

Escola Politécnica da Universidade de São Paulo

Departamento de Engenharia Mecânica

Avaliação do Desempenho Dinâmico de Estruturas

Aluno: Fúlvio Vittorino

Orientador: Prof. Dr. José Maria Saiz Jabardo

-1988-

Sumário

Neste trabalho foram desenvolvidos estudos acerca do comportamento, sob condições dinâmicas de exposição, de elementos típicos de edificações, através do "Método dos Fatores de Resposta Térmica". Para tanto, todas as equações envolvidas no referido método foram detalhadas, e foi ainda, desenvolvido um "software" para microcomputadores da linha IBM-PC que emprega o método.

O fundamento do "Método dos Fatores de Resposta Térmica" está no fato de se buscar a resposta em termos de fluxo de calor a uma excitação de temperatura sobre um determinado elemento de uma edificação. Admite-se que os elementos são planos, homogêneos, isotérmico em suas superfícies e que o fluxo de calor seja unidirecional.

DEDICATORIA

Aos meus pais, duas pessoas maravilhosas, que sempre me deram tanto amor e carinho.

Agradecimentos

Ao Prof.Dr. José Maria Saiz Jabardo, por todos o
ensinamentos e pela brilhante orientação dada;

A amiga Maria Akutsu, pelo apoio e incentivo.

Índice

I-Introdução

II-Modelagem Matemática

II.1-Conceituação do problema

II.2-Elemento com múltiplas camadas

II.3-Procedimento de cálculo

II.3.1-O Teorema da Inversão

II.3.2-Aplicação do teorema da inversão ao fenômeno modelado

III-Estudo da Função $B(s)$

III.1-Equação de Recorrência para o Cálculo de $B(s)$

III.2-Derivadas da função $B(s)$

III.2.1-Derivação dos Coeficientes A,B,C e D e Alterações no Método de "Newton-Raphson"

III.2.2-Fórmulas de recorrência para o cálculo de dB/ds

III.3-Unidade de ϕ_j

IV-Descrição do programa "FATORES"

V- Resultados

VI- Bibliografia

ANEXO - A: Listagem do Programa "FATORES"

ANEXO - B: Listagem do Programa "RESFX"

I-Introdução

O estudo da transferência de calor por condução em regime transitório sempre representou um problema de difícil solução analítica, a não ser para algumas geometrias simples e condições de contorno bem definidas.

Com a grande difusão dos computadores, ganharam destaque nesta área, alguns métodos numéricos que exigem a realização de grande número de cálculos e iterações. Entre estes métodos pode-se destacar o "Método dos Elementos Finitos", o "Método das Diferenças Finitas" e o "Método dos Fatores de Resposta Térmica", sendo que este último apresenta algumas vantagens em relação aos anteriores, principalmente quando aplicado aos elementos de uma edificação como por exemplo paredes e telhados.

As principais vantagens apresentadas pelo "Método dos Fatores de Resposta Térmica" são:

- boa precisão;
- um único cálculo para cada elemento estudado, resultando em um tempo de processamento relativamente baixo.

Quanto às desvantagens, pode-se citar o fato de que o "Método dos Fatores de Resposta Térmica" possibilita apenas a realização de cálculos com fluxos de calor unidimensionais.

Em função do exposto acima, será detalhado aqui o "Método dos Fatores de Resposta Térmica", que foi

desenvolvido na década de 60 por Arsenault e Mitalas e posteriormente por Kusuda.

Este método utiliza o princípio da superposição que admite que a resposta térmica global de um elemento, num dado instante, é a soma das respostas causadas por vários pulsos individuais de temperatura nos instantes anteriores, ou seja, a simulação de uma condição de contorno transitória de temperaturas é feita considerando-se uma sequência de pulsos consecutivos, e a resposta do elemento será obtida pela soma dos fluxos de calor, causados pelos pulsos, individualmente.

Matematicamente, o princípio da superposição será modelado através da utilização de conceitos da álgebra matricial e das Transformadas de Laplace.

Neste trabalho, o "Método dos Fatores de Resposta Térmica" foi adaptado para microcomputadores compatíveis com a linha IBM-PC, já que este método existe estruturado de maneira a ser usado apenas em computadores de grande porte. Um exemplo destes programas que usa o referido método, adaptado para o computador de grande porte CYBER 170-720 do IPT (Instituto de Pesquisas Tecnológicas do Estado de São Paulo S. A.), encontra-se listado no anexo B.

A adaptação para microcomputadores não foi apenas uma transcrição dos programas já existentes, mas sim fruto de um estudo detalhado das funções envolvidas, visando a obtenção de resultados confiáveis, com tempos de processamento

reduzidos. No anexo A, é apresentado o programa FATORES, resultante deste trabalho.

No capítulo II são apresentados os modelos físicos e matemáticos empregados na formulação do problema, bem como as equações empregadas na sua resolução. No capítulo III é apresentada a função característica do modelo, equações de recorrência para sua determinação, gráficos, bem como sua derivada, e unidades. No capítulo IV encontra-se uma descrição do programa desenvolvido, através dos fluxogramas das principais rotinas. Finalmente, no capítulo V são apresentados os resultados obtidos pelo programa FATORES, e uma comparação dos mesmos com aqueles obtidos pelo programa listado no anexo B.

