaram a estudar outras bases numéricas – um estudante do 2º grau, um neófito em informática, um estagiário de sistemas etc. Se um desses é o caso do leitor, mãos à obra!

USANDO O PROGRAMA:

O programa requer o *INPUT* de apenas três dados: a nova base, NB, o primeiro número a converter, P, e o último, U, nessa ordem.

→ Exemplo 1: Converter o intervalo entre os números $(134)_{10}$ e $(138)_{10}$ para base 8 (oito).

Após "*RUN*", a pergunta-*interface* que se mostra no vídeo é:

QUAL A NOVA BASE?

Fazemos:

8 <NEWLINE>

DISPLAY

O primeiro e o último número a converter?

134 <NEWLINE>
138 <NEWLINE>

e obteremos como resposta:

134	2Ø6
135	2Ø7
136	21Ø
137	211
138	212

QUER MAIS CONVERSÕES?

→ Exemplo 2: Converter os números inteiros de $(35000)_{10}$ a $(35005)_{10}$ para a base 16.

De maneira similar, faremos a introdução dos números 16, 35000 e 35005 após respondermos com "S" (SIM) à pergunta "QUER MAIS CONVERSÕES?". Como resposta, temos:

35ØØØ	88B8
35ØØ1	88B9
35ØØ2	88BA
35ØØ3	88BB
35ØØ4	88BC
35ØØ5	88BD

QUER MAIS CONVERSÕES?

→ Exemplo 3: Transformar de $(2500)_{10}$ a $(2504)_{10}$ para a base 3.

RESPOSTA:

25ØØ	1Ø1Ø2121
25Ø1	1Ø1Ø2122
25Ø2	1Ø1Ø22ØØ
25Ø3	1Ø1Ø22Ø1
25Ø4	1Ø1Ø22Ø2

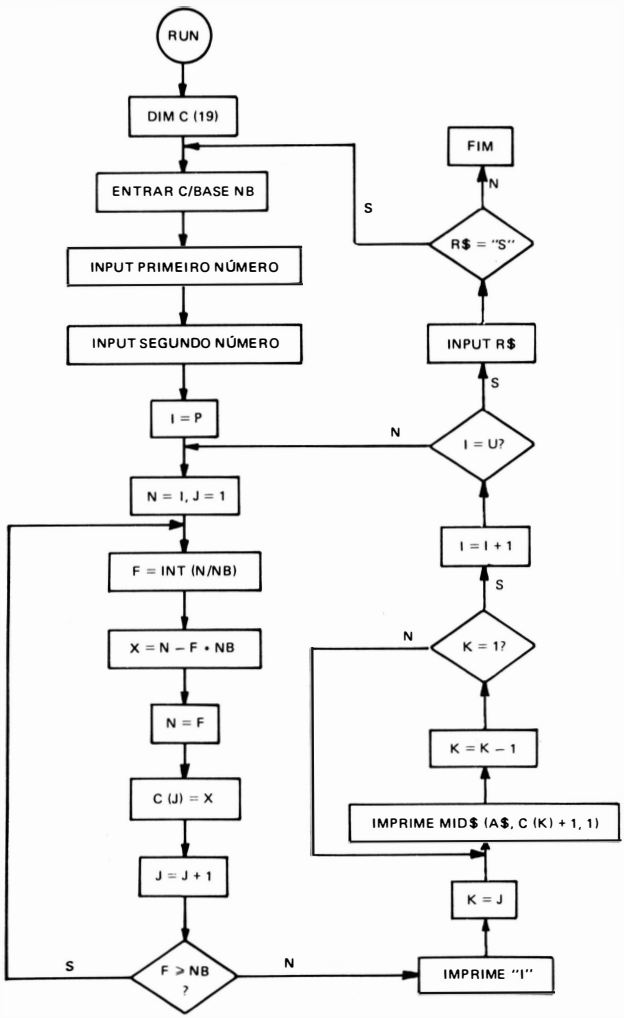
→ Exemplo 4: Transformar de $(7800)_{10}$ a $(7803)_{10}$ para a base 17.

RESPOSTA:

78Ø0	19GE
78Ø1	19GF
78Ø2	19GG
78Ø3	1AØØ

Para encerrar, responder "N" (NÃO).

FLUXOGRAMA:



```
QUAL A NOVA BASE? 18
O PRIMEIRO E O U'LTIMO NU'MERO A CONVERTER? 100
?? 110

 100     5A
 101     5B
 102     5C
 103     5D
 104     5E
 105     5F
 106     5G
 107     5H
 108     60
 109     61
 110     62
QUER MAIS CONVERSOES? .
```

```
10 'PROGRAMA PARA CONVERSAO DE BASE 10 PARA BASE N (N ATE' 20).
20 'POR: FAUSTO A. DE A. BARBUTO.   DATA: 25/JAN/84
30 DIM C(19)
40 A$="0123456789ABCDEFGHIJ"
60 CLS: INPUT"QUAL A NOVA BASE";NB
70 INPUT"O PRIMEIRO E O U'LTIMO NU'MERO A CONVERTER";P,U
80 FOR I=P TO U
90 PRINT: GOSUB 160
100 PRINT I;TAB(8);
110 FOR K=J TO 1 STEP -1
120 PRINT MID$(A$,C(K)+1,1);
125 NEXT K
130 NEXT I
140 PRINT: INPUT"QUER MAIS CONVERSOES";R$
150 IF R$="S" THEN 60 ELSE END
160 N=I: J=1
170 F=INT(N/NB)
180 X=N-F*NB
190 N=F
200 C(J)=X
210 J=J+1
220 IF F)=NB THEN 170
230 C(J)=F
240 RETURN
```

MÉDIA, VARIÂNCIA E DESVIO-PADRÃO

15

A média, variância e desvio-padrão são medidas de dispersão, isto é, aferem o grau de homogeneidade dos valores de uma amostra de dados em torno de um valor médio.

A média é definida por:

$$M = \sum_{i=1}^{N} \frac{A_i}{N} = \frac{1}{N} \sum_{i=1}^{N} A_i$$

A variância:

$$V = \frac{1}{N-1} \sum_{i=1}^{N} (A_i - M)^2$$

e o desvio-padrão:

$$DP = \sqrt{V}$$

A_i é o valor de cada dado da amostra.

USANDO O PROGRAMA: Após *RUN* <NEWLINE>, vem:

'NU'MERO DE DADOS?

e entramos com o número de variáveis da amostra. Essas variáveis são previamente inseridas na declaração *"DATA"* da linha número 100. Vejamos o exemplo a seguir:

→ Exemplo 1: Calcular média, variância e desvio-padrão dos dados:

$A_1 = 12, \quad A_2 = 20, \quad A_3 = 48,$
$A_4 = 44, \quad A_5 = 76, \quad A_6 = 79.$

Após a introdução do programa, faremos:

EDIT 100
<NEWLINE>
X

e a linha 100 fica

100 DATA –

Entramos no modo direto de edição; EDIT 100 seguido de "X" nos permitirá inserir dados no final da linha; assim, digitamos

12, 20, 48, 44, 76, 79 <NEWLINE>

de forma que a linha 100 ficará com o seguinte aspecto:

100 DATA 12, 20, 48, 44, 76, 79

O passo seguinte: RUN

NU'MERO DE DADOS? –
6 <NEWLINE>

RESPOSTA:

NU'MERO DE DADOS: 6 MÉDIA: 46.5
VARIÂNCIA: 765.5 DESVIO-PADRÃO: 27.6677

QUER ANALISAR OUTROS DADOS (S/N)?

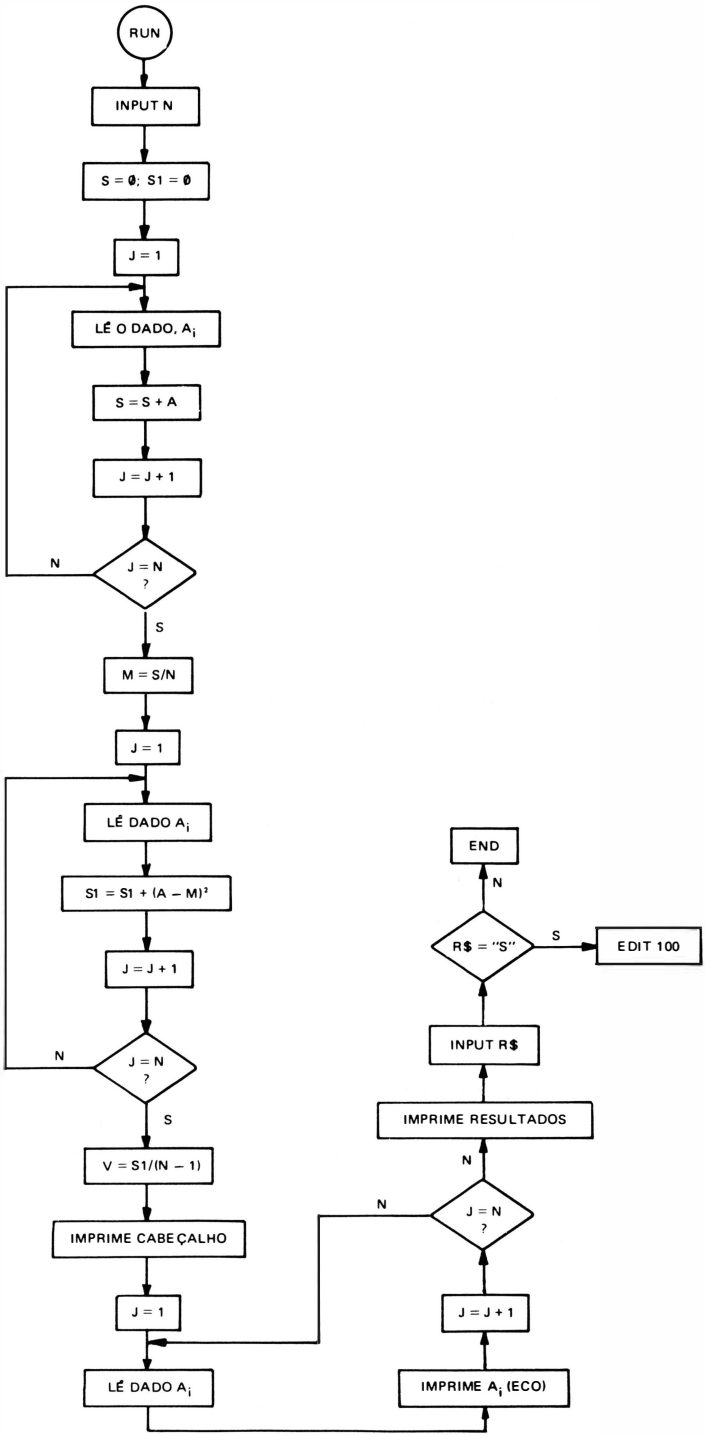
Se respondermos "S" (SIM), voltamos ao modo direto de edição, na linha 100.

→ Exemplo 2: Analisar os dados:

$A_1 = 1{,}752; \quad A_2 = 2{,}333; \quad A_3 = 5{,}1; \quad A_4 = 3{,}568$

RESPOSTA: M = 3,18825
V = 2,19776
DP = 1,48248

FLUXOGRAMA:


RUN
INPUT N
S = 0; S1 = 0
J = 1
LÊ O DADO, Ai
S = S + A
J = J + 1
J = N ?
N
S
M = S/N
J = 1
LÊ DADO Ai
S1 = S1 + (A – M)²
J = J + 1
J = N ?
N
S
V = S1/(N – 1)
IMPRIME CABEÇALHO
J = 1
LÊ DADO Ai
J = N ?
N
J = J + 1
IMPRIME Ai (ECO)
IMPRIME RESULTADOS
INPUT R$
R$ = "S"
S
EDIT 100
END
N

```
NU'MERO DE DADOS?









***** 'ECO' DOS DADOS FORNECIDOS *****
 12     20     48     44     76     79
************************************************

NU'MERO DE DADOS:  6              MEDIA:  46.5

VARIANCIA:  765.5                 DESVIO PADRAO:  27.6677

QUER ANALISAR OUTROS DADOS (S/N)? S
```

```
10 'PROGRAMA MEDIA, VARIANCIA E DESVIO PADRAO.
20 'DATA: 22/JAN/84
30 'AUTOR: FAUSTO A. DE A. BARBUTO.
40 CLS: INPUT"NU'MERO DE DADOS";N
50 S=0: S1=0
60 FOR J=1 TO N
70 READ A
80 S=S+A
90 NEXT J
100 DATA12,20,48,44,76,79
110 M=S/N
120 RESTORE
130 FOR J=1 TO N
140 READ A
150 S1=S1+(A-M)[2
160 NEXT J
170 V=S1/(N-1)
180 CLS: PRINT "***** 'ECO' DOS DADOS FORNECIDOS *****"
190 RESTORE
200 FOR J=1 TO N
205 READ A
210 PRINT A;"  ";
220 NEXT J
230 PRINT: PRINT STRING$(50,"*")
240 PRINT: PRINT"NU'MERO DE DADOS: ";N,"MEDIA: ";M
260 PRINT: PRINT"VARIANCIA: ";V,"DESVIO PADRAO: ";SQR(V)
280 PRINT
290 INPUT"QUER ANALISAR OUTROS DADOS (S/N)";R$
300 IF R$="S"THEN EDIT100 ELSE END
```

RAIZ DE EQUAÇÕES
– MÉTODO DE NEWTON-RAPHSON

16

Seja F (X), contínua e derivável no intervalo $[X_A, X_B]$

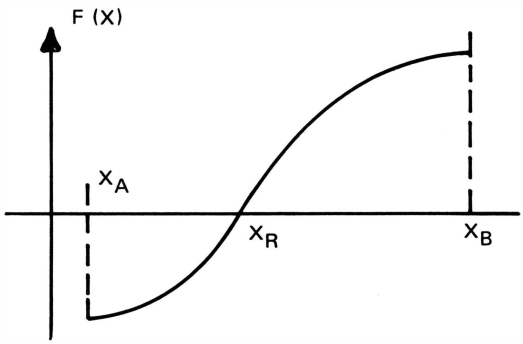


FIG. 1. F (X) no intervalo $[X_A, X_B]$.

Seja também $X_R \epsilon [X_A, X_B]$, tal que $F(X_R) = 0$, ou $F(X) = 0$ para $X = X_R$. X_R é então a *raiz* da função F (X) no intervalo $[X_A, X_B]$.

Algumas funções exibem uma certa facilidade para se calcular a raiz, por exemplo, as equações do segundo grau do tipo $Ax^2 + Bx + C$ onde a fórmula de Baskhara

$$X = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

nos fornece a raiz.

Como calcular, no entanto, raízes para equações do tipo

$$F(X) = X^3 - 4X^2 + 5X - 2?$$

Existem vários métodos: o da corda, da dicotomia, Newton-Raphson etc. Vamos nos deter, a partir de agora, neste último. Suponha-se uma F (X) de acordo com a Fig. 1.

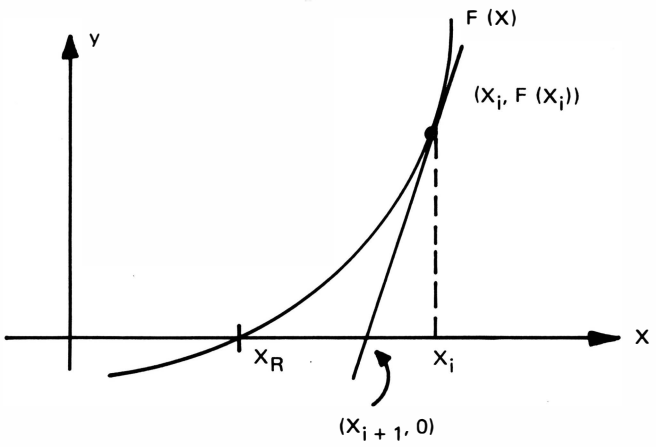


FIG. 2. F (X), um ponto $(X_i, F(X_i))$ e a reta tangente.

Na figura acima, representamos também um ponto $(X_i, F(X_i))$ e uma reta que tangencia este ponto e vai cortar o eixo X no ponto $(X_{i+1}, 0)$. Esta reta pode ser representada da seguinte maneira:

$$y = mx + C$$

onde 'm' é o coeficiente angular e 'C' o linear. Para o ponto $(X_i, F(X_i))$, "m" é a derivada da função neste ponto, ou:

$$m = F'(X_i)$$

então,

$$y = F'(X_i) \cdot x + C$$

Quando $X = X_i$, $y = F(X_i)$ e logo, temos:

$$F(X_i) = F'(X_i) \cdot X_i + C \qquad \text{(Equação 1)}$$

O ponto $(X_{i+1}, 0)$ também faz parte, melhor dizendo, pertence à reta acima. Então, quando $x = X_{i+1}$, $y = 0$. Logo:

$$0 = F'(X_i) \cdot X_{i+1} + C \qquad \text{(Equação 2)}$$

Diminuindo a equação 1 da equação 2, temos:

$$F(X_i) - 0 = F'(X_i)(X_i - X_{i+1}) + C - C$$

que fica:

$$F(X_i) = F'(X_i)(X_i - X_{i+1})$$

Remanejando os termos, temos:

$$X_{i+1} = X_i - \frac{F(X_i)}{F'(X_i)} \qquad \text{(Equação 3)}$$

Vejamos na Fig. 2, o que acontece se subirmos com X_{i+1} em F (X):

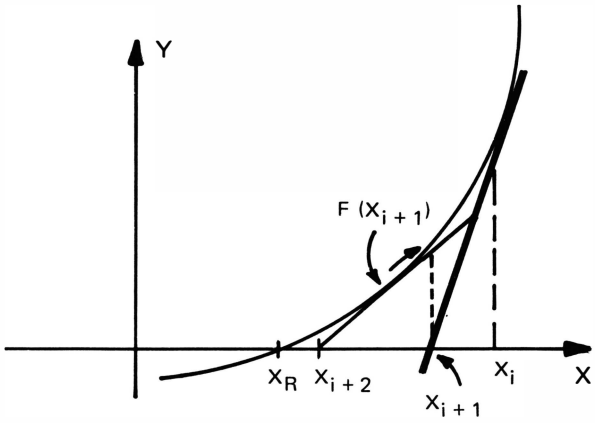


FIG. 3. Pontos X_{i+1} e X_{i+2}.

Note que X_{i+2} pode ser calculado usando-se o mesmo algoritmo usado para calcular X_{i+1} (Equação 3). Um outro fato interessante: X_{i+2} é um valor mais próximo da raiz do que X_{i+1}, que também está mais próximo dela do que X_i. O raciocínio lógico que se impõe agora é que se continuarmos o processo repetidamente por um certo número de vezes, acabaremos encontrando a raiz com uma precisão apreciável. Quanto mais vezes repetirmos o processo, maior também será a precisão. Podemos fixar um erro, de modo que quando o valor da função a ele se igualar, consideramos que a raiz foi atingida, tal que:

$$F(X_{i+N}) \leqslant \epsilon, \quad X_{i+N} \sim X_R$$

Outra alternativa: Fazemos *N* um número grande (no programa, N = 10 – mas o usuário pode variar *N* conforme o seu desejo). De modo a assegurar-nos que X_{i+N} é realmente um bom valor para a raiz. Essa é a opção escolhida pelo autor para o programa que ora descrevemos. Se necessário for o usuário pode – e deve – fazer as modificações que achar interessantes, inclusive optando pela fixação de um erro para calcular a raiz.

Nas linhas 120 e 130 colocamos a função e sua derivada, respectivamente. Como exemplo, na listagem do programa elas ficaram como

```
120 FX    = X[2+10*X+25
130 DFDX = 2*X+10
```

Correspondendo a $F(X) = X^2 + 10X + 25$ e $F'(X) = 2X + 10$. Só há necessidade de entrar com um dado para a execução do programa: o valor de X_i (é muito importante o critério de escolha desse dado. Falaremos sobre isso mais tarde).

→ Exemplo 1: Calcular a raiz de $X^2 + 10X + 25$, usando $X_i = 2$ e $X_i = -9$

Primeiramente, fazemos

RUN <NEWLINE>

DISPLAY

ABSCISSA INICIAL?

Fazemos:

2 <NEWLINE>

e obtemos a resposta:

RAIZ: 4,99354

Para $X_i = -9$, obtemos:

RAIZ: 5.00354

→ Exemplo 2: Calcular raiz de $X^2 - 8X + 15$ para $X_i = 10$ e $X_i = -3$.

A derivada $F'(X)$ é: $2X - 8$

As linhas 120 e 130:

```
120 FX    = X[2-8*X+15
130 DFDX = 2*X-8
```

a) Para $X_i = 10$:

RESPOSTA: RAIZ = 5

b) Para $X_i = -3$:

RESPOSTA: RAIZ = 3

Neste ponto é interessante fazer uma observação: neste exemplo encontramos duas raízes. O que acontece fica mais claro graficamente.

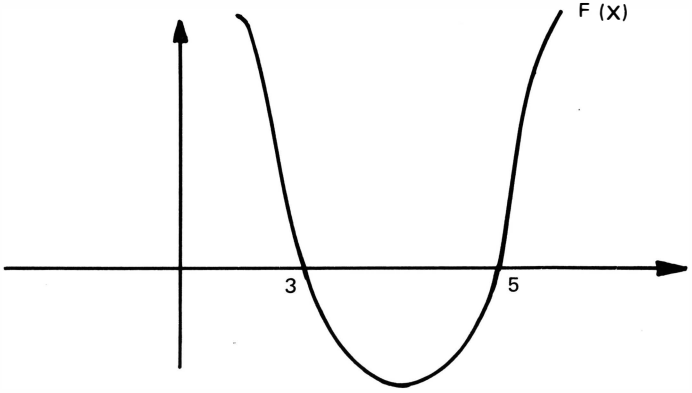


FIG. 4. $F(X) = X^2 - 8X + 15$.

É lógico que temos duas raízes distintas, $X_{R_1} = 3$ e $X_{R_2} = 5$. Quando escolhemos $X_i = 10$, convergimos para $X_{R_2} = 5$ pois $10 > 5$. Quando fazemos $X_i = -3$, a convergência tende para $X_{R_1} = 3$.

→ Exemplo 3: Calcular a raiz de $X^2 - 6X + 9$, com $X_i = -1$ e $X_i = +6$.

(FX = X [2 − 6 * X + 9 e DFDX = 2 * X − 6)

RESPOSTA:

A) Com $X_i = -1$;

RAIZ = 2,9962

B) Com $X_i = +6$

RAIZ = 3,00265

Repare agora: achamos a mesma raiz (~ 3), de acordo com a figura a seguir

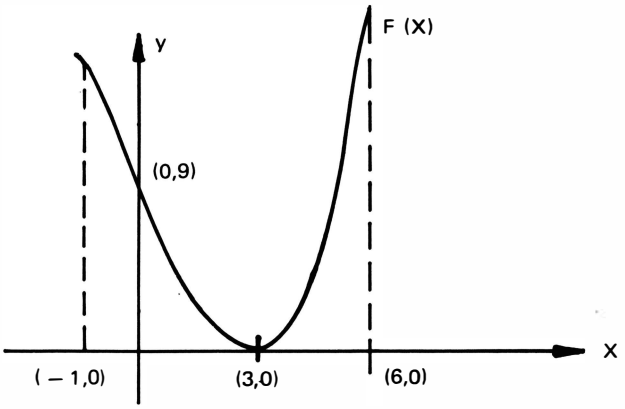


FIG. 5. F (X) = X^2 – 6X + 9.

Com $X_i = -1$ ou $X_i = 6$, convergimos para a (única) raiz.

→ Exemplo 4: Calcular a raiz de F (X) = LN (X) + X, com $X_i = 0{,}1$ e $X_i = 2$.

FX = LOG (X) + X e DFDX = 1/X + X

RESPOSTA:

Ambos os valores de X_i convergem para 0,567143. É importante uma observação, neste ponto: o que poderia ocorrer se o usuário, para calcular a raiz X_R da função mostrada na figura, escolhesse os seguintes pontos:

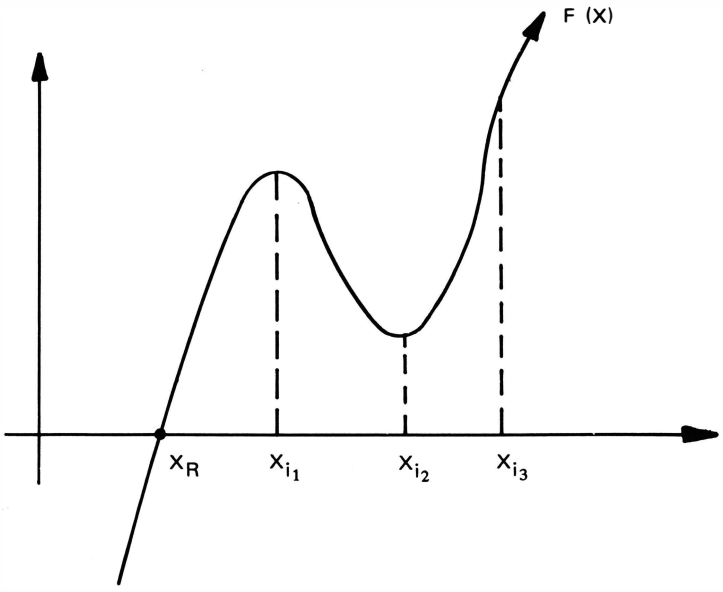


FIG. 6. Pontos X_i que não fornecem raízes corretas.

Primeiramente, X_{i_1} e X_{i_2} são, respectivamente, pontos de máximo e mínimo de F (X) no intervalo assim representado e assim, em ambos os casos, $F'(X) = 0$. Ocorreria, portanto, um erro na linha 8Ø (/Ø, ou divisão por zero).

Para X_{i_3}, bem, aí não há remédio. A raiz final calculada ficaria oscilando até o final do *LOOP* entre X_{i_1} e X_{i_3}, não importando que se aumente o valor final de "I" no laço *FOR-NEXT* (linha 6Ø) para 20, 50 ou 100. A explicação é simples: X_{i3+1} cairia no intervalo $[X_{i_1}, X_{i_2}]$. Na próxima iteração, X_{i_3+2} estaria contido no intervalo $[X_{i_2}, X_{i_3}]$ e assim indefinidamente. Para evitar contratempos, é sempre bom checar se a raiz calculada realmente zera F (X). Vamos, portanto, incluir mais uma linha ao programa.

```
85 PRINT @ 576, "- - - - - - - - - - → F (X) = "; FX; "← - - - - - - - - "
```

Que sempre nos fornece o último valor calculado de F (X). Se F (X) não for zero — ou quase zero — é justo desconfiar do valor escolhido de X_i.

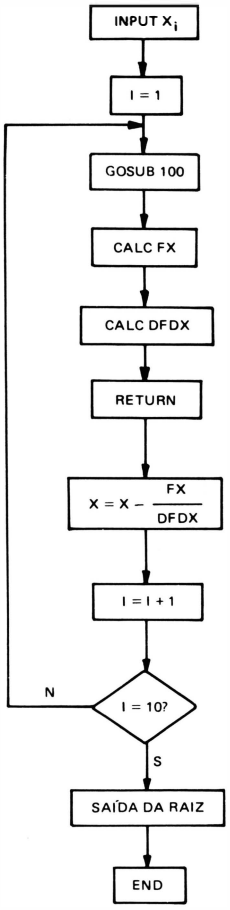


FIG. 7. Fluxograma de Newton-Raphson.

```
ABCISSA INICIAL? 2



==========) RAIZ: -4.99354 (==========
READY
```

```
10 'RAIZ DE EQUACOES PELO ME'TODO DE NEWTON-RAPHSON.
20 'AUTOR: FAUSTO A. DE A. BARBUTO.   DATA: 25/JAN/84.
30 CLS: INPUT"ABCISSA INICIAL";X
60 FOR I=1 TO 10
75 GOSUB120
80 X=X-FX/DFDX
90 NEXT I
100 PRINT @448,"==========) RAIZ: ";X;"(==========="
110 END
115 'SUBROTINA PARA CALCULO DA FUNCAO E SUA DERIVADA
120 FX=X[2+10*X+25
130 DFDX=2*X+10
140 RETURN
```

RAIZ DE UMA EQUAÇÃO – MÉTODO DA CORDA

17

Já vimos anteriormente como calcular a raiz de uma equação do tipo $Y = F(X)$, $X \epsilon R$, pelo método de Newton. Desta feita vamos fazê-lo de outro modo: usando o método da corda (chord method), também bastante interessante do ponto de vista matemático.

Como funciona?

Seja F (X), contínua e derivável, de acordo com a Fig. 1:

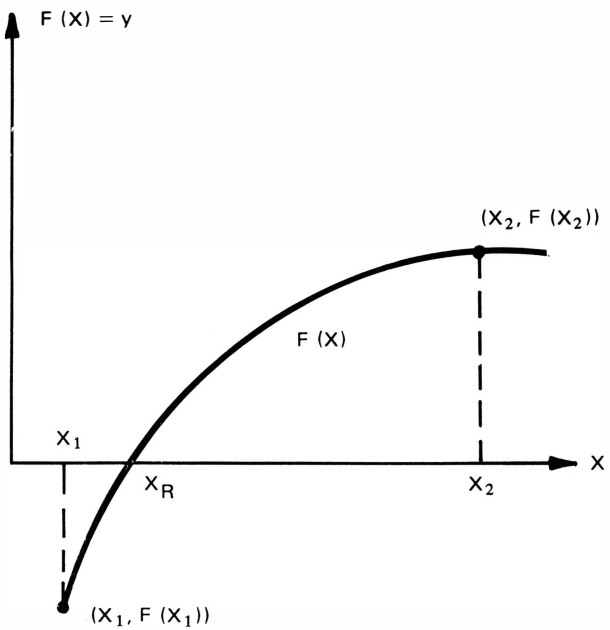


FIG. 1. Função F (X).

Vemos que X_R, a raiz, está contida no intervalo $[X_1, X_2]$ e que $F(X_R) = 0$.

Vamos agora ligar os pontos $(X_1, F(X_1))$ e $(X_2, F(X_2))$ por uma reta, conforme a Fig. 2.

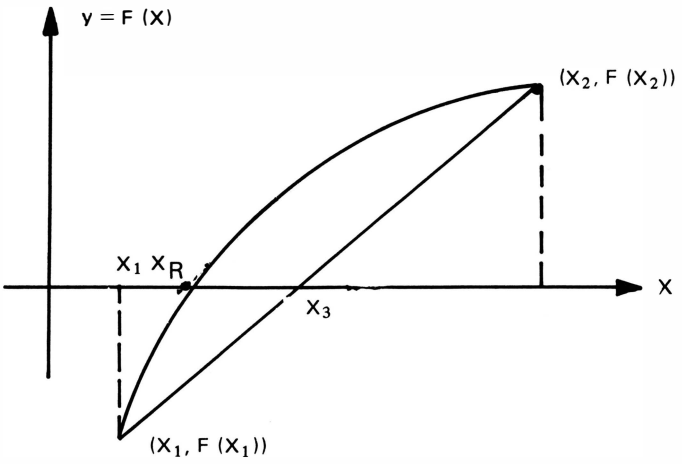


FIG. 2. F (X) e a corda.

O segmento de reta que liga $(X_1, F(X_1))$ a $(X_2, F(X_2))$ é chamado de corda. A sua intersecção com o eixo das abscissas (eixo X) determina um ponto X_3, que pode ser calculado analiticamente como se segue.

A equação que une os pontos $(X_1, F(X_1))$ e $(X_2, F(X_2))$ é a seguinte:

$$F(X) - F(X_1) = \frac{(F(X_2) - F(X_1))}{X_2 - X_1} (X - X_1)$$

Quando $X = X_3$, $F(X) = 0$; logo

$$0 - F(X_1) = \frac{(F(X_2) - F(X_1))}{X_2 - X_1} (X_3 - X_1)$$

Explicitando,

$$X_3 = X_1 - F(X_1) \frac{(X_2 - X_1)}{(F(X_2) - F(X_1))}$$

Se, com X_3, calcularmos $F(X_3)$ e novamente ligarmos esse novo ponto a $(X_1, F(X_1))$ ficaremos com: (Fig. 3).

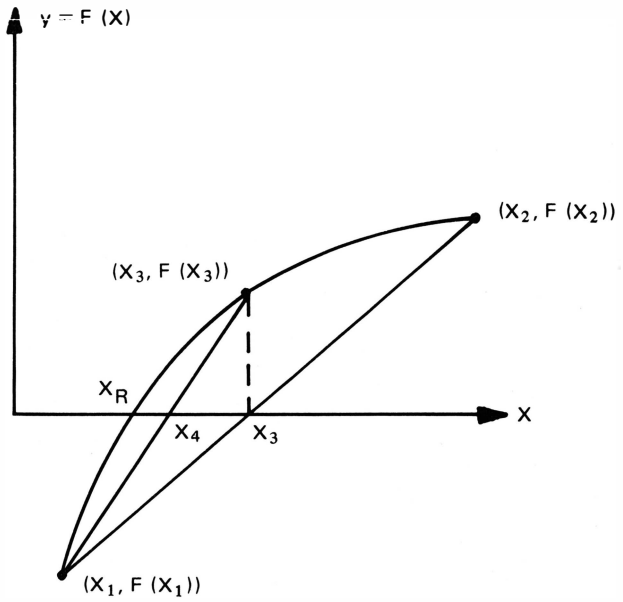


FIG. 3. F (X) e a nova corda.

Observe que determinamos uma nova corda entre os pontos $(X_1, F(X_1))$ e $(X_3, F(X_3))$ e ficou também estabelecida uma nova intersecção desta corda com o eixo-X, X_4. Note também que $|X_R - X_4| < |X_R - X_3|$, ou em outras palavras X_4 está mais próximo da raiz que X_3. Se continuarmos o processo aproximar-nos-emos cada vez mais da raiz X_R. É fácil notar visualmente isso. E mais: com umas dez iterações já teremos obtido um valor razoável para a raiz.

A equação para calcular X_4 fica assim:

$$X_4 = X_1 - F(X_1) \frac{X_3 - X_1}{(F(X_3) - F(X_1))}$$

Generalizando,

$$X_{i+1} = X_1 - F(X_1) \frac{(X_i - X_1)}{(F(X_i) - F(X_1))}$$

Se a concavidade da curva está voltada para cima, fica:

$$X_{i+1} = X_2 - F(X_2) \frac{(X_i - X_2)}{(F(X_i) - F(X_2))}$$

O programa apresentado discerne essas situações, vide linha **100**.

Na linha 22Ø colocamos a função a ter sua raiz calculada. Se $y = F(X) = 4 - X^2$, temos:

```
22Ø Y (K) = 4 – X (K) [ 2
```

ou

```
22Ø Y (K) = 4 – X (K) * X (K)
```

VARIÁVEIS USADAS NO PROGRAMA:

NOME	*TIPO*	*OBSERVAÇÕES*
X (K)	VETOR REAL, 16 POSIÇÕES	ABSCISSA
Y (K)	IDEM	ORDENADA (F (X))
K	INTEIRA	ÍNDICE DOS VETORES X, Y

→ Exemplo 1: Calcular a raiz de $Y = F(X) = 4 - X^2$ no intervalo $[-4, -1]$.

Antes de tudo, é bom saber que a escolha dos limites do intervalo é muito criteriosa: é condição "sine qua non" que o valor de F (X) se anule para $X \epsilon [X_1, X_2]$. Desse modo, vemos que:

$$F(-4) = 4 - (-4)^2 = 4 - 16 = -12$$
$$F(-1) = 4 - (-1)^2 = 4 - 1 = +3$$

Houve uma mudança de sinal de F (– 4) para F (– 2). Como F (X) é contínua e diferenciável neste intervalo, essa mudança de sinal passou, certamente, por um valor intermediário e nulo de F (X).

Fazemos a linha 220 como

```
22Ø Y (K) = 4 – X (K) [ 2
```

e, após isto, *RUN.*

Aparecem no topo do vídeo:

```
K, X (K)?
```

Digitamos 1, e, *NEWLINE* e desta forma:

```
K, X (K)? 1
??
```

Fizemos K = 1 (1º ponto) e agora faremos

– 4 <NEWLINE>

Que assume X (1) como – 4.

Para o segundo ponto, faremos:

2 <NEWLINE>
– 1 <NEWLINE>

e finalmente:

A raiz da equação é: X = – 2

Vamos testar agora um novo intervalo: [+ Ø.5, + 4.5]

Os passos serão:

<RUN>
1 <NEWLINE>
Ø.5 <NEWLINE>
2 <NEWLINE>
4.5 <NEWLINE>

que nos dá uma raiz X = 2.

OBS.: Usamos o programa para calcular as raízes de $Y = 4 - X^2$ apenas por motivos didáticos: esta equação é trivial e não há necessidade real de se usar artifícios numéricos para se calcular as suas raízes!

Vamos partir agora para a resolução de uma equação não-trivial.

→ Exemplo 2: Calcular a raiz de Y = F (X) = LOG (X) + X (logaritmo neperiano) no intervalo [Ø.1, 2.5].

RESPOSTA: Raiz calculada = 0,566912.

Agora que sabemos que a raiz está em torno do valor acima, vamos "apertar" mais um pouco o intervalo e obter um resultado mais preciso para a raiz. Façamos agora o "GAP" variar de 0,1 a 1,0 ([Ø.1, 1.Ø]) e executemos novamente o programa assim:

RUN <NEWLINE>
1 <NEWLINE>
1 <NEWLINE>
2 <NEWLINE>
1 <NEWLINE>

e obtemos:

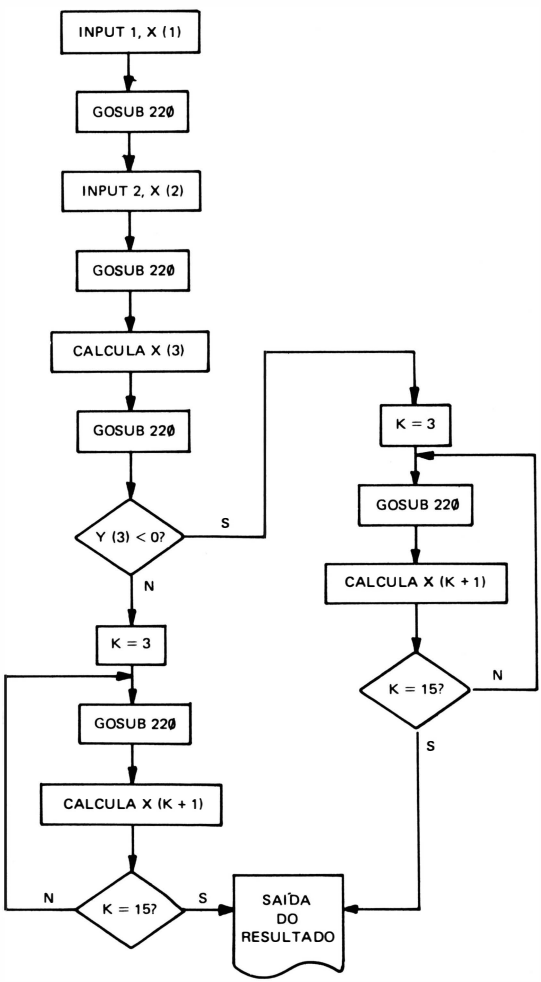
A raiz da equação é: 0,567143

Poderíamos melhorar o *SOFT* modificando um pouco as linhas números 35 a 70, assim:

```
35 DIM X(16), Y(16)
40 INPUT "X (1)"; X (1)
50 K = 1: GOSUB 220
60 INPUT "X (2)"; X (2)
70 K = 2: GOSUB 220
```

Isso simplifica um pouco as coisas para o usuário.