II-Modelagem Matemática

Neste item, o "problema" será configurado, primeiramente, para um elemento constituído por uma única camada de um material homogêneo; depois a análise será expandida para elementos com várias camadas e a seguir, são apresentadas as equações para a determinação dos "Fatores de Resposta Térmica".

II.1-Conceituação do problema

O sistema físico estudado é constituído por meios que estão a temperaturas distintas, porém uniformes, mais um elemento de separação colocado entre eles.

Na modelagem do problema, para se estudar a transferência de calor por condução em regime transitório em edificações, admite-se que os elementos são planos, que os fluxos de calor se dão na direção perpendicular a estes, e que as propriedades físicas de transporte sejam constantes para os materiais constituintes destes elementos. Assim, podemos escrever a seguinte equação, que caracteriza o fenómeno:

$$\rho c \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2} \quad (1)$$

O sistema de coordenadas usado para escrever esta equação encontra-se representado na figura 1.

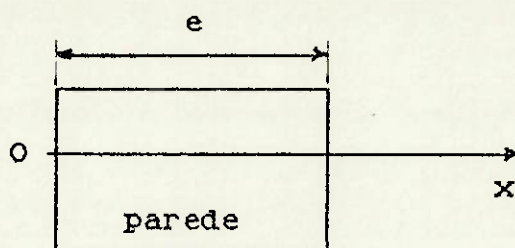


Figura 1: Sistema de coordenadas usado na eq (1)

Como condições iniciais e de contorno, tem-se:

a) superfícies isotérmicas em qualquer instante:

$$\begin{aligned} T(0, t) &= T_0(t), \quad t > 0 \\ T(e, t) &= T_e(t), \quad t > 0 \end{aligned}$$

b) o elemento todo está a uma mesma temperatura, antes de se iniciarem as observações :

$$T(x, t) = T_1(t), \quad t \leq 0$$

Fazendo-se a seguinte mudança de variáveis $\theta = T - T_1$, e aplicando-se a definição de difusividade térmica, α , a equação (1) toma a seguinte forma:

$$\frac{\partial \theta}{\partial t} = \alpha \frac{\partial^2 \theta}{\partial x^2} \quad (2)$$

com as condições:

$$\begin{aligned} \theta(0, t) &= T_0 - T_1 = \theta_0 \\ \theta(e, t) &= T_e - T_1 = \theta_e \\ \theta(x, 0) &= 0 \end{aligned}$$

Aplicando-se a Transformada de Laplace à equação (2) e às condições de contorno e iniciais, resulta:

$$s\theta(s, x) = \alpha \frac{\partial^2 \theta(s, x)}{\partial x^2} \quad \frac{\partial^2 \theta(s, x)}{\partial x^2} - m^2 \theta(s, x) = 0 \quad (3)$$

$$\begin{cases} \theta(0, t) = \theta_0 \\ \theta(0, t) = \theta_e \end{cases} \quad (4) \quad \begin{array}{l} \text{com } s = \text{parâmetro da transformada e} \\ m^2 = s/\alpha \end{array}$$

Resolvendo-se a equação (3), obtem-se:

$$\theta(s, x) = C_1 \sinh(mx) + C_2 \cosh(mx) \quad (5)$$

Considerando-se agora a equação de Fourier para o cálculo do fluxo de calor, em uma dada posição x :

$$q(s, x) = -k \frac{\partial \theta(s, x)}{\partial x}$$

e aplicando-a, com as condições (4) à equação (5), tem-se:

$$\begin{cases} \theta_e = -\frac{q_0}{mk} \sinh(me) + \theta_0 \cosh(me) \\ q_e = -q_0 \cosh(me) - \theta_0 m k \sinh(me) \end{cases}$$

que pode ser escrita sob a seguinte forma matricial:

$$\begin{vmatrix} \theta \\ q_0 \\ 0 \end{vmatrix} = \begin{vmatrix} \cosh(me) & \sinh(me)/(mk) \\ m k \sinh(me) & \cosh(me) \end{vmatrix} \begin{vmatrix} \theta \\ q_e \\ e \end{vmatrix}$$

$$\begin{vmatrix} \theta \\ q_0 \\ 0 \end{vmatrix} = \begin{vmatrix} A & B \\ C & D \end{vmatrix} \begin{vmatrix} \theta \\ q_e \\ e \end{vmatrix} \quad (6), \text{ com}$$

$$\begin{array}{ll} A = \cosh(me); & C = \sinh(me)/(mk) \\ B = m k \sinh(me); & D = \cosh(me) \end{array}$$

II.2-Elemento com múltiplas camadas

Se for considerado um elemento de várias camadas de materiais e espessuras diferentes, conforme representado na

figura 2, que é comum na prática, e desprezando-se as resistências térmicas de contato entre as distintas camadas, tem-se, aplicando-se a equação (6):