FLUXOGRAMA DO PROGRAMA:



```
K, X(K)? 1
?? .1
K, X(K)? 2
?? 2.5.
```

```
A RAIZ E': X= .566912
READY
```

```
10 'RAIZ DE UMA EQUACAO - ME'TODO DA CORDA.
20 'POR: FAUSTO A. DE A. BARBUTO.   DATA:12/FEV/84.
30 CLS
40 DIM X(16), Y(16)
50 INPUT"K, X(K)"; K, X(K)
60 GOSUB 220
70 INPUT"K, X(K)"; K, X(K)
80 GOSUB 220
90 X(3)=X(1)-Y(1)*(X(2)-X(1))/(Y(2)-Y(1))
100 GOSUB 220
110 IF Y(3)<0 THEN 150
120 FOR K=3 TO 15
130 GOSUB 220
140 X(K+1)=X(1)-Y(1)*(X(K)-X(1))/(Y(K)-Y(1))
150 NEXT K
160 FOR K=3 TO 15
170 GOSUB 220
180 X(K+1)=X(2)-Y(2)*(X(K)-X(2))/(Y(K)-Y(2))
190 NEXT K
200 CLS:PRINT@448,"A RAIZ E': X=";X(16)
210 END
215 'SUBROTINA-FUNCAO.
220 Y(K)=LOG(X(K))+X(K)
230 RETURN
```

OPERAÇÕES COM MATRIZES: SOMA DE DUAS MATRIZES, SUBTRAÇÃO DE DUAS MATRIZES, MULTIPLICAÇÃO POR UM ESCALAR, MULTIPLICAÇÃO DE UMA MATRIZ POR OUTRA

18

As operações matriciais que veremos doravante são extremamente importantes para aqueles que lidam com sistemas de equações e transformações lineares, vetores etc.

Numa matriz, os elementos estão dispostos em *linhas* e *colunas,* e cada elemento tem a forma genérica A_{ij} (para uma matriz $\overline{A}$) onde *i* é o número da linha e *j* o da coluna. Assim, A_{24} é o quarto elemento — pertecente à quarta coluna — da segunda linha. Dizemos também que uma matriz tem dimensões (m X n) quando tem m linhas e n colunas. Seja portanto a matriz abaixo representada na sua forma usual:

$$\overline{A} = \begin{bmatrix} 0 & 2 \\ -3 & 7 \end{bmatrix}$$

O elemento A_{21} é igual a -3, e a matriz acima tem dimensões (2×2). Quando o número de linhas é igual ao de colunas, diz-se que a matriz é quadrada. Se agora definirmos a matriz $\overline{B}$ como:

$$\overline{B} = \begin{bmatrix} 3 & -2 \\ 5 & -4 \end{bmatrix}$$

A soma de $\overline{A}$ com $\overline{B}$ será

$$\overline{A} + \overline{B} = \begin{bmatrix} 0+3 & 2-2 \\ -3+5 & 7-4 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 2 & 3 \end{bmatrix}$$

e a subtração:

$$\overline{A} - \overline{B} = \begin{bmatrix} -3 & 4 \\ -8 & 11 \end{bmatrix}$$

Podemos observar que $\overline{A} + \overline{B} = \overline{B} + \overline{A}$ e que $\overline{A} - \overline{B} = -(\overline{B} - \overline{A})$.

(OBS.: Só podemos somar e/ou substituir uma matriz (m X n) de uma (p X q) se m = p e n = q).

A multiplicação escalar é igualmente simples; após a multiplicação por um escalar, cada elemento da matriz passará a ser $K \cdot A_{ij}$. Exemplo: Multiplicar a matriz (1 X 3) abaixo por seis (6).

$$\overline{A} = [\ -1 \quad 0 \quad 3\ ]$$

$$\overline{A} \cdot K = [\ -6 \quad 0 \quad 18\ ]$$

A multiplicação por um escalar apresenta a propriedade distributiva, ou seja:

$$K\ (\overline{A} + \overline{B}) = K\overline{A} + K\overline{B}$$

A multiplicação de uma matriz por outra é tarefa um pouco mais complexa. Multiplicamos uma matriz (m X n) por uma (p X q) se n = p, e o resultado é uma matriz (m X q). Como exemplo, sejam as duas matrizes abaixo:

$$\overline{A} = \begin{bmatrix} 1 & 2 \\ 5 & 9 \end{bmatrix} \text{ e } \overline{B} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

O produto $\overline{A} \times \overline{B}$ é:

$$\overline{A} \times \overline{B} = \begin{bmatrix} 3 \times 1 + 4 \times 2 \\ 5 \times 3 + 4 \times 9 \end{bmatrix} = \begin{bmatrix} 11 \\ 51 \end{bmatrix}$$

A multiplicação matricial possui propriedades associativas e distributivas:

$$\overline{A} \times \overline{B} \times \overline{C} = \overline{A} \times (\overline{B} \times \overline{C}) = (\overline{A} \times \overline{B}) \times \overline{C}$$

$$(\overline{A} + \overline{B}) \times \overline{C} = \overline{C} \times \overline{A} + \overline{C} \times \overline{B}$$

Com o programa que listamos adiante podemos verificar todas as propriedades acima descritas.

→ Exemplo 1: Somar as matrizes $\overline{A}$ e $\overline{B}$ a seguir:

$$\overline{A} = \begin{bmatrix} 5 & 2 & 0 & -1 \\ 7 & 0 & 6 & -6 \\ 5 & 4 & -4 & 8 \end{bmatrix} \quad \overline{B} = \begin{bmatrix} 10 & 7 & -2 & -5 \\ 4 & 0 & -9 & 6 \\ 4 & -3 & -3 & 9 \end{bmatrix}$$

Uma vez executado o comando *RUN*, e tendo-se escolhido o código da operação que vamos fornecer, há que se fornecer a *dimensão* das matrizes que vão ser somadas; no caso, (3, 4).

```
?
3 <ENTER>
??
4 <ENTER>
```

Após, os elementos da 1ª matriz, linha por linha:

```
 5 <ENTER>
 2 <ENTER>
 Ø <ENTER>
-1 <ENTER>
 ...........
```

e assim por diante, inclusive a 2ª matriz. Não há como errar, as mensagens-interface são claras. A resposta é a seguinte:

$$\overline{A} + \overline{B} = \begin{bmatrix} 15 & 9 & -2 & -6 \\ 11 & 0 & -3 & 0 \\ 9 & 1 & -6 & 17 \end{bmatrix}$$

A saída – elementos C (J, K) – é também fornecida linha por linha. Quando uma linha se acaba, aparece a mensagem:

BREAK IN <NÚMERO DA LINHA>

Executando-se *"CONT"*, obteremos os elementos da linha seguinte, até a última. O resultado da subtração das duas matrizes é:

$$\overline{A} - \overline{B} = \begin{bmatrix} -5 & -5 & 2 & 4 \\ 3 & 0 & 15 & -12 \\ 1 & 7 & 0 & -1 \end{bmatrix}$$

→ Exemplo 2: Multiplicar por cinco a matriz $\overline{A}$ abaixo:

$$\overline{A} = \begin{bmatrix} 5 & -1 \\ 0 & -2 \\ -1 & 0{,}5 \\ 2 & 6 \end{bmatrix}$$

A opção no "menu" é o número 3

3 <ENTER>

Em seguida, as dimensões da matriz: (4 X 2)

4 <ENTER>

2 <ENTER>

e, por último e de modo semelhante ao exemplo anterior, os elementos da matriz:

```
  5 <ENTER>
– 1 <ENTER>
  0 <ENTER>
– 2 <ENTER>
  ----------
```

até o último elemento. A matriz-resposta é:

$$\overline{P} = \begin{bmatrix} 25 & -5 \\ 0 & -10 \\ -5 & 2{,}5 \\ 10 & 30 \end{bmatrix}$$

Também dada linha a linha, necessita-se de "*CONT*" para listar as outras linhas.

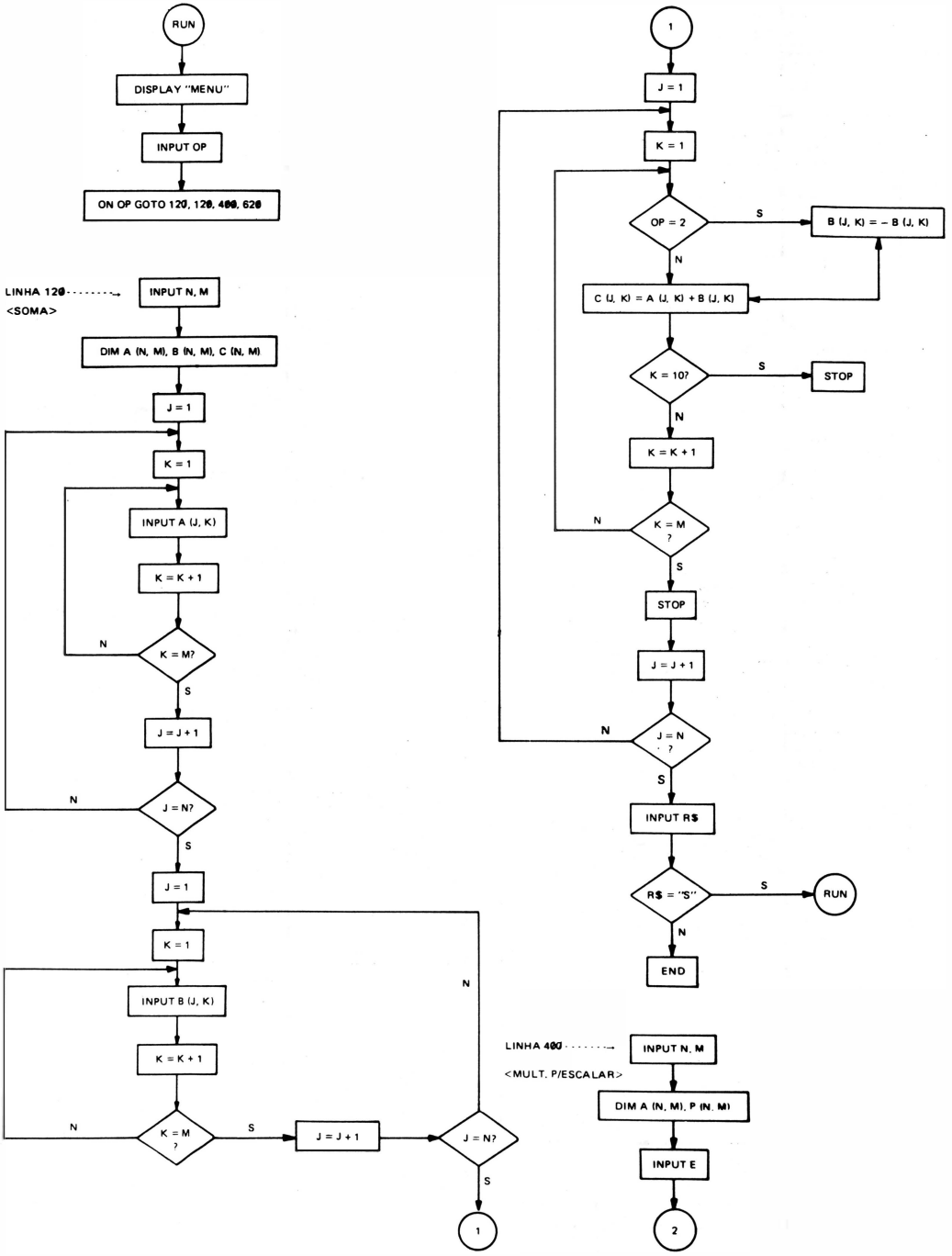
→ Exemplo 3: Multiplicar a matriz $\overline{A}$ pela $\overline{B}$.

$$\overline{A} = \begin{bmatrix} 5 & 2 & -3 \\ 7 & -7 & 1 \end{bmatrix} \quad \overline{B} = \begin{bmatrix} 3 \\ 2 \\ -1 \end{bmatrix}$$

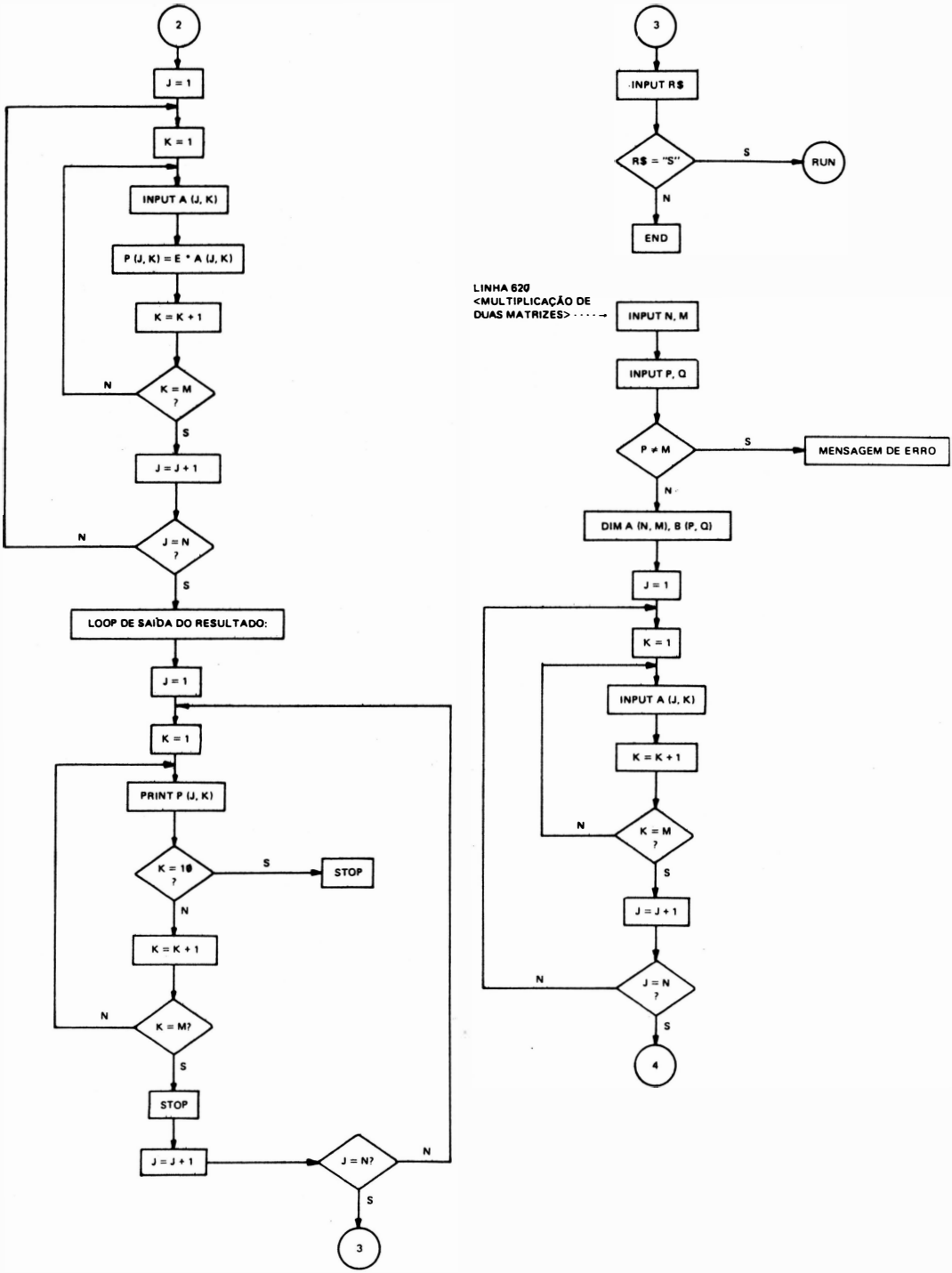
RESPOSTA: $\overline{P} = \overline{A} \times \overline{B} = \begin{bmatrix} 22 \\ 6 \end{bmatrix}$

FLUXOGRAMA:

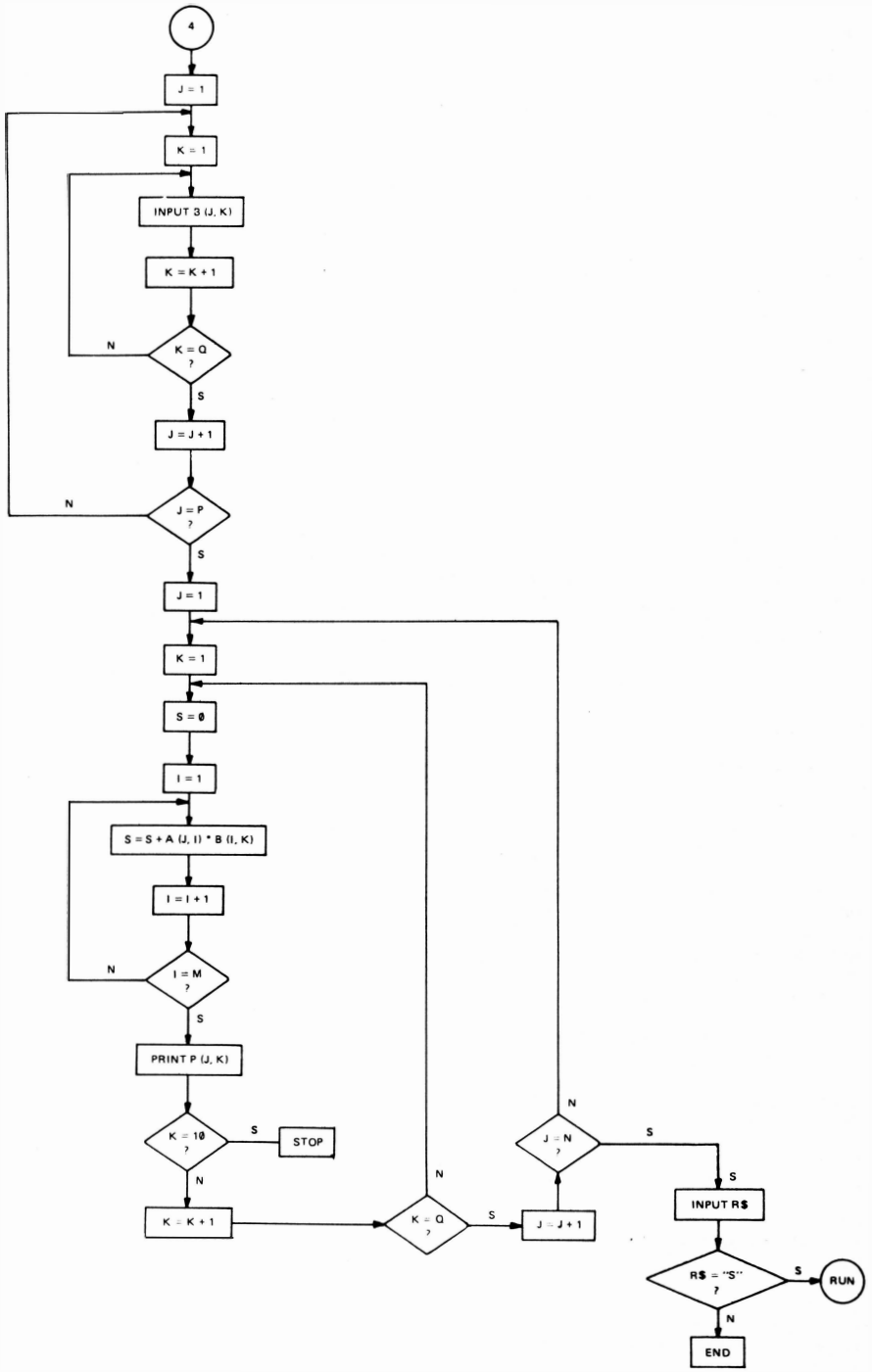
OPERAÇÃO C/MATRIZES



OP. C/MATRIZES (CONT. 1).



OP. C/MATRIZES (CONT. 4).



```
                              MENU DE OPCOES:

SOMA DE DUAS MATRIZES: 1

SUBTRACAO DE DUAS MATRIZES: 2

MULTIPLICACAO POR UM ESCALAR: 3

MULTIPLICACAO MATRIZ VS. MATRIZ: 4

OPCAO: ? 1




                              RESPOSTAS:
C( 1 , 1 )= 15
C( 1 , 2 )= 9
C( 1 , 3 )=-2
C( 1 , 4 )=-6
Break na 350
READY
>
```

```
10 'OPERACOES COM MATRIZES: SOMA, SUBTRACAO, MULTIPLICACAO
20 'POR UM ESCALAR E MULTIPLICACAO MATRIZ VS. MATRIZ.
30 'POR: FAUSTO A. DE A. BARBUTO.   DATA:11/JUL/84.
40 CLS:PRINT@25,"MENU DE OPCOES:"
50 PRINT:PRINT"SOMA DE DUAS MATRIZES: 1"
60 PRINT:PRINT"SUBTRACAO DE DUAS MATRIZES: 2"
70 PRINT:PRINT"MULTIPLICACAO POR UM ESCALAR: 3"
80 PRINT:PRINT"MULTIPLICACAO MATRIZ VS. MATRIZ: 4"
90 PRINT:INPUT"OPCAO: ";OP
100 ON OP GOTO 120, 120, 400, 620
110 'SOMA DE DUAS MATRIZES.
120 CLS:INPUT"DIMENSAO DA MATRIZ (LINHAS * COLUNAS)"; N, M
130 DIM A(N,M), B(N,M), C(N,M)
135 'PRIMEIRA MATRIZ.
140 FOR J=1 TO N
150 CLS:PRINT"PRIMEIRA MATRIZ: LINHA NO."; J
160 FOR K=1 TO M
170 PRINT"ELEMENTO DA COLUNA NO."; K
180 INPUT A(J,K)
190 NEXT K, J
200 'SEGUNDA MATRIZ.
210 FOR J=1 TO N
220 CLS:PRINT"SEGUNDA MATRIZ: LINHA NO."; J
230 FOR K=1 TO M
240 PRINT"ELEMENTO DA COLUNA NO."; K
250 INPUT B(J,K)
260 NEXT K, J
270 'SOMA/SUBTRACAO DAS MATRIZES.
280 FOR J=1 TO N
285 CLS:PRINT@24,"RESPOSTAS:"
290 FOR K=1 TO M
300 IF OP=2 THEN B(J,K)=-B(J,K)
310 C(J,K)=A(J,K)+B(J,K)
320 PRINT"C(";J;",";K;")=";C(J,K)
330 IF K=10 THEN STOP
340 NEXT K
350 STOP
360 NEXT J
370 INPUT"QUER EXECUTAR DE NOVO (S/N)"; R$
380 IF R$="S" THEN RUN ELSE END
390 'MULTIPLICACAO POR UM ESCALAR.
400 CLS:INPUT"DIMENSAO DA MATRIZ (LINHAS * COLUNAS)"; N, M
410 DIM A(N,M), P(N,M)
```

```
420 PRINT:INPUT"ESCALAR"; E
430 FOR J=1 TO N
440 CLS:PRINT"MATRIZ: LINHA NO."; J
450 FOR K=1 TO M
460 PRINT"ELEMENTO NO."; K
470 INPUT A(J,K)
480 P(J,K)=E*A(J,K)
490 NEXT K, J
500 'SAIDA DOS RESULTADOS.
510 CLS:PRINT@27,"RESULTADOS:"
520 FOR J=1 TO N
530 FOR K=1 TO M
540 PRINT"P(";J;",";K;")="; P(J,K)
550 IF K=10 THEN STOP
560 NEXT K
570 STOP
580 NEXT J
590 INPUT"QUER EXECUTAR DE NOVO (S/N)"; R$
600 IF R$="S" THEN RUN ELSE END
610 'MULTIPLICACAO MATRIZ VS. MATRIZ.
620 CLS:INPUT"DIMENSAO DA MATRIZ A SER MULTIPLICADA (LINHAS * COLUNAS)"; N, M
630 PRINT:INPUT"DIMENSAO DA MATRIZ MULTIPLICADORA (L * C)"; P, Q
640 IF P()M THEN PRINT"MULTIPLICACAO IMPOSSI'VEL: P()M."
650 DIM A(N,M), B(P,Q)
660 CLS:PRINT"MATRIZ NO. 1"
670 FOR J=1 TO N
680 PRINT:PRINT"LINHA NO."; J
690 FOR K=1 TO M
700 PRINT"ELEMENTO NO.";K
710 INPUT A(J,K)
720 NEXT K, J
730 CLS:PRINT"MATRIZ NO. 2"
740 FOR J=1 TO P
750 PRINT:PRINT"LINHA NO.";J
760 FOR K=1 TO Q
770 PRINT"ELEMENTO NO.";K
780 INPUT B(J,K)
790 NEXT K, J
795 CLS
800 FOR J=1 TO N
810 FOR K=1 TO Q
820 S=0
830 FOR I=1 TO M
840 S=S+A(J,I)*B(I,K)
850 NEXT I
860 PRINT"P(";J;",";K;")="; S
870 IF K=10 THEN STOP
880 NEXT K
900 NEXT J
910 PRINT:INPUT"QUER EXECUTAR DE NOVO? (S/N)"; R$
920 IF R$="S" THEN RUN ELSE END
```

SOLUÇÃO DE UM SISTEMA LINEAR DE EQUAÇÕES PELO MÉTODO DE GAUSS-JORDAN (até 40 Equações)

19

Vamos agora iniciar a falar sobre um programa que utiliza o conhecido algoritmo de Gauss-Jordan para resolver um sistema de até 40 equações lineares a 40 variáveis. Este programa é uma conversão para o *BASIC* de um programa em *FORTRAN* de autoria do brigadeiro Tércio Pacitti (FORTRAN MONITOR – PRINCÍPIOS, 3ª edição, Livros Técnicos e Científicos Editora S.A.). Não convém nos estendermos demais na base teórica do Método (Diagonalização das Matrizes que representam um sistema linear), sob riscos de tornarmos a discussão árida demais e transcendente ao objetivo desta obra. Quem, no entanto, se interessar pela teoria, pode se reportar ao livro citado anteriormente, que lá encontrará os subsídios que deseja.

USANDO O PROGRAMA:

Em primeiro lugar, entramos com a ordem do sistema, "N"; depois com os coeficientes das variáveis e, por último, com o vetor constante. O que são cada um desses dados? Seja a seguinte equação pertencente a um sistema de equações lineares:

$$a_{11}X_1 + a_{12}X_2 + a_{13}X_3 + \ldots + a_{1\,(N-1)}X_{N-1} + a_{1N}X_N = C$$

Deste modo, "N" é a ordem do sistema: é o seu número de variáveis. $a_{11}, a_{12}, a_{13}, \ldots, a_{1\,(N-1)}, a_{1N}$ são os coeficientes, e "C" é um dos "N" componentes do vetor constante.

→ Exemplo 1: Resolver o sistema abaixo:

$$\begin{cases} 2X_1 + 4X_2 - X_3 = 7 \\ X_1 + X_2 - 4X_3 = -9 \\ 5X_1 - 7X_2 + 2X_3 = -3 \end{cases}$$

Após *RUN*, respondemos com 3 <NEWLINE> à pergunta:

ORDEM DO SISTEMA?

Em seguida, entramos com os coeficientes da primeira até a n-ésima equação deste modo:

```
  2 <NEWLINE>
  4 <NEWLINE>
- 1 <NEWLINE>
  1 <NEWLINE>
  ------------
- 7 <NEWLINE>
  2 <NEWLINE>
```

Após o último coeficiente, vamos à derradeira etapa: entrada do vetor constante:

```
  7 <NEWLINE>
- 9 <NEWLINE>
- 3 <NEWLINE>
```

Agora vemos listado no *DISPLAY* o "eco" dos coeficientes das equações fornecidas anteriormente. Pressione qualquer tecla para continuar (exceto *BREAK* e *SHIFT*). No vídeo aparece o "eco" do vetor constante. Uma vez mais, apertamos uma tecla qualquer e é dada a resposta; as raízes do sistema são:

$$X_1 = 1,\ X_2 = 2,\ X_3 = 3.$$

Há, é claro, a hipótese da solução ser indeterminada ou impossível. Nesse caso, a resposta é remetida às linhas 660 e 680, respectivamente.

O "eco" dos dados é para simples conferência do usuário. Afinal, não é de se estranhar que um dado possa ser introduzido errado, ainda mais quando a ordem do sistema é grande. Quando acontecer um equívoco desse tipo, pressione "*BREAK*" e, após "*READY*", coloque diretamente (vamos supor que você errou no 5º coeficiente):

```
A (2,2) = 1
<NEWLINE>
```

e depois voltamos à linha onde paramos com *GOTO* n (no caso, n = 255). Se houver mais de um erro, repetimos o processo de correção, antes de GOTO n.

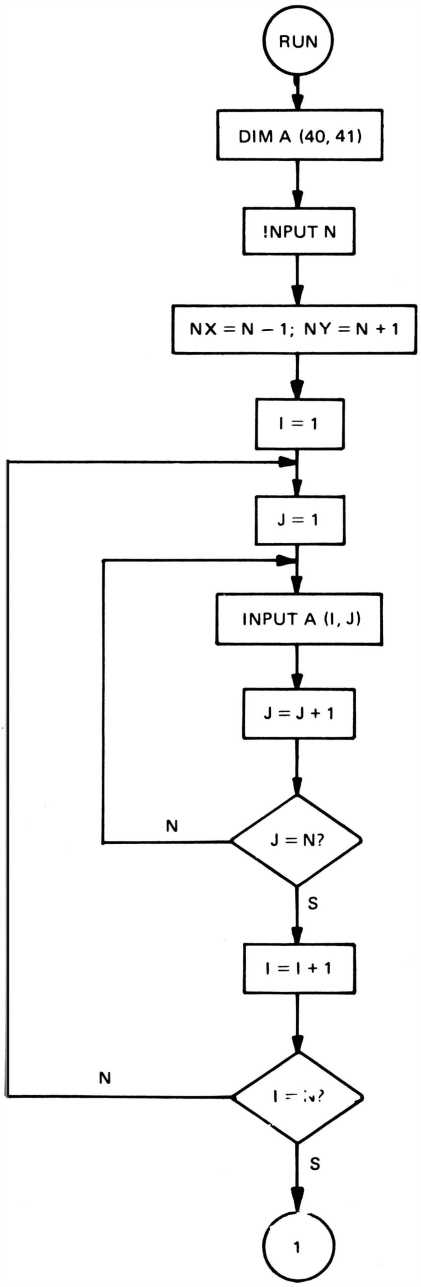
→ Exemplo 2: Resolver o sistema:

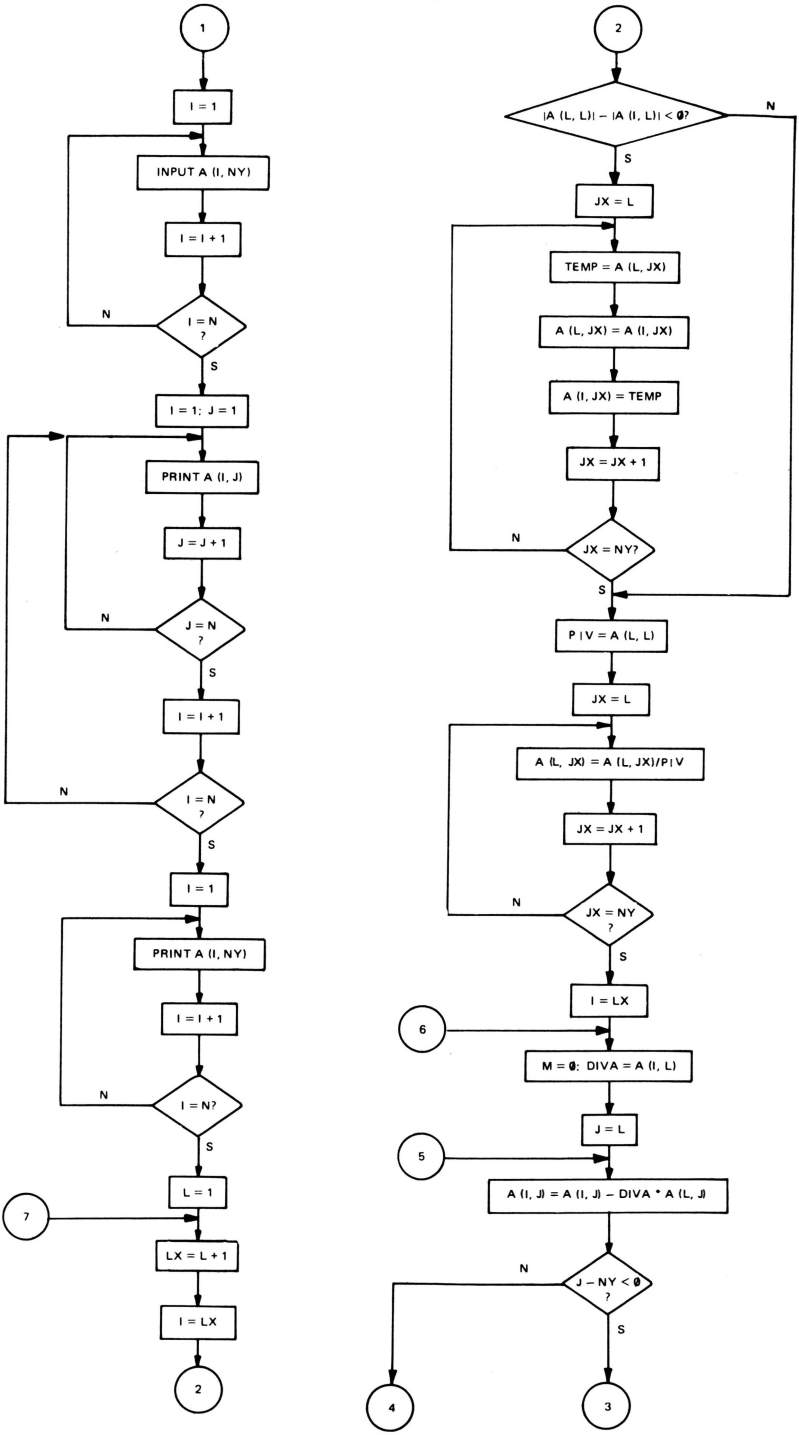
$$\begin{aligned} 2X_1 + X_2 + 5X_3 - X_4 &= -6 \\ 5X_1 + 3X_2 - X_3 + 4X_4 &= 3 \\ 10X_1 - 5X_2 + 2X_3 + 5X_4 &= 43 \\ 6X_1 + 3X_2 - 3X_3 + X_4 &= 4 \end{aligned}$$

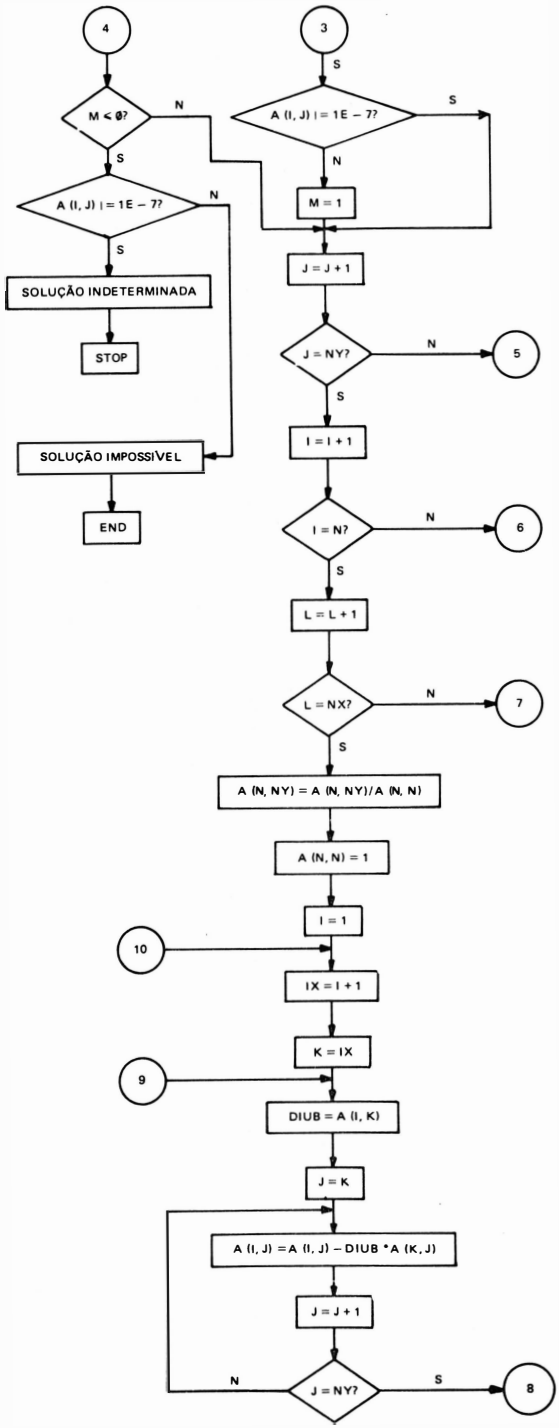
Jogando os coeficientes e constantes acima no programa, obtemos:

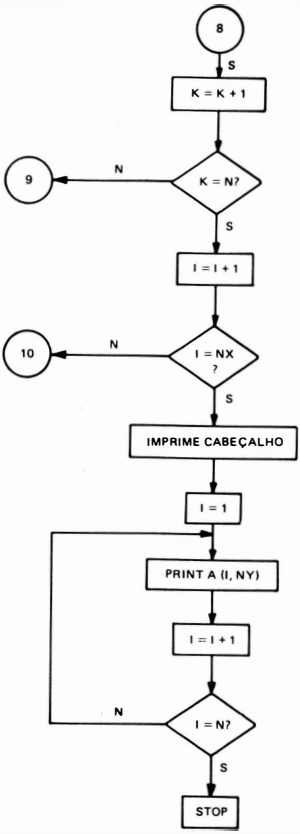
$$X_1 = 2;\ X_2 = -4;\ X_3 = -1;\ X_4 = 1$$

FLUXOGRAMA:




1
I = 1
INPUT A (I, NY)
I = I + 1
I = N ?
N
S
I = 1; J = 1
PRINT A (I, J)
J = J + 1
J = N ?
N
S
I = I + 1
I = N ?
N
S
I = 1
PRINT A (I, NY)
I = I + 1
I = N?
N
S
L = 1
7
LX = L + 1
I = LX
2
2
|A (L, L)| – |A (I, L)| < Ø?
N
S
JX = L
TEMP = A (L, JX)
A (L, JX) = A (I, JX)
A (I, JX) = TEMP
JX = JX + 1
JX = NY?
N
S
PIV = A (L, L)
JX = L
A (L, JX) = A (L, JX)/PIV
JX = JX + 1
JX = NY ?
N
S
I = LX
6
M = Ø; DIVA = A (I, L)
J = L
5
A (I, J) = A (I, J) – DIVA * A (L, J)
J – NY < Ø ?
N
S
4
3


4
M < 0?
N
S
A (I, J) = 1E – 7?
N
S
SOLUÇÃO INDETERMINADA
STOP
SOLUÇÃO IMPOSSÍVEL
END
3
S
A (I, J) = 1E – 7?
S
N
M = 1
J = J + 1
J = NY?
N
5
S
I = I + 1
I = N?
N
6
S
L = L + 1
L = NX?
N
7
S
A (N, NY) = A (N, NY)/A (N, N)
A (N, N) = 1
I = 1
10
IX = I + 1
K = IX
9
DIUB = A (I, K)
J = K
A (I, J) = A (I, J) – DIUB * A (K, J)
J = J + 1
N
J = NY?
S
8

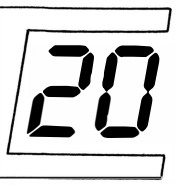

8
S
K = K + 1
9
N
K = N?
S
I = I + 1
10
N
I = NX ?
S
IMPRIME CABEÇALHO
I = 1
PRINT A (I, NY)
I = I + 1
N
I = N?
S
STOP

```
 'ECO DOS COEFICIENTES
 2    4  -1    1    1  -4    5  -7    2
SISTEMA DE ORDEM  3

PRESSIONE QUALQUER TECLA PARA CONTINUAR









 'ECO' DO VETOR CONSTANTE
 7   -9    3
PRESSIONE QUALQUER TECLA PARA CONTINUAR









********** RAIZES DO SISTEMA **********
 1    2    3
Break na 650
READY
).
```

```
10 'PRGRAMA PARA RESOLUCAO DE EQUACOES SIMULTANEAS.
20 'CAPACIDADE: ATE' 40 EQUACOES.
30 'AUTOR: FAUSTO A. DE A. BARBUTO  DATA:26/JAN/84.
40 'ADAPTADO PARA O 'BASIC' DE UM PROGRAMA DO BRIG. T. PACITTI.
50 CLS: DIM A(40,41)
60 CLEAR
100 INPUT"ORDEM DO SISTEMA";N
110 NX=N-1: NY=N+1
120 FOR I=1 TO N
125 CLS:PRINT"ENTRE COM OS COEFICIENTES"
130 FOR J=1 TO N
140 INPUT A(I,J)
150 NEXT J: NEXT I
170 FOR I=1 TO N
175 CLS:PRINT"ENTRE COM O VETOR CONSTANTE"
180 INPUT A(I,NY)
190 NEXT I
195 CLS:PRINT" 'ECO DOS COEFICIENTES"
200 FOR I=1 TO N
210 FOR J=1 TO N
220 PRINT A(I,J);" ";
230 NEXT J: NEXT I
240 PRINT:PRINT"SISTEMA DE ORDEM ";N
```

```
250 PRINT:PRINT"PRESSIONE QUALQUER TECLA PARA CONTINUAR"
255 IF INKEY$="" THEN 255
260 CLS:PRINT" 'ECO' DO VETOR CONSTANTE"
270 FOR I=1 TO N
280 PRINT A(I,NY);" ";
285 NEXT I
290 PRINT:PRINT"PRESSIONE QUALQUER TECLA PARA CONTINUAR"
295 IF INKEY$="" THEN 295
300 FOR L=1 TO NX
310 LX=L+1
320 FOR I=LX TO N
330 IF(ABS(A(L,L))-ABS(A(I,L)))<0 THEN 360 ELSE 410
360 FOR JX=L TO NY
370 TEMP=A(L,JX)
380 A(L,JX)=A(I,JX)
390 A(I,JX)=TEMP
400 NEXT JX
410 PIV=A(L,L)
420 FOR JX=L TO NY
430 A(L,JX)=A(L,JX)/PIV
435 NEXT JX
440 FOR I=LX TO N
450 M=0: DIVA=A(I,L)
460 FOR J=L TO NY
470 A(I,J)=A(I,J)-DIVA*A(L,J)
480 IF J-NY<0 THEN 490 ELSE 510
490 IF ABS(A(I,J))-1E-7<=0 THEN 530  ELSE 500
500 M=1: GOTO 530
510 IF M<=0 THEN 520 ELSE 530
520 IF ABS(A(I,J))-1E-7<=0 THEN 660 ELSE 680
530 NEXT J: NEXT I: NEXT L
540 A(N,NY)=A(N,NY)/A(N,N)
550 A(N,N)=1
560 FOR I=1 TO NX
570 IX=I+1
580 FOR K=IX TO N
590 DIUB=A(I,K)
600 FOR J=K TO NY
610 A(I,J)=A(I,J)-DIUB*A(K,J)
615 NEXT J: NEXT K: NEXT I
620 CLS: PRINT"********** RAIZES DO SISTEMA **********"
625 FOR I=1 TO N
630 PRINT A(I,NY);" ";
640 NEXT I
650 STOP
660 CLS:PRINT@448,"SOLUCAO INDETERMINADA."
670 STOP
680 CLS:PRINT@448,"SOLUCAO IMPOSSI'VEL."
690 END
```

PROGRAMA CURVE FITTING



Suponhamos agora que você tenha obtido em laboratório um "set" de dados experimentais que representem, por exemplo, a variação da velocidade em queda livre de uma partícula em relação ao tempo, ou o crescimento de uma colônia de microorganismo "in vitro". Você pode plotar esses pontos num gráfico, mas a dispersão dos pontos nem sempre permite uma precisão razoável quando se traça a curva representativa dos pontos. Mais próximo do nosso dia-a-dia, esses pontos poderiam ser o consumo mensal de energia elétrica em uma residência.

Existe uma outra maneira de se relacionar esses dados, que é encontrar uma equação que os represente. Isso pode ser feito linearizando os pontos, encontrando os parâmetros da reta obtida e, ao final, fazer a operação inversa, que é ajustar os parâmetros à equação proposta inicialmente. Pode parecer complexo demais, mas não é — e as explanações seguintes vão nos ajudar a compreender melhor esse mecanismo.

TEORIA:

UM POUCO DE MATEMÁTICA:

Seja $Y = F(X)$, $X \epsilon R$; então:

$$\text{LOG } Y = \text{LOG } F(X)$$

Como também:

$$Y^A = [\ F(X)\ ]^A,\ A = C^{TE}.$$

Se, por exemplo,

$Y = AX^M$, logaritmizando, temos:

LOG (Y) = LOG (AX^M), e aplicando as propriedades dos logaritmos, obtemos:

$$\text{LOG (Y)} = \text{LOG (A)} + \text{LOG } X^M \text{ ou:}$$
$$\text{LOG (Y)} = \text{LOG (A)} + M \text{ LOG (X)}$$

Fazendo LOG (Y) = Y′ e LOG (X) = X′, temos finalmente:

$$Y' = C + MX'$$

que é a equação de uma reta, fácil de se ajustar, usando o método dos mínimos quadrados.

O PROGRAMA:

Consiste de um programa principal que lineariza os dados recebidos e uma sub-rotina que submete os pontos linearizados ao método dos mínimos quadrados, calculando os parâmetros que são reajustados quando do retorno dos mesmos ao programa principal.

Oito equações são propostas:

Reta:	$Y = MX + A$
Potencial:	$Y = AX^M$
Exponencial:	$Y = Ae^{MX}$
Logarítmica:	$Y = M \text{ LOG (X)} + A$
Segundo grau:	$Y = M^2 X^2 + 2AMX + A^2 = (MX + A)^2$
Potencial em X:	$Y = A \cdot M^X$
Hipérbole:	$Y = 1/(MX + A)$
Hipérbole 2º grau:	$Y = 1/(MX + A)^2$

Como durante a linearização empregamos logaritmos ou raiz quadrada, é preciso ter cautela ao introduzirmos os dados. Como sabemos, não existe log de zero ou número menor que zero, assim como não existe raiz quadrada de número negativo. Consulte a tabela de utilização abaixo:

TIPO DA EQUAÇÃO:

	Reta	Potencial	Exponencial	Log.	2º gr.	Potência X	Hipérbole	Hip. 2º gr.
X	>0, <0 =0	>0	>0, <0, =0	>0	>0, <0 =0	>0, <0, =0	>0, >0, =0	>0, <0, =0
Y	>0, <0 =0	>0	>0	>0, <0 =0	⩾0	>0	≠0	>0

Por exemplo, seja o "set" abaixo:

Ponto	X	Y
A	– 3.2	8.1
B	– 0.7	6.5
C	0.0	5.1
D	1.4	4.8
E	2.5	3.7
F	3.3	1.2
G	4.1	0.0
H	5.5	– 1.6
I	6.8	– 3.0
J	7.1	– 4.1

Poderíamos usar os seguintes pontos para tentar os ajustes:

Reta: Todos.
Potencial: D, E, F.
Exponencial: A, B, C, D, E, F.
Logaritmo: D, E, F, G, H, I, J.
Segundo grau: A, B, C, D, E, F, G.
Potência em X: A, B, C, D, E, F.
Hipérboles: A, B, C, D, E, F, H, I, J.
Hipérbole 2º grau: A, B, C, D, E, F.

O fato de às vezes não podermos utilizar todos os pontos para determinado tipo de *FITTING,* não quer dizer que os pontos que foram excluídos não façam parte do ajuste calculado; apenas não podem ser usados no processo de linearização. A insistência em usar dados fora dos limites estabelecidos ocasionará uma mensagem de erro do tipo "chamada ilegal de função".

Como *"OUTPUT"*, temos os coeficientes "M", "A" e "R". Este último nos dá a indicação da aderência do modelo proposto. Quanto mais próximo de um (1) for "R", melhor o ajuste. A unidade nos traduz um ajuste perfeito.

→ 1º Exemplo:

Qual a equação que melhor representa os pontos:

X	0,5	1,Ø	1,5	2,0	3,5	4,0	4,5
Y	0,756	0,592	0,476	0,391	0,238	0,207	0,181

Inicialmente, há a solicitação do *"INPUT"* do número de pontos, que são sete (7). A seguir, introduzimos os pontos X, Y até o último par. Após executarmos todos os ajustes, obteremos as seguintes saídas:

	M	A	R
Reta	− 0,133703	0,729034	0,963518
Potência	− 0,670071	0,552174	0,980395
Exponencial	− 0,3516	0,839106	0,996286
Logaritmo	− 0,267319	0,579561	0,999415
Segundo grau	− 0,106409	0,875906	0,984226
Potência X	0,703562	0,839106	0,996286
Hipérbole	1,0552	0,612762	0,996727
Hipérbole 2º grau	0,29991	0,999747	1

O que nos leva a concluir que o melhor ajuste é o da hipérbole de potência 2. A equação escolhida fica então, assim:

$$Y = \frac{1}{(0{,}29991\,X + 0{,}999747)^2}$$

ou, aproximando,

$$Y = \frac{1}{(0{,}3X + 1)^2}$$

O pior ajuste é a reta, com R = 0,963518.

→ 2º Exemplo: Os pontos

X	0	0,5	3	7
Y	−5	0	1	3

representam (bem) uma reta?

RESPOSTA: Não, pois R = 0,815326, um valor um tanto baixo.

→ 3º Exemplo: Quais os parâmetros da potencial que passa pelos pares:

X	1	2	3
Y	2.1	16.8	56.7

RESPOSTA: Executando o programa, obtemos M = 3 e A = 2.1.

OBSERVAÇÃO: No caso de não podermos usar todos os pontos de um grupo de dados devido ao fato de alguns deles serem negativos, não é um problema insolúvel. Uma substituição de eixos pode ser empregada. Desse modo,

$$X' = X + C1 \quad e \quad Y' = Y + C2$$

onde C1 e C2 são constantes que fazem os valores de Y' e X' serem positivos. Então, se

$$Y' = F(X')$$

Teremos, por fim

$$Y + C2 = F(X + C1).$$

```
ESCOLHA O CODIGO DO MODELO QUE VOCE QUER AJUSTAR
MODELO                    EXPRESSAO                 CODIGO


RETA                      Y=M*X+A                   1
POTENCIAL                 Y=A*X[M                   2
EXPONENCIAL               Y=A*EXP(M*X)              3
LOGARITMICA               Y=M*LOG(X)+A              4
SEGUNDO GRAU              Y=(M*X+A)[2               5
POTENCIAL EM X            Y=A*M[X                   6
HIPERBOLE                 Y=1/(M*X+A)               7
HIPERBOLE GRAU 2          Y=(MX+A)[-2               8

CODIGO?




HIPERBOLE GRAU 2

M=   .29991               A=   .999747              R=  1


QUER TENTAR OUTRO AJUSTE? (S/N)
? S




RETA

M= -.133073               A=   .729034              R=  .963518


QUER TENTAR OUTRO AJUSTE? (S/N)
? N
```

```
10 'PROGRAMA CURVE FITTING
20 'AUTOR: FAUSTO A. DE A. BARBUTO
30 'DATA: 22/NOV/83
40 CLS:INPUT"NUMERO DE PARES (X,Y)";N
50 IFN(=0 THEN END
60 DIM X(N),Y(N),U(N),V(N)
70 CLS:PRINT@320,"INTRODUZA OS PARES DE PONTOS X,Y"
80 FORI=1TON
90 PRINT@448,"PAR NO. ";I
100 INPUT"X";X(I):INPUT"Y";Y(I)
110 NEXT I
120 CLS:PRINT"ESCOLHA O CODIGO DO MODELO QUE VOCE QUER AJUSTAR":PRINT@64,"MODELO
"TAB(25)"EXPRESSAO"TAB(50)"CODIGO"
130 PRINT@320,"RETA"TAB(25)"Y=M*X+A"TAB(50)"1"
140 PRINT@384,"POTENCIAL"TAB(25)"Y=A*X[M"TAB(50)"2"
150 PRINT@448,"EXPONENCIAL"TAB(25)"Y=A*EXP(M*X)"TAB(50)"3"
160 PRINT@512,"LOGARITMICA"TAB(25)"Y=M*LOG(X)+A"TAB(50)"4"
170 PRINT@576,"SEGUNDO GRAU"TAB(25)"Y=(M*X+A)[2"TAB(50)"5"
180 PRINT@640,"POTENCIAL EM X"TAB(25)"Y=A*M[X"TAB(50)"6"
190 PRINT@704,"HIPERBOLE"TAB(25)"Y=1/(M*X+A)"TAB(50)"7"
200 PRINT@768,"HIPERBOLE GRAU 2"TAB(25)"Y=(MX+A)[-2"TAB(50)"8"
210 PRINT:INPUT"CODIGO";CD
215 ON CD GOTO 250, 300, 400, 480, 550, 650, 700, 800
220 '
230 'MODELO RETA
240 '
250 FORI=1TON
260 U(I)=X(I):V(I)=Y(I)
270 NEXT I:GOSUB890
280 PRINT@320,"RETA":PRINT@448,"M= ";MTAB(25)"A= ";CTAB(50)"R= ";R
290 GOTO1010
300 '
310 'POTENCIAL
320 '
330 FOR I=1TON
340 U(I)=LOG(X(I)):V(I)=LOG(Y(I))
350 NEXT I:GOSUB890
360 PRINT@320,"POTENCIA":PRINT@448,"M= ";MTAB(25)"A= ";EXP(C)TAB(50)"R= ";R
370 GOTO1010
380 '
390 'EXPONENCIAL
400 '
410 FOR I=1TON
420 U(I)=X(I):V(I)=LOG(Y(I))
430 NEXT I:GOSUB890
440 PRINT@320,"EXPONENCIAL":PRINT@448,"M= ";MTAB(25)"A= ";EXP(C)TAB(50)"R= ";R
450 GOTO1010
460 '
470 'LOGARITMICO
480 '
490 FOR I=1TON
500 U(I)=LOG(X(I)):V(I)=Y(I)
510 NEXT I:GOSUB890
520 PRINT@320,"LOGARITMICO":PRINT@448,"M= ";MTAB(25)"A= ";CTAB(50)"R= ";R
530 GOTO1010
540 '
550 'SEGUNDO GRAU
560 '
570 FOR I=1TON
580 U(I)=X(I):V(I)=Y(I)[.5
590 NEXT I: GOSUB890
600 PRINT@320,"SEGUNDO GRAU":PRINT@448,"M= ";MTAB(25)"A= ";CTAB(50)"R= ";R
610 GOTO1010
620 '
630 'POTENCIAL EM X
640 '
650 FOR I=1TON
660 U(I)=X(I):V(I)=LOG(Y(I))
670 NEXT I:GOSUB890
680 PRINT@320,"POTENCIAL EM X":PRINT@448,"M= ";EXP(M)TAB(25)"A= ";EXP(C)TAB(50)"
R= ";R
```

```
690 GOTO1010
700 '
710 'HIPERBOLE
720 '
730 FOR I=1TON
740 U(I)=X(I):V(I)=1/Y(I)
750 NEXT I:GOSUB890
760 PRINT@320,"HIPERBOLE":PRINT@448,"M= ";MTAB(25)"A= ";CTAB(50)"R= ";R
770 GOTO1010
780 '
790 'HIPERBOLE GRAU 2
800 '
810 FOR I=1TON
820 U(I)=X(I):V(I)=1/(Y(I)[.5)
830 NEXT I:GOSUB890
840 PRINT@320,"HIPERBOLE GRAU 2":PRINT@448,"M= ";MTAB(25)"A= ";CTAB(50)"R= ";R
850 GOTO1010
860 '
870 'SUBROTINA MINIQUAD
880 '
890 CLS:S1=0:S2=0:S3=0:S4=0:S5=0
900 FOR I=1TON
910 S1=S1+U(I)
920 S2=S2+V(I)
930 S3=S3+U(I)[2
940 S4=S4+V(I)[2
950 S5=S5+U(I)*V(I)
960 NEXT I
970 M=(S5-S1*S2/N)/(S3-S1[2/N)
980 C=(S2-M*S1)/N
990 R=ABS(M*SQR((S3-S1[2/N)/(S4-S2[2/N)))
1000 RETURN
1005 'NOVO AJUSTE.
1010 PRINT@704,"QUER TENTAR OUTRO AJUSTE? (S/N)"
1020 INPUT R$
1030 IF R$="S" THEN 120
1035 'NOVOS PONTOS.
1040 CLS:PRINT@704,"QUER AJUSTAR OUTROS PONTOS? (S/N)"
1050 INPUT S$
1060 IF S$="S" THEN RUN
1070 CLS:PRINT@475,"F I M"
1080 END
```

CÁLCULO DE INTEGRAIS DEFINIDAS – MÉTODO DOS RETÂNGULOS

21

Quando nos deparamos com a necessidade de resolvermos uma integral definida entre dois pontos – durante o desenvolvimento de uma questão de engenharia ou estatística, por exemplo – os métodos de integração (fórmulas, tabelas etc) sempre nos conduzem a um resultado, desde que a equação seja até certo ponto simples de integrar. Às vezes, é preciso "suar" um pouco, mas sempre "chegamos lá".

O mesmo não acontece se a função não é integrável. (Sabemos que toda função é diferenciável, mas existem algumas impossíveis de serem integradas). Assim, não existe F (X) tal que:

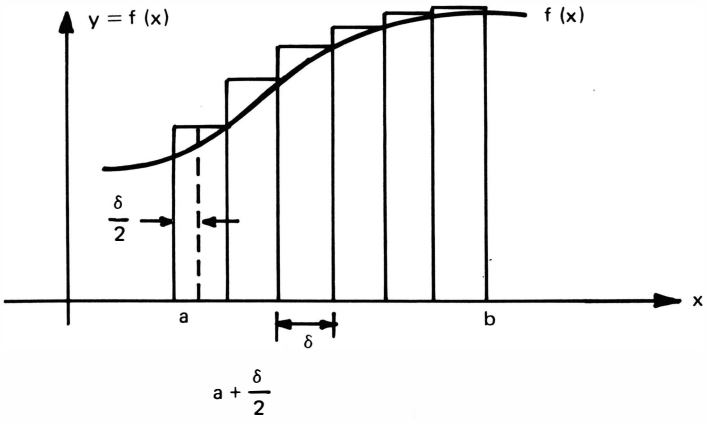
$$F(X) = \int_0^x e^{-x^2}\, dx$$

Isso ocorre bastante em estatística (veremos isso mais adiante), em engenharia química (transferência de calor) e engenharia de petróleo (avaliação de jazidas). No caso da função acima, existem tabelas, o que no entanto não ocorre com a integral

$$I = \int_c^b \frac{\text{sen } x}{x}\, dx$$

que é por alguns conhecida como "senóide amortecida".

TEORIA: Para resolver toda e qualquer integral de função, propomos que se a subdivida em "N" intervalos iguais, assim:



A cada intervalo está associado um retângulo. Como a integral é muitas vezes definida como a "área sob a curva", a soma da área dos "N" retângulos compreendidos entre "a" e "b" nos forneceria a integral aproximada da função. É claro que, quanto maior "N", mais próximo do valor real estaria o nosso cálculo. A área do primeiro retângulo é:

$$A_1 = \delta \cdot f\left(a + \frac{\delta}{2}\right)$$

A área do segundo e do terceiro retângulos, similarmente, ficaria:

$$A_2 = \delta \cdot f\left(a + \frac{3\delta}{2}\right) \quad A_3 = \delta \cdot f\left(a + \frac{5\delta}{2}\right)$$

Generalizando, vem:

$$A_i = \delta \cdot f\left(a + (2i - 1)\ \frac{\delta}{2}\right)$$

A integral fica:

$$I = \sum_{i=1}^{N} A_i = \sum_{i=1}^{N} \delta \cdot f\left(a + (2i - 1)\ \frac{\delta}{2}\right) = \delta \cdot \sum_{i=1}^{N} f\left(a + (2i - 1)\ \frac{\delta}{2}\right)$$

onde:

$$\delta = \frac{b - a}{N}\ ,\quad b > a \qquad (1)$$

De acordo com (1),

$$\lim_{N \to \infty} \frac{b-a}{N} = 0$$

Logo, quando N cresce diminui a largura dos retângulos, e mais precisa fica a integral, conseqüentemente. Isto fica mais fácil de visualizar se mostrarmos em destaque dois retângulos genéricos. Ver Fig. 1.

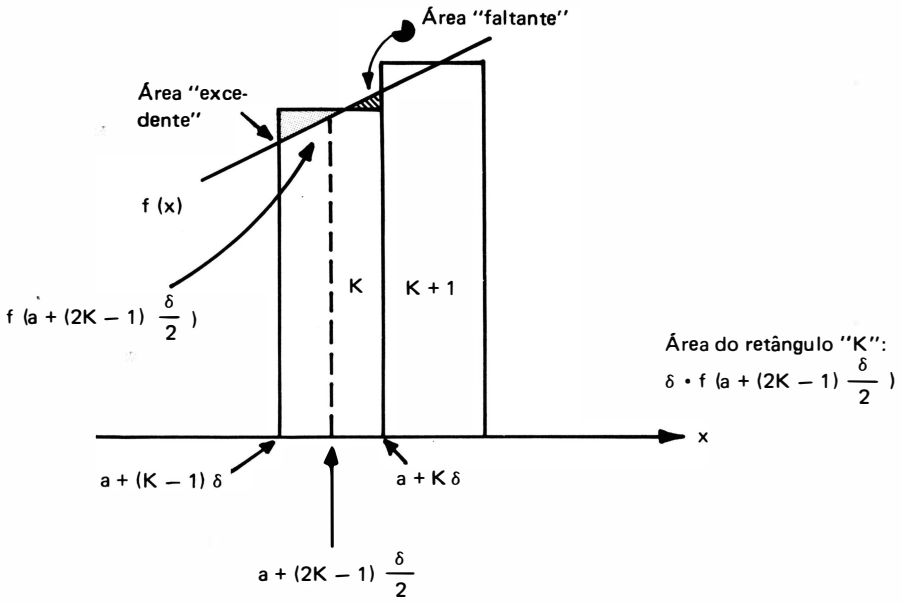


FIG. 1. Retângulos genéricos 'K' e 'K + 1'.

Com hachura, é mostrado um pedaço da área sob a curva que não fica incluída nem na área do retângulo "K" nem na do "K + 1". Esta área "faltante" é em parte compensada pela área "excedente", marcada ao lado. Quando $N \to \infty$, $[a + (K - 1)\,\delta]$ e $[a + K\delta]$ tendem para $[a + (2K - 1)\,\frac{\delta}{2}]$, e as áreas "faltante" e "excedente" de cada retângulo tendem a se anular, tornando exata a integral. No programa *BASIC* que veremos a seguir, não podemos fazer $N = \infty$, mas podemos usar $N = 1000$, por exemplo, que nos dá um valor muito bom (erro menor que 0,5%) na maioria dos casos. Com $N = 1000$, o tempo de processamento é elevado (mais de 1 1/2 min.), mas o resultado é compensador. Podemos usar um "N" menor, de acordo com as necessidades de tempo de processamento e precisão que exigimos. Em tempo: este método também é excelente para máquinas de calcular programáveis, principalmente aquelas com poucos passos.

USANDO O PROGRAMA:

O quadro seguinte nos mostra os dados de entrada

NOME DA VARIÁVEL	*DIMENSIONAMENTO*	*COMENTÁRIOS*
XØ	Não-dimensionada	Limite inferior (a)
XN	Não-dimensionada	Limite superior (b)
N	Não-dimensionada	Número de intervalos (N)

Na linha 115, colocamos a função Y = F (X), por exemplo:

→ 1º Exemplo: 115 Y = X [2

que, no *BASIC D-8000* significa $Y = X^2$. O limite inferior é zero e o superior é dois. Trata-se portanto de

$$I = \int_0^2 x^2\,dx = 2{,}66666\ldots$$

Iniciamos com

RUN

<NEWLINE> (ou <RETURN>)

Temos:

Limite inferior?
Ø <NEWLINE>
Limite superior?
2 <NEWLINE>
Número de intervalos?
1ØØ <NEWLINE>

Após alguns instantes, temos:

Valor da integral: 2.5874Ø

O erro, portanto, é de

$$\frac{2.58740 - 2.66667}{2.66667} \cdot 100 = -2{,}97262\%$$

Usando N = 800, temos:

Valor da integral: 2.65868

O erro se torna:

$$\frac{2.65868 - 2.66667}{2.66667} \cdot 100 = -0,29962\%$$

A despeito do tempo de processamento aumentar bastante, aumentamos também a precisão do resultado.

→ 2º Exemplo: Integrar

$$\int_0^1 e^{-x^2}\,dx, \text{ com } N = 1000$$

Então, XØ = 0, XN = 1 e iN = 1000. A linha 115 fica

115 Y = EXP (— X * X)

(Por que fazemos assim, e não Y = EXP (— X [2)?)

Obtemos como resposta:

Valor da integral: Ø.746491

O valor (tabelado) até seis casas decimais é 0,746842. O erro, então, é:

$$= \frac{0,746491 - 0,746842}{0,746842} * 100 = -0,0469979\%,$$

que é um valor de boa precisão. Se usarmos N = 200, o erro aumenta para — 0,22548% — e que, mesmo assim, é um bom resultado.

Existem métodos de integração mais apurados e com tempo de processamento muito menor, como o de Simpson (vamos vê-lo adiante), Romberg etc. O método que acabamos de apresentar não deixa de ser interessante do ponto de vista didático, uma vez que ilustra um artifício banal, mas eficiente e estimula o nosso *FEELING,* no campo do cálculo numérico, cujo uso os técnicos modernos não podem prescindir.

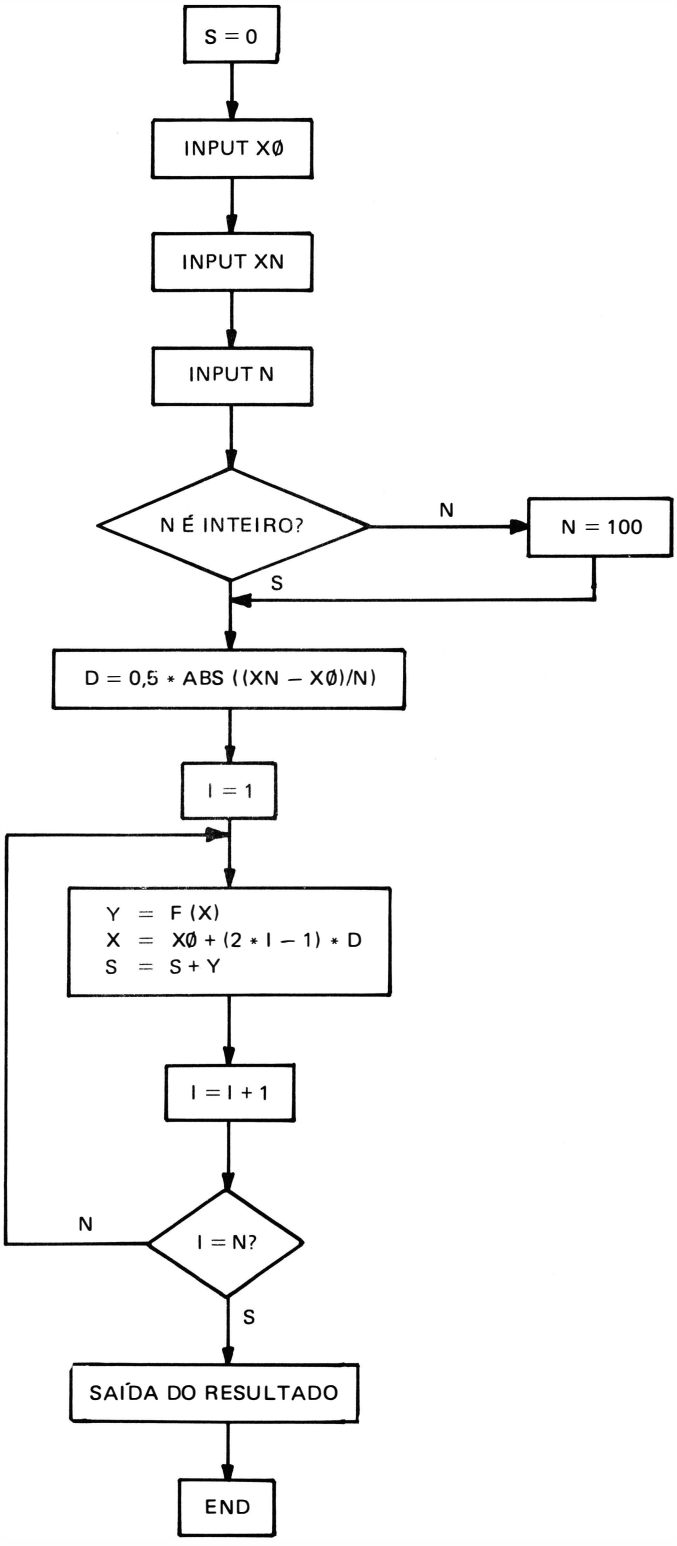


FIG. 2. Diagrama de blocos do programa integrador.

```
LIMITE INFERIOR? 0
LIMITE SUPERIOR? 2
NUMERO DE INTERVALOS? 100


VALOR DA INTEGRAL:  2.5874
READY
>
```

```
10 'PROGRAMA INTEGRADOR  -  ME'TODO DOS RETANGULOS.
20 'AUTOR: FAUSTO A. DE A. BARBUTO   ***   DATA: 23/DEZ/83
30 CLS: S=0
40 INPUT"LIMITE INFERIOR";X0
50 INPUT"LIMITE SUPERIOR";XN
60 INPUT"NUMERO DE INTERVALOS";N
70 IF(N-INT(N))<>0 THEN N=100
80 D=.5*ABS((XN-X0)/N)
100 'CALCULO DA ALTURA DOS RETANGULOS
110 FOR I=1 TO N
115 Y=X^2
120 X=X0+(2*I-1)*D
130 S=S+Y
140 NEXT I
150 'IMPRESSAO DOS RESULTADOS
160 PRINT@320,"VALOR DA INTEGRAL: ";S*D*2
170 END
```

CÁLCULO DE INTEGRAIS POR SIMPSON

22

Vejamos agora um outro método para calcular integrais definidas num intervalo de função contínua e diferenciável, chamado método de Simpson.

Seja o seguinte intervalo (X_0, X_N) a ser integrado. Vamos subdividi-lo em "N" intervalos de igual largura "H", de tal forma que:

$$H = ABS\left(\frac{X_N - X_0}{N}\right)$$ "N" inteiro e par.

Adiante veremos por que isso é feito.

A precisão do valor da integral a ser calculada depende intrinsecamente do valor de "N". Quanto maior este for, mais apurada a resposta. Para intervalos não muito grandes, N = 32 já é suficiente. O autor geralmente usa $N = 2^K$ (K = 3, 4, 5, 6, 7, . . .), mas nada nos impede que façamos N = 40, por exemplo. Este método é mais preciso que o visto anteriormente.

USANDO O PROGRAMA:

Em primeiro lugar, falemos sobre as variáveis do programa

NOME	*TIPO/DIMENSIONAMENTO*	*COMENTÁRIOS*
N	INTEIRA/NÃO-DIMENSIONADA	NÚMERO DE INTERVALOS
X (I)	REAL/DIMENSIONADA	ABSCISSAS
Y (I)	REAL/DIMENSIONADA	VALORES DA FUNÇÃO
S1	REAL/NÃO-DIMENSIONADA	SOMATÓRIO DE 4Y (I)
S2	REAL/NÃO-DIMENSIONADA	SOMATÓRIO DE 2Y (I)
H	REAL/NÃO-DIMENSIONADA	LARGURA DOS INTERVALOS
ÁREA	REAL/NÃO-DIMENSIONADA	ÁREA TOTAL

O programa só solicita como dados de entrada N, X (0) e X (N). Na linha 220, colocamos a função a ser integrada como Y (I). Para $Y = X^2$, ficamos com:

```
28Ø Y (I) = X (I) [ 2
```

ou, melhor ainda:

```
28Ø Y (I) = X (I) * X (I)
```

(O símbolo "[" no *BASIC D-8000* significa exponenciação; outros micros usam "↑").

Poderíamos usar, também, a declaração *DEFFN* para definir a função, no início do programa. Algumas versões nacionais do *TRS-80* — como a do autor, por exemplo — não possuem essa declaração; daí a opção por um exemplo genérico.

É lógico que ninguém em sã consciência vai utilizar Simpson para integrar $F(X) = X^2$, que é uma função trivial e solúvel por métodos usuais. Vamos resolver um exemplo mais adiante usando essa função, mas com caráter exclusivamente didático.

Para os que possuem *DEFFN,* poder-se-ia fazer:

```
25 DEFFN Y (X) = X [ 2
```

Isso acarretaria em algumas outras mudanças no programa. Mudanças simples, sobre as quais não discorreremos aqui.

Se por um lado $F(X) = X^2$ é fácil de integrar, já não o podemos dizer de $F(X) = e^{-x^2}$. Sabemos que toda a função possui derivada, mas nem todas são integráveis, e e^{-x^2} é uma dessas. Em outras palavras, não existe função tal que:

$$F(x) = \int e^{-x^2}\, dx,$$

ou

$$\frac{dF}{dx} = e^{-x^2}$$

No entanto, há um valor numérico para

$$I = \int_a^b e^{-x^2}\, dx$$

(Já falamos antes sobre isso, no programa "método dos retângulos"), que pode ser calculado numericamente por Simpson. Outras funções, como a função erro e erro complementar também são não integráveis. Falaremos delas depois.

Vamos aos exemplos de demonstração:

→ Exemplo 1:

Seja $F(x) = e^{-x^2}$. Determinar $\int_0^1 F(x)\,dx$, usando 32 intervalos.

Sigamos as ordens dadas pela tabela abaixo. A linha 28Ø ficaria como

28ØY (I) = EXP (– X (I) * X (I)).

COMANDO	*MENSAGEM*	*INTRODUZIR*	*COMENTÁRIOS*
<RUN>	–	–	–
<NEWLINE>	NÚMERO DE DIVISÕES?	32	N = 32
<NEWLINE>	X (Ø)?	Ø	LIMITE INFERIOR
<NEWLINE>	X (N)?	1	LIMITE SUPERIOR
<NEWLINE>			

Após o último *NEWLINE,* obtemos: (após alguns segundos)

Valor da Integral: Ø.746824

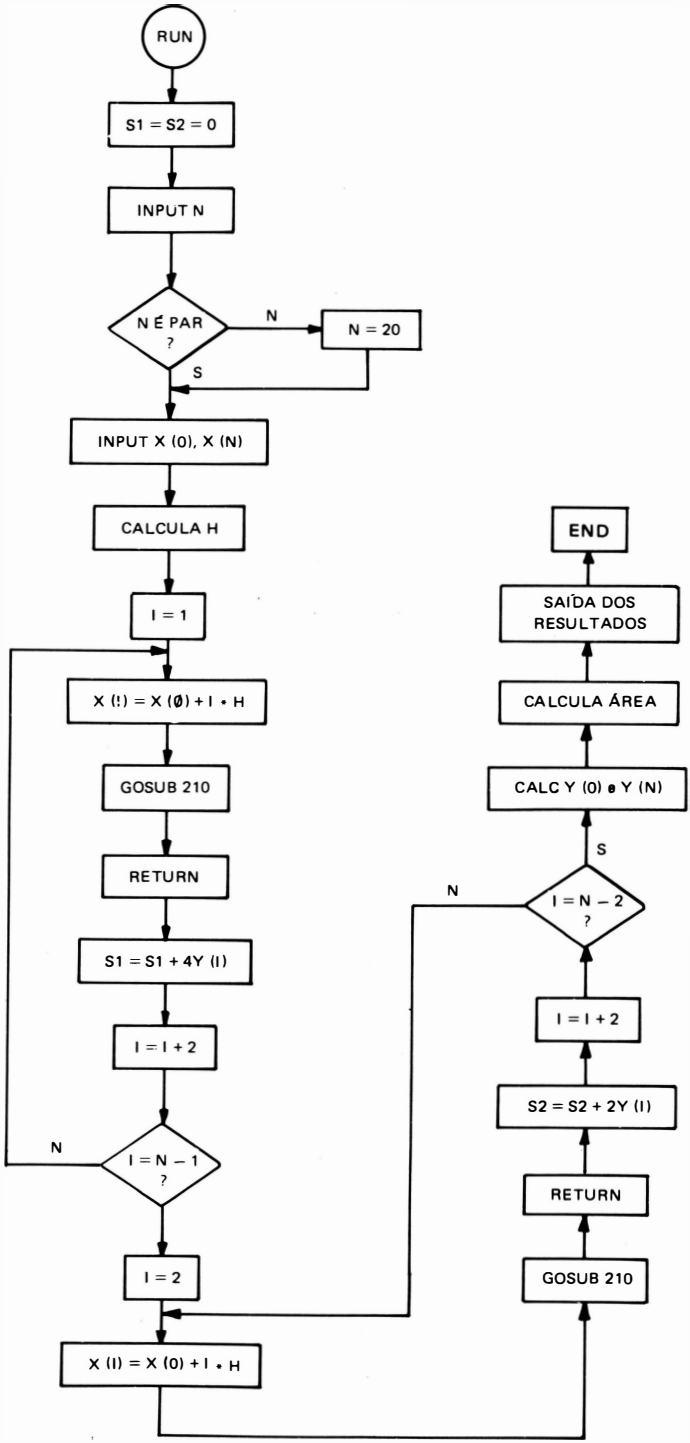
→ Exemplo 2: Integrar $F(X) = X^2$ no intervalo [0, 2] com N = 32.