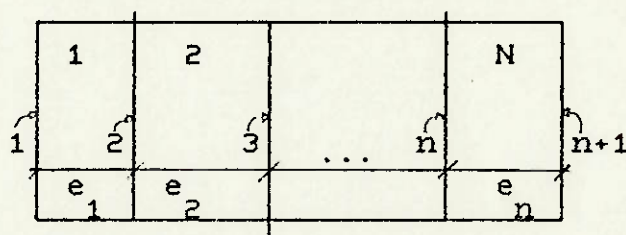


Figura 2: Elemento com N camadas

$$\begin{vmatrix} \bar{\theta} \\ \bar{\rho}^1 \\ 1 \end{vmatrix} = \begin{vmatrix} A & B \\ C^1 & D^1 \\ 1 & 1 \end{vmatrix} \begin{vmatrix} \bar{\theta} \\ \bar{\rho}^2 \\ 2 \end{vmatrix}; \quad \begin{vmatrix} \bar{\theta} \\ \bar{\rho}^2 \\ 2 \end{vmatrix} = \begin{vmatrix} A & B \\ C^2 & D^2 \\ 2 & 2 \end{vmatrix} \begin{vmatrix} \bar{\theta} \\ \bar{\rho}^3 \\ 3 \end{vmatrix}; \dots$$

$$\dots; \begin{vmatrix} \bar{\theta} \\ \bar{\rho}^n \\ n \end{vmatrix} = \begin{vmatrix} A & B \\ C^{n+1} & D^{n+1} \\ n+1 & n+1 \end{vmatrix} \begin{vmatrix} \bar{\theta} \\ \bar{\rho}^{n+1} \\ n+1 \end{vmatrix}$$

ou

$$\begin{vmatrix} \bar{\theta} \\ \bar{\rho}^1 \\ 1 \end{vmatrix} = \begin{vmatrix} A & B \\ C^1 & D^1 \\ 1 & 1 \end{vmatrix} \begin{vmatrix} A & B \\ C^2 & D^2 \\ 2 & 2 \end{vmatrix} \dots \begin{vmatrix} A & B \\ C^{n+1} & D^{n+1} \\ n+1 & n+1 \end{vmatrix} \begin{vmatrix} \bar{\theta} \\ \bar{\rho}^{n+1} \\ n+1 \end{vmatrix}$$

ou ainda

$$\begin{vmatrix} \bar{\theta} \\ \bar{\rho}^1 \\ 1 \end{vmatrix} = \begin{vmatrix} A & B \\ C & D \end{vmatrix} \begin{vmatrix} \bar{\theta} \\ \bar{\rho}^{n+1} \\ n+1 \end{vmatrix}$$

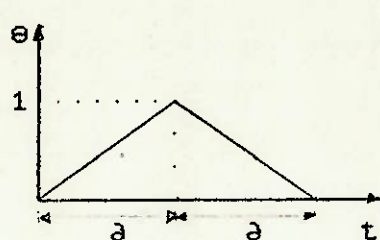
que pode ser reescrita como:

$$\begin{vmatrix} \bar{\rho} \\ \bar{\rho}^1 \\ N+1 \end{vmatrix} = \begin{vmatrix} D/B & -1/B \\ 1/B & -A/B \end{vmatrix} \begin{vmatrix} \bar{\theta} \\ \bar{\rho}^1 \\ n+1 \end{vmatrix} \quad (7)$$

II.3-Procedimento de cálculo

Até aqui, foi apresentado o fundamento matemático que modela o fenômeno de transferência de calor unidimensional em regime transitório de uma forma relativamente genérica, sendo que daqui em diante serão aplicados os conceitos do "Método dos Fatores de Resposta Térmica".

O referido método apresenta a resposta do sistema estudado, em termos de fluxo de calor, a uma perturbação de temperatura na forma de um pulso triangular (Kusuda demonstrou que este formato para o pulso apresentava um tempo de processamento menor do que se o mesmo fosse retangular) unitário conforme representado na figura 3. Os "fatores de resposta térmica" são esta resposta em fluxo, citada anteriormente, e terão a forma de uma série infinita, cujos elementos são os valores das funções resposta em intervalos regulares de tempo.



$$e = \begin{cases} 0 & p/ t \leq 0 \\ t/a & p/ 0 < t \leq a \\ 2-t/a & p/ a < t \leq 2a \\ 0 & p/ t > 2a \end{cases} \quad (8)$$

Figura 3: Representação do pulso triangular unitário

Ou aplicando-se a Transformada de Laplace a equação (8), tem-se:

$$\bar{e} = \begin{cases} 0 & p/ t \leq 0 \\ 1/as^2 & p/ 0 < t \leq a \\ 1/as^2(1-e^{-sa}) & p/ a < t \leq 2a \\ 1/as^2(1-e^{-s2a})^2 & p/ t > 2a \end{cases} \quad (9)$$

O procedimento a seguir será substituir $\bar{e}$ por (9) na equação (7) e aplicar o teorema da inversão para obter os fluxos de calor.

Primeiramente faremos $\bar{e}_1 = \bar{e}$ e $\bar{e}_{n+1} = 0$ para determinar φ_1 e φ_{n+1} , ou seja, determinar a resposta em

fluxo a uma perturbação de temperatura na superfície 1. Depois repete-se o procedimento porém com $\bar{\theta}_{n+1} = \bar{\theta}$ e $\bar{\theta}_1 = 0$.

Como a resposta da superfície 1, φ_1 , para uma perturbação na superfície $n+1$ é igual, mas de sinal contrário à resposta da superfície $n+1$ para uma perturbação na superfície 1, teremos 3 fatores que caracterizarão as respostas do sistema. Na figura 4 temos a representação do pulso e das respostas em fluxo.

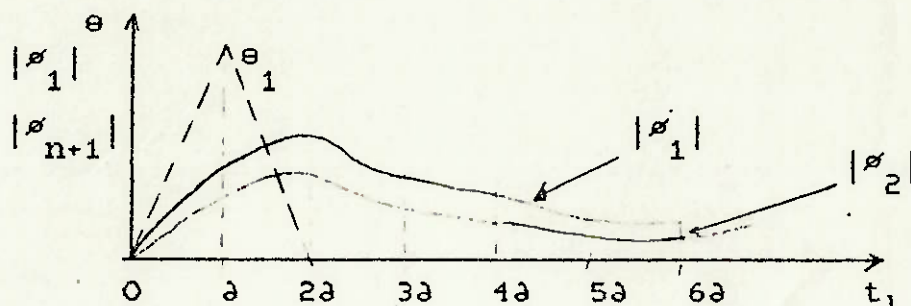


Figura 4: Representação de um pulso de temperatura e sua respostas em fluxo.

II.3.1-O Teorema da Inversão

O teorema da inversão tem a seguinte forma:

$$f(x, t) = \frac{1}{2\pi i} \lim_{L \rightarrow \infty} \int_{\gamma - iL}^{\gamma + iL} e^{ts} \bar{f}(x, s) ds \quad \text{com}$$

i = unidade imaginária

$f(x, s)$ = transformada de Laplace da função

γ = parâmetro arbitrário, suficientemente alto, para que a integral: $\int_0^{\infty} e^{-\gamma c} f(c) dc$ exista.

Por outro lado, do teorema dos resíduos de Cauchy, tem-se:

$$\int f(z)dz = 2\pi i \sum_{j=1}^n \text{Res}(a_j)$$

Usa-se então a curva representada na figura 5 para o teorema da inversão.

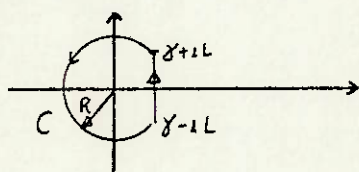


Figura 5: Curva usada no teorema da inversão

Do teorema da inversão tem-se:

$$\begin{aligned} \lim_{L \rightarrow \infty} \int_{\gamma - iL}^{\gamma + iL} e^{ts} f(x, s) ds + \lim_{R \rightarrow \infty} \int_C e^{ts} f(x, s) ds &= \\ &= 2\pi i \lim_{n \rightarrow \infty} \sum_{j=1}^n \text{Res}(a_j) \end{aligned}$$

Pode-se demonstrar que neste caso a integral para o círculo de raio R é nula, portanto tem-se:

$$f(x, t) = \lim_{n \rightarrow \infty} \sum_{j=1}^n \text{Res}(a_j) \quad (10)$$

onde os resíduos correspondem a função $e^{st}f(x, s)$.

II.3.2-Aplicação do teorema da inversão ao fenômeno modelado

Retomando-se a equação (6), pode-se escreve-la na seguinte forma:

$$s = \frac{R}{B} \theta \quad \text{com } R = \begin{vmatrix} D & -1 \\ 1 & -A \end{vmatrix}$$

Para se obter a resposta em fluxo na superfície 1 para uma perturbação nesta superfície, faz-se $R=D$. Já para a resposta em fluxo na superfície $n+1$, para esta mesma perturbação, faz-se $R=1$.

$$\bar{\varphi}_1 = D/B \bar{\theta}_1 \quad (11)$$

$$\bar{\varphi}_{n+1} = 1/B \bar{\theta}_1 \quad (12)$$

Se a perturbação for na superfície $n+1$, tem-se:

$$\bar{\varphi}_1 = -1/B \bar{\theta}_{n+1} \quad (13)$$

$$\bar{\varphi}_{n+1} = -A/B \bar{\theta}_{n+1} \quad (14)$$

Pode-se agora, aplicar a equação (10) às equações (11), (12) e (14) (como já foi dito anteriormente, e como $\bar{\theta}_1 = \bar{\theta}_{n+1}$, as equações (12) e (13) são iguais a menos do sinal) e determinar as respostas em fluxo ao pulso triangular unitário.