Similarmente, introduzimos 32, 0 e 2, obtendo:

Valor da Integral: 2.66667

O valor exato (pelos métodos do cálculo) é a dízima 2,66666 . . .

FLUXOGRAMA DE SIMPSON:



CONCLUSÕES: Simpson é um método simples, rápido, eficiente e fácil de usar. Seu tempo de processamento é muito menor que os outros, notadamente Gauss. Compare a exatidão dos resultados por Gauss com N = 32 e Simpson. Compare também os tempos de execução dos dois métodos.

TEORIA:

Para aqueles que se interessam por matemática e querem "ir um pouco mais além", vejamos o embasamento teórico do método Simpson. Seja o seguinte intervalo da função F (X), $[X_0, X_N]$.

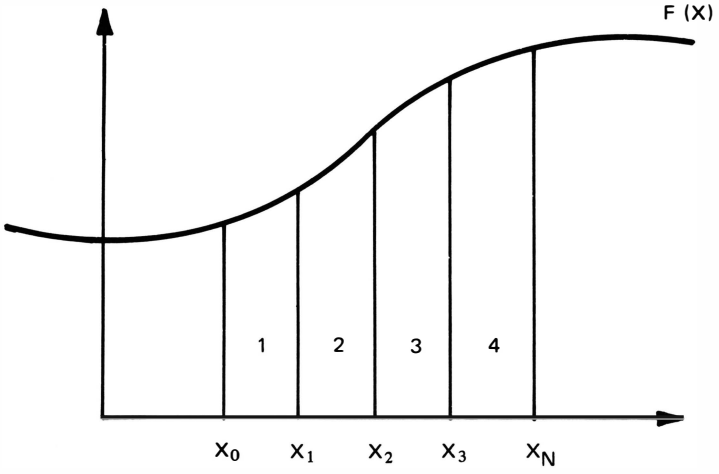


FIG. 1. F (X) e suas subdivisões.

O intervalo foi subdivido em 4 (quatro) intervalos de largura igual a:

$$H = \left| \frac{X_N - X_0}{4} \right|$$

No caso acima, "N" é igual a quatro. Segundo Simpson, o algoritmo que deve ser usado para calcular o valor da integral é:

$$\int_{X_C}^{X_N} F(X)\,dx = \frac{H}{3}\left(F(X_0) + F(X_N) + \sum_{\substack{i=1 \\ (i\ \text{ímpar})}}^{N-1} F(X_i) + \sum_{\substack{j=2 \\ (j\ \text{par})}}^{N-2} F(X_j)\right)$$

Se aplicarmos a fórmula acima, a integral da Fig. 1 será:

$$I = \frac{(X_N - X_0)}{12}\left\{F(X_0) + F(X_N) + [F(X_1) + F(X_3)] + F(X_2)\right\}.$$

```
NUMERO DE SUBDIVISOES? 32
X(0)? 0
X(N)? 1.
```

```
VALOR DA INTEGRAL:  .746824
READY
).
```

```
10 'PROGRAMA INTEGRADOR  -  ME'TODO DE SIMPSON.
20 'AUTOR: FAUSTO A. DE A. BARBUTO.     DATA:23/DEZ/83.
30 CLS
40 IF(N/2-INT(N/2))<>0 THEN N=32
50 INPUT"NUMERO DE SUBDIVISOES";N
60 IF(N/2-INT(N/2))<>0 THEN N=20
70 DIM X(N), Y(N)
80 INPUT"X(0)";X(0)
90 INPUT"X(N)";X(N)
100 H=ABS((X(N)-X(0))/N)
110 FOR I=1 TO N-1 STEP 2
120 X(I)=X(0)+I*H
130 GOSUB270
140 S1=S1+4*Y(I)
150 NEXT I
160 FOR I=2 TO N-2 STEP 2
170 X(I)=X(0)+I*H
180 GOSUB270
190 S2=S2+2*Y(I)
200 NEXT I
210 I=0: GOSUB270
220 I=N: GOSUB270
230 AREA=H*(Y(0)+Y(N)+S1+S2)/3
240 CLS
250 PRINT@320,"VALOR DA INTEGRAL: ";AREA
260 END
270 'CHAMADA DA FUNCAO;
280 Y(I)=X(I)[2
290 RETURN
```

VOLUME DE FIGURAS DE REVOLUÇÃO

23

Uma figura de revolução (também chamada sólido de revolução) é, a priori, uma figura tridimensional obtida fazendo-se girar em volta de um eixo uma segunda figura, geralmente bidimensional. A figura gerada, deste modo, é simétrica em relação ao eixo gerador. Veja a figura abaixo:

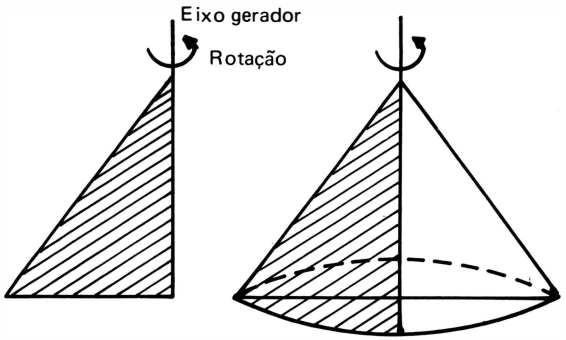


FIG. 1. Rotação de um triângulo, gerando um cone.

Examinando as figuras acima, vemos que um triângulo gerou, sob revolução em torno de um eixo coincidente com um de seus catetos, um cone. Um quadrado ou um retângulo gerariam um cilindro. Existem figuras mais complexas, como o "tronco de cone" ou a "trombeta de Gabriel". Como calcular, analiticamente, seu volume?

Vamos supor que, na Fig. 1, o eixo gerador seja o eixo dos X (abscissas) e que o outro cateto jaz sobre o eixo dos Y (ordenadas) de acordo com a Fig. 2:

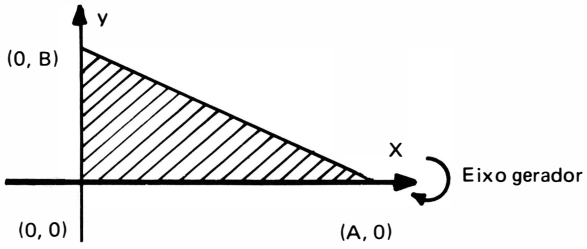


FIG. 2. O triângulo e o eixo gerador.

Assim definimos o triângulo pelos seus vértices: (0, 0); (A, 0); (0, B). Tal equação é:

$$f(X) = y = -\frac{B}{A} X + B$$

(Trata-se da equação analítica de uma reta).

Vamos imaginar também um elemento infinitesimal de área do triângulo de modo a podermos analisar microscopicamente como ele contribui na geração da figura de revolução. Veja a Fig. 3:

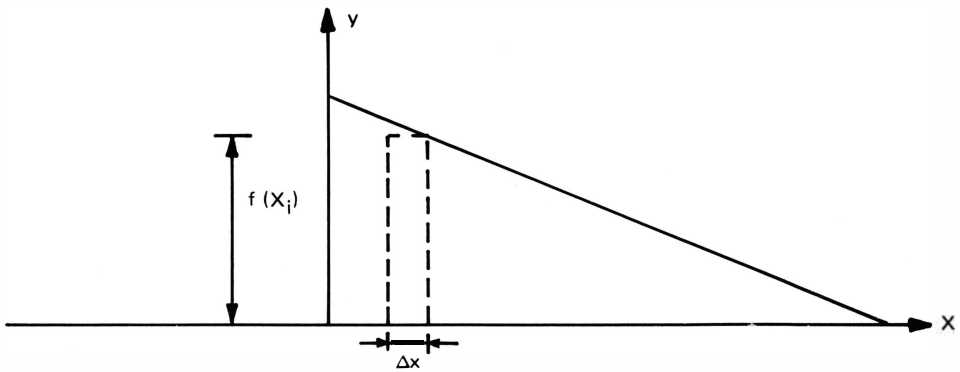


FIG. 3. O retângulo infinitesimal.

Este elemento infinitesimal é um retângulo de base Δx (delta x) e altura $f(X_i)$. Vamos supor agora que se lhe imprima uma rotação em torno do eixo X:

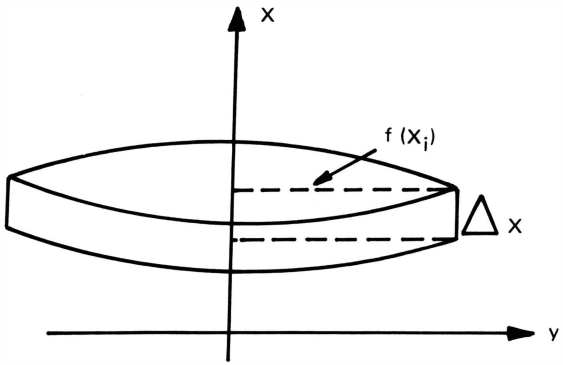


FIG. 4. Rotação do retângulo infinitesimal.

E então notamos que a rotação deste retângulo gerou como sólido de revolução um cilindro. (Já havíamos dito isso anteriormente). O volume desse cilindro infinitesimal é:

$$V_i = \pi R^2 \, h$$

onde $R = f(X_i)$ e $h = \Delta X$. Assim, temos:

$$V_i = \pi \, (f(X_i))^2 \, \Delta X$$

O volume total do cone gerado pela revolução do triângulo será:

$$V_T = \Sigma V_i = \sum_{i=1}^{N} \pi \, [f(X_i)]^2 \, \Delta X$$

que é o somatório de todos os cilindros infinitesimais. Passando o limite e explicitando Pi, temos finalmente:

$$V_T = \pi \int_{X_1}^{X_2} [f(X)]^2 \, dX$$

No caso do nosso triângulo, $X_1 = 0$, $X_2 = A$ e $f(X) = -\frac{B}{A} X + B$; por fim, obtemos:

$$V = \pi \int_0^A \left(-\frac{Bx}{A} + B \right)^2 dx$$

Desenvolvendo:

$$V = \pi \int_0^A \left(\frac{B^2}{A^2} X^2 - 2 \frac{B^2}{A} X + B^2 \right) dX$$

Se usarmos Simpson (com pequenas alterações), fica fácil calcular o volume de qualquer figura de revolução, mesmo aquelas para as quais não existem fórmulas tabeladas. As alterações são simples; é só alterar a linha 250, que ficaria assim (no programa Simpson):

250 PRINT @ 320, "VOLUME DA FIGURA DE REVOLUÇÃO:"; 3.14159 * ÁREA

Na linha 280, escreveríamos a função, tomando o cuidado de elevar ao quadrado. Se, no exemplo do triângulo, B = 4 e A = 8, ter-se-ia:

$$y = -\frac{4}{8} X + 4 = -\frac{1}{2} X + 4$$

e, na linha 280;

280 Y (I) = (– • 5 * X (I) + 4) [2

→ Exemplo 1: Calcular o volume de revolução gerado por $f(X) = \frac{1}{X}$ no intervalo [1, 10] com N = 64.

Os *INPUTS* que requer o programa são os mesmos, obviamente, do programa Simpson.

A figura gerada é vulgarmente chamada "trombeta de Gabriel".

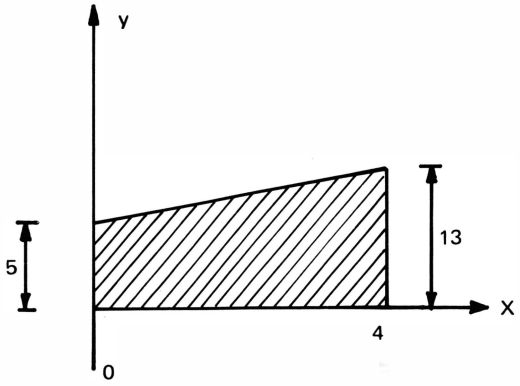
Fazemos, após "RUN", NEWLINE, 64, 1 e 10, que são os dados que o *SOFT* processa. A resposta:

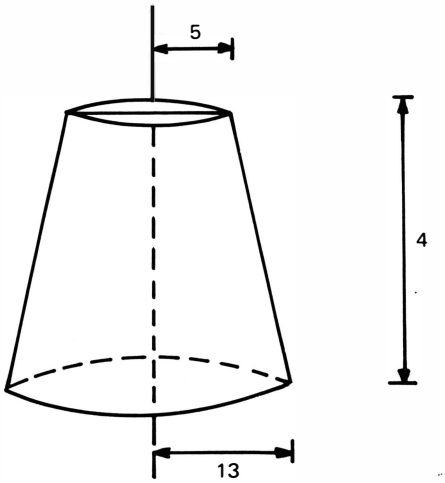
Volume da figura de revolução: 2.82759

A resposta é em unidades de volume: 2,82759 cm^3, por exemplo, se nossa unidade é o centímetro. Para este exemplo, a linha 280 ficaria assim:

280 Y (I) = 1/X(I)/X (I)

→ Exemplo 2: Calcular o volume do seguinte *tronco de cone* (gerado pela figura abaixo):





A equação que descreve a aresta é:

$$y - 5 = \frac{13 - 5}{4 - 0} (X - 0)$$

$$y = 2X + 5.$$

Na linha 28Ø,

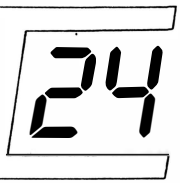
28Ø Y (I) = (2 * X (I) + 5) [2

Com N = 24, a resposta é:

Volume da figura de revolução: 1Ø84.9

Poderíamos também usar o método dos retângulos, mas Simpson nos fornece maior precisão. "A LISTAGEM É A MESMA DO PROGRAMA SIMPSON, COM AS ALTERAÇÕES JÁ DESCRITAS."

FUNÇÃO ERRO E ERRO COMPLEMENTAR



Falamos anteriormente sobre Simpson como método eficiente e preciso. Vamos agora aplicá-lo mais uma vez, para calcular duas funções muito importantes e que encontram amplo uso na engenharia nuclear, química, de petróleo e metalúrgica (para citar apenas algumas), que são as funções erro, definida como

$$ERF\ (A) = \frac{2}{\sqrt{\pi}} \int_0^A e^{-x^2}\, dx$$

e erro complementar, que tem a forma:

$$ERFC\ (A) = \frac{2}{\sqrt{\pi}} \int_A^{\infty} e^{-x^2}\, dx$$

onde "A" é o seu argumento. Falaremos um pouco mais sobre teoria no final do capítulo.

As funções ERF (X) e ERFC (X) são muito usadas em difusão gasosa (enriquecimento do urânio), transmissão de calor e avaliação de jazidas de petróleo.

Para o cálculo destas funções, usamos um programa Simpson, modificado especialmente para isso.

USANDO O PROGRAMA:

→ Exemplo 1: Calcular ERF (X) e ERFC (X) para X = 0,7 e N = 32.

Já que o programa é uma adaptação de Simpson, ele solicita os mesmos dados de entrada daquele programa, à exceção de X (0) que para o programa é de início assumido como zero. Usaremos 32 subdivisões e X (N) = 0,7. A resposta – no *DISPLAY* – é:

VALOR DA FUNÇÃO ERRO: Ø.677802

VALOR DA FUNÇÃO ERRO COMPLEMENTAR: Ø.322198

→ Exemplo 2: Calcular ERF (X) e ERFC (X) para X = 0,3 e N = 64.

RESPOSTA:

ERF (0, 3) = Ø,328627

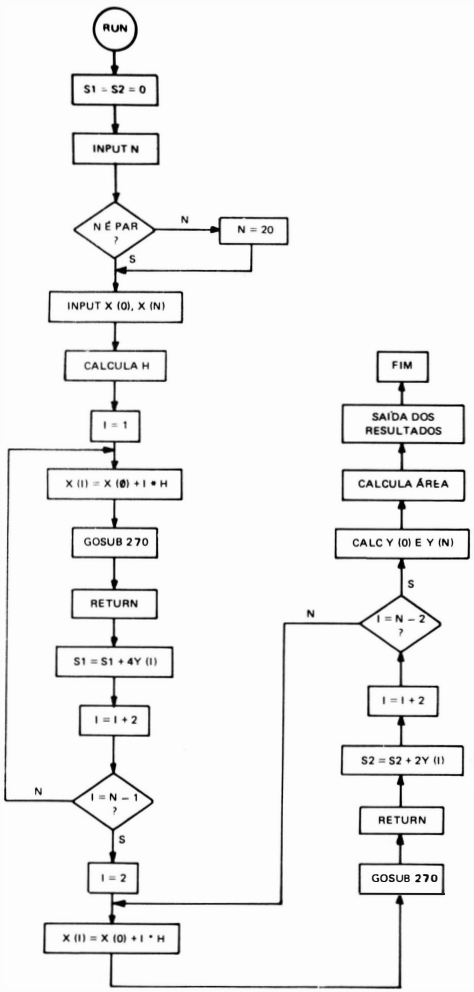
ERFC (0, 3) = Ø,671373

→ Exemplo 3: Idem para X = 2 e N = 64

RESP: ERF (2,0) = 0,995323 e ERFC (2,0) = $4{,}6768 * 10^{-3}$

(É fácil para o leitor perceber que ERF (X) + ERFC (X) = 1. Veremos isso a seguir.)

FLUXOGRAMA:

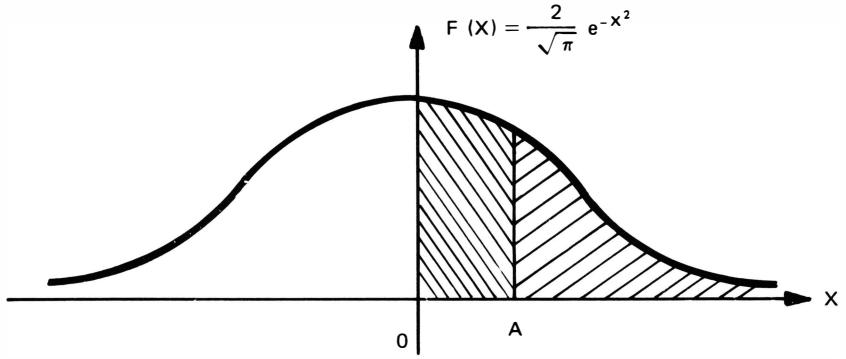


TEORIA:

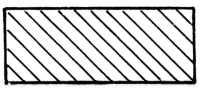
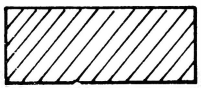
Conforme vimos, a função erro e erro complementar são definidas por:

$$ERF(X) = \frac{2}{\sqrt{\pi}}\int_0^X e^{-x^2}\,dx; \qquad ERFC(X) = \frac{2}{\sqrt{\pi}}\int_X^\infty e^{-x^2}\,dx$$

Como já dissemos, e^{-x^2} não pode ser integrada pelos métodos normais de cálculo. No caso de ERFC (X), o limite superior é infinito — um complicador a mais para nós, pois nem com Simpson é possível calcular uma integral com limite superior infinito, embora possa se usar um L. S. "muito grande", com 1000, 5000 etc. Existe, no entanto, uma maneira mais simples de calcular ERFC (X) através de ERF (X). Representemos graficamente $F(X) = \frac{2}{\sqrt{\pi}} e^{-x^2}$:



As áreas hachuradas representam:

 : ERF (A)  : ERFC (A)

Pode-se provar matematicamente que:

$$ERF(\infty) = \frac{2}{\sqrt{\pi}}\int_0^\infty e^{-x^2}\,dx = 1$$

Não é do escopo do livro demonstrar isso; no entanto, o leitor pode experimentar usar o programa usando um limite superior igual a oito (8) e N = 256 e verificar que o valor realmente

tende para a unidade. Oito não é infinito; mas a função $e^{x^{-2}}$ decresce exponencialmente com o quadrado de X. Destarte,

$$F(8) = e^{-8^2} = \frac{1}{e^{64}} = 1{,}604 * 10^{-28}$$

Experimente com outros números maiores do que oito, com o cuidado para não haver *"OVERFLOW"*.

Agora façamos uma igualdade:

$$\frac{2}{\sqrt{\pi}}\int_0^{\infty} e^{-x^2}\,dx = \frac{2}{\sqrt{\pi}}\int_0^{A} e^{-x^2}\,dx + \frac{2}{\sqrt{\pi}}\int_A^{\infty} e^{-x^2}\,dx$$

O primeiro termo é igual a um; o segundo é ERF (A) e o terceiro é ERFC (A). Logo,

$$1 = ERF(A) + ERFC(A)$$

ou então:

$$ERFC(A) = 1 - ERF(A)$$

e é assim que calculamos ERFC (A): através de ERF (A). Veja no programa as linhas 250 e 255 e procure entender o seu mecanismo.

```
NUMERO DE SUBDIVISOES? 32
X(N)? .7.



VALOR DA FUNCAO ERRO:  .677802

VALOR DA FUNCAO ERRO COMPLEMENTAR:  .322198
READY
).
```

```
10 'PROGRAMA PARA CALCULO DA FUNCAO ERRO E ERRO COMPLEMENTAR.
20 'AUTOR: FAUSTO A. DE A. BARBUTO.    DATA:23/DEZ/83
30 CLS
40 S1=0: S2=0
50 INPUT"NUMERO DE SUBDIVISOES";N
60 IF(N/2-INT(N/2))<>0 THEN N=20
70 DIM X(N), Y(N)
80 X(0)=0
90 INPUT"X(N)";X(N)
100 H=ABS((X(N)-X(0))/N)
110 FOR I=1 TO N-1 STEP 2
120 X(I)=X(0)+I*H
130 GOSUB270
140 S1=S1+4*Y(I)
150 NEXT I
160 FOR I=2 TO N-2 STEP 2
170 X(I)=X(0)+I*H
180 GOSUB270
190 S2=S2+2*Y(I)
200 NEXT I
210 I=0: GOSUB270
220 I=N: GOSUB270
230 AREA=H*(Y(0)+Y(N)+S1+S2)/3
240 CLS
250 PRINT@320,"VALOR DA FUNCAO ERRO: ";AREA
255 PRINT@448,"VALOR DA FUNCAO ERRO COMPLEMENTAR: ";1-AREA
260 END
270 'CHAMADA DA FUNCAO
280 Y(I)=1.12838*EXP(-X(I)*X(I))
290 RETURN
```

DISTRIBUIÇÃO NORMAL DE PROBABILIDADES

25

A função distribuição normal de probabilidades é definida como:

$$P(X) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{X} e^{-\frac{x^2}{2}} \, dx$$

Podemos notar, na fórmula acima, que o limite inferior é igual a $-\infty$ (menos infinito) e que do jeito que se nos foi apresentada, fica difícil (se não impossível) calcular o valor de $P(X)$, $\forall x$, por um método qualquer (Simpson, retângulos, trapézio etc.). Para contornar o problema, usamos um polinômio que aproxima a função com erro no sétimo algarismo significativo, e que tem a seguinte forma:

$$P(x) \cong 1 - \frac{1}{\sqrt{2\pi}} e^{-x^2/2} [(a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5)]$$

onde:

$$t = \frac{1}{1 + px}$$

e:

$a_1 = 0{,}31938$ $\quad a_2 = -0{,}35656$ $\quad a_3 = 1{,}78148$

$a_4 = -1{,}82126$ $\quad a_5 = 1{,}33027$ $\quad p = 0{,}23164$

Na apresentação, oferecemos ao usuário a possibilidade de optar por um modelo generalizado ou não-generalizado. Neste último caso, temos:

$$X' = \frac{X - \mu}{\sigma}$$

onde "μ" é a média da amostragem e "σ", o desvio-padrão.

No programa não existem variáveis indexadas. Todas são reais, simples precisão. "XA" é uma variável auxiliar.

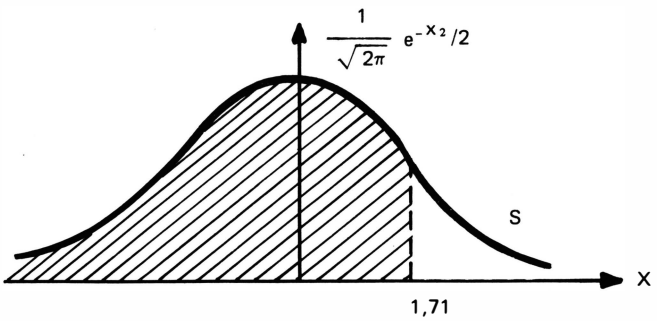
→ Exemplo 1: Calcular a probabilidade de uma variável aleatória estar contida no intervalo $[-\infty, 1{,}71]$. Calcular também a probabilidade da variável ser maior que 1,71. A distribuição é generalizada.

RESPOSTA: O código de operação é 1 (distribuição generalizada) e X = 1,71.

Assim, P (X) = 0,956367, para $-\infty < X \leqslant 1{,}71$. Para P (X), $X > 1{,}71$, temos:

$$1 - 0{,}956367 = 0{,}043633$$

(OBS.: Para executar de novo, responder com "S"). De acordo com a Fig. 1:



A área hachurada tem um valor igual a 0,956367. A área não-hachurada tem um valor de 0,043633, e corresponde a P ($X > 1{,}71$).

→ Exemplo 2: Idem para X = 0,725

RESPOSTA: Generalizada, P (X) = 0,765774 e P ($X > 0{,}725$) = 0,234226.

→ Exemplo 3: A média de uma amostragem é 32, e o desvio-padrão é 10. Qual o valor de P (X) para X = 37?

RESPOSTA: O código de operação é 2, não generalizada. A média é 32, e o desvio, 10. Com esses dados, temos:

$$P(X) = 0{,}691463$$

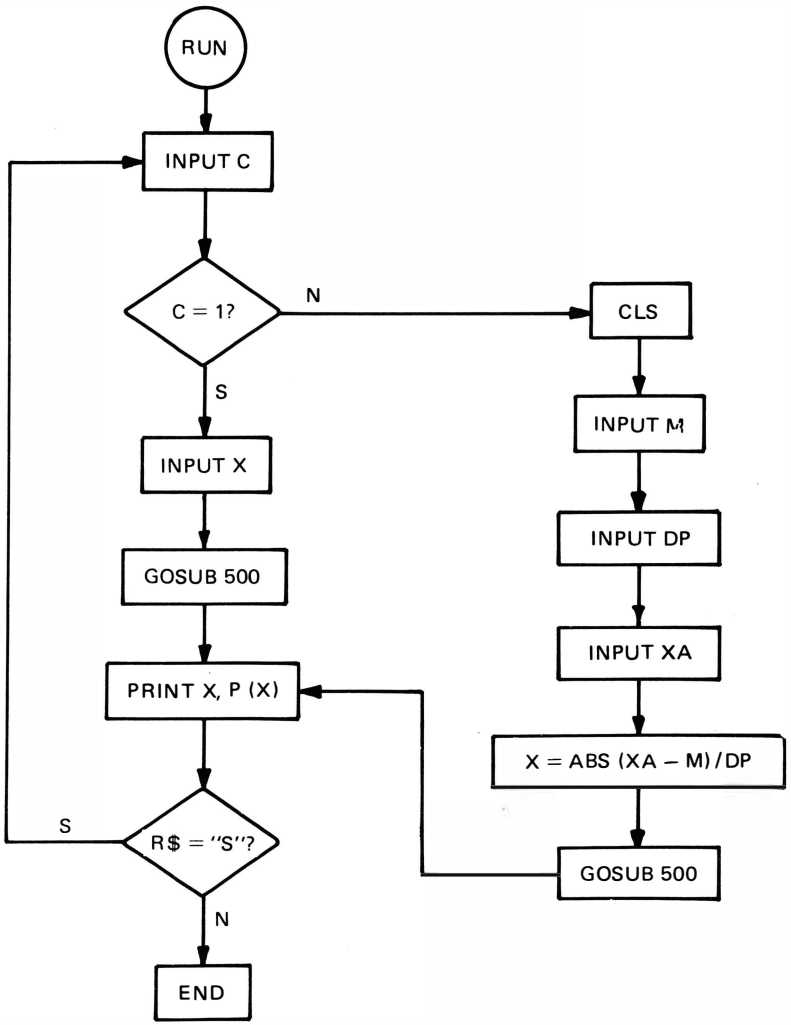
→ Exemplo 4: No caso acima, qual a probabilidade de "X" ocorrer entre 34 e 37 ($34 \leqslant X \leqslant 37$), para a mesma média e desvio-padrão?

$$X = 37, \quad P(X) = 0{,}691463$$
$$X = 34, \quad P(X) = 0{,}579261$$

Para $34 \leqslant X \leqslant 37$, $P(X) = 0{,}691463 - 0{,}579261 = 0{,}112202$

(OBS.: Para 100% de probabilidade, temos $P(X) = 1$).

FLUXOGRAMA:



A sub-rotina que se inicia no endereço 500 executa o cálculo de P (X).

```
************************************************
DISTRIBUICAO NORMAL DE PROBABILIDADES.
************************************************

ESCOLHA O CO'DIGO DA OPCAO:
CO'DIGO 1: GENERALIZADA.
CO'DIGO 2: NAO-GENERALIZADA.? 1




X= ? 1.71.




================================================
DISTRIBUICAO NORMAL DE PROBABILIDADES.
================================================

X= 1.71          P(X)= .956367
------------------------------------------------

QUER EXECUTAR NOVAMENTE? (S/N)? S
```

```
10 'DISTRIBUICAO NORMAL DE PROBABILIDADES.
20 'GENERALIZADA E NAO-GENERALIZADA.
30 'POR: FAUSTO A. DE A. BARBUTO.   DATA:08/MAIO/'84.
40 '
50 CLEAR 100: CLS: PRINT STRING$(50,"*"): PRINT"DISTRIBUICAO NORMAL DE PROBABILI
DADES.": PRINT STRING$(50,"*")
60 PRINT: PRINT"ESCOLHA O CO'DIGO DA OPCAO:": INPUT"CO'DIGO 1: GENERALIZADA.
CO'DIGO 2: NAO-GENERALIZADA."; C
70 A0=.398942 : A1=.31938 : A2=-.35656 : A3=1.78148
80 A4=-1.82126 :'A5=1.33027
90 ON C GOTO 110, 200
110 'DISTRIBUICAO GENERALIZADA.
120 CLS: INPUT"X= "; X
130 XA=X: GOSUB 500
140 CLS: PRINT STRING$(50,"="): PRINT"DISTRIBUICAO NORMAL DE PROBABILIDADES.": P
RINT STRING$(50,"=")
150 PRINT@320,"X=";XA,"P(X)=";P
160 PRINT: PRINT STRING$(50,"-")
170 PRINT: INPUT"QUER EXECUTAR NOVAMENTE? (S/N)"; R$
180 IF R$="S" THEN RUN ELSE END
190 'DISTRIBUICAO NAO-GENERALIZADA.
200 CLS: INPUT"ME'DIA DA AMOSTRAGEM"; M
210 INPUT"DESVIO PADRAO"; DP
220 INPUT"X="; XA
230 X=ABS(XA-M)/DP
240 GOSUB 500
250 GOTO 140
500 'SUBROTINA PARA CA'LCULO DA DNP.
510 T=1/(1+.23164*X)
520 P=1-A0*T*(A1+T*(A2+T*(A3+T*(A4+A5*T))))*EXP(-.5*X*X)
530 RETURN
```

FATORIAIS

26

Fatoriais de números são amplamente empregados em estatística, análise combinatória, termodinâmica molecular, cálculo etc. Por fatorial de um número define-se: (o ponto de exclamação denota o fatorial)

$$n! = \prod_{j=0}^{n-1} (n - j)$$

Assim, o fatorial de cinco (5) é:

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

Poderíamos também calcular o fatorial de um número através da propriedade distributiva dos logaritmos, isto é, o logaritmo de um produto é igual à soma dos logaritmos dos fatores. Desta maneira, o logaritmo do fatorial de cinco seria:

$$\log (5!) = \log 5 + \log 4 + \log 3 + \log 2 + \log 1$$

Usamos este artifício para calcular o fatorial de números maiores que 33, e que ocasionariam um "*OV ERROR*" (erro de *OVERFLOW*) denotando que o fatorial é maior que o maior número real que o computador consegue operar: $1{,}701411 \times 10^{38}$ em precisão simples. Deste modo, o programa se divide em duas partes: uma para números maiores e outra para números menores ou iguais a 33. Uma observação: fatoriais só são definidos para números inteiros positivos.

→ Exemplo 1: Calcular fatorial de 12.

Executamos *RUN* e entramos com o número 12.

RESPOSTA: Fatorial = 479001600

→ Exemplo 2: Calcular o fatorial de 30.

Do mesmo modo, obtemos:

$$\text{Fatorial} = 2{,}652528598121911 \times 10^{32}$$

(OBS.: O programa usa variáveis de precisão dupla). Um número um tanto grande, mas . . .

→ Exemplo 3: Calcular fatorial de 1000.

Após aproximadamente 30 seg. de processamento, temos:

$$1000! = 4{,}02411413192749 \times 10^{2567}$$

Um número com 2568 algarismos!

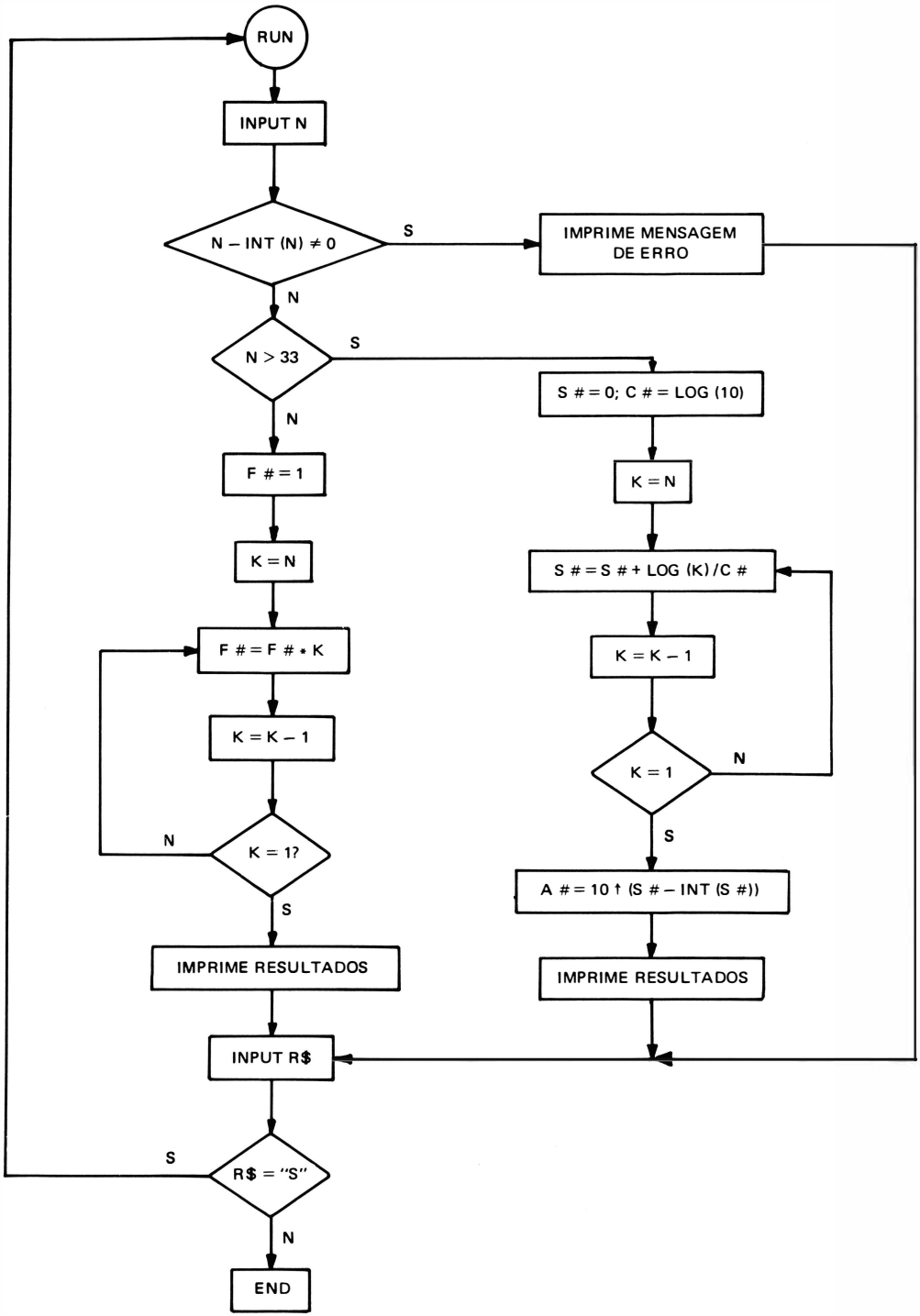
A resposta é dada dividida em duas partes: o argumento (número em dupla precisão) e o expoente ou potência de dez — no exemplo, 2567.

→ Exemplo 4: Calcular fatorial de 100.

RESPOSTA: ARG = 9,332673072814942
EXP = 157

Um número igualmente respeitáve! em tamanho, com 158 algarismos.

FLUXOGRAMA:


RUN
INPUT N
N – INT (N) ≠ 0
S
IMPRIME MENSAGEM DE ERRO
N
N > 33
S
S # = 0; C # = LOG (10)
N
F # = 1
K = N
S # = S # + LOG (K) / C #
K = K – 1
F # = F # * K
K = K – 1
K = 1
N
K = 1?
N
S
S
A # = 10 ↑ (S # – INT (S #))
IMPRIME RESULTADOS
IMPRIME RESULTADOS
INPUT R$
S
R$ = "S"
N
END

```
NUMERO=? 1000.

******************************
NUMERO= 1000
ARGUMENTO= 4.02411413192749
EXPOENTE= 2567
******************************

NOVA EXECUCAO (S/N)?
```

```
10 'PROGRAMA FATORIAL.
20 'QUALQUER NUMERO INTEIRO PODE SER INTRODUZIDO.
30 'AUTOR: FAUSTO A. DE A. BARBUTO;     DATA: 30/06/84.
40 CLS: INPUT"NUMERO="; N
50 IF (N-INT(N))<>0 OR N<0 THEN 220
55 IF N>33 THEN GOTO 120
60 F#=1
70 FOR K=N TO 1 STEP -1
80 F#=F#*K
90 NEXT K
95 CLS: PRINT@320, STRING$(30,"*")
100 PRINT@384,"NUMERO=";N : PRINT@448,"FATORIAL=";F#
105 PRINT@512, STRING$(30,"*")
110 GOTO 230
120 'FATORIAL PARA NUMEROS MAIORES QUE 33.
130 S#=0 : C#=LOG(10)
140 FOR K=N TO 1 STEP -1
150 S#=S#+LOG(K) : NEXT K
160 S#=S#/C#
165 A#=10^(S#-INT(S#))
170 CLS: PRINT@320, STRING$(30,"*")
175 PRINT@384,"NUMERO=";N
180 PRINT@448,"ARGUMENTO=";A#
190 PRINT@512,"EXPOENTE=";INT(S#)
200 PRINT@576, STRING$(30,"*")
210 GOTO 230
220 CLS: PRINT@448,"NU'MERO NAO-INTEIRO E/OU NEGATIVO; EXECUCAO ABORTADA"
230 PRINT@704,"":INPUT"NOVA EXECUCAO (S/N)"; R$
240 IF R$="S" THEN RUN ELSE END
```

FUNÇÃO GAMA

27

A função gama é definida por:

$$\Gamma(n) = \int_0^{\infty} x^{n-1} e^{-x} dx$$

Devido ao limite ser infinito (∞) não podemos calcular esta integral pelos métodos triviais de cálculo numérico — Simpson, por exemplo — mas através de uma correlação denominada *Fórmula de Stirling,* que permite calcular o valor de $\Gamma(n)$, $n > 0$.

$$\Gamma(n) = \sqrt{2\pi}\, n^{n-0,5}\, e^{z}$$

onde

$$Z = \frac{1}{12x} - \frac{1}{360x^3} - x$$

A função gama apresenta também a propriedade seguinte:

$$\Gamma(n+1) = n\Gamma(n) = n!$$

onde n! é o fatorial de n. (Vide programa fatorial). Como $n \epsilon R$ e $n > 0$ podemos desta forma calcular o "fatorial" de um número real, não inteiro. Por exemplo:

$$\Gamma(5,5) = 4,5\Gamma(4,5) = 4,5!$$

(Sabemos que, pela definição, a função fatorial só se aplica a números inteiros). Lançaremos mão desta propriedade para calcular a função gama de números reais não inteiros negativos. Desenvolvendo um pouco mais esta propriedade, temos que:

$$\Gamma(n+2) = \Gamma((n+1)+1) = n+1\,\Gamma(n+1)$$

e que:

$$\Gamma (n + 1) = n \Gamma (n)$$

Logo,

$$\Gamma (n + 2) = n (n + 1) \Gamma (n)$$

e finalmente,

$$\Gamma (n) = \frac{\Gamma (n + 2)}{n (n + 1)}$$

Generalizando:

$$\Gamma (n) = \frac{\Gamma (n + K)}{\prod_{i=0}^{K-1} (n + i)}$$

Se $n = -4,5$, por hipótese:

$$\Gamma (-4,5) = \frac{\Gamma (0,5)}{-4,5 (-4,5 + 1) (-4,5 + 2) (-4,5 + 3) (-4,5 + 4)}$$

Fizemos, acima, $K = 5$. Desta forma:

$$\Gamma (-4,5) = \frac{\Gamma (0,5)}{-29,5313}$$

→ Exemplo 1: Calcular $\Gamma (n)$ para $n = 7$.

Executando, obtemos:

$$\Gamma (7) = 720,001$$

OBS.: Como 720 é igual a 6!, notamos que é válida a propriedade da função gama abordada anteriormente.

→ Exemplo 2: Calcular $\Gamma(0,5)$.

RESPOSTA: $\Gamma(0,5) = 1,75661$

OBS.: $\Gamma(0,5) \sim \sqrt{\pi}$

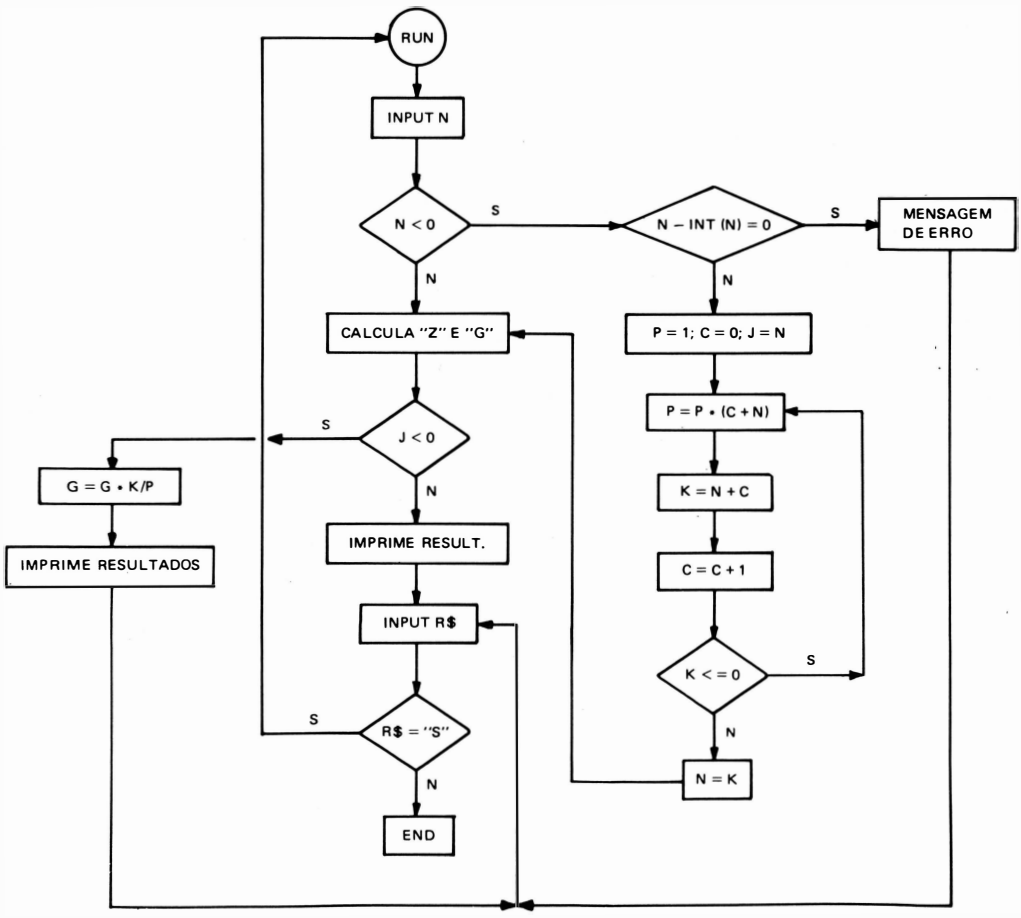
→ Exemplo 3: Calcular $\Gamma(-0,5)$.

RESPOSTA: $\Gamma(-0,5) = -3,51321$

→ Exemplo 4: Calcular $\Gamma(-8,2)$.

RESPOSTA: $\Gamma(-8,2) = -8,6077 \times 10^{-5}$.

FLUXOGRAMA:



```
*****************************
NUMERO= 7
G(N)= 720.001
*****************************


QUER EXECUTAR DE NOVO (S/N)? .
```

```
10 'PROGRAMA PARA CA'LCULO DA FUNCAO GAMA;
20 'USANDO A APROXIMACAO DE STIRLING.
30 'AUTOR: FAUSTO A. DE A. BARBUTO.     DATA: 30/06/84.
40 'OBS: GAMA NAO PODE SER CALCULADA PARA NU'MEROS INTEIROS
50 '      NEGATIVOS.
60 CLS:INPUT"N="; N
70 IF N<0 THEN 150
80 Z=1/12/N-1/360/N[3-N
90 G=2.50663*EXP(Z)*N[(N-.5)
95 IF J<0 THEN 230
100 CLS: PRINT@384, STRING$(30,"*")
110 PRINT@448,"NUMERO=";N
120 PRINT@512,"G(N)=";G
130 PRINT@576, STRING$(30,"*")
140 GOTO 280
150 'CA'LCULO DE GAMA PARA N<0 E NAO-INTEIRO.
155 IF(N-INT(N))=0 THEN 270
160 P=1 : C=0 : J=N
170 P=P*(C+N)
180 K=N+C
190 C=C+1
200 IF K<=0 THEN 170
210 N=K
220 GOTO 80
230 G=G*K/P
240 CLS: PRINT@384,STRING$(30,"*")
250 PRINT@448,"NUMERO=";J
260 PRINT@512,"G(N)=";G
270 PRINT@576,STRING$(30,"*")
280 PRINT@704,"" : INPUT"QUER EXECUTAR DE NOVO (S/N)"; R$
290 IF R$="S" THEN RUN ELSE END
300 PRINT@448,"NUMERO INTEIRO NEGATIVO; SOLUCAO IMPOSSIVEL."
310 GOTO 280
```

DERIVADA DE F (X) EM UM PONTO (X, Y)

28

Já havíamos visto, em capítulos anteriores, técnicas para efetuar a integração de uma função em X, Y = F (X), contínua e diferenciável num intervalo $[X_1, X_2]$. A operação inversa da integração é a *derivação* e esta é relacionada não mais a um intervalo, mas a um ponto (X, Y) ϵ F (X) desde que a função seja diferenciável neste ponto. A definição de derivada é um limite, qual seja:

$$F'(X) = \lim_{\Delta X \to 0} \frac{F(X + \Delta X) - F(X)}{\Delta X}$$

Em termos de geometria analítica, a derivada é a tangente do ângulo da reta que intercepta a função no ponto (X, Y). (Fig. 1).

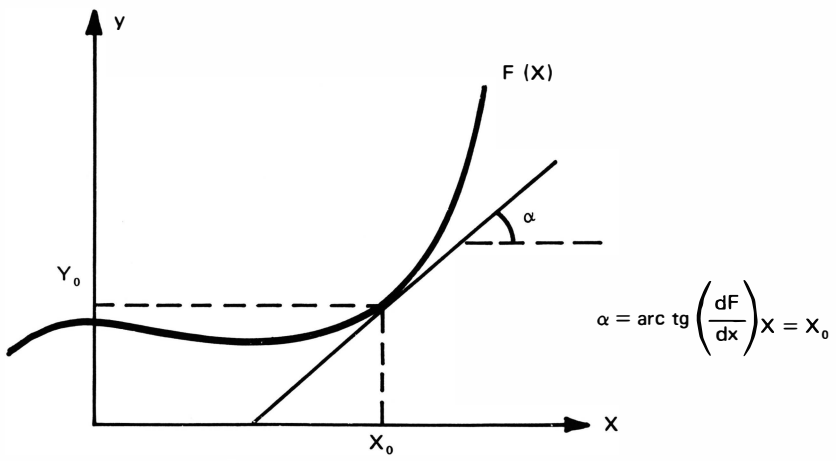
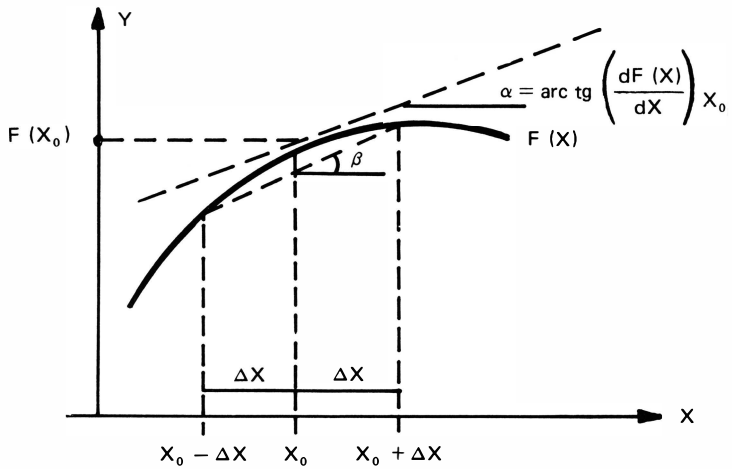


FIG. 1. A função F (X), o ponto (X_0, Y_0) e a reta tangente a F (X) em (X_0, Y_0).

Usamos a notação $\frac{dF(X)}{dx}$ para representar a derivada genérica de F (X) e $\left(\frac{dF(X)}{dx}\right)_{X_0}$ para o valor da derivada no ponto (X_0, Y_0). Deste modo, $tg\,\alpha = \left(\frac{dF(X)}{dx}\right)_{X_0}$, de acordo com a Fig. 1. No computador, não podemos calcular o limite, mas podemos em troca tornar Δx pequeno o suficiente para obtermos um valor para a derivada com precisão excelente, e tempo de execução mínimo. O cálculo é feito levando-se em conta que nas proximidades do ponto aonde se quer calcular a derivada, todos os pontos têm derivadas com a mesma ordem de grandeza e valores bastante próximos. De acordo com a Fig. 2, temos:



Como podemos observar da figura, X_0 é o ponto médio do intervalo $[\, X_0 - \Delta X, X_0 + \Delta X\,]$. No desenho, β é o ângulo determinado pela equação

$$tg\,\beta = \frac{F(X_0 + \Delta X) - F(X_0 - \Delta X)}{(X_0 + \Delta X) - (X_0 - \Delta X)} = \frac{F(X_0 + \Delta X) - F(X_0 - \Delta X)}{2\Delta X}$$

Se fizermos ΔX se aproximar de zero, isto é, estreitando o intervalo $[\, X_0 - \Delta X, X_0 + \Delta X\,]$, teremos que:

$$tg\,\alpha \cong tg\,\beta$$

Já falando sobre o programa em si, colocamos a função F (X) na linha 240 como, por exemplo, para a função exponencial:

```
240 Y = EXP (X)
```

Além disso, admite-se um valor para Δx igual a 0,005, desde que o usuário, durante a execução, não arbitre um outro valor para este parâmetro.

→ Exemplo 1: Calcular derivada de $F(X) = X^2$ no ponto $X = 5$, com DX = 0,0025 e DX = = 0,005.

```
RUN <ENTER>
X = ?
5 <ENTER>
DELTA = ?
.0025 <ENTER>
```

A resposta é: $F'(X) = 10{,}0006$

```
S <ENTER>
X = ?
5 <ENTER>
DELTA = ?
<ENTER>
```

RESPOSTA: $F'(X) = 10{,}0004$

Como a resposta exata é $F'(X) = 10$, temos como erro, respectivamente, 0,006% e 0,004%.

→ Exemplo 2: Para $X = 1{,}5$, calcular a derivada neste ponto na função $y = \frac{2}{\sqrt{\pi}} e^{-x^2}$, com delta = 0,005.

Linha 240:

```
240 Y = 2 * EXP (– X * X)/SQR (3.1415926)
```

```
RUN
X = ?
1.5 <ENTER>
DELTA = ?
<ENTER>
```

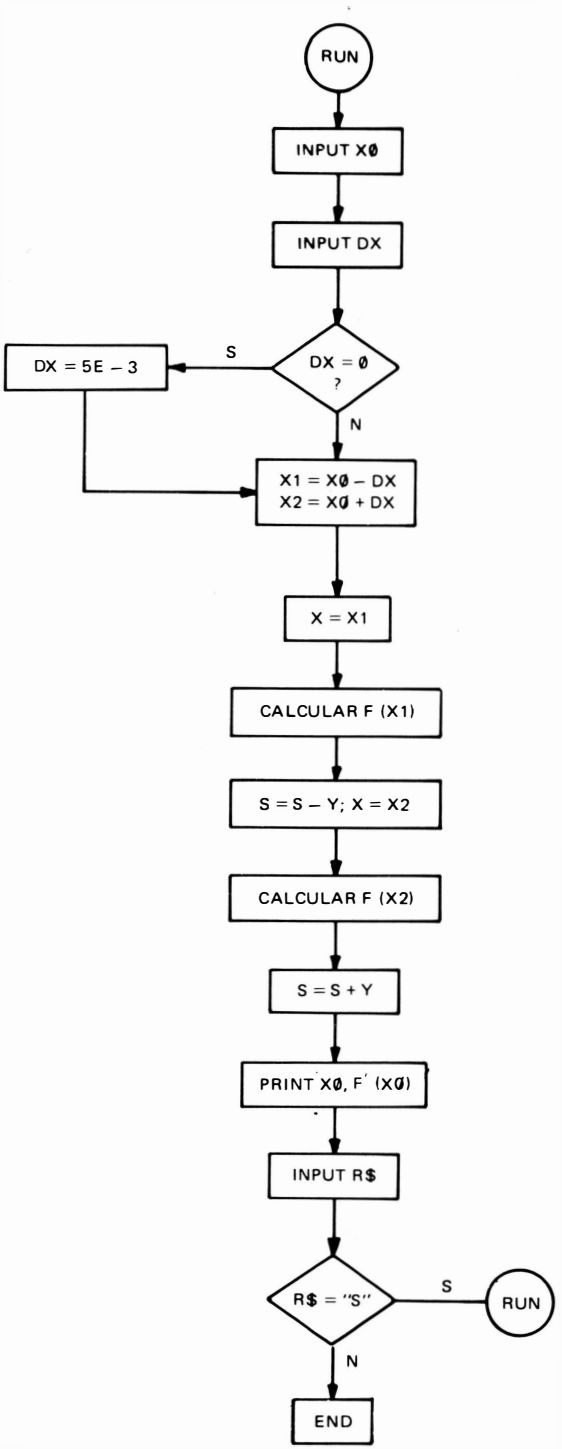
RESPOSTA: $F'(X) = -0{,}356795$

Como a resposta exata é $-0{,}356791$, o erro é 0,0011%.

→ Exemplo 3: Seja $F(X) = \frac{e^{\text{sen } x} - e^{\cos x}}{X^2 \text{ tg } X^2}$. Calcular a derivada no ponto $X = 0{,}5$, com delta = 0,005.

Do mesmo modo, obtemos $F'(X) = -2288{,}7$ com erro de 0,1168% (Confira!)

FLUXOGRAMA:


RUN
INPUT XØ
INPUT DX
DX = Ø ?
S
DX = 5E – 3
N
X1 = XØ – DX
X2 = XØ + DX
X = X1
CALCULAR F (X1)
S = S – Y; X = X2
CALCULAR F (X2)
S = S + Y
PRINT XØ, F (XØ)
INPUT R$
R$ = "S"
S
RUN
N
END

```
**************************************
X= .5
DERIVADA NO PONTO 'X':-2288.7
**************************************


QUER EXECUTAR DE NOVO? (S/N)
? .
```

```
10 'PROGRAMA PARA CALCULO DE DERIVADAS EM DETERMINADO PONTO
20 'DE UMA FUNCAO Y=F(X) USANDO O TEOREMA DO VALOR ME'DIO.
30 'A FUNCAO A SER DERIVADA DEVE SER COLOCADA NA LINHA 240.
40 'POR: FAUSTO ARINOS DE ALMEIDA BARBUTO.   DATA: 06/JUL/84.
50 CLS:INPUT"X= "; X0
60 PRINT:INPUT"DELTA= "; DX
70 IF DX=0 THEN DX=5E-3
80 X1=X0-DX
90 X2=X0+DX
100 X=X1 : GOSUB 240
110 S=S-Y
120 X=X2 : GOSUB 240
130 S=S+Y
140 CLS:PRINT@384, STRING$(40,"*")
160 PRINT@448,"X=";X0
170 PRINT@512,"DERIVADA NO PONTO 'X':";S/2/DX
180 PRINT@576, STRING$(40,"*")
190 PRINT@832,"QUER EXECUTAR DE NOVO? (S/N)"
200 INPUT R$
210 IF R$="S" THEN RUN ELSE END
230 'DEFINICAO DA FUNCAO "Y".
240 Y=(EXP(SIN(X))-EXP(COS(X)))/(X*X-TAN(X*X))
250 RETURN
```

INTERPOLAÇÃO
– MÉTODO LINEAR, LOGARÍTMICO E DE AITKEN

29

A interpolação é um método numérico bastante empregado em vários ramos da Engenharia, Matemática, Química e outras ciências e consiste em obter valores de uma propriedade ou função num determinado ponto que esteja contido num intervalo de cujos extremos se conhecem os pontos e suas respectivas propriedades. Quanto mais estreito for este intervalo, melhor, idem se se conhecem mais valores da função dentro do intervalo. Neste último caso, pode-se tentar estabelecer uma função que passe pelos pontos ou que, ao menos, represente (bem) o intervalo em questão. (Ver programa *CURVE FITTING*). À primeira vista, é uma tarefa relativamente simples.

Examinando a questão por outro lado, verificamos que o conhecimento do comportamento da função e a posse de mais de dois pontos pode ser valioso. Se a função se comporta como uma reta, a interpolação linear é mais do que suficiente. Caso contrário, uma linearização com logaritmos pode vir bem a calhar. De acordo com a dimensão do intervalo que se usa, a diferença em termos de precisão de um ou outro método pode ser – ou não – relevante.

O *método linear* admite que o intervalo usado para a interpolação de um terceiro ponto pode ser representado exata ou aproximadamente bem por uma reta de equação conforme segue:

$$y = ax + c$$

Pode-se admitir isso sem restrições para funções lineares ou para não lineares, desde que para intervalos $[X_1, X_2]$ muito pequenos. As constantes "a" e "c" acima são características de cada função.

O *método logarítmico* supõe uma curva linearizada através de logaritmos:

$$\log y = a \log x + c$$

As constantes "a" e "c" são diferentes das suas correspondentes no método linear, mas são calculadas de maneira análoga. Levando-se em conta que raras são as propriedades em Engenharia ou Ciências que apresentam comportamento linear, optar por este método de cálculo pode ser interessante.

O *método de Aitken* é interessante se dispusermos de mais de dois dados conhecidos. Pode ser usado tanto para comportamento linear como não-linear. Considerando n pontos, podemos calcular linearmente as propriedades nos intervalos (X_0, X_i), i = 1, 2, 3, . . . , n. Assim:

$$F(X_j) = \frac{(X_j - X)\,F(X_0) - (X_0 - X)\,F(X_j)}{X_j - X_0}$$

Se dispomos de mais pontos, podemos aumentar o grau de interpolações – todas lineares – repetindo-as, de acordo com o seguinte algoritmo:

Para K = 1, 2, 3, . . . , n – 1

Fazer j = K + 1, K + 2, . . . , n

$$F_{K,j} = \frac{(X_j - X)\,F_{K-1,K} - (X_K - X)\,F_{K-1,j}}{X_j - X_K}$$

Até que se atinja a tolerância (erro) preestabelecida.

Os três métodos acima citados são perfeitamente válidos dentro de suas possibilidades, e todos encontram seu lugar e vez. Passemos aos exemplos.

→ Exemplo 1: Dados os pontos (X, F (X)) abaixo, calcular F (X) para X = 3,5, métodos linear e log.

$$A = (3, 9);\ B = (4, 16)$$

Quando executamos, a primeira providência é fornecer o código da operação de interpolação a ser executada: "L" para linear, "LG" para logarítmica e "A" para *Aitken* escolhendo o método linear, entramos com o primeiro par (X, Y).

```
 ?
 3 <ENTER>
??
 9 <ENTER>
```

e então, com o segundo:

```
 ?
 4 <ENTER>
??
16 <ENTER>
```

Quando entrarmos com o ponto a interpolar, colocamos zero como valor de y, porque este é o valor de que não dispomos e queremos calcular. Poderíamos ter o inverso: calcular "X" para y = 10, por exemplo. Neste caso, entraríamos com zero seguido de 10.

```
?
3.5 <ENTER>
??
Ø <ENTER>
```

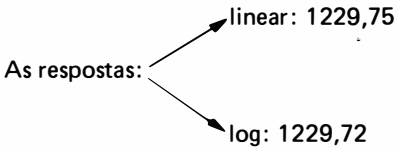
A resposta é y = 12,5.

Executando para logarítmica, obtemos y = 12,25.

→ Exemplo 2: Considere os seguintes dados para o vapor d'água superaquecido, a 640°F:

Pressão: 1500 PSIA. → Entalpia: 1226,4 BTU/LB.
Pressão: 1450 PSIA. → Entalpia: 1233,1 BTU/LB.
Pressão: 1425 PSIA. → Entalpia: 1236,2 BTU/LB.

Calcular a entalpia a 1475 psia e 640°F, usando os três métodos. (OBS.: Nos métodos linear e log. usamos os dados para 1450 e 1500 psia.



(Resultados muito semelhantes).

Vamos agora usar o método de *AITKEN* pela primeira vez:

O NÚMERO DE PONTOS É TRÊS:

```
?
3 <ENTER>
```

Quanto à precisão, se introduzirmos um valor nulo apertando simplesmente a tecla <ENTER>, o programa assume-o como sendo 1×10^{-5}. A escolha – ou não – de um erro fica a cargo do usuário. Assim:

PRECISÃO = ?
<ENTER>

A abscissa a interpolar é X = 1475; X (1) = 1425, Y (1) = 1236; X (2) = 1450, Y (2) = = 1233,1; X (3) = 1500, Y (3) = 1226,4. Após fornecermos estes dados, obtemos a resposta:

Y = 1229,83

Pelo método de *AITKEN,* H = 1229,83 BTU/LB.

O valor correto retirado de uma *STEAM TABLE* é: H = 1229,7 BTU/LB.

Todos os três métodos apresentaram resultados com boa confiabilidade, sendo que o menor erro ficou a cargo do método logarítmico: ϵ = 0,0016%, e em seguida o linear: ϵ = = 0,0041%.

→ Exemplo 3: Calcular a entalpia do vapor d'água a 660°F, em função da temperatura, à pressão constante de 1500 psia. Dados:

Temperatura: 640°F	Entalpia: 1226,4 BTU/LB.
Temperatura: 680°F	Entalpia: 1268,6 BTU/LB.
Temperatura: 700°F	Entalpia: 1286,8 BTU/LB.

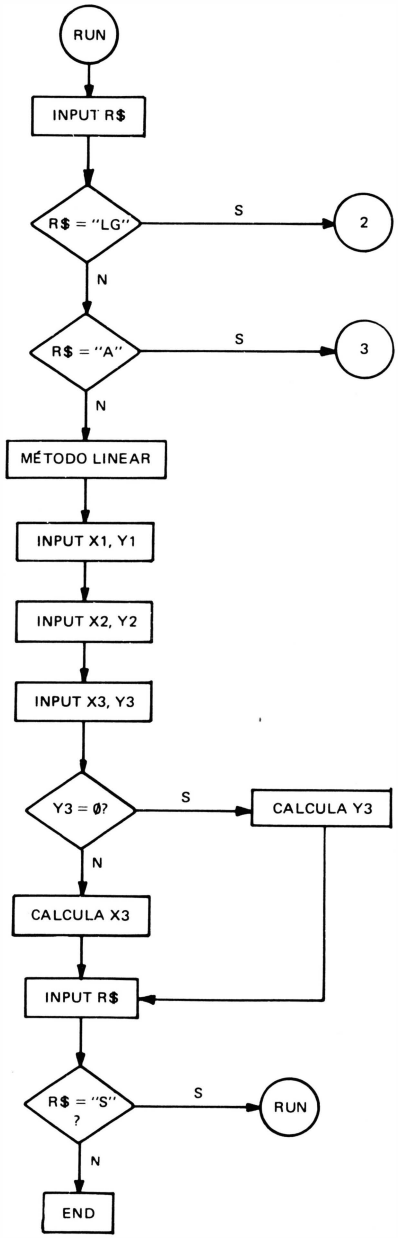
RESPOSTA:	Método linear:	1247,5 BTU/LB.
	Método log:	1247,64 BTU/LB.
	Método Aitken:	1248,47 BTU/LB.

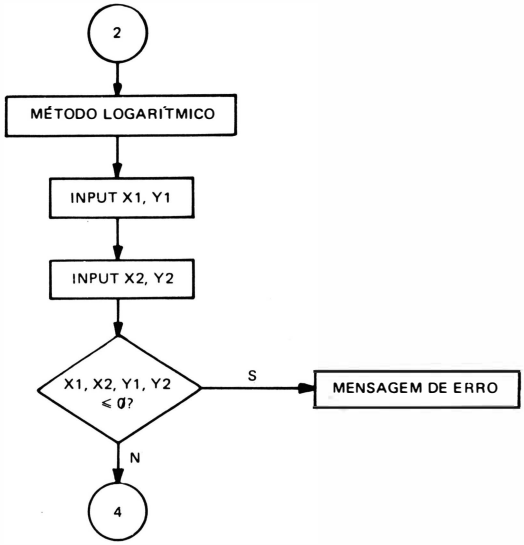
Pela *STEAM TABLE,* o valor tabelado e exato é: 1248,5 BTU/LB. Desta vez, *AITKEN* exibiu o menor erro, ϵ = – 0,0024%.

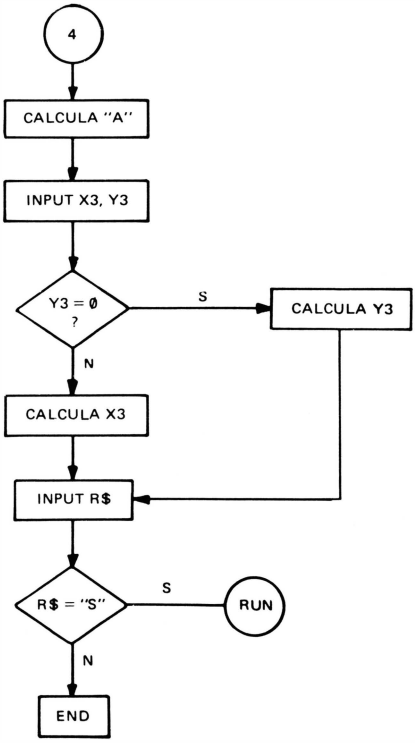
Por último, é bom lembrar que o método log. não permite entrada de dados nulos; apenas o valor a ser interpolado o pode (e deve) ser. Tentativas de transgressão ocasionarão uma mensagem de erro. E mais: os métodos acima também servem para extrapolação, embora extrapolações só devam ser feitas em último caso, dando-se preferência a interpolações.

FLUXOGRAMA:

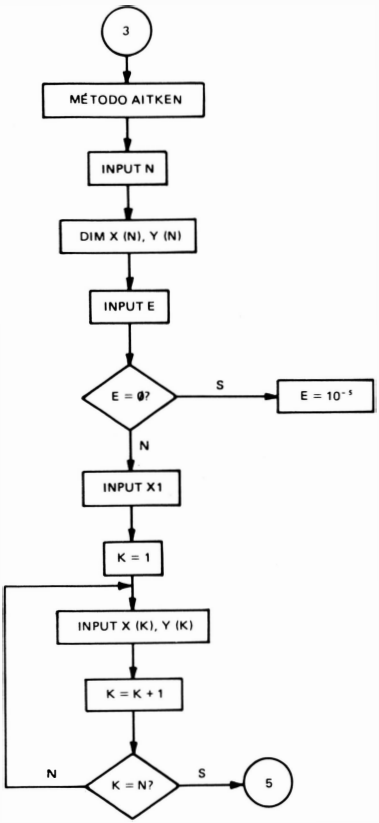
INTERPOLAÇÃO



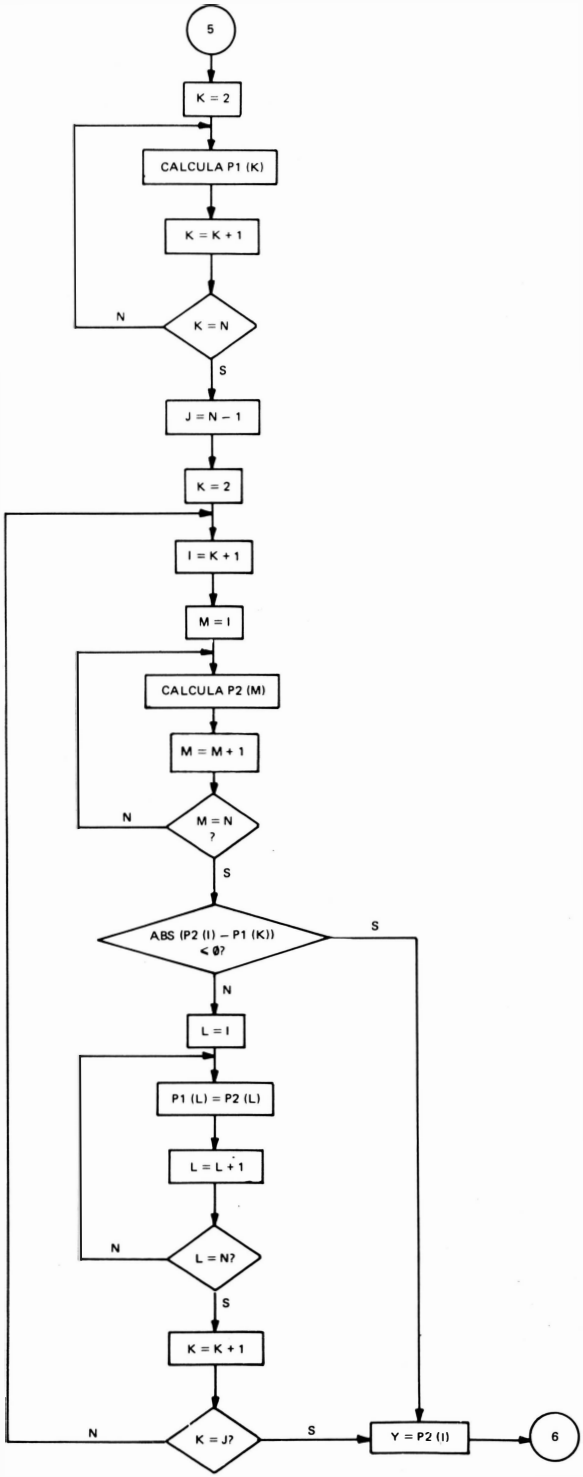




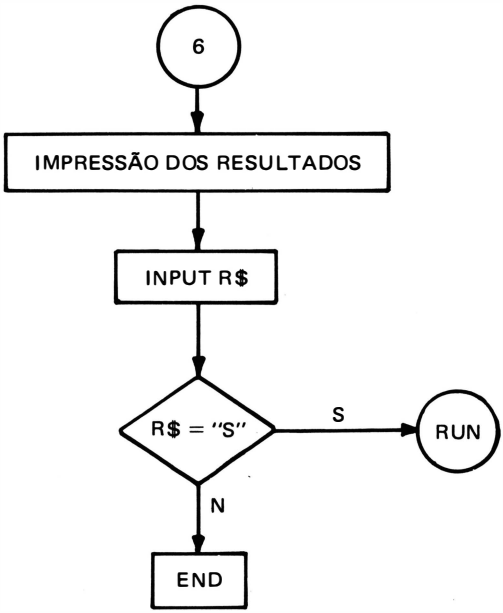
INTERPOLAÇÃO (CONT. 1)



INTERPOLAÇÃO (CONT. 2)



INTERPOLAÇÃO (CONT. 3)




```
ENTRE COM O PRIMEIRO PAR X,Y? 3
?? 9

ENTRE COM O SEGUNDO PAR X,Y? 4
?? 16

ENTRE COM O PONTO X,Y A INTERPOLAR? 3.5
?? 0.
```

```
****************************************
X1= 3           Y1= 9
X2= 4           Y2= 16

X3= 3.5         -) Y3= 12.25
****************************************

QUER EXECUTAR DE NOVO? (S/N)
? .
```

```
10 'PROGRAMA INTERPOLACAO.
20 'TRES METODOS SAO ADOTADOS: LINEAR, LOGARITMICA E AITKEN.
30 'POR: FAUSTO A. DE A. BARBUTO.      DATA: 10/JUL/84.
40 CLS
50 PRINT:INPUT"ESCOLHA O ME'TODO: LINEAR, LOG, AITKEN (L/LG/A)"; R$
60 IF R$="LG" THEN 160 ELSE IF R$="A" THEN 270
70 'ME'TODO LINEAR.
80 CLS:INPUT"ENTRE COM O PRIMEIRO PAR X,Y"; X1, Y1
90 PRINT:INPUT"ENTRE COM O SEGUNDO PAR X,Y"; X2, Y2
100 PRINT:INPUT"ENTRE COM O PONTO X,Y A INTERPOLAR"; X3, Y3
110 CLS:PRINT@256, STRING$(40,"*")
120 PRINT"X1=";X1,"Y1=";Y1
130 PRINT"X2=";X2,"Y2=";Y2
140 PRINT:IF Y3=0 THEN PRINT"X3=";X3,"-) Y3="; Y1+(Y2-Y1)*(X3-X1)/(X2-X1) ELSE P
RINT"-) X3=";(Y3-Y1)*(X2-X1)/(Y2-Y1)+X1,"Y3=";Y3 ELSE 550
145 PRINT STRING$(40,"*")
150 GOTO 550
160 'ME'TODO LOGARITMICO.
170 CLS:INPUT"ENTRE COM O PRIMEIRO PAR X,Y"; X1, Y1
180 PRINT:INPUT"ENTRE COM O SEGUNDO PAR X,Y"; X2, Y2
190 A=(LOG(Y2)-LOG(Y1))/(LOG(X2)-LOG(X1))
200 PRINT:INPUT"ENTRE COM O PONTO X,Y A INTERPOLAR"; X3, Y3
210 CLS:PRINT@256, STRING$(40,"*")
220 PRINT"X1=";X1,"Y1=";Y1
230 PRINT"X2=";X2,"Y2=";Y2
240 PRINT:IF Y3=0 THEN PRINT"X3=";X3,"-) Y3=";EXP(A*(LOG(X3)-LOG(X1))+LOG(Y1)) E
LSE PRINT"-) X3=";EXP((LOG(Y3)-LOG(Y1))/A+LOG(X1)),"Y3=";Y3 ELSE 550
250 PRINT STRING$(40,"*")
260 GOTO 550
270 'ME'TODO DE AITKEN.
280 CLS:INPUT"NU'MERO DE PONTOS X,Y"; N
290 DIM X(N), Y(N)
300 PRINT:INPUT"PRECISAO= "; E
305 IF E=0 THEN E=1E-5
310 PRINT:INPUT"ABCISSA PARA INTERPOLAR="; X1
320 FOR K=1 TO N
330 CLS:PRINT"X(";K;")="
340 INPUT X(K)
350 PRINT:PRINT"Y(";K;")="
360 INPUT Y(K)
370 NEXT K
380 FOR K=2 TO N
390 P1(K)=((X(K)-X1)*Y(1)-(X(1)-X1)*Y(K))/(X(K)-X(1))
400 NEXT K
410 J=N-1
420 FOR K=2 TO J
430 I=K+1
440 FOR M=I TO N
450 P2(M)=((X(M)-X1)*P1(K)-(X(K)-X1)*P1(M))/(X(M)-X(K))
460 NEXT M
470 IF ABS(P2(I)-P1(K))(=0 THEN 510
480 FOR L=I TO N
490 P1(L)=P2(L)
500 NEXT L, K
510 Y=P2(I)
520 CLS:PRINT@320, STRING$(40,"*")
530 PRINT"X=";X1,"Y=";Y
540 PRINT STRING$(40,"*")
550 PRINT@768,"QUER EXECUTAR DE NOVO? (S/N)"
560 INPUT R$
570 IF R$="S" THEN RUN ELSE END
```

PARTE III

ENGENHARIA

PROPRIEDADES DE MISTURAS GASOSAS: FATOR DE COMPRESSIBILIDADE (Z), VISCOSIDADE, MOLARIDADE E DENSIDADE

30

Passaremos a partir de agora a discutir um assunto bastante específico e relacionado aos profissionais da área de química industrial, que é o cálculo das propriedades de misturas gasosas mais importantes industrialmente, e que são citadas no título acima. Para tal, necessitamos de saber os seguintes dados sobre a mistura:

1 – Pressão absoluta (ATM)
2 – Temperatura (Kelvin)
3 – Número de componentes
4 – Fração molar (%) de cada componente
5 – Pressão crítica de cada componente
6 – Temperatura crítica de cada componente
7 – Molaridade de cada componente

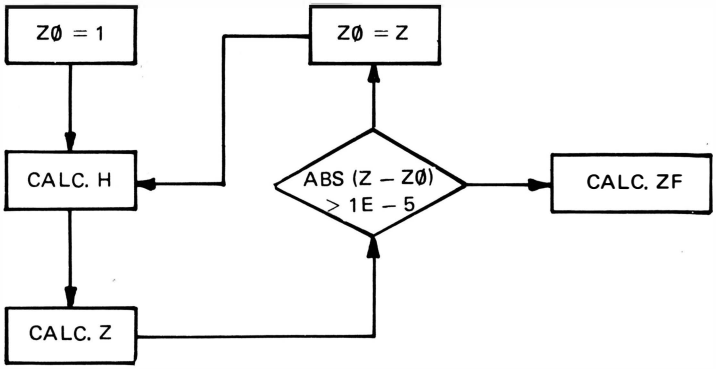
Os quatro primeiros itens da lista acima são relacionados com o processo que se vai analisar, e são medidos "in situ". Os três últimos são tabelados em "*handbooks*" de Química e Física. (Veja bibliografia citada).