Então a função a ser usada no teorema da inversão será:

$$e^{st} \frac{R}{B} \bar{\theta}(s) \quad (15)$$

com $\bar{\theta}(s)$ dada pela equação (9), que apresenta polos de ordem 2 na origem, resultantes do fator $1/as^2$. Neste ponto, para se calcular o resíduo da função, aplica-se a equação (16) com " $a = 0$ ".

$$\text{Res}(a) = \frac{1}{(k-1)!} \left[\frac{d^k}{dz^{k-1}} [(z-a)^k f(z)] \right]_{z=0} \quad (16)$$

onde k = ordem do polo.

Para os demais polos da equação (15), pode-se considera-la com a forma:

$$f(z) = \frac{P(z)}{z^k Q(z)} \quad (17)$$

com $P(z)$ sendo um polinômio; e $Q(z)$ outro polinômio de raízes a_j , com $j = 1, \dots, N$ e $a_j \neq 0, \forall j$. Dessa forma, a equação (17) apresentará polos simples ($k=1$) nos pontos $z = a_j$, cujos resíduos são calculados por:

$$\text{Res}(a_j) = \lim_{z \rightarrow a_j} \frac{(z-a_j)P(z)}{z^k Q(z)}$$

que pode ser reescrita como:

$$\text{Res}(a_j) = \frac{P(a_j)}{a_j^k (dQ/dz)_{z=a_j}}$$

$$\text{Logo } \sum_{j=1}^n \text{Res}(a_j) = [\text{Res}(0) + \sum_{j=1}^n \text{Res}(a_j)]$$

Portanto a função φ será dada por:

$$\varphi = \lim_{s \rightarrow 0} \left[\frac{d(s^2 \bar{\Theta} e^{ts} R/B)}{ds} \right] + \sum_{j=1}^{\infty} \left[\frac{\bar{\Theta} R e^{ts}}{dB/ds} \right]_{s=-\phi_j} \quad (18)$$

onde $-\phi_j$ ($\phi_j > 0$) são as raízes reais da equação $B(s) = 0$.

O primeiro termo da equação (18) resulta em:

$$\lim_{s \rightarrow 0} \left[\frac{d(s^2 \bar{\Theta} e^{ts} R/B)}{ds} \right] = \begin{cases} U/\partial (dR/ds + R/t_1 - R/B \cdot dB/ds) s=0 \\ p/ 0 < t_1 \leq \partial \\ -U/\partial (dR/ds - R/B \cdot dB/ds) s=0 \\ p/ \partial < t_1 \leq 2\partial \\ 0 \\ p/ t > 2\partial \end{cases} \quad (19)$$

Para $s = 0$, os fatores A, B, C e D podem ser escritos na seguinte forma:

$$m = \sqrt{(s/\alpha)}; \text{ pois } s \rightarrow 0 \implies m \rightarrow 0$$

$$A = \cosh(me) = \cosh(0) = 1 \quad (20)$$

$$B = \lim_{m \rightarrow 0} \frac{\sinh(me)}{mk} = \lim_{m \rightarrow 0} \frac{e \cdot \cosh(me)}{k} = \frac{e}{k} \quad (21)$$

$$C = mk \sinh(me) = 0 \cdot \sinh(0) = 0 \quad (22)$$

$$D = \cosh(me) = \cosh(0) = 1 \quad (23)$$

Por outro lado, pode-se ter $m=0$ quando a camada em questão tiver inércia térmica desprezível, já que $\rho c \rightarrow 0$
 $\alpha = \infty$ e portanto $m = 0$

Desta forma, se a estrutura estudada for constituída de camadas de inércia térmica desprezível, tem-se:

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix} = \begin{vmatrix} 1 & e/\sqrt{k_1} \\ 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & e/\sqrt{k_2} \\ 0 & 1 \end{vmatrix} \dots \begin{vmatrix} 1 & e/\sqrt{k_n} \\ 0 & 1 \end{vmatrix}$$

que resulta:

$$\begin{aligned} A &= 1; & B &= e_1/\sqrt{k_1} + e_2/\sqrt{k_2} + \dots + e_n/\sqrt{k_n} = 1/U \\ C &= 0; & D &= 1 \end{aligned}$$

sendo U o coeficiente global de transferência de calor associado à estrutura, entre suas superfícies limite.

Para se avaliar o 2º termo da equação (18), deve-se calcular as raízes de $B(s) = 0$. A única possibilidade para que isso ocorra é que s seja um número complexo puro, ou seja, para isso deve-se ter $s = -\phi_j$ ($\phi_j > 0$), pois

$$m = \sqrt{(s/\alpha)} = \sqrt{(-\phi_j/\alpha)} = i \sqrt{(\phi_j/\alpha)}$$

Assim os coeficientes terão a seguinte forma:

$$A = \cosh(me) = \cosh(\sqrt{(\phi_j e^2/\alpha)}i) = \cos \sqrt{(\phi_j e^2/\alpha)} \quad (24)$$