As equações representativas do comportamento da mistura gasosa são:

$$Z = \frac{1}{1-H} - \frac{4{,}934}{T_R^{1,5}} \cdot \frac{H}{1+H}$$

$$H = \frac{0{,}0867\, P_R}{Z\, T_R}$$

Observando mais cuidadosamente as duas equações, notamos que ambas contêm "Z" e "H", e que não é possível explicitar qualquer uma das duas variáveis. Empregamos um método iterativo para o cálculo de "Z", descrito pelo fluxograma a seguir, *para cada componente.*



Quando ABS (Z – Z∅) for maior que 0,00001, ZF (ponderado) é calculado. Enquanto isso não ocorrer, permaneceremos dentro do "*LOOP*".

Nas equações, P_R e T_R são, respectivamente, a pressão reduzida e a temperatura reduzida do sistema. (O programa calcula esses parâmetros).

$$P_R = \frac{P}{P_C} \qquad T_R = \frac{T}{T_C}$$

T_c = Temperatura crítica; P_c = pressão crítica.

→ Exemplo 1: Calcular as propriedades de uma mistura gasosa contendo 75% de metano (CH_4) e 25% de etano (C_2H_6).

Dados:

Do sistema: Temperatura: 300 K
Pressão: 5,5 ATM ABS.
Nº de componentes: 2

Do metano: (Componente Nº 1) Pressão crítica: 45,4 ATM
Temperatura crítica: 190 K
Fração molar: 75%
Mol: 16

Do etano: (Componente Nº 2) Pressão crítica: 48,2 ATM
Temperatura crítica: 305,4 K
Fração molar: 25%
Mol: 30

Com esses dados, obtemos:

Z = 0,991884
Viscosidade = 6,12067 * 10^{-3} centipoise
Molaridade = 19,5
Densidade = 4,39254 gramas/litro
Mol = 19,5

→ Exemplo 2: Calcular as mesmas propriedades para o ar (79% de N_2 e 21% de O_2) a 1,5 ATM e 40°C.

Dados do sistema:

Pressão: 1,5 ATM ABS
Temperatura: 313 K (40°C)
Número de componentes: 2

Dados da mistura:

Componente Nº 1 (Nitrogênio):
Pressão crítica: 33,5 ATM
Temperatura crítica: 126,2 K
Fração molar: 79%
Mol: 28

Componente Nº 2 (Oxigênio):
Pressão crítica: 49,8 ATM
Temperatura crítica: 154,6 K
Fração molar: 21%
Mol. 32

RESPOSTAS:

Z = 0,999705
Viscosidade = 5,27678 X 10^{-3} centipoise (CP)
Molaridade = 28,84
Densidade = 1,68489 gramas/litro

(OBS.: A ordem de entrada dos componentes não afeta o resultado).

→ Exemplo 3: Calcular propriedades do neon a 2 ATM e 400 K.

Dados do sistema: Pressão: 2 ATM ABS
Temperatura: 400 K
Número de componentes: 1

Dados da mistura: Pressão crítica: 27,2 ATM
Temperatura crítica: 44,4 K
Mol: 20,2
Fração molar: 100%

(OBS.: Não se trata de uma mistura, mas de um gás "puro": um só componente).

RESPOSTAS:

Z = 1,00058
Viscosidade = $8{,}024 \times 10^{-3}$
Molaridade = 20,2
Densidade = 1,23019

Vamos agora apresentar um pouco mais de teoria para aqueles que se interessarem. A densidade (no programa, a variável RO) é calculada pela fórmula:

$$\rho = \frac{M}{V} = \frac{P.MOL}{Z.R.T.}$$

Onde "R" é a constante universal dos gases, 0,082054 $\frac{\text{ATM} \cdot \text{LITRO}}{\text{GMOL} \cdot \text{K}}$

A viscosidade é calculada pela seguinte correlação:

$$VIS = D * EXP\ (C)$$

onde:

$$A = 3{,}5 + \frac{547{,}778}{T} + 0{,}01 * MF$$

$$B = 2{,}4 * A - 0{,}2 * A^2$$

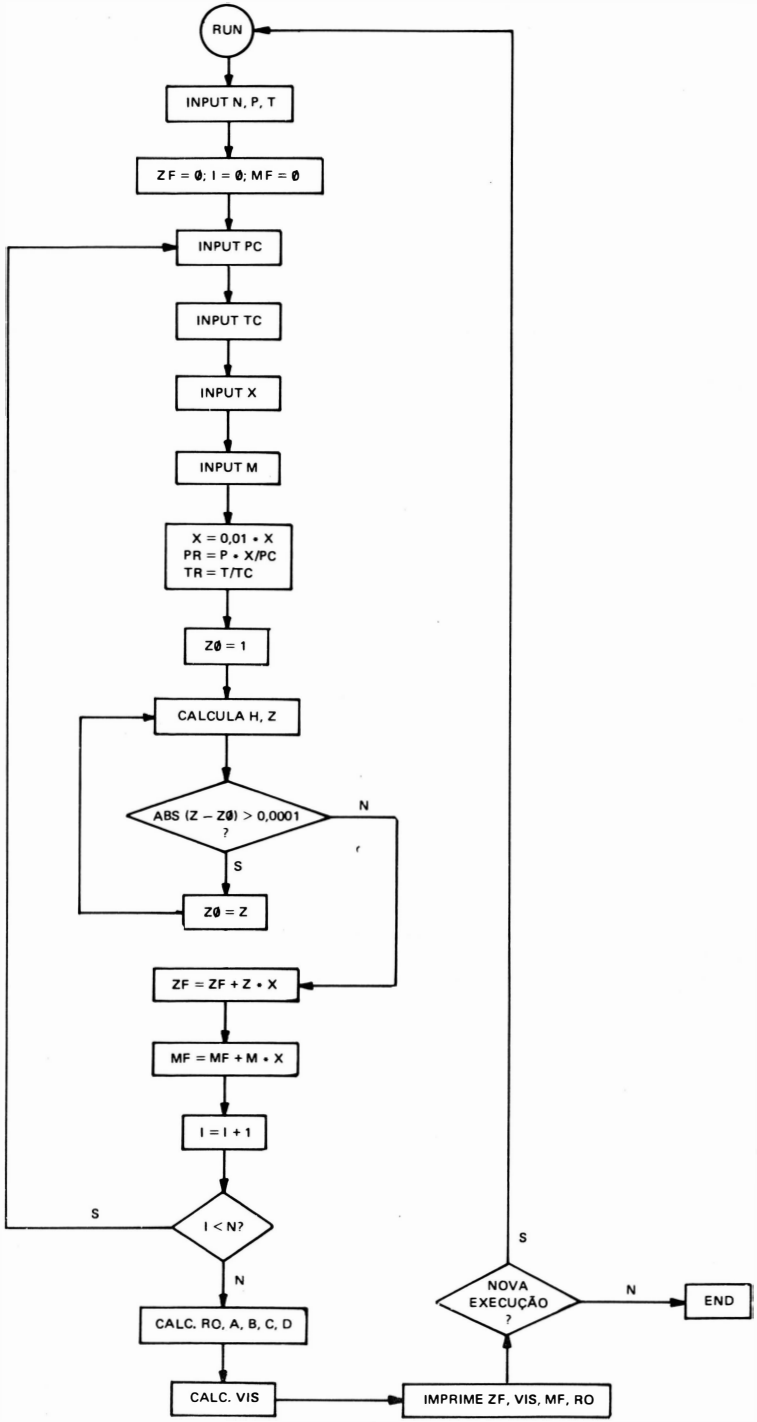
$$C = \frac{0{,}0219409 * P * MF * B}{ZF * T}$$

$$D = \frac{0{,}0001 * ((9{,}4 + 0{,}02 * MF) * T^{1{,}5})}{209 + 19 * MF + T}$$

Na verdade, esta correlação fica menos precisa à medida que se aumenta a temperatura do sistema. Até 100°C, temos um erro de aproximadamente 5% ou mais, em certos casos.

Existem outros métodos para se calcular o fator de compressibilidade, porém, o que vimos agora — conhecido como *REDLICH-KWONG* — é de amplo espectro de uso e bastante utilizado quando outros não o podem ser, por limitações inerentes a eles mesmos. Pode ser empregado tanto para gases polares como apolares (hidrocarbonetos, por exemplo).

FLUXOGRAMA:


RUN
INPUT N, P, T
ZF = Ø; I = Ø; MF = Ø
INPUT PC
INPUT TC
INPUT X
INPUT M
X = 0,01 • X
PR = P • X/PC
TR = T/TC
ZØ = 1
CALCULA H, Z
ABS (Z – ZØ) > 0,0001 ?
N
S
ZØ = Z
ZF = ZF + Z • X
MF = MF + M • X
I = I + 1
I < N?
S
N
CALC. RO, A, B, C, D
CALC. VIS
IMPRIME ZF, VIS, MF, RO
NOVA EXECUÇÃO ?
S
N
END

```
NU'MERO DE COMPONENTES DO SISTEMA? 2
PRESSAO ABSOLUTA DO SISTEMA (ATMÓSFERAS)? 5.5
TEMPERATURA DO SISTEMA (KELVIN)? 300



COMPONENTE 1
PRESSAO CRI'TICA (ATM)? 45.4
TEMPERATURA CRI'TICA (KELVIN)? 190
FRACAO MOLAR (%)? 75
MOL? 16.



COMPONENTE 2
PRESSAO CRI'TICA (ATM)? 48.2
TEMPERATURA CRI'TICA (KELVIN)? 305.4
FRACAO MOLAR (%)? 25
MOL? 30



**************************************************
         PROPRIEDADES DA MISTURA GASOSA:
**************************************************

Z= .991884
VISCOSIDADE= 6.12067E-03 CP
MOL= 19.5
DENSIDADE= 4.39254 GRAMAS/LITRO
QUER EXECUTAR NOVAMENTE? (S/N)? .
```

```
10 'PROPRIEDADES DOS GASES: MOL, DENSIDADE, VISCOSIDADE E
20 'FATOR "Z" DE COMPRESSIBILIDADE.
30 'POR: FAUSTO ARINOS DE ALMEIDA BARBUTO.  DATA:06/MAIO/'84.
40 '
50 CLS:INPUT"NU'MERO DE COMPONENTES DO SISTEMA"; N
60 PRINT:INPUT"PRESSAO ABSOLUTA DO SISTEMA (ATMOSFERAS)"; P
70 PRINT:INPUT"TEMPERATURA DO SISTEMA (KELVIN)"; T
80 ZF=0 : I=0 : MF=0
90 CLS:PRINT"COMPONENTE";I+1
100 PRINT:INPUT"PRESSAO CRI'TICA (ATM)";PC
110 PRINT:INPUT"TEMPERATURA CRI'TICA (KELVIN)"; TC
120 PRINT:INPUT"FRACAO MOLAR (%)"; X
130 PRINT:INPUT"MOL"; M
140 X=.01*X
150 PR=P*X/PC
160 TR=T/TC
170 Z0=1
180 H=.0867*PR/Z0/TR
190 Z=1/(1-H)-4.934*H/(1+H)/TR[1.5
200 IF ABS(Z-Z0)).0001 THEN 210 ELSE 220
210 Z0=Z : GOTO 180
220 ZF=ZF+Z*X
230 MF=MF+M*X
240 I=I+1
250 IF I(N THEN 90
260 RO=P*MF*12.1871/ZF/T
270 A=3.5+547.778/T+.01*MF
280 B=2.4*A-.2*A*A
290 C=.0219409*P*MF*B/ZF/T
300 D=.0001*((9.4+MF/50)*T[1.5)/(209+19*MF+T)
310 VIS=D*EXP(C)
315 ' SAI'DA DOS RESULTADOS.
320 CLS : PRINT STRING$(50,"*")
330 PRINT@73,"PROPRIEDADES DA MISTURA GASOSA:":PRINT STRING$(50,"*")
340 PRINT@320,"Z=";ZF : PRINT@448,"VISCOSIDADE=";VIS;"CP"
350 PRINT@576,"MOL=";MF:PRINT@704,"DENSIDADE=";RO;"GRAMAS/LITRO"
360 PRINT : INPUT"QUER EXECUTAR NOVAMENTE? (S/N)"; R$
370 IF R$="S" THEN RUN ELSE END
```

TUBULAÇÕES INDUSTRIAIS: CÁLCULO DA PERDA DE CARGA, VAZÃO E DIÂMETRO

31

Os fluidos na indústria são transportados continuamente por meio de tubos de um ponto a outro do sistema seja para armazenamento, mistura, processamento, fornecimento ao consumidor etc. Como partes auxiliares nestas tubulações, temos as válvulas, conexões, luvas, tês, check valves etc. Devido ao atrito do fluido transportado com as paredes dos tubos e partes auxiliares (doravante chamados *acidentes*), há uma queda de pressão ao longo de todo o tubo, ou seção de tubo em questão. Assim, se em dois pontos distintos de uma tubulação medimos um diferencial de 1 ATM, dizemos que neste trecho houve uma *perda de carga* de 1 ATM. Vários fatores influenciam a perda de carga, aumentando-a ou diminuindo-a. Entre eles, temos:

A) Da tubulação:

– Diâmetro: quanto menor, menor a área de escoamento do fluido, e maior a resistência ao fluxo. Equivale dizer, maior perda de carga. Ao contrário, acontece o efeito inverso.

– Rugosidade (ϵ): nenhum tubo é perfeitamente liso; quanto mais rugoso for, maior o atrito e a perda de carga. A rugosidade relativa (ϵ/D) é a razão entre a rugosidade absoluta e o diâmetro da tubulação. É admensional.

– Comprimento: quanto mais longa for a tubulação, maior a perda de carga.

– Acidentes: acidentes como válvulas globo, válvulas gaveta, luvas, tês etc. são às vezes responsáveis por 70-80% da perda de carga total num trecho, quando não mais. É mais fácil representar um acidente como um comprimento equivalente, L_{EQ}, ou seja, o comprimento de uma seção de tubo com o mesmo diâmetro do acidente que ocasionaria a mesma perda de carga que o acidente proporciona.

B) Do fluido:

– Viscosidade: podemos defini-la como a resistência que um fluido oferece ao deslocamento. Quanto mais viscoso o fluido, mais difícil é escoá-lo através de uma tubulação.

— Densidade: é a razão entre a massa e o volume de um corpo. No caso de um fluido, a densidade varia bastante com a temperatura e pouco com a pressão, no caso deste ser incompressível. *Numa mesma classe química de compostos,* a viscosidade tem relação direta com a densidade: quanto mais denso, mais viscoso. Ex: derivados de petróleo.

C) Do processo:

— Vazão: quanto maior, maior a turbulência, o atrito e a perda de carga.

Podemos caracterizar o fluxo de um fluido através de uma tubulação pela equação de Bernoulli:

$$\frac{P_1}{\rho g} + \frac{V_1^2}{2g} + Z_1 = \frac{P_2}{\rho g} + \frac{V_2^2}{2g} + Z_2 + h$$

Se o fluido for incompressível, temos $V_1 = V_2$ e

$$\frac{P_1}{\rho g} + Z_1 = \frac{P_2}{\rho g} + Z_2 + h$$

onde:

g = aceleração da gravidade.
P_1 = pressão no ponto 1.
P_2 = pressão no ponto 2.
V_1 = velocidade do fluido no ponto 1.
V_2 = velocidade do fluido no ponto 2.
Z_1 = altura do ponto 1.
Z_2 = altura do ponto 2.
h = perda de carga.
ρ = densidade do fluido.

A perda de carga, h, é calculada pela fórmula:

$$h = f \frac{L}{D} \frac{V^2}{2g}$$

onde:

V = velocidade do fluido no duto

L = comprimento do duto

D = diâmetro do duto

f = fator de atrito, admensional

g = aceleração da gravidade

Se fizermos uma análise dimensional, veremos que h tem dimensão de comprimento, em concordância com a equação de Bernoulli. Para transformarmos esta altura em atmosferas, dividimo-la por 10,33. O fator f, de atrito, é calculado através de gráficos (diagrama de Moddy) a partir da rugosidade relativa do duto e do número de Reynolds, expressado por:

$$Rey = \frac{DV\rho}{\mu} \quad \text{ou} \quad Rey = \frac{4Q\rho}{\pi\mu D}$$

onde:

D = diâmetro do tubo

V = velocidade do fluido

ρ = densidade do fluido

μ = viscosidade do fluido

Q = vazão do fluido

em dimensões homogêneas. Reynolds é um número admensional e nos indica se o fluxo é turbulento ou laminar. Se $Rey > 2300$, o fluxo é turbulento. Quanto a f, é função apenas de Reynolds e da rugosidade relativa, ϵ/D.

Ao invés de usarmos gráficos e tabelas para calcular f, usaremos a correlação de *COLEBROOK,* válida para $REY > 10000$. Assim,

$$f = A + B \cdot REY^{-C}$$

onde:

$$A = 0{,}094\left(\frac{\epsilon}{D}\right)^{0{,}225} + 0{,}53\left(\frac{\epsilon}{D}\right)$$

$$B = 88\left(\frac{\epsilon}{D}\right)^{0,44}$$

$$C = 1,62\left(\frac{\epsilon}{D}\right)^{0,134}$$

O programa que analisaremos em breve contém uma sub-rotina que dado o tipo de material de que é feito o tubo e seu diâmetro interno, calcula a rugosidade relativa e as constantes A, B, C. A filosofia de trabalho do programa pode ser definida em três partes:

- tendo diâmetro e vazão, calcular perda de carga;
- tendo vazão e perda de carga, calcular diâmetro;
- tendo diâmetro e perda de carga, calcular vazão.

As duas últimas partes são iterativas.

VARIÁVEIS DO PROGRAMA:

NOME	*TIPO*	*OBSERVAÇÃO*
Q	Real, simples	Vazão, metros cúbicos/hora
D	Real, simples	Diâmetro, polegadas
H	Real, simples	Perda de carga, ATM
L	Real, simples	Comprimento do tubo, metros
VIS	Real, simples	Viscosidade, centipoise
RO	Real, simples	Densidade, gramas/cm^3
I	Inteira, maior que zero	Código do tubo (até 6)
N	Real, simples	Coeficiente do tubo, adm
REY	Real, simples	Reynolds, admensional
F, F1	Real, simples	Fator de atrito, adm
J	Inteira, simples	Controle de loop, de zero a nove

USANDO O PROGRAMA:

À semelhança dos outros programas, as variáveis que este requer, entram via *"INPUT"*, com mensagem. As dimensões das variáveis (m^3/h, pol etc.) são as mais usadas industrialmente. É necessário muita atenção ao entrar com os dados: as dimensões devem ser coerentes com as pedidas nas mensagens. A variável a ser calculada deve constar como zero. Assim, se quisermos calcular o diâmetro, este deverá ser nulo, quando da entrada dos dados.

→ Exemplo 1: Calcular perda de carga, h, a partir dos seguintes dados:

Vazão: 150 m^3/h, água doce
Diâmetro: 4 polegadas

Viscosidade: 1 centipoise (CP) = 0,01 G/CM/SEG
Densidade: 1
Comprimento: 450 m
Tipo de tubo: ferro fundido (código '4')

RESPOSTAS:

Reynolds = 522162 F = 0,0257956
H = 14,905 atmosferas

Assim, existe entre estes dois pontos da tubulação uma queda de pressão devido à fricção do fluido ao longo do trecho.

→ Exemplo 2: Calcular o diâmetro de tubulação que atende às seguintes condições de processo:

Vazão: 180 m^3 /h
Perda de carga: 13,4 ATM
Comprimento: 1000 m
Viscosidade: 3,5 CP
Densidade: 1,5
Tipo de tubo: ferro galvanizado

RESPOSTAS:

Reynolds = 215241 F = 0,0224266
Diâmetro = 4,9905 polegadas

que significa um tubo de aprox. 5", comercialmente.

→ Exemplo 3: Calcular a vazão que ocasiona as seguintes condições de transporte:

Diâmetro: 12 polegadas
Perda de carga: 15 ATM
Viscosidade: 5 CP
Densidade: 0,95
Comprimento: 5200 m

RESPOSTAS:

Reynolds = 179306 F = 0,0172955
Vazão = 842,723 m^3 /h

→ Exemplo 4: Este é um exemplo mais complexo e, conseqüentemente, mais completo.

— Tubulação: 3/4 de polegadas, com:

⇒ 3 válvulas globo, 100% aberta. (L/D = 340)
→ 1 válvula retenção, (L/D = 135)
→ 1 válvula macho, 50% aberta (L/D = 260)
→ 2 válvulas macho, 75% aberta (L/D = 50)

— Vazão: 3 m^3/h de solução salina
— Viscosidade: 1,2 CP
— Densidade: 1,1
— Comprimento: 120 m
— Tubo: ferro fundido

Aqui entramos com o conceito de comprimento equivalente. Quando uma válvula ou qualquer outro acidente tem uma razão L/D = 100, por exemplo, significa dizer que cada unidade de diâmetro dela é equivalente a cem unidades de comprimento do tubo equivalente, em média. Assim, se D = 2", temos

$$L' = \left(\frac{L}{D}\right) * D = 100 * 2'' = 200'' = 508 \text{ cm} = 0{,}508 \text{ m}.$$

Em outras palavras, a perda de carga ocasionada por este acidente é equivalente àquela causada por 0,508 metros da tubulação de 2". No exemplo presente, temos:

$$L_{EQ} = 120 + 0{,}75 * 0{,}0254\,(3 * 340 + 135 + 260 + 2 * 50) = 148{,}8608 \text{ m}$$

(OBS.: 1 pol = 0,0254 m).

Destarte, os acidentes são equivalentes a 28,8608 metros de tubo 3/4". O comprimento total equivalente do tubo é de 120 + 28,8608; igual a 148,8608 metros.

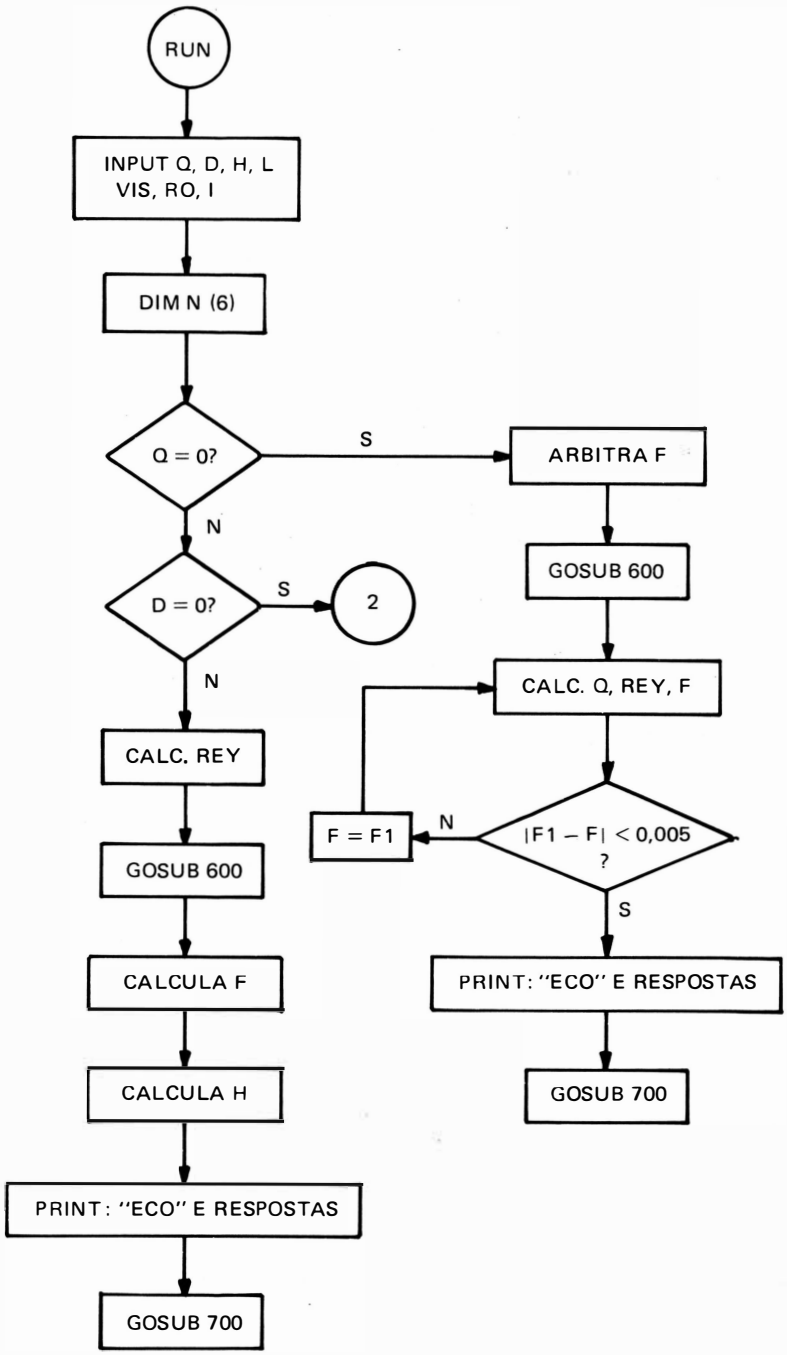
Temos agora todos os dados; e, uma vez alimentando-se com eles o programa, obtemos:

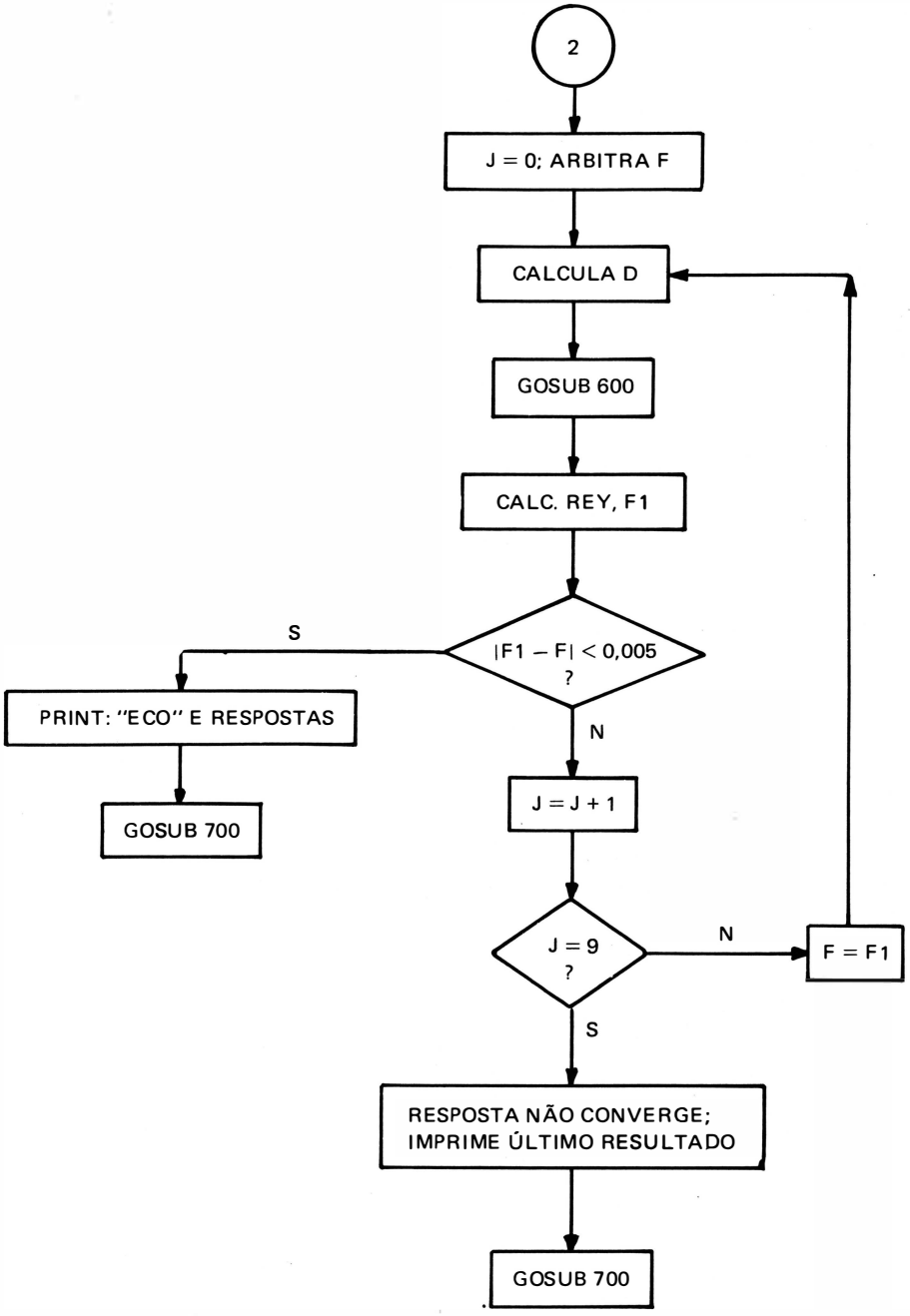
RESPOSTAS:

Reynolds = 51055,8 F = 0,0436205
H = 14,3912 ATM

Experimente executar o programa sem incluir os acidentes, isto é, tomando a tubulação apenas como 120 metros de tubo de ferro fundido. Calcule o h. Quanto (por cento) é menor que a calculada anteriormente? Verifique.

FLUXOGRAMA:


RUN
INPUT Q, D, H, L
VIS, RO, I
DIM N (6)
Q = 0?
S
ARBITRA F
N
GOSUB 600
D = 0?
S
2
N
CALC. Q, REY, F
CALC. REY
F = F1
N
|F1 – F| < 0,005
?
GOSUB 600
S
CALCULA F
PRINT: "ECO" E RESPOSTAS
CALCULA H
GOSUB 700
PRINT: "ECO" E RESPOSTAS
GOSUB 700



Por fim, é bom frisar que pode ser que nenhuma das opções de escolha do tipo de tubo satisfaça o usuário. Sendo assim, na linha 115 deixamos em aberto a possibilidade do usuário colocar a constante N (6) do modo que bem entender, bastando para isso mudar o seu valor para o desejado. O valor de N (6) foi colocado como zero pelo autor apenas para efeito de

sintaxe. Abaixo listamos o valor de outras constantes de tubo para cálculo da rugosidade relativa:

Concreto ($\epsilon = 0{,}01$): N = — 1,8779
Aço rebitado ($\epsilon = 0{,}03$): N = — 0,3948
Aço rebitado ($\epsilon = 0{,}003$): N = — 3,2058
Madeira aparelhada ($\epsilon = 0{,}003$): N = — 3,2058

Os comprimentos equivalentes são facilmente encontráveis na literatura, porém listaremos aqui alguns:

ACIDENTE	L/D
Válvula Globo Standard, 100% aberta	340
Válvula Globo — Tipo X, haste 60°, 100% aberta	175
Válvula Globo — Tipo X, haste 45°, 100% aberta	175
Válvula Angular, 100% aberta	175
Válvula Macho, 100% aberta	13
Válvula Macho, 75% aberta	35
Válvula Macho, 50% aberta	160
Válvula Macho, 25% aberta	900
Válvula Macho, "Quick Closing", 100% aberta	17
Válvula Macho, "Quick Closing", 75% aberta	50
Válvula Macho, "Quick Closing", 50% aberta	260
Válvula Macho, "Quick Closing", 25% aberta	1200
Check Valve, Standard, 100% aberta	135
Curva, 90°, Standard	30
Curva, 90°, Raio longo	20
Curva, 90°, Raio curto	50
Curva, 45°, Standard	16
Curva, 45°, Curva curta	26
Tê, Standard, fluxo direto	20
Tê, Standard, fluxo na derivação	60
Curva de 180°	75
Luva (Uniões)	7

```
VAZAO (M3/H)? 150
DIAMETRO (POL)? 4
PERDA DE CARGA (ATM)?
COMPRIMENTO (M)? 450
VISCOSIDADE (CP)? 1
DENSIDADE (G/CM3)? 1

CO'DIGO DA TUBULACAO:
        ACO COMERCIAL OU FERRO FORJADO: '1'
        FERRO FUNDIDO ASFALTADO: '2'
        FERRO GALVANIZADO: '3'
        FERRO FUNDIDO: '4'
        CONCRETO (EPS=.001): '5'
        ESCOLHA DO USUARIO: '6'?


DADOS DO SISTEMA:

VAZAO= 150        DIAM.= 4          COMPR.= 450

VISC.= 1          DENS.= 1          RUG. REL.= 2.35095E-03



RESPOSTAS:

REY.= 522162      F= .0257956       H= 14.905 ATM
PARA RODAR NOVAMENTE APERTE QUALQUER TECLA.
```

```
10 'PROGRAMA TUBULACOES.
20 'RESOLUCAO PARA VAZAO, PERDA DE CARGA E DIAMETRO.
30 'AUTOR: FAUSTO A. DE A. BARBUTO.    DATA: 05/MAIO/84.
40 CLEAR:CLS:INPUT"VAZAO (M3/H)";Q
50 INPUT"DIAMETRO (POL)";D
60 INPUT"PERDA DE CARGA (ATM)";H
70 INPUT"COMPRIMENTO (M)";L
80 INPUT"VISCOSIDADE (CP)";VIS
90 INPUT"DENSIDADE (G/CM3)";RO
100 PRINT:INPUT"CO'DIGO DA TUBULACAO:
        ACO COMERCIAL OU FERRO FORJADO: '1'
        FERRO FUNDIDO ASFALTADO: '2'
        FERRO GALVANIZADO: '3'
        FERRO FUNDIDO: '4'
        CONCRETO (EPS=.001): '5'
        ESCOLHA DO USUARIO: '6'"; I
105 DIM N(6)
110 N(1)=-6.3362 : N(2)=-5.5215 : N(3)=-5.116
115 N(4)=-4.6001 : N(5)=-4.3836 : N(6)=0
120 Q$="VAZAO=" : D$="DIAM.=" : L$="COMPR.=" : R$="REY.="
130 H$="H=" : V$="VISC.=" : W$="DENS.=" : K$="RUG. REL.="
150 IF Q=0 GOTO 260
160 IF D=0 GOTO 360
170 'SOLUCAO PARA PERDA DE CARGA (H).
180 REY=13924.317*Q*RO/VIS/D
190 GOSUB 600
200 F=A+B*REY[C
210 H=.60366*F*L*Q*Q/D[5
220 CLS:PRINT"DADOS DO SISTEMA:":PRINT@128,Q$;Q,D$;D,L$;L
230 PRINT@256,V$;VIS,W$;RO,K$;K
240 PRINT@512,"RESPOSTAS:":PRINT@640,R$;REY,"F=";F,H$;H/10.33;"ATM"
250 GOSUB 700
260 'SOLUCAO PARA VAZAO (Q).
270 F=.05 : S=SQR(17.1123*H*D[5/L)
275 GOSUB 600
280 Q=S/SQR(F)
290 REY=13924.317*Q*RO/VIS/D
300 F1=A+B*REY[C
310 IF ABS(F1-F)(=.005 THEN 330 ELSE 320
320 F=F1 : GOTO 280
```

```
330 CLS:PRINT"DADOS DO SISTEMA:":PRINT@128,H$;H,D$;D,L$;L
335 PRINT@256,V$;VIS,W$;RO,K$;K
340 PRINT@512,"RESPOSTAS:":PRINT@640,R$;REY,"F=";F1,Q$;S/SQR(F1)
350 GOSUB 700
360 'SOLUCAO PARA DIAMETRO (D).
370 J=0 : F=.05 : S=(.0584376*L*Q*Q/H)[.2
380 D=S*F[.2
390 GOSUB 600
400 REY=13924.317*Q*RO/VIS/D
410 F1=A+B*REY[C
420 IF ABS(F1-F)(=.005 THEN 490 ELSE 430
430 J=J+1
440 IF J=9 THEN 470
450 F=F1
460 GOTO 380
470 CLS:PRINT@320,"A RESPOSTA NAO CONVERGE; U'LTIMO DIAMETRO IGUAL A",D;"POLEGAD
AS"
480 GOSUB 700
490 CLS:PRINT"DADOS DO SISTEMA:":PRINT@128,Q$;Q,H$;H,L$;L
495 PRINT@256,V$;VIS,W$;RO,K$;K
500 PRINT@512,"RESPOSTAS:":PRINT@640,D$;D,R$;REY,"F=";F1
510 GOSUB 700
600 'SUBROTINA DOS PARAMETROS A, B, C, K.
610 K=EXP(-1.048*LOG(D)+N(I))
620 A=.094*K[.225+.53*K
630 B=88*K[.44
640 C=-1.62*K[.134
650 RETURN
700 PRINT@704,"PARA RODAR NOVAMENTE APERTE QUALQUER TECLA."
710 IF INKEY$="" THEN 710 ELSE 40
```

VAZÃO DE GASES INDUSTRIAIS EM DUTOS

32

A mecânica dos fluidos constitui assunto de interesse para aqueles ligados às áreas de engenharia química, de petróleo, de-processamento, química industrial, entre outras. Já abordamos anteriormente questões sobre o transporte de fluidos incompressíveis. Agora é a vez dos gases, compressíveis que são e com propriedades termodinâmicas e cinemáticas, apresentando variações bem distintas daquelas experimentadas por fluidos incompressíveis — qualitativa e quantitativamente — quando sujeitos a modificações de pressão e temperatura dentro do processo em que estão envolvidos. Assim, se para um líquido um aumento de temperatura implica em redução da sua viscosidade, para um gás o efeito é o inverso.

Muito já se falou sobre o transporte de gases, e existem muitas fórmulas e equações empíricas para representar este fenômeno. Entre as mais usadas industrialmente está a *Equação de Weymouth:*

$$Q = K \frac{T_B}{P_B} \left(\frac{P_1^2 - P_2^2}{L\, d_G\, T_f\, Z_m} \right)^{0,5} \cdot D^{8/3} \cdot E$$

onde:

Q = vazão de gás nas condições T_B e P_B, m^3/dia.
T_f = temperatura média de fluxo, Kelvin.
P_1, P_2 = pressões de fluxo, ATM ABS.
L = comprimento do duto, km.
T_B = temperatura-base, k.
P_B = pressão-base, ATM ABS.
d_G = densidade do gás em relação ao ar (ar = 1,0).
E = eficiência do duto, admensional.
D = diâmetro do duto, cm.
Z_m = fator de compressibilidade, calculado a Pm.
K = constante de transformação; K = 1,7396

$$Pm = \frac{2}{3} \left(\frac{P_1^3 - P_2^3}{P_1^2 - P_2^2} \right)$$

Sobre a eficiência do duto, normalmente se usa e = 0,92 para gases secos (sem condensáveis), e = 0,77 para gás com condensáveis (mas ainda sem condensado) e e = 0,6 para fluxo de gás e condensado (fluxo em duas fases).

Quanto ao fator Z_m pode ser obtido através de gráficos (existem bons para misturas de hidrocarbonetos) ou dos métodos usuais. Quanto ao programa que veremos a seguir, se o usuário não tiver um valor de "Z" para com ele alimentar o computador, este dado é calculado internamente pelo próprio programa numa rotina interna. Se Z_m for fornecido, esta rotina é by-passada durante a execução do programa.

A densidade do gás (d_G) é calculada pelo programa quando introduzimos os dados relativos ao(s) componente(s) do gás/mistura gasosa. Esse cálculo toma como base $d_{AR} = 1,0$.

A pressão e temperatura-base são arbitradas pelo usuário. Como usualmente P_B = = 1 ATM ABS e $T_B = 15,5^\circ C$, são esses os valores atribuídos a esses parâmetros, se houver *"DEFAULT"* de parte do usuário.

→ Exemplo 1: Calcular a vazão de gás num duto sob as condições de processo abaixo:

T_B e P_B: admite-se 15,5°C e 1 ATM · ABS.
P_1 = 100 ATM ABS.
P_2 = 50 ATM ABS.
TF = 30°C.

Duto: L = 350 km
D = 30,48 cm (12")
E = 92%

Gás:

Dois componentes: CH_4 (metano) e C_2H_6 (etano) metano, CH_4: mol = 16; percentual molar = 80,4%, PC = 45,4 ATM, TC = 190,6 K etano, C_2H_6: mol = 30; percentual molar = = 19,6%, PC = 48,2 ATM, TC = 305,4 K.

A introdução dos dados começa com P_1 e P_2, $P_1 > P_2$. Se $P_1 < P_2$ o programa volta à linha 40, do *INPUT* desses dados:

```
?
100 <ENTER>
??
50 <ENTER>
```

A temperatura de fluxo é 30°C.

```
?
30 <ENTER>
```

Omitiremos pressão e temperatura-base: o programa admitirá $T_B = 15,5°C$ e 1 ATM ABS.

```
? <ENTER>
? <ENTER>
```

O diâmetro é 30,48 cm.

```
?
30.48 <ENTER>
```

O comprimento é L = 350 km

```
?
350 <ENTER>
```

A princípio, o fator de compressibilidade é zero; deixemos que o programa o calcule internamente.

```
?
<ENTER>
```

O número de componentes é igual a dois.

O mol do componente nº 1 (o metano, CH_4), é 16.

```
?
16 <ENTER>
```

Seu percentual molar, 80,4%

```
?
80.4 <ENTER>
```

A pressão crítica é 45,4 ATM.

```
?
45.4 <ENTER>
```

A temperatura crítica, 190,6 K.

Introduzimos os dados do componente nº 2 do mesmo modo como fizemos para o componente nº 1.

A eficiência do duto é 92%: não há condensáveis na mistura gasosa.

```
?
92 <ENTER>
```

A resposta é: Vazão = 1,45084 × 10^6 m^3/dia a P_B, T_B.

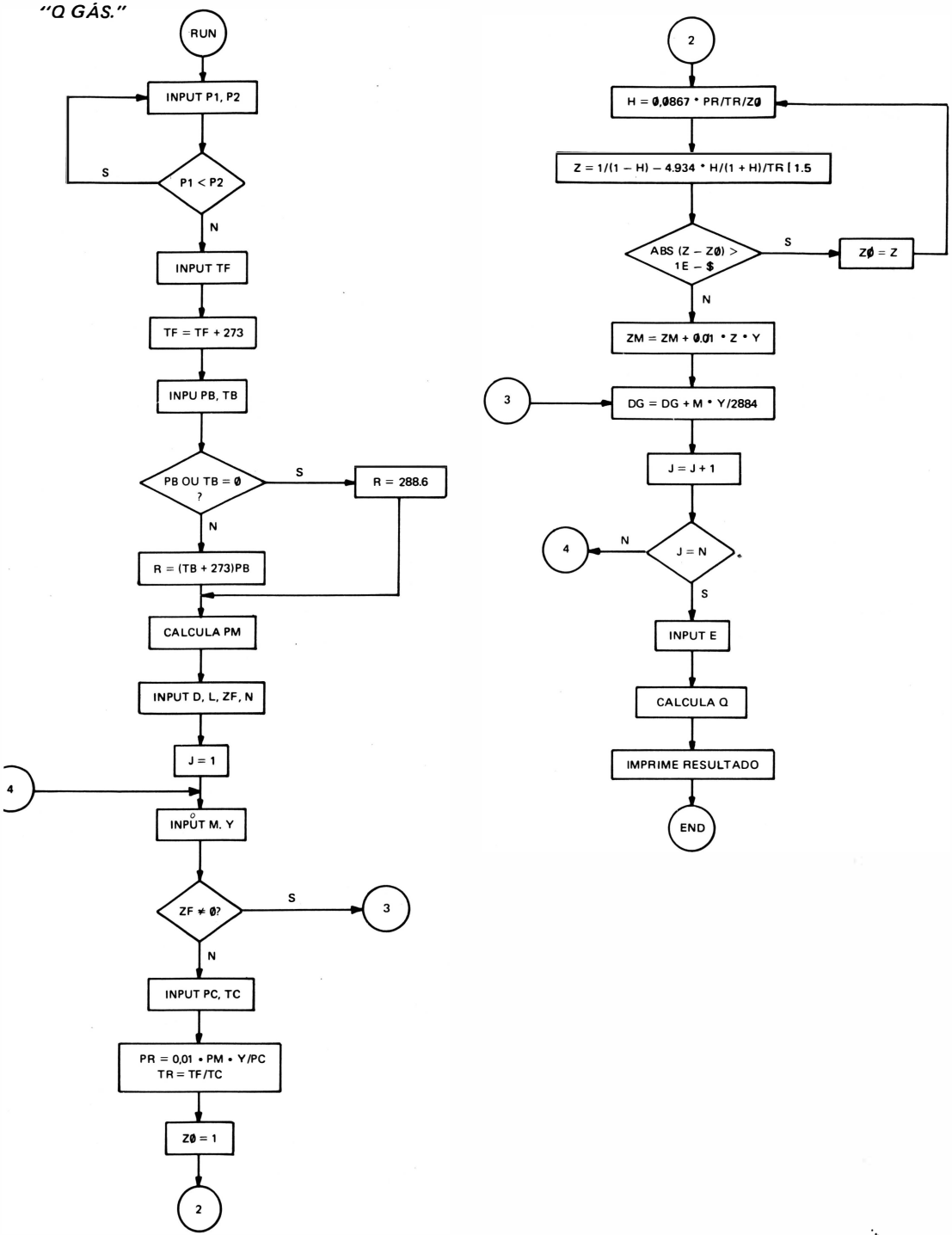
→ Exemplo 2: Nas mesmas condições do exemplo, calcular a vazão de gás com Z = 0,87.

A única diferença agora é que não é mais necessário introduzir pressão e temperatura crítica para cada componente da mistura: esses dados só eram usados para calcular o fator de compressibilidade.

A vazão de gás é igual a 1,48078 × 10^6 m^3/dia.

FLUXOGRAMA:

"Q GÁS."



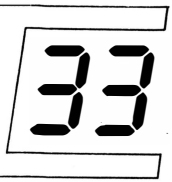
```
P1 E P2 (EM ATM ABS; P1)P2)? 100
?? 50

TEMPERATURA DE FLUXO (GRAUS CELSIUS)? 30

PRESSAO BASE(ATM ABS)?

TEMPERATURA BASE(CELSIUS)?

DIAMETRO (EM CM)? 30.48

COMPRIMENTO (EM KM)? 350

FATOR DE COMPRESSIBILIDADE ? .


COMPONENTE NO. 1 :

MOL: ? 16

PERCENTUAL MOLAR? 80.4

PRESSAO CRITICA(ATM)? 45.4
TEMPERATURA CRITICA(K)? 190.6.




COMPONENTE NO. 2 :

MOL: ? 30

PERCENTUAL MOLAR? 19.6

PRESSAO CRITICA(ATM)? 48.2

TEMPERATURA CRITICA(K)? 305.4




EFICIENCIA DO DUTO (%)? 92.




**********************************************
VAZAO DE GA'S: 1.45084E+06 M3/DIA (CONDICOES BASE)
**********************************************
READY
).
```

```
10 'PROGRAMA QGAS - VAZAO DE GA'S EM UM DUTO - UTILIZANDO-SE
20 'A FORMULA DE WEYMOUTH.
30 'AUTOR: FAUSTO A. DE A. BARBUTO.   DATA: 04/JUL/84.
40 CLS:INPUT"P1 E P2 (EM ATM ABS; P1)P2)"; P1, P2
50 IF P1(P2 THEN 40
60 PRINT:INPUT"TEMPERATURA DE FLUXO (GRAUS CELSIUS)"; TF
70 TF=TF+273
80 PRINT:INPUT"PRESSAO BASE(ATM ABS)"; PB
90 PRINT:INPUT"TEMPERATURA BASE(CELSIUS)"; TB
100 IF PB=0 OR TB=0 THEN R=288.6 ELSE R=(TB+273)/PB
110 PM=2*((P1[3-P2[3)/(P1[2-P2[2))/3
120 PRINT:INPUT"DIAMETRO (EM CM)"; D
130 PRINT:INPUT"COMPRIMENTO (EM KM)"; L
135 PRINT:INPUT"FATOR DE COMPRESSIBILIDADE "; ZM
140 CLS:INPUT"NU'MERO DE COMPONENTES"; N
150 FOR J=1 TO N
160 CLS:PRINT"COMPONENTE NO.";J;":"
170 PRINT:INPUT"MOL: "; M
180 PRINT:INPUT"PERCENTUAL MOLAR"; Y
185 IF ZM()0 THEN 270
190 PRINT:INPUT"PRESSAO CRITICA(ATM)"; PC
200 PRINT:INPUT"TEMPERATURA CRITICA(K)"; TC
210 PR=.01*PM*Y/PC : TR=TF/TC : Z0=1
220 H=.0867*PR/TR/Z0
230 Z=1/(1-H)-4.934*H/(1+H)/TR[1.5
240 IF ABS(Z-Z0))1E-5 THEN Z0=Z
250 IF ABS(Z-Z0))1E-5 THEN 220
260 ZF=ZF+.01*Z*Y
270 DG=DG+M*Y/2884
280 NEXT J
288 IF ZM()0 THEN ZF=ZM
290 CLS:INPUT"EFICIENCIA DO DUTO (%)"; E
300 Q=.017396*R*E*D[(8/3)*SQR((P1[2-P2[2)/L/DG/TF/ZF)
310 CLS:PRINT@448, STRING$(50,"*")
320 PRINT@512,"VAZAO DE GA'S:";Q;"M3/DIA (CONDICOES BASE)"
330 PRINT@576, STRING$(50,"*")
340 END
```

APLICAÇÃO GERAL

PARTE IV

MANUTENÇÃO E GERAÇÃO DE ARQUIVOS



Os arquivos constituem certamente o tipo de *software* que a maioria dos usuários de microcomputadores ou *home computers* deseja ter em sua fitoteca, devido ao seu largo espectro de uso. Uma agenda telefônica, um catálogo da biblioteca, os títulos de uma discoteca, uma carteria de ações etc., podem muito bem ser arquivados para posterior consulta num pequeno espaço de fita K7. É verdade que no caso dos exemplos citados acima, cada caso é um caso particular. No entanto, a dinâmica é a mesma.

Podemos exemplificar essa filosofia com um programa simples, mas eficiente, e que nos faz compreender de maneira inequívoca o funcionamento de um programa-arquivo básico.

Em primeiro lugar, o usuário deve reservar em fita dois locais distintos: um, para armazenar o programa-arquivo; o outro, para guardar os dados que vão ser arquivados. Como não é de bom alvitre misturar programas e arquivos de dados numa só fita, o autor deixa aqui duas sugestões interessantes: a primeira seria reservar um lado da fita para os programas do usuário e o outro lado seria utilizado para o armazenamento de dados. A segunda — e melhor — seria a manutenção de uma fita só para guardar dados. Isso eliminaria a possibilidade de confusão na hora de carregar/descarregar os dados ou o programa. Nosso programa-arquivo simples permite guardar 39 títulos (nomes) e dados relativos a esses nomes. Não é muito, mas é o suficiente para compreendermos o seu funcionamento. Com o decorrer do tempo, elaboraremos arquivos mais específicos e sofisticados.

O nosso programa é constituído de duas partes:

— Programa de entrada/gravação dos dados (linhas 01 a 160)
— Programa verificador/descarregador dos dados (linhas 170 a 300)*

As variáveis dos dois programas são:

NOME DA VARIÁVEL	*DIMENSIONAMENTO*	*COMENTÁRIOS*
NP	Variável não-dimensionada, mas que vai dimensionar outras	Número de títulos a ser arquivado

NOME DA VARIÁVEL	*DIMENSIONAMENTO*	*COMENTÁRIOS*
D$	8 (Oito)	Data do dia em que o arquivo foi gerado
N$	NP	Nome do(s) título(s)
PA$	NP	Dado relativo ao título (nome) a ser arquivado

USANDO O PROGRAMA:

Após digitarmos o programa-arquivo (linhas ØØ1 a 3ØØ). Informamos ao computador que o mesmo está findo com *"BREAK"*. Após obtermos *"READY"* com N. L. (*NEWLINE*) digitamos:

RUN

E em seguida, N. L. aparece no topo do vídeo a pergunta:

NÚMERO DE TÍTULOS?

Esse dado — NP — dimensionará outras variáveis (já vimos isso antes) e comandará os *LOOPS FOR-NEXT* (linhas 19Ø a 21Ø, 7Ø a 9Ø, 12Ø a 14Ø e 26Ø a 28Ø e as linhas de gravação/descarga da própria variável (linhas 11Ø, 17Ø) para controle do programa. Esses títulos compõem uma matriz de NP elementos, bem como os dados relativos a esses títulos. Como exemplo, introduzimos

3 (três)

e NEWLINE. O vídeo então mostra uma interrogação, solicitando o 1º título/nome. Por exemplo:

Carlos (NEWLINE)

Abaixo da 1ª interrogação, aparecem outras duas, que se referem ao título (que pode ser o telefone do "Carlos").

712-4419 (NEWLINE)

Similarmente, introduzimos mais dois nomes e seus respectivos telefones, ex.:

Marlene	<NEWLINE>
5ØØ-2825	<NEWLINE>
Fernando	<NEWLINE>
266-1Ø56	<NEWLINE>

Após este último dado (lembre-se, NP = 3), cria-se o seguinte *DISPLAY*

DATA?

Como a data corresponde a um dimensionamento de oito caracteres, entramos com a data no formato XX/YY/ZZ:

16/Ø2/83

e, *muito importante,* antes de teclar *NEWLINE,* posicionamos o gravador numa área de fita disponível para gravar dados. Assim que o gravador estiver em modo de gravação (*RECORD* + + *PLAY*) executamos *NEWLINE* (o D-8000, por exemplo, tem comando automático de gravação) e os dados são gravados na fita. A mensagem

READY
>–

assinala o final da gravação. Temos agora os dados gravados em fita. Como consultá-los agora? Em primeiro lugar, descarregar na memória do computador se já não o fez o programa-arquivo (não se esqueça – este programa está numa área reservada para programas) e, uma vez que isto tenha sido feito, digite

RUN 17Ø

mas não execute ainda *NEWLINE*. Posicione a fita pouco antes do início da gravação dos dados (se o seu gravador tem contagiros, esta tarefa fica facílima!), coloque o gravador no modo "*PLAY*" e, imediatamente após, *NEWLINE*. Aguarde alguns instantes (o tempo de descarga depende do número de títulos, NP) e os nomes e respectivos telefones aparecerão no vídeo. Se NP > 10, o comando *CONT* listará os próximos dez nomes e telefones. Se NP > 20, deveremos usar *CONT* duas vezes para listar todos os nomes e telefones. Se usarmos a capacidade máxima de armazenamento do programa, devemos usar *CONT* três vezes, até obtermos o 39º nome. O nosso rudimentar arquivo tem as seguintes vantagens:

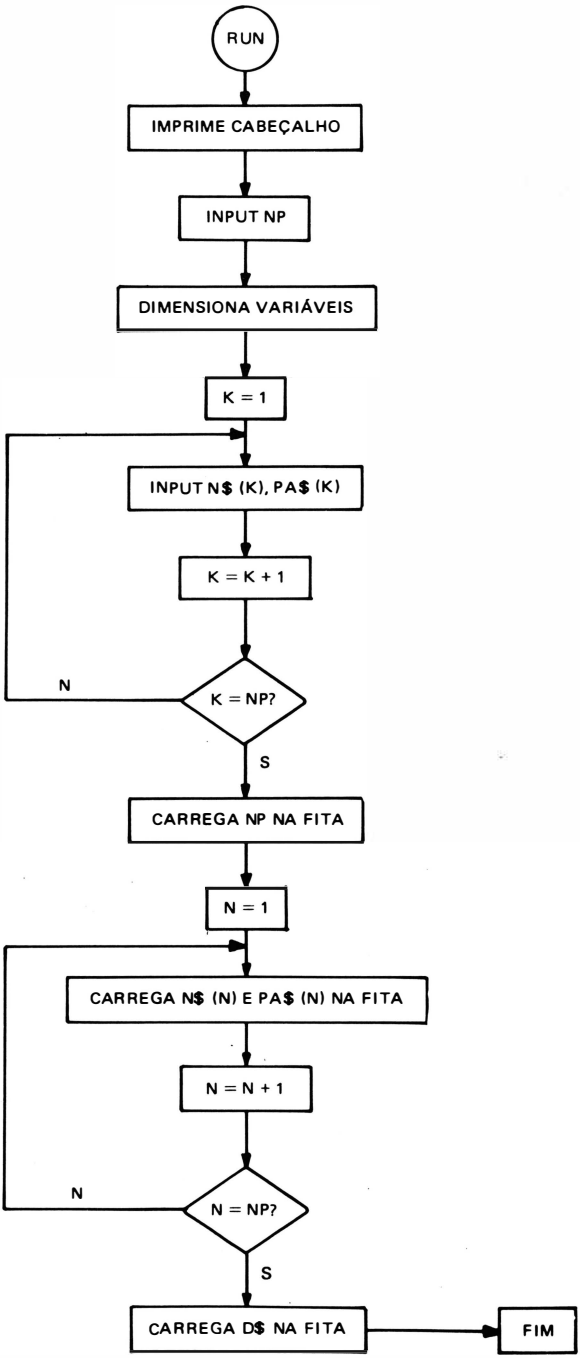
– Simplicidade
– Fácil de usar e digitar
– Ocupa pouca memória.

Por outro lado, notemos que:

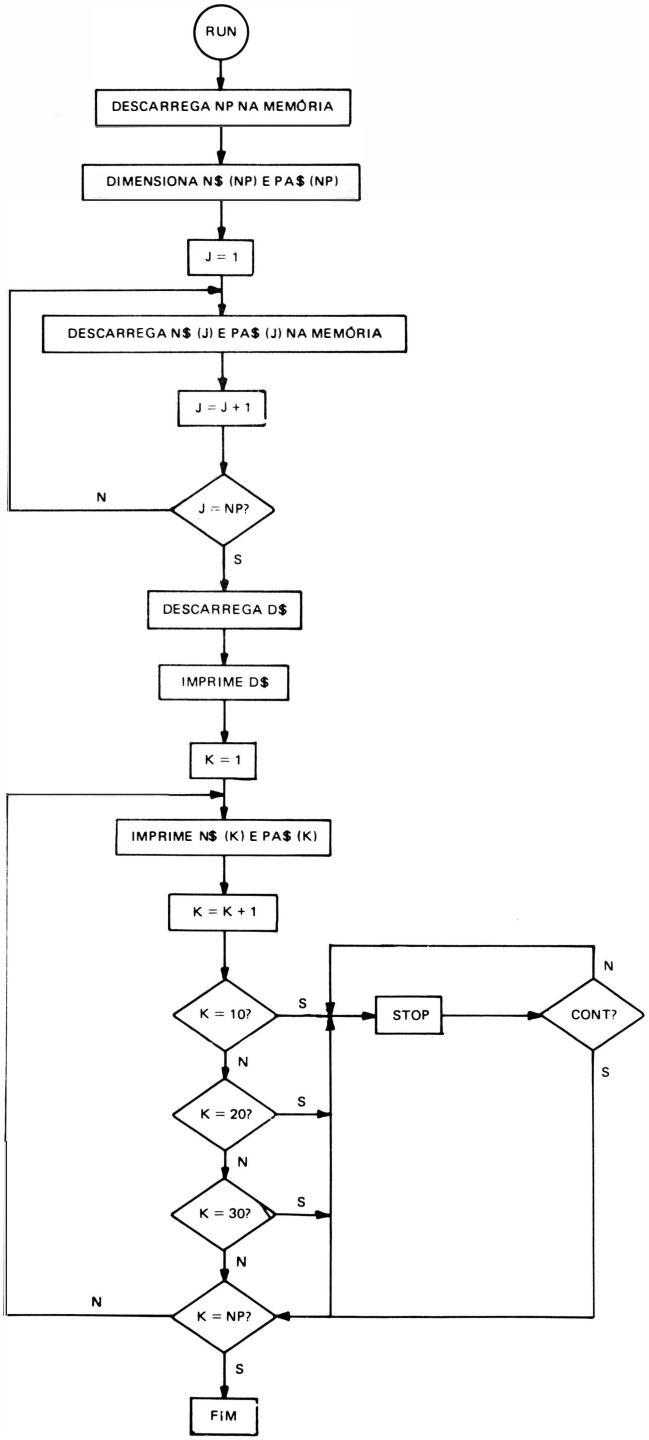
– Só podemos colocar até 39 títulos.
– Uma vez colocados NP títulos (NP < 39), se quisermos colocar mais quantos outros forem necessários, temos de aumentar a área de memória em fita e recolocar os títulos e dados novamente, *um a um.* Aconselhamos a quem usar o arquivo simples que procure esgotar sua capacidade, fazendo NP = 39.

Nos próximos arquivos que apresentaremos, vamos ver como manter as virtudes do programa que acabamos de apresentar, tentando, porém, eliminar suas deficiências.

FLUXOGRAMA 1: PROGRAMA GRAVADOR



FLUXOGRAMA 2: PROGRAMA DESCARREGADOR



```
10 'ARQUIVO SIMPLES.
20 'POR: FAUSTO A. DE A. BARBUTO.   DATA: 06/FEV/84.
30 CLS:PRINT TAB(25)"ARQUIVO SIMPLES."
40 CLEAR 2000
50 INPUT"NU'MERO DE TI'TULOS"; NP
60 DIM D$(8), N$(NP), PA$(NP)
70 FOR K=1 TO NP
80 INPUT N$(K), PA$(K)
90 NEXT K
100 INPUT"DATA:"; D$(8)
110 PRINT#-1,NP
120 FOR N=1 TO NP
130 PRINT#-1,N$(N),PA$(N)
140 NEXT N
150 PRINT#-1,D$(8)
160 END
170 INPUT#-1,NP
180 DIM N$(NP), PA$(NP)
190 FOR J=1 TO NP
200 INPUT#-1,N$(J), PA$(J)
210 NEXT J
220 INPUT#-1,D$(8)
230 CLS
240 PRINT@28,"DATA: ";D$(8)
250 PRINT@64,"TITULO","CR$"
260 FOR K=1 TO NP
270 PRINT N$(K), PA$(K)
280 IF K=10 OR K=20 OR K=30 THEN STOP
290 NEXT K
300 END
```

EDITOR ASSEMBLY

34

"ASSEMBLY", em inglês, significa montar, construir, juntar. Na língua dos computadores, *ASSEMBLY* é a linguagem do microprocessador, através da qual podemos "falar" diretamente com ele. Os microcomputadores que usam *BASIC* possuem, na *ROM* (*READ ONLY MEMORY*) um programa tradutor que transforma o *BASIC* numa série de comandos inteligíveis pelo microprocessador, e vice-versa. Esse programa tradutor ou interpretador, não é volátil, isto é, não desaparece quando se desliga o micro. Os programas em *BASIC* são voláteis; eles são colocados na *RAM* (*RANDOM ACCESS MEMORY*) e extinguem-se quando desenergiza-se o aparelho. Como a *RAM* é a única área livre na memória do micro para colocarmos nossos programas, é aí também que vamos colocar nosso programa em linguagem de máquina (leia-se *ASSEMBLY*).

A primeira coisa a fazer é proteger na memória uma área destinada a receber nossas rotinas em *ASSEMBLY*. Entenda-se por "proteger" a tarefa de reservar um espaço na *RAM* de modo a que nada — nem o programa *BASIC* — nele interfira, e, conseqüentemente, altere as rotinas que aí estão. A área de memória *RAM* nos TRS-80 16K modelo I organiza-se assim:

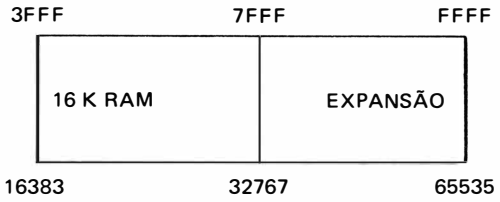


FIG. 1. Organização de memória *RAM* no TRS-80 16 K Mod. I.

Vemos que a memória inicial tem 32767 − 16383 = 16384 *BYTES*, ou 16 K. Os *BYTES* são alocados em potências de dois. Assim, 1 K *BYTE* = 2^{10} = 1024 *BYTES*. Assim, 16 *BYTES* são 16384 *BYTES*, ou 2^{14}. Em cima do gráfico, pusemos o endereço *HEX* das posições de memória. A memória *RAM* vai do endereço 16383 ao 32767, ou de 3FFF a 7FFF em hexade-

cimal. De 7FFF a FFFF é a área que será preenchida pela expansão de memória; com ela, o micro terá sua *RAM* ampliada para 65535 − 16383 = 49152, ou 48 *KBYTES*. Como o "*KEY LEVEL*" deste livro é para micros com 16 K, de agora em diante não falaremos mais em expansão de memória e todas as nossas considerações serão feitas para os 16 *KBYTES* iniciais.

Bem, e como proteger uma área na *RAM* para a rotina em linguagem de máquina?

Em primeiro lugar, vamos definir o local da memória que vamos utilizar. Geralmente, utilizam-se os últimos endereços da memória, e vamos supor que a nossa rotina *ASSEMBLY* vai iniciar no endereço 30000. Temos, portanto, 2767 *BYTES* para colocar nosso programa em código de máquina. Além disso, vamos transformar 29999 de forma a poder armazená-lo como dois números de oito *BITS* na memória. Isto é necessário porque cada posição de memória pode conter no máximo 8 *BITS*. Assim, o maior número que podemos representar numa posição de memória é:

$$1\,1\,1\,1\,1\,1\,1\,1 = (255)_{10} = (FF)_{16}$$

Convém recordar que é um número binário. Dessa maneira, números maiores que 255 têm que ser "quebrados" para serem colocados na memória. Isto é feito assim:

$$N = 256 * INT\,(N/256) + [N - 256 * INT(N/256)]$$

BYTE MAIS SIGNIFICANTE — *BYTES* MENOS SIGNIFICANTE

(Se somarmos as duas parcelas da equação acima teremos, é lógico, N!)

Após isto ter sido feito colocamos, com uma instrução "*POKE*" o *BYTE* menos significante na memória 16526 e o mais significante na memória 16527. No caso da memória 29999, fica:

BYTE MAIS SIGNIFICANTE: INT (30000/256) = 117
BYTE MENOS SIGNIFICANTE: 30000 − 117 * 256 = 47

e, então:

POKE 16526,48
POKE 16527,117

Feito isto, reservamos a área de memória a partir de 30000. Inicialmente isto pode parecer um tanto ou quanto complicado para o leitor, embora não haja razões para desânimo, uma vez que o programa se encarrega desta etapa inicial, bastando que se introduza o endereço inicial (ADR) onde irá ficar a nossa rotina *ASSEMBLY*. De mais a mais, tentar explicar como e por que tudo isso acontece, e é necessário que aconteça, obrigar-nos-á a nos estendermos muito mais no assunto e não é essa a nossa intenção.

Bem, estamos aptos a introduzir nossa rotina *ASSEMBLY.* Aqui neste ponto o programa se bifurca em dois caminhos diferentes: introdução em hexadecimal ou decimal. Relembrando: colocaremos os *códigos* das instruções de máquina a partir de determinado endereço, de modo a formar a rotina *ASSEMBLY.*

A introdução em hexa tem uma vantagem: podemos colocar os códigos como uma seqüência, formando uma *STRING.* Em se tratando dos códigos:

```
ØE
11
AF
ØD
CD
12
BE
BØ
19
F7
C9
```

podemos introduzi-los como:

```
ØE11AFØDCD12BEBØ19F7C9   <NEWLINE>
```

ou

```
ØE11AFØDCD12      <NEWLINE>
BEBØ19F7C9        <NEWLINE>
```

ou então de par em par:

```
ØE   <NEWLINE>
11   <NEWLINE>
.
.
.
.
.
.
C9   <NEWLINE>
```

Importante: "ØE" é diferente de "E". A *STRING* a ser colocada para introdução tem que ter número par de caracteres – veja linha 180 – não importando o seu tamanho. Espaços em branco no início, meio ou fim da cadeia também darão erro.

É óbvio que estamos falando em termos de programação; matematicamente, ØE = E = = 14, pois o zero à esquerda não é significativo, isto é, é nulo para a 1ª potência de 16 (16^1).

É uma boa prática introduzirmos de 20 em 20 pares, a fim de que possamos colocar todo o "eco" dos dados introduzidos dentro do *DISPLAY* e também evitando os possíveis erros.

A *STRING* recebida como *INPUT* é quebrada, da esquerda para a direita, em pedaços de 2 caracteres (número máximo representado: FF = 255), representando um nº hexadecimal que são transformados para base 10 e por meio de um *POKE* colocados na memória reservada no início da execução. Vemos isso melhor representado assim:

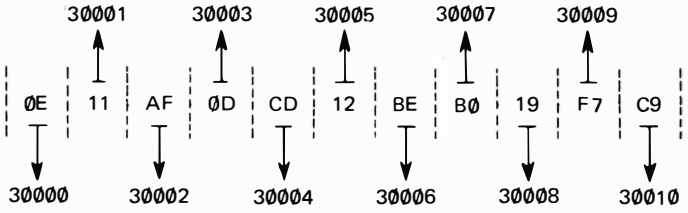


FIG. 1. Colocando a *STRING* nos endereços a partir de 30000.

O *POKE* não se faz em hexa; temos por exemplo:

POKE 30002, 175

e não:

POKE 30002, AF

VARIÁVEIS DO PROGRAMA:

ENDEREÇADOR HEXA

NOME	*TIPO/FUNÇÃO*
P	Simples inteira; incremento de ADR
ADR	Inteira simples; endereço inicial da rotina *ASSEMBLY*
MS	Inteira simples; byte mais significante
MN	Inteira simples; byte menos significante
R	Inteira simples; controle de escolha de dados
A$	Alfanumérica; string hexa a decompor e introduzir
X	Alfanumérica; finaliza rotina de entrada hexa

NOME	*TIPO/FUNÇÃO*
N	Inteira simples; controla o loop de decomposição e introdução da string hexa A$
J	Inteira simples; controle do lopp de decomposição e introdução de A$; controla a quebra e conversão de A$
B$	Alfanumérica; 1º caracter dos pares de caracteres obtidos a partir da "quebra" de A$
C$	Alfanumérica; 2º caracter dos pares acima
B	Inteira simples; valor numérico do 1º caracter dos pares da string A$
C	Inteira simples; valor numérico do 2º caracter dos pares de A$
S	Inteira simples; número decimal relativo a cada par de A$. Valor máximo: $(255)_{10}$

ENDEREÇADOR DECIMAL

NOME	*TIPO/FUNÇÃO*
P	Inteira simples; mesma função do endereçador hexa
D	Inteira simples; código decimal a ser colocado na(s) memória(s)
R$	Alfanumérica; controle de delação do editor *ASSEMBLY;* comum à rotina hexa.

→ Exemplo 1: Colocar, a partir do endereço 30000, os seguintes códigos:

Em hexa	*Em decimal*
Ø1	1
FF	255
Ø3	3
21	33
ØØ	Ø
3C	6Ø
36	54
4Ø	64
11	17
Ø1	1
3C	6Ø
ED	237
BØ	176
C9	2Ø1

Estes códigos correspondem aos mnemônicos *ASSEMBLER:*

ENDEREÇO	*HEX*	*MNEMÔNICO*	*COMENTÁRIOS*
30000	01FF03	LD BC; 1023	; Coloca 1023 ('03FF') no par BC
30003	21003C	LD HL, 15360	; Coloca 15360 ('3C00') no par HL
30006	3640	LD (HL), 64	; Coloca 64 ('40') no endereço indicado por HL
30008	11013C	LD DE, 15361	; Coloca 15361 ('3C01') no par DE
30011	EDB0	LDIR	; Incrementa ponteiros e decrementa BC
30013	C9	RET	; Retorna ao *BASIC*

Explicaremos mais tarde como funciona este programa.

(Linguagem *ASSEMBLY* não é exatamente a especialidade do autor, embora de vez em quando ele "dê seus pulinhos" nesta área).

Vamos rodar o programa como de costume: com <*RUN*> e <*NEWLINE*>. A primeira mensagem pede que se defina o endereço inicial. Respondamos:

? 29999 <NEWLINE>

A segunda pede a maneira como vamos fornecer os dados: em hexa ou em decimal. O número "1" executa entrada com hexadecimal, e "2" entrada decimal. Faremos inicialmente entrada com hexadecimal; portanto:

? 1 <NEWLINE>

A terceira mensagem do *DISPLAY* solicita a(s) *STRING* (S) que é o conjunto de códigos hexadecimais que constitui o programa.

CÓDIGO HEXA? –

De uma só vez, faremos:

01FF0321003C3640110 13CEDB0C9 <NEWLINE>

e, desta forma, aparece um mapa da memória, com o endereço, os códigos hexadecimal e decimal para que o leitor confira se teve sucesso na introdução dos códigos. Como a rotina já foi totalmente introduzida, respondemos com "X" à pergunta:

CÓDIGO HEXA? X <NEWLINE>

A tela se limpa e surge no centro da tela:

SEU PROGRAMA EM *ASSEMBLY* JÁ ESTÁ INTRODUZIDO
A PARTIR DA MEMÓRIA 29999.

DESEJA APAGAR O PROGRAMA EDITOR?
(S/N)? –

Por enquanto, faremos

N <NEWLINE>

Após "*READY*", estamos prontos para executar a rotina *ASSEMBLER* que está no endereço 30000; digitamos diretamente

K = USR (0) <NEWLINE>

e podemos observar o que faz a nossa rotina *ASSEMBLY*. (Trata-se de um resultado gráfico, e não numérico).

Compare a velocidade de execução com o seguinte programa equivalente em *BASIC:*

```
10 FOR K = 1 TO 1023
20 PRINT "@";
30 NEXT K
40 END
```

Como o programa está em código-máquina, poderíamos executá-lo fazendo uso do comando "*SYSTEM*", assim ao obtermos

READY
>–

executaríamos "*SYSTEM*":

>SYSTEM <NEWLINE>

responderíamos à interrogação abaixo

* ?

com

/30000 (Barra, endereço inicial da rotina) e <NEWLINE>.

ATENÇÃO: Deste modo saímos de *BASIC* por um simples *BY-PASS* ao interpretador contido na *ROM*. Para voltarmos ao *BASIC,* executaremos "*RESET*" e tudo OK.

Para introduzirmos os códigos em decimal voltemos ao início, com "*RUN*". Não importa que já tenha sido escrita uma rotina a partir do endereço 30000; vamos escrever o mesmo programa, desta vez em decimal, "por cima" do primeiro. Não importa o modo de edição: o programa é o mesmo, só muda a forma de introduzí-lo.

A única diferença é que se escolha "2" como senha no "menu" apresentado no *DISPLAY*. Os dados são introduzidos seqüencialmente e intercalados por <NEWLINE>. Assim, faremos:

```
CO'DIGO DECIMAL? 1      <NEWLINE>
30000 - - - - - - - - - - - → 1
CO'DIGO DECIMAL? 255    <NEWLINE>
30001 - - - - - - - - - - - → 255
CO'DIGO DECIMAL? 3      <NEWLINE>
30002 - - - - - - - - - - - → 3
```

e assim por diante, até o último código, 201 (RET). Cuidado para não errar na introdução! São ao todo 14 códigos. Qualquer erro ou esquecimento fará com que as coisas se compliquem e o programa perderá sua finalidade ou então tudo vai ficar "muito louco". Após este encerramos com um número maior que 255, por exemplo, 333 e <NEWLINE>.

Tudo certo, agora. A sub-rotina já foi totalmente colocada nas posições de memória 30000 a 30013. A execução é feita de modo idêntico:

```
K = USR (0)
```

ou "*SYSTEM*" com /30000 após "*?".

→ Exemplo 2: Experimente trocar o código $(40)_{16}$ — correspondente a 64 decimal por $(2A)_{16}$, ou $(40)_{10}$. O que acontece? Por quê?

Vamos para um exemplo mais interessante.