$$B = \frac{\sinh(me)}{mk} = \frac{\sinh(\sqrt{(1-\phi_j e^2/\alpha)})}{i\sqrt{(\phi_j/\alpha)}k} = \frac{1}{i} \frac{\sin(\sqrt{(\phi_j e^2/\alpha)})}{\sqrt{(\phi_j/\alpha)}k} = \frac{\sin(\sqrt{(\phi_j e^2/\alpha)})}{\sqrt{(\phi_j/\alpha)}k} \quad (25)$$

$$C = mk \sinh(me) = k \sqrt{(\phi_j/\alpha)} i^2 \sin \sqrt{(\phi_j e^2/\alpha)} = -k \sqrt{(\phi_j/\alpha)} \sin \sqrt{(\phi_j e^2/\alpha)} \quad (26)$$

$$D = A \quad (27)$$

O valor da derivada dB/ds será calculado através do seguinte procedimento:

$$\begin{aligned} \frac{d}{ds} \begin{vmatrix} A & B \\ C & D \end{vmatrix} &= \begin{vmatrix} dA/ds & dB/ds \\ dC_1/ds & dD_1/ds \end{vmatrix} \begin{vmatrix} A & B \\ C_2 & D_2 \end{vmatrix} \dots \begin{vmatrix} A & B \\ C_n & D_n \end{vmatrix} + \\ &+ \begin{vmatrix} A & B \\ C_1 & D_1 \end{vmatrix} \begin{vmatrix} dA/ds & dB/ds \\ dC_2/ds & dD_2/ds \end{vmatrix} \dots \begin{vmatrix} A & B \\ C_n & D_n \end{vmatrix} + \\ &+ \begin{vmatrix} A & B \\ C_1 & D_1 \end{vmatrix} \begin{vmatrix} A & B \\ C_2 & D_2 \end{vmatrix} \dots \begin{vmatrix} dA/ds & dB/ds \\ dC_n/ds & dD_n/ds \end{vmatrix} \end{aligned}$$

sendo que a derivada de cada coeficiente é dado por:

$$dA/dz = \sinh(me) \cdot e \cdot \frac{1}{2} \cdot [\sqrt{(z/\alpha)}]^{-1} \cdot 1/\alpha = \frac{\sinh(me) \cdot e^2}{2 m \alpha}$$

$$p/z = -\phi_j \quad dA/dz = \frac{e^2}{2\alpha} \frac{\sin(\sqrt{(\phi_j e^2/\alpha)})}{\sqrt{(\phi_j e^2/\alpha)}} \quad (28)$$

Para os demais coeficientes será apresentada apenas a forma final de suas derivadas, já que para obtê-las, basta seguir o procedimento aplicado acima.

$$dB/dz = \frac{e^3}{2 k \alpha} \frac{\cos \sqrt{(\phi_j e^2/\alpha)} - \sin[\sqrt{(\phi_j e^2/\alpha)}] / \sqrt{(\phi_j e^2/\alpha)}}{(\phi_j e^2/\alpha)^2} \quad (29)$$

$$dC/dz = -\frac{e}{2 k \alpha} \sin[\sqrt{(\phi_j e^2/\alpha)}] / \sqrt{(\phi_j e^2/\alpha)} + \cos \sqrt{(\phi_j e^2/\alpha)} \quad (30)$$

$$dD/dz = dA/dz \quad (31)$$

Assim o 2º termo da equação (17) toma a seguinte forma:

$$\sum_{j=1}^{\infty} \frac{\bar{\theta} e^{ts} R}{dB/ds \quad s=-\phi_j} = \begin{cases} \sum_{j=1}^{\infty} \psi(\phi_j) e^{-\phi_j t} & p/ \quad 0 < t_1 \leq \partial \\ \sum_{j=1}^{\infty} \psi(\phi_j) (1-2 e^{-\phi_j \partial}) e^{-\phi_j t} & p/ \quad \partial < t_1 \leq 2\partial \\ \sum_{j=1}^{\infty} \psi(\phi_j) (1-2 e^{-\phi_j \partial})^2 e^{-\phi_j t} & p/ \quad t > 2\partial \end{cases} \quad (32)$$

com

$$\psi(\phi_j) = \frac{1}{\phi_j^2 \partial} \left[\frac{R}{dB/ds} \right]_{s=-\phi_j} \quad (33)$$

Montando as equações (19) e (32) na forma de uma série temporal infinita, tem-se:

$$t_1 = 0\partial, 1\partial, 2\partial, \dots, 1\partial, \dots$$

$$FR_0 = \left[\frac{R}{B} + \frac{dR/ds}{B\partial} - \frac{R \cdot dB/ds}{B^2 \cdot \partial} \right]_{s=0} + \sum_{j=1}^{\infty} \psi(\phi_j) e^{-\phi_j t} \quad p/ \quad 1 = 1 \quad (34)$$

$$FR_1 = - \left[\frac{dR/ds}{B \cdot \partial} - \frac{R \cdot dB/ds}{B^2 \cdot \partial} \right]_{s=0} + \sum_{j=1}^{\infty} \psi(\phi_j) (1-2 e^{-\phi_j \partial}) e^{-\phi_j t} \quad p/ \quad 1 = 2 \quad (35)$$

$$FR_{1-1} = \sum_{j=1}^{\infty} \psi(\phi_j) (1-2 e^{-\phi_j \partial})^2 e^{-\phi_j t} \quad p/ \quad 1 \geq 3 \quad (36)$$

Quando se faz $R = 1 \rightarrow FR = Y$
 $R = D \rightarrow FR = X$
 $R = A \rightarrow FR = Z$

considerando-se, agora, uma situação em que as temperaturas variam dinamicamente ao longo do tempo, pode-se então representar estas variações através da superposição de pulsos triangulares de amplitude igual ao valor da temperatura em um dado instante, como representado na figura 6.

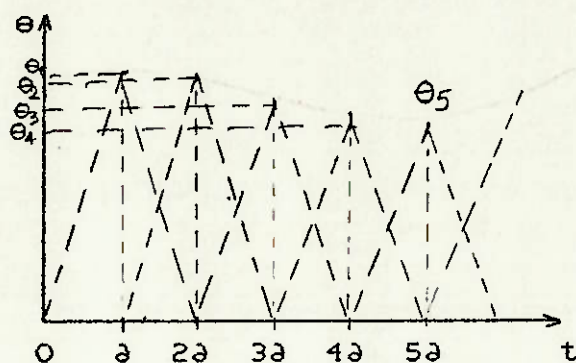


Figura 6: Representação de uma função temporal por pulsos triangulares

Então o fluxo de calor no instante t é dado pela soma dos fluxos, nesta superfície nos instantes anteriores:

$$(\varphi_i)_t = \sum_{j=1}^{\infty} (\varphi_i)_{t-j}, \text{ e escrevendo a equação (7) na sua forma}$$

"anti-transformada":

$$\begin{vmatrix} \varphi_1 \\ \varphi_{N+1}^1 \end{vmatrix} = \begin{vmatrix} D/B & -1/B \\ 1/B & -A/B \end{vmatrix} \begin{vmatrix} \Theta \\ \Theta_{n+1}^1 \end{vmatrix} \quad (37)$$

, tem-se como forma final para os fluxos de calor nas superfícies 1 e $n+1$ (considerando-se as variações de temperatura nas duas superfícies):

$$(\vartheta_1)_t = \sum_{j=0}^{\infty} X_j(\vartheta_1)_{t-j} - \sum_{j=0}^{\infty} Y_j(\vartheta_{n+1})_{t-j} \quad (38)$$

$$(\vartheta_{n+1})_t = \sum_{j=0}^{\infty} Y_j(\vartheta_1)_{t-j} - \sum_{j=0}^{\infty} Z_j(\vartheta_{n+1})_{t-j} \quad (39)$$

Quando o valor de j é muito elevado, pode-se provar que os fatores de resposta térmica apresentam a seguinte propriedade:

$$\frac{X_j}{X_{j+1}} = \frac{Y_j}{Y_{j+1}} = \frac{Z_j}{Z_{j+1}} = RC = \text{razão comum} = \text{constante} \quad (40)$$

Como as equações (38) e (39) não estão dispostas de uma maneira prática para serem manipuladas, por empregarem somatórias infinitas, define-se os "fatores des resposta térmica modificados", dados por

$$\begin{cases} X'_0 = X_0 \\ Y'_0 = Y_0 \\ Z'_0 = Z_0 \end{cases} \quad (41)$$

$$\begin{cases} X'_1 = X_1 - RC \cdot X_{1-1} \\ Y'_1 = Y_1 - RC \cdot Y_{1-1} \\ Z'_1 = Z_1 - RC \cdot Z_{1-1} \end{cases} \quad (42) \quad p/1 \quad X > 0$$

Com a aplicação destes "fatores modificados", as equações (38) e (39) se reduzem a:

$$(\vartheta_1)_t = (\vartheta_1)_{t-1} \cdot RC + \sum_{j=0}^N [X'_j(\vartheta_1)_{t-j} - Y'_j(\vartheta_{n+1})_{t-j}] \quad (43)$$

$$(\vartheta_{n+1})_t = (\vartheta_{n+1})_{t-1} \cdot RC + \sum_{j=0}^N [Y'_j(\vartheta_1)_{t-j} - Z'_j(\vartheta_{n+1})_{t-j}] \quad (44)$$

sendo N o número de fatores de resposta calculados até
'atingir-se a razão comum RC .

III-Estudo da Função B(s)

O cálculo dos fatores de resposta térmica mostrava-se "muito demorado" devido, principalmente, ao método iterativo empregado para se determinar as raízes da equação $B(s) = 0$, já que para se determinar o valor da função era necessário realizarem-se vários produtos matriciais.

Para se solucionar esta dificuldade pensou-se em se fazer o produto matricial para um determinado número de camadas pré-estabelecido, por exemplo 4, que cobriria as principais situações encontradas na prática, e com isso determinar uma função matemática, $y = B(s)$, que agilizaria os cálculos, além de permitir o uso do método de NEWTON-RAPHSON para se obter com menos iterações a precisão desejada, pois poderia-se obter a derivada de B em função de s (dB/ds) que é usada no referido método.