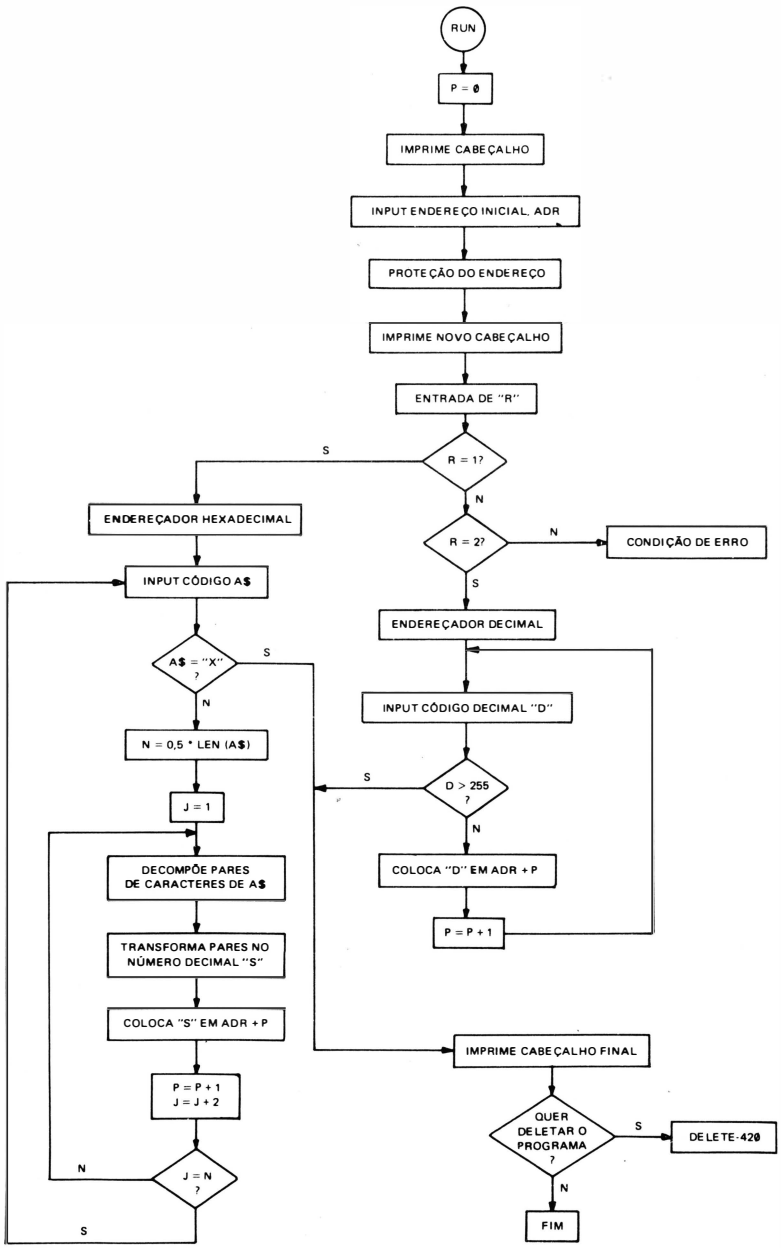
→ Exemplo 3: Introduzir, a partir da memória 32000, os seguintes códigos em hexadecimal:

```
CD C9 01
CD B3 1B
23
CD 6C 0E
CD 77 09
CD B1 0A
CD 09 08
CD BD 0F
21 30 41
CD A7 28
C3 B2 02
```

Após a introdução, rodar com "*SYSTEM*" e /32000. Surge no alto da tela uma interrogação (pedido de dado). Digitaremos 152.46 e <NEWLINE>. Abaixo aparece: 5.0269, que é o logaritmo natural (neperiano) de 152.46. Temos agora, a partir da memória 32000, um programa que calcula o *LOG* natural de qualquer número menor que 1.7E38. Após a execução, os comandos finais C3B202 remetem o sistema ao modo de "*SYSTEM*", aparecendo abaixo da

resposta a interrogação "*?". Quer executar novamente? Ótimo, faça /32000 novamente. Experimente outro número, por exemplo, 32,079. A resposta é 3,4682.

Se trocarmos o comando C3B202 (JP SYSTEM) para C3037D (JP 32003) eliminaríamos o inconveniente de o sempre termos de fazer /32000 para reexecutar o programa. Após a resposta, colocaríamos um novo número e assim até cansar e acionar *"BREAK"* ou *"RESET"* (único jeito de sairmos do programa sem erro de sintaxe). Experimente! Procure também descobrir o que representam os códigos de máquina acima.



```
**************** E D I T O R   A S S E M B L Y ****************

A) ENTRE COM O ENDERECO A PARTIR DO QUAL VOCE VAI COLOCAR O SEU
PROGRAMA ASSEMBLY
? 30000
B) ESCOLHA AGORA COMO INTRODUZIR OS CO'DIGOS:
            HEXADECIMAL: ENTRE COM '1'.

            DECIMAL: ENTRE COM '2'.
? 1.



CO'DIGO HEXA? 0E11AF0DCD12BEB019F7C9.
```

```
CO'DIGO HEXA? 0E11AF0DCD12BEB019F7C9
ENDERECO      HEXA      DECIMAL    ENDERECO      HEXA      DECIMAL
 30001   ----) 0E ----)   14        30002   ----) 11 ----)   17
 30003   ----) AF ----)   175       30004   ----) 0D ----)   13
 30005   ----) CD ----)   205       30006   ----) 12 ----)   18
 30007   ----) BE ----)   190       30008   ----) B0 ----)   176
 30009   ----) 19 ----)   25        30010   ----) F7 ----)   247
 30011   ----) C9 ----)   201
CO'DIGO HEXA? X.
```

```
CO'DIGO HEXA? 01FF0321003C3640011013C
ENDERECO      HEXA      DECIMAL    ENDERECO      HEXA      DECIMAL
 30001   ----) 01 ----)   1         30002   ----) FF ----)   255
 30003   ----) 03 ----)   3         30004   ----) 21 ----)   33
 30005   ----) 00 ----)   0         30006   ----) 3C ----)   60
 30007   ----) 36 ----)   54        30008   ----) 40 ----)   64
 30009   ----) 11 ----)   17        30010   ----) 01 ----)   1
 30011   ----) 3C ----)   60
CO'DIGO HEXA? EDB0C9
ENDERECO      HEXA      DECIMAL    ENDERECO      HEXA      DECIMAL
 30012   ----) ED ----)   237       30013   ----) B0 ----)   176
 30014   ----) C9 ----)   201
CO'DIGO HEXA? X
```

```
10 'EDITOR ASSEMBLY; ENTRADA EM HEXADECIMAL OU DECIMAL.
20 'POR: FAUSTO ARINOS DE ALMEIDA BARBUTO.   DATA: 28/JAN/84
30 CLS: P=0
40 PRINT"**************** E D I T O R   A S S E M B L Y ****************"
50 PRINT@192,"A) ENTRE COM O ENDERECO A PARTIR DO QUAL VOCE VAI COLOCAR O SEU PR
OGRAMA ASSEMBLY"
60 INPUT ADR
70 MS=INT(ADR/256)
80 MN=ADR-256*MS
90 POKE 16526,MN: POKE 16527,MS
100 PRINT@384,"B) ESCOLHA AGORA COMO INTRODUZIR OS CO'DIGOS:"
110 PRINT@460,"HEXADECIMAL: ENTRE COM '1'."
120 PRINT@588,"DECIMAL: ENTRE COM '2'."
130 INPUT R
140 ON R GOTO 150, 300
150 'ENDERECADOR HEXA.
155 CLS
160 INPUT"CO'DIGO HEXA";A$
165 PRINT"ENDERECO     HEXA     DECIMAL   ENDERECO     HEXA     DECIMAL"
170 IF A$="X" THEN 380
180 N=.5*LEN(A$)
190 FOR J=1 TO N
200 B$=MID$(A$,J*2-1,1)
210 C$=MID$(A$,J*2,1)
220 IF ASC(B$)>57 THEN B=ASC(B$)-55 ELSE B=VAL(B$)
230 IF ASC(C$)>57 THEN C=ASC(C$)-55 ELSE C=VAL(C$)
240 S=16*B+C
250 POKE ADR+P,S
260 PRINT ADR+P;" ----) ";B$+C$;" ----) ";S,
265 P=P+1
270 NEXT J
280 PRINT: GOTO160
300 'ENDERACADOR DECIMAL.
310 CLS: P=0
320 INPUT"CO'DIGO DECIMAL";D
330 IF D>255 THEN 380
340 POKE ADR+P,D
350 PRINT ADR+P;" ----) ";D,
360 P=P+1
370 GOTO320
380 CLS:PRINT@320,"SEU PROGRAMA EM ASSEMBLY JA' ESTA' INTRODUZIDO A PARTIR DA","
MEMO'RIA";ADR;"."
390 PRINT@448,"DESEJA APAGAR O PROGRAMA EDITOR?"
400 INPUT"(S/N)";R$
410 IF R$="S" THEN DELETE-420 ELSE 420
420 END
```

PROGRAMA ARQUIVO, PARA NOMES E TELEFONES

35

Um arquivo rudimentar e limitado já foi apresentado com o subtítulo de "manutenção e geração de arquivos". Este arquivo tinha limitações para quem o tentasse usar, entre elas incapacidade de gravar mais de 39 títulos, impossibilidade de inserção e/ou deleção de títulos etc., contrabalançando com sua simplicidade e pouco uso de memória. Desta vez exibiremos um programa-arquivo que cobre muitas deficiências do arquivo simples a que nos referimos. Com efeito, poderemos agora inserir, gravar em fita, listar ou deletar 50 nomes e respectivos telefones contendo 10 e 11 caracteres alfanuméricos, respectivamente. Apesar de, a princípio, só podermos usar 100 matrizes alfanuméricas, pode o usuário "abrir" mais memória para inserir mais títulos ou então um maior número de títulos, ou ambas as coisas. De início, 2 *KBYTES* bastam para esta agenda telefônica (ver linhas 60 e 290). O programa é fácil de usar, auto-explicativo (embora exija um conhecimento menos superficial do *BASIC TRS-80* para melhor compreensão do seu funcionamento) e, à medida que se o for usando, simples. A forma de agenda telefônica é apenas um formato básico e pode ser alterada de forma a atender os interesses do usuário.

Quando se está usando o programa pela primeira vez, a seqüência lógica é entrar com os nomes e telefones, (opção 1), listar de novo (opção 4) e não simplesmente listar (opção 3), coisa que só deve ser feita para recuperar os dados e gravar após verificar que não houve nenhum erro (opção 2). Se houver erro, corrigi-los após listagem teclando diretamente.

Se o caso é recuperar os dados em fita, a seqüência é: listar (opção 3). No fim da listagem, responder convenientemente aos desvios para deleção/inserção com "D", "I" ou <ENTER> se não houver nada a deletar e/ou inserir. Se quiser listar novamente, opção 4. Esta última termina com "*BREAK IN 390*". O comando "*CONT*" devolve o menu no *DISPLAY*.

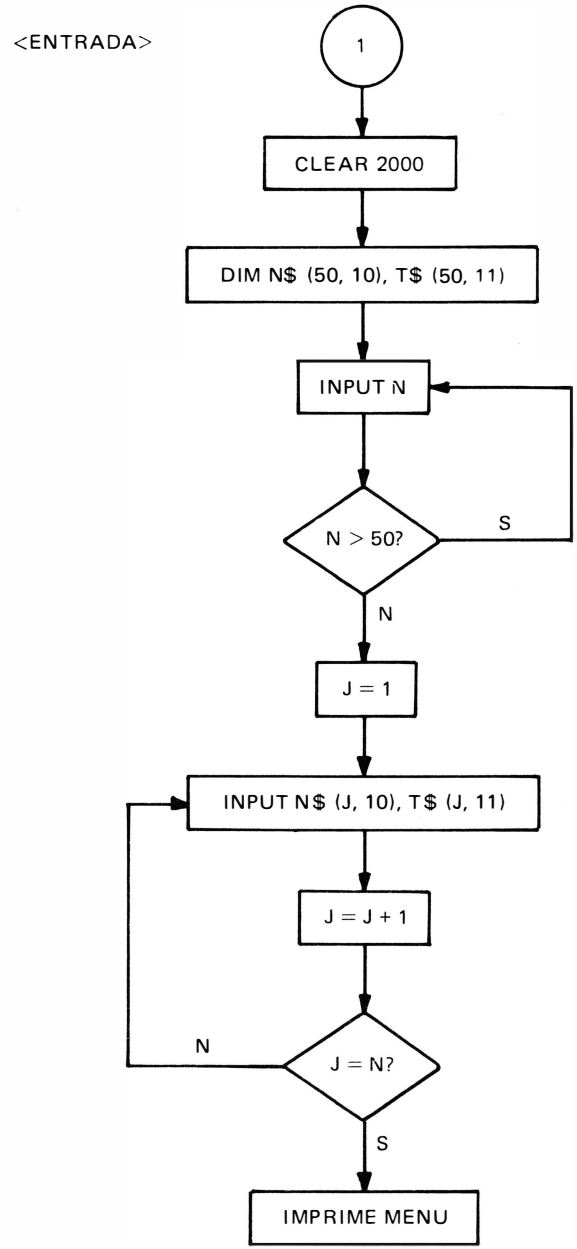
Deleções ou inserções também necessitam ser recuperadas em fita após serem feitas, isto é, qualquer modificação no arquivo original deve ser regravada por cima deste, ou ao menos regravada à parte.

A opção 7 encerra a execução do programa.

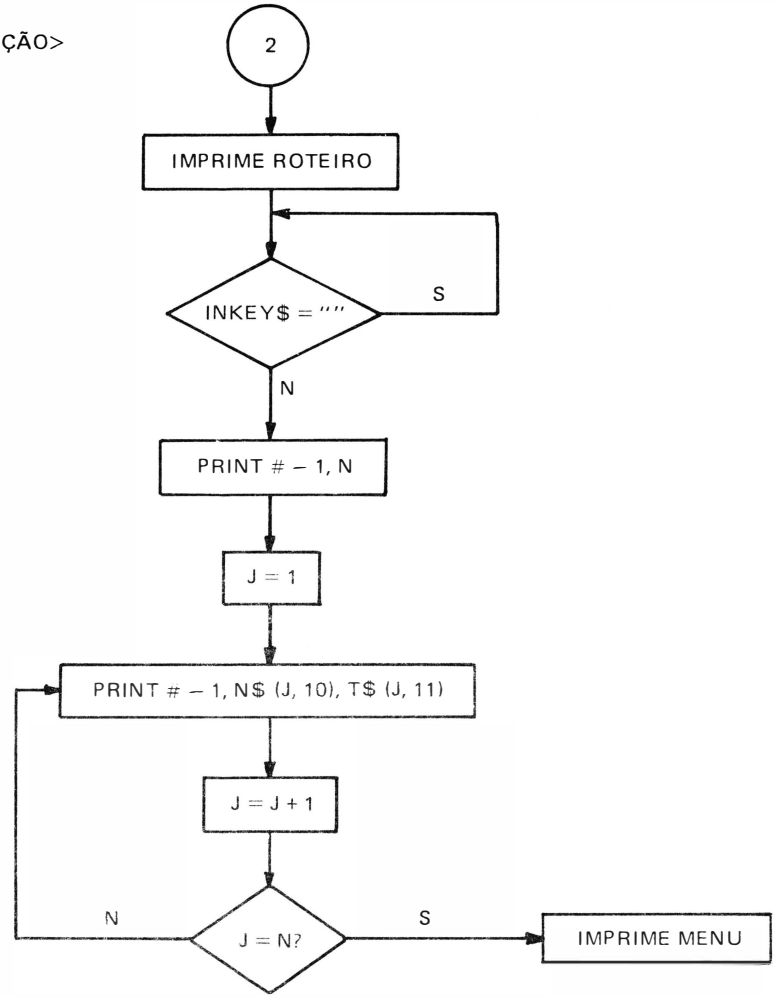
Uma última observação: se, a princípio, uma vez introduzidos os nomes e telefones, desejar o usuário deletar/inserir alguns nomes e respectivos telefones pode fazê-lo antes de gravar. Em qualquer ponto do programa podemos retornar ao menu com <*BREAK*> e *GOTO* 660.

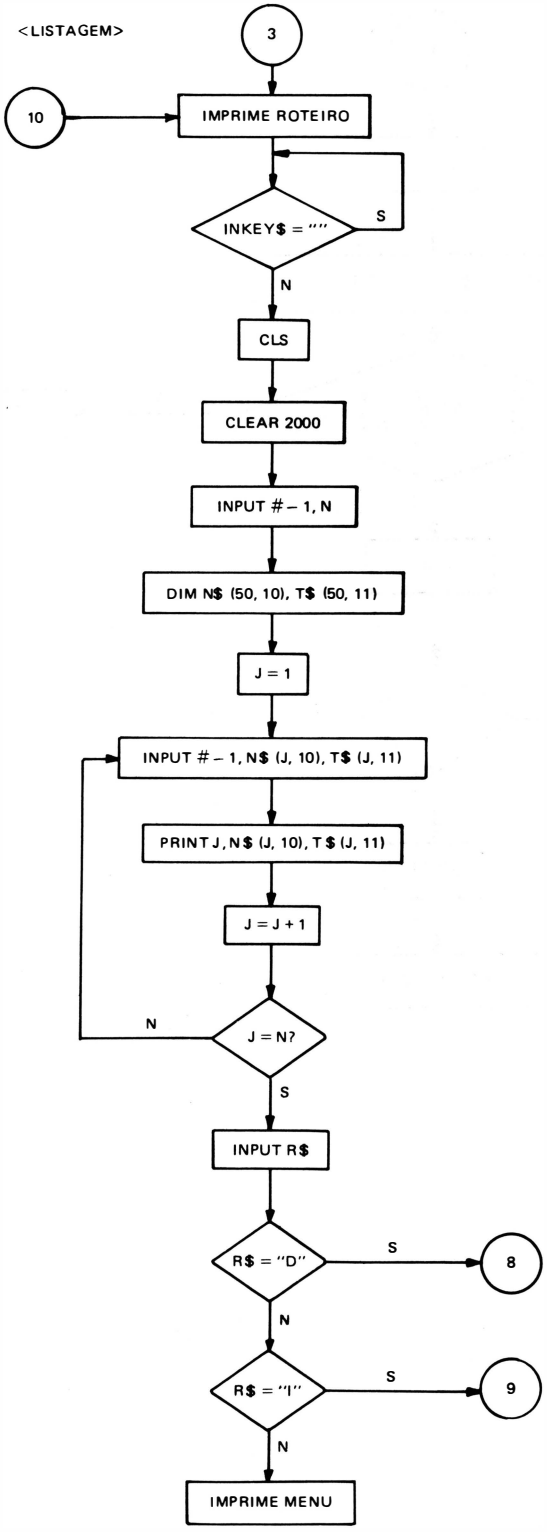
Conforme frisamos anteriormente, só com o uso contínuo do programa será possível dominar de modo total sua filosofia e explorar suas possibilidades e até mesmo modificá-lo aqui ou ali, no sentido de melhorá-lo.

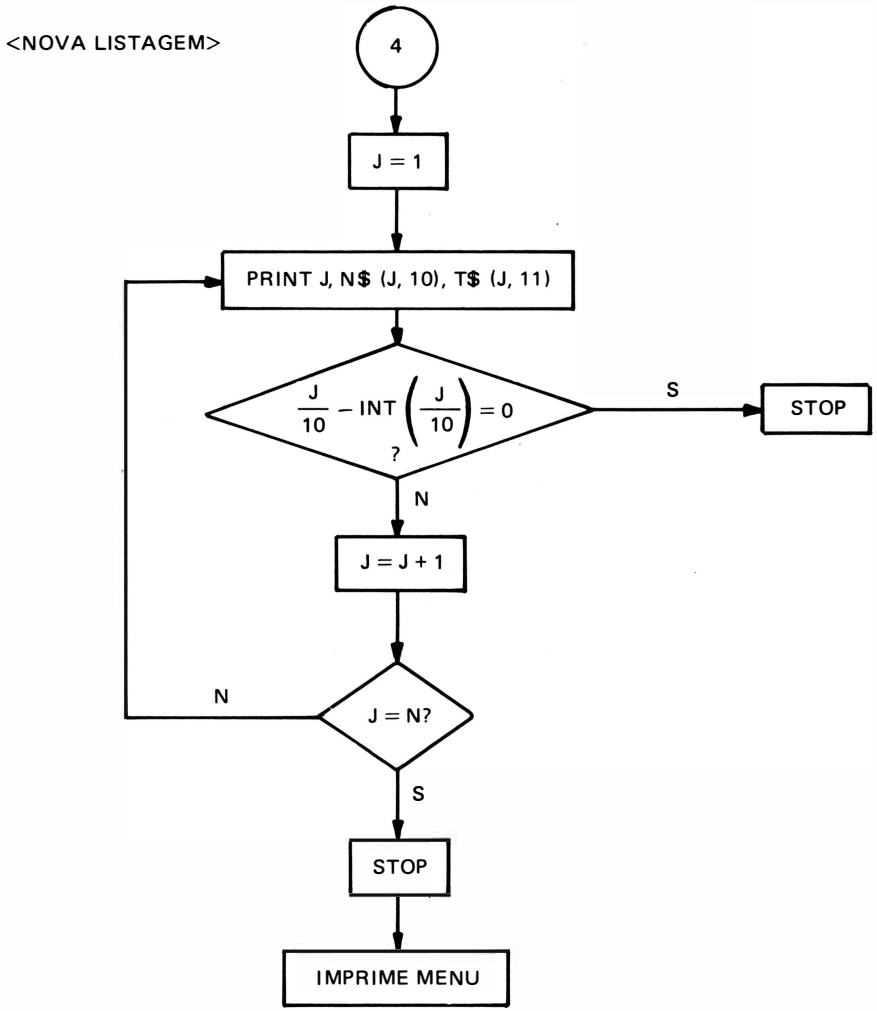
FLUXOGRAMA: PROGRAMA ARQUIVO



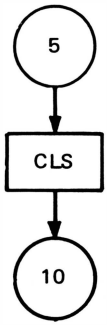
<GRAVAÇÃO>

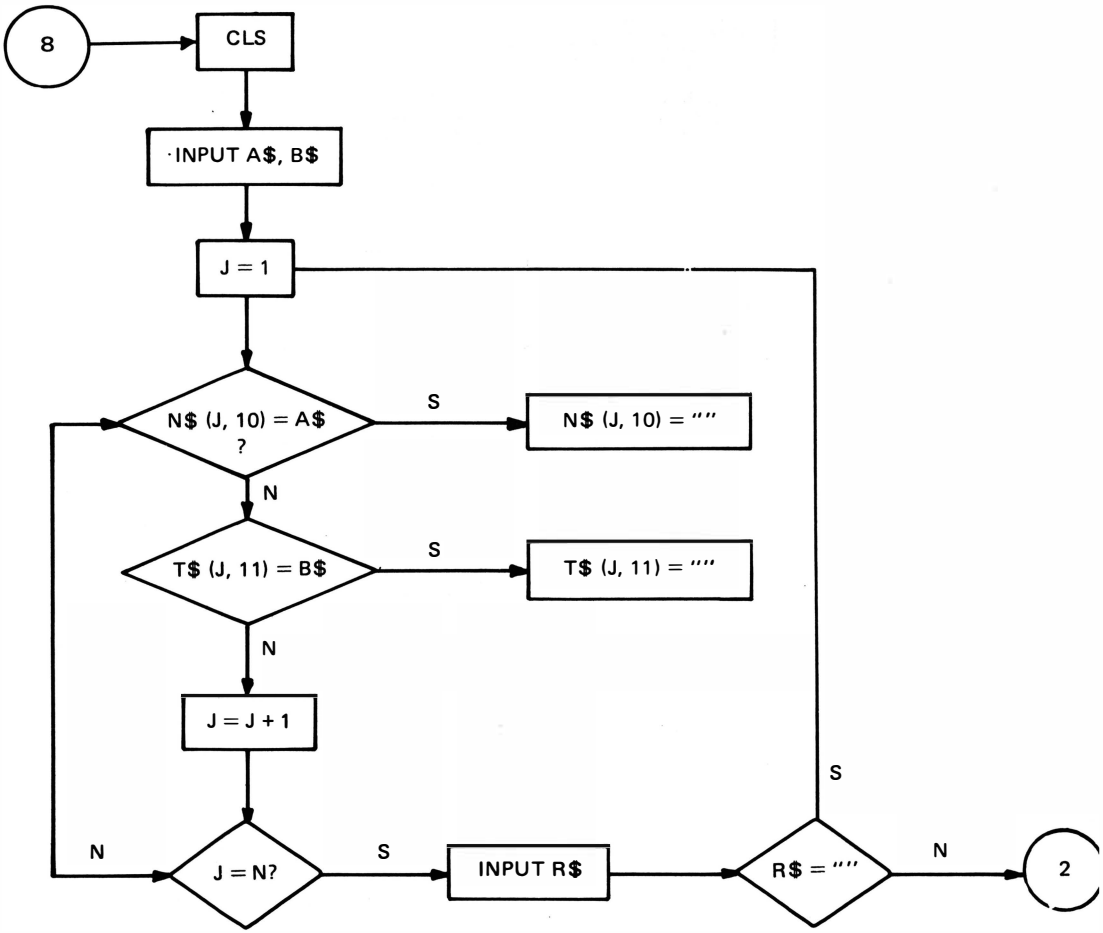

2
IMPRIME ROTEIRO
INKEY$ = ""
S
N
PRINT # – 1, N
J = 1
PRINT # – 1, N$ (J, 10), T$ (J, 11)
J = J + 1
J = N?
N
S
IMPRIME MENU

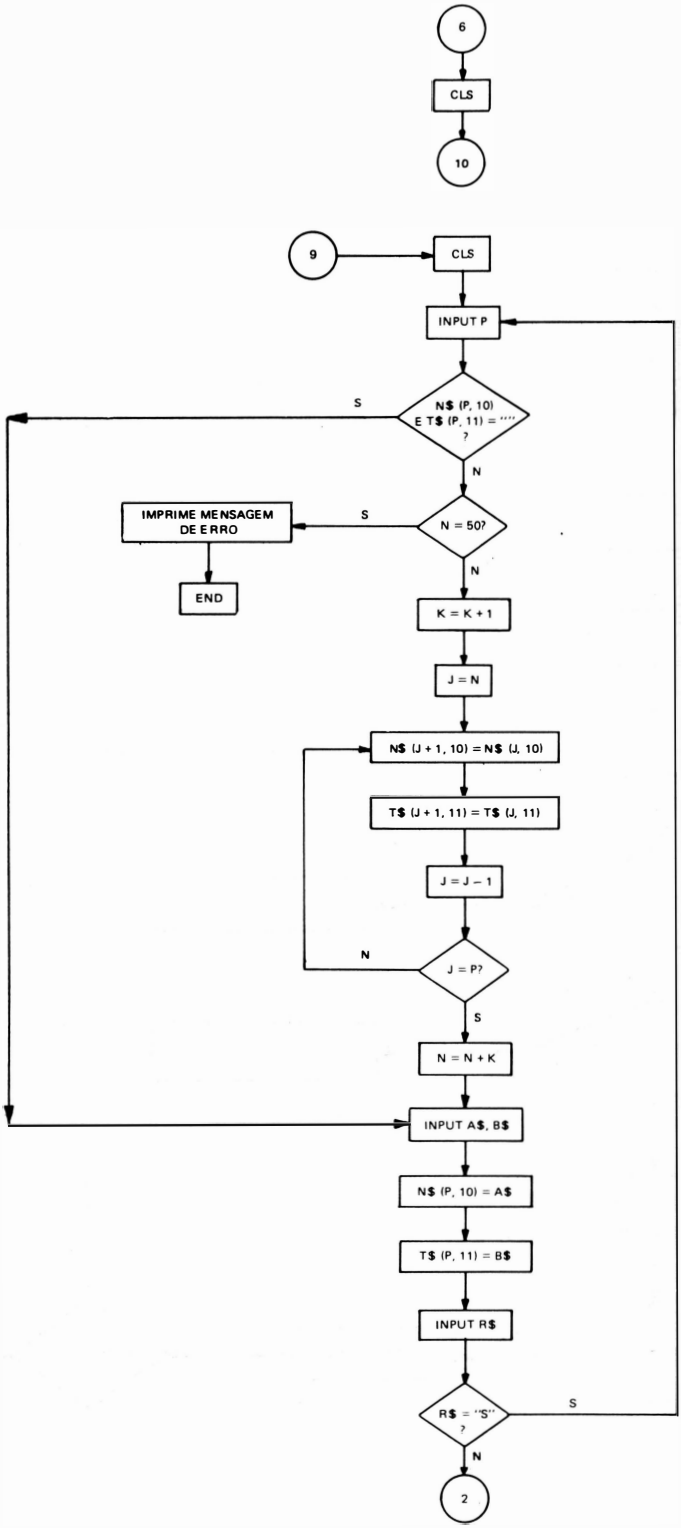

<LISTAGEM>
3
10
IMPRIME ROTEIRO
INKEY$ = ""
S
N
CLS
CLEAR 2000
INPUT #– 1, N
DIM N$ (50, 10), T$ (50, 11)
J = 1
INPUT #– 1, N$ (J, 10), T$ (J, 11)
PRINT J, N$ (J, 10), T$ (J, 11)
J = J + 1
J = N?
N
S
INPUT R$
R$ = "D"
S
8
N
R$ = "I"
S
9
N
IMPRIME MENU


<NOVA LISTAGEM>
4
J = 1
PRINT J, N$ (J, 10), T$ (J, 11)
J/10 − INT (J/10) = 0 ?
S
STOP
N
J = J + 1
J = N?
N
S
STOP
IMPRIME MENU

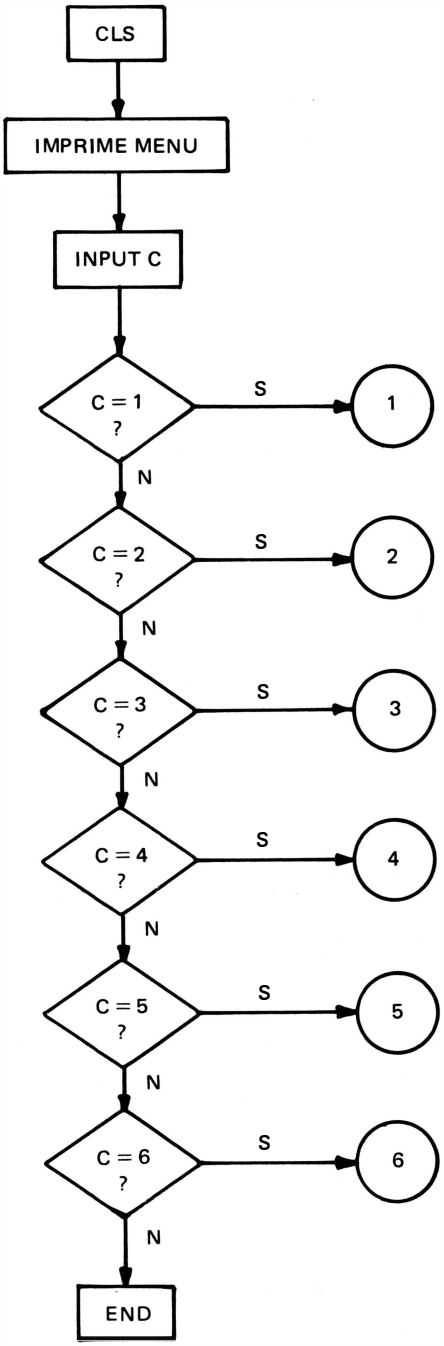
)ELEÇÃO>


5
CLS
10


8
CLS
INPUT A$, B$
J = 1
N$ (J, 10) = A$?
S
N$ (J, 10) = ""
N
T$ (J, 11) = B$
S
T$ (J, 11) = ""
N
J = J + 1
N
J = N?
S
INPUT R$
R$ = ""
S
N
2


6
CLS
10
9
CLS
INPUT P
N$ (P, 10) E T$ (P, 11) = "" ?
S
N
IMPRIME MENSAGEM DE ERRO
S
N = 50?
END
N
K = K + 1
J = N
N$ (J + 1, 10) = N$ (J, 10)
T$ (J + 1, 11) = T$ (J, 11)
J = J – 1
N
J = P?
S
N = N + K
INPUT A$, B$
N$ (P, 10) = A$
T$ (P, 11) = B$
INPUT R$
R$ = "S" ?
S
N
2

<MENU>


CLS
IMPRIME MENU
INPUT C
C = 1 ?
S
1
N
C = 2 ?
S
2
N
C = 3 ?
S
3
N
C = 4 ?
S
4
N
C = 5 ?
S
5
N
C = 6 ?
S
6
N
END

```
ESCOLHA A OPCAO:

ENTRAR COM NOMES=1              GRAVAR=2

LISTAR=3                        LISTAR DE NOVO=4

DELETAR NOMES=5                 INSERIR NOVOS NOMES=6

NADA MAIS=7

OPCAO=? .
```

```
10 'PROGRAMA ARQUIVO; PARA NOMES E TELEFONES.
20 'PERMITE ENTRAR, GRAVAR, LISTAR, DELETAR E INSERIR ATE'
30 '50 NOMES E TELEFONES.
40 'POR: FAUSTO A. DE A. BARBUTO.   DATA: 08/JUL/84.
50 GOTO 660
60 CLEAR 2000
70 DIM N$(50,10), T$(50,11)
80 CLS:INPUT"NU'MERO DE NOMES"; N
90 IF N)50 THEN 80
100 FOR J=1 TO N
110 CLS:INPUT"NOME"; N$(J,10)
120 INPUT"TELEFONE"; T$(J,11)
130 NEXT J
140 GOTO 660
150 CLS:PRINT@320,"PONHA A FITA DE DADOS (NO INI'CIO) NO GRAVADOR."
160 PRINT@448,"APERTE REC + PLAY"
170 PRINT@576,"TUDO PRONTO? APERTE QUALQUER TECLA."
180 IF INKEY$="" THEN 180
190 PRINT#-1,N
200 FOR J=1 TO N
210 PRINT#-1, N$(J,10), T$(J,11)
220 NEXT J
230 GOTO 660
240 CLS:PRINT@15,"PARA DESCARREGAR:"
250 PRINT@64,"(1) POSICIONE A FITA NO INI'CIO DOS DADOS;"
260 PRINT@192,"(2) APERTE PLAY;"
270 PRINT@320,"(3) APERTE QUALQUER TECLA."
280 IF INKEY$="" THEN 280 ELSE CLS
290 CLEAR 2000: INPUT#-1, N
300 DIM N$(50,10), T$(50,11)
310 FOR J=1 TO N
320 INPUT#-1, N$(J,10), T$(J,11): PRINT J, N$(J,10),T$(J,11)
330 NEXT J
335 PRINT:PRINT:INPUT"DELETAR OU INSERIR (D/I)"; R$
340 IF R$="D" THEN 420 ELSE IF R$="I" THEN 520 ELSE 660
345 GOTO 660
350 FOR J=1 TO N
360 PRINT J,N$(J,10),T$(J,11)
370 IF .1*J-INT(.1*J)=0 THEN STOP
380 NEXT J
390 STOP:GOTO 660
400 CLS:PRINT@15,"PARA DELETAR:"
410 GOTO 250
420 CLS:INPUT"NOME E NU'MERO A DELETAR"; A$, B$
430 FOR J=1 TO N
440 IF N$(J,10)=A$ THEN N$(J,10)=""
450 IF T$(J,11)=B$ THEN T$(J,11)=""
460 NEXT J
470 CLS:INPUT"OK. MAIS ALGUM NOME (S/N)"; R$
480 IF R$="S" THEN 420
490 GOTO 150
500 CLS:PRINT@15,"PARA INSERIR:"
510 GOTO 250
520 CLS:INPUT"POSICAO DO NOME/TEL. A INSERIR"; P
```

```
530 IF N$(P,10)="" AND T$(P,11)="" THEN 600
540 IF N=50 THEN 640
550 K=K+1
560 FOR J=N TO P STEP -1
570 N$(J+1,10)=N$(J,10) : T$(J+1,11)=T$(J,11)
580 NEXT J
590 N=N+K
600 PRINT:INPUT"NOME E TELEFONE A INSERIR"; A$, B$
610 N$(P,10)=A$ : T$(P,11)=B$
620 PRINT:INPUT"MAIS ALGUMA INSERCAO (S/N)"; R$
630 IF R$="S" THEN 520 ELSE 150
640 CLS:PRINT@448,"IMPOSSIVEL INSERIR: N=50."
650 END
660 CLS:PRINT@64,"ESCOLHA A OPCAO:"
670 PRINT:PRINT"ENTRAR COM NOMES=1","";"GRAVAR=2"
680 PRINT:PRINT"LISTAR=3","","LISTAR DE NOVO=4"
690 PRINT:PRINT"DELETAR NOMES=5","","INSERIR NOVOS NOMES=6"
700 PRINT:PRINT"NADA MAIS=7"
710 PRINT:INPUT"OPCAO="; C
720 CLS: ON C GOTO 60, 150, 240, 350, 400, 500, 650
```

BIBLIOGRAFIA

BARBUTO, F. A. A. – *Calcule os Juros e Escolha o Crediário com o TK-82C*. Revista Micro Sistemas, nº 8, ano I, maio 1982.

BARBUTO, F. A. A. – *Calcule o Tempo de Retorno do seu Investimento*. Revista Micro Sistemas, nº 17, ano II, fevereiro 1983.

BARBUTO, F. A. A. – *Química: O Fator Z no TK-82C*. Revista Micro Sistemas, nº 11, ano I, agosto 1982.

BARBUTO, F. A. A. – *Um Programa para o Pequeno Investidor*. Revista Micro Sistemas, nº 13, ano II, outubro 1982.

KREYSZIG, Erwin – *Matemática Superior*, 2º Vol., LTC – Livros Técnicos e Científicos Editora S.A., 1976.

LIMA, Léo da Rocha – *Elementos Básicos de Engenharia Química*. Editora McGraw-Hill do Brasil Ltda., 1978.

McCAUL, Earles L. – *TRS-80 ASSEMBLY LANGUAGE MADE SIMPLE*, 1st Ed., 1st Printing. Howard W. Sams & Co., Inc., 1981.

OLIVEIRA, Antônio Marmo; SILVA, Agostinho – *Biblioteca da Matemática Moderna*. Livros Irradiantes S.A., 1969.

PACITTI, Tércio – *FORTRAN Monitor, Princípios*, 3ª edição, LTC – Livros Técnicos e Científicos Editora S.A., 1977.

PUCCINI, A; HESS, G.; ROCHA PAES, L. C.; MARQUES, J. L. – *Engenharia Econômica*, 9ª edição, Difusão Editorial S.A., 1977.

SANTOS, José A. – *Engenharia Econômica* (Apostila). Petrobrás, 1981.

SEELEY, Robert T. – *Cálculo de uma Variável*, 1º Volume, LTC – Livros Técnicos e Científicos Editora S.A., 1973.

SHAMES, Irving H. – *Mecânica dos Fluidos*, Vols. 1 e 2 – 2ª Reimpressão. Editora Edgard Blücher, 1973.

SMITH, J. M.; VAN NESS, H. C. – *Introduction to Chemical Engineering Thermodynamics* – 3rd edition – McGraw-Hill Kogakusha Ltd., 1975.

THOMAS JR, George B. – *Cálculo*, Vols. 1, 2, 3 e 4. LTC – Livros Técnicos e Científicos Editora S.A., 1976.

Impressão e Acabamento
IONELI - Indústrias Gráficas Ltda.
Rua Viúva Cláudio, 81 - Tel. 201-2644

35 PROGRAMAS BASIC PARA MICROCOMPUTADORES

FAUSTO ARINOS DE ALMEIDA BARBUTO

O objetivo desta obra é levar ao leitor alguns programas selecionados entre os de maior interesse e importância pertencentes a áreas de Economia e Finanças, Matemática, Estatística, Engenharia e Aplicações gerais.

São trinta e cinco programas acompanhados de um resumido embasamento teórico, assim como exemplos já resolvidos e sua aplicação.

Os programas apresentados são orientados para a linguagem BASIC com que trabalham os micros compatíveis com o TRS-80, ou seja: CP-300, CP-500, D-8000, D-8001, JR Sysdata, DGT-100, DGT-101, entre outros.

MAIS UM LANÇAMENTO
DA
LIVROS TÉCNICOS E CIENTÍFICOS EDITORA S.A.

ISBN: 85-216-0388-6