III.1-Equação de Recorrência para o Cálculo de B(s)

Fazendo-se explicitamente o produto matricial para a determinação de B(s) as seguintes equações foram obtidas:

$$B(s) = \frac{\sin[\sqrt{s}.f_1]}{[\sqrt{s}.g_1]} \quad \text{p/ um elemento c/1 camada;}$$

$$B(s) = \frac{\{\sin[\sqrt{s}.f_1].\cos[\sqrt{s}.f_2].g_2/g_1 + \sin[\sqrt{s}.f_2].\cos[\sqrt{s}.f_1]\}}{\sqrt{s}} \quad \text{p/ um elemento c/2 camadas;}$$

$$\begin{aligned}
 B(s) = & \operatorname{sen}[f(s).f_3] \{ \cos[f(s).f_1] . \cos[f(s).f_2] + \\
 & - g_2/g_1 . \operatorname{sen}[f(s).f_1] . \operatorname{sen}[f(s).f_2] \} + g_3 . \operatorname{sen}[f(s).f_3] . \\
 & . \{ \operatorname{sen}[f(s).f_2] . \cos[f(s).f_1] + g_2/g_1 . \operatorname{sen}[f(s).f_1] . \\
 & . \operatorname{sen}[f(s).f_2] \} \quad \text{p/ um elemento c/3 camadas;}
 \end{aligned}$$

Observando-se as equações acima, nota-se que se fizer $f_3 = 0$ e $g_3 = 1$ para a função $B(s)$, correspondente a um elemento com 3 camadas, esta se reduzirá a função àquela representativa de um elemento com 2 camadas. A partir daí, se for feito $f_2 = 1$ e $g_2 = 0$, esta se reduzirá a função que caracteriza os sistemas de uma única camada.

Contudo, analisando-se o produto matricial, pode-se obter uma "fórmula de recorrência" para os coeficientes A, B, C e D.

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix} = \begin{vmatrix} A & B \\ C^1 & D^1 \end{vmatrix}_1 \begin{vmatrix} A & B \\ C^2 & D^2 \end{vmatrix}_2 \begin{vmatrix} A & B \\ C^3 & D^3 \end{vmatrix}_3 \begin{vmatrix} A & B \\ C^4 & D^4 \end{vmatrix}_4$$

Assim se o elemento estudado tivesse apenas uma camada, tem-se:

$B_{T1} = B_1$, onde B_{T1} é a função B acumulada até a i-ésima camada e B_i é dada pela equação (24)

$A_{T1} = A_1$, onde A_{T1} é a função A acumulada até a i-ésima camada e A_i é dada pela equação (25)

Para um elemento com 2 camadas, tem-se:

$$B_{T2} = A_1 B_2 + B_1 D_2$$

$$A_{T2} = A_1 A_2 + B_1 C_2$$

ou

$$B_{T2} = A_{T1}B_2 + B_{T1}D_2$$

$$A_{T2} = A_{T1}A_2 + B_{T1}C_2$$

No caso do elemento ter 3 camadas:

$$B_{T2} = (A_1B_2 + B_1D_2)D_3 + (A_1A_2 + B_1C_2)B_3 = A_{T2}B_3 + B_{T2}D_3$$

$$A_{T2} = (A_1A_2 + B_1C_2)A_3 + (B_1D_2 + A_1B_2)C_2 = A_{T2}A_3 + B_{T2}C_3$$

Logo pode-se escrever:

$$B_{Tj} = A_{Tj-1}B_j + B_{Tj-1}D_j \quad (45)$$

para $j > 1$

$$A_{Tj} = A_{Tj-1}A_j + B_{Tj-1}C_j \quad (46)$$

Analogamente, tem-se:

$$C_{Tj} = C_{Tj-1}A_j + D_{Tj-1}C_j \quad (47)$$

para $j > 1$

$$D_{Tj} = C_{Tj-1}B_j + D_{Tj-1}D_j \quad (48)$$

III.2-Derivadas da função B(s)

No item II.3.2 derivou-se os coeficientes A, B, C e D a partir de sua forma complexa, aqui as derivadas serão apresentadas quando a substituição $z = -\phi_j$ é feita antes da derivação e não após a mesma.

III.2.1-Derivação dos Coeficientes A, B, C e D e Alterações no Método de "Newton-Raphson"

Derivando-se os coeficientes A, B, C e D, tem-se:

$$dA/dz = - \frac{e^2}{2\alpha} \frac{\text{sen}(\sqrt{\phi_j \cdot e^2/\alpha})}{\sqrt{\phi_j \cdot e^2/\alpha}} \quad (49)$$

$$dB/dz = \frac{e^2}{2 k \alpha} \frac{\text{sen}[\sqrt{\phi_j \cdot e^2/\alpha}] / \sqrt{\phi_j \cdot e^2/\alpha} - \cos \sqrt{\phi_j \cdot e^2/\alpha}}{(\phi_j \cdot e^2/\alpha)^2} \quad (50)$$

$$dC/dz = \frac{e}{2 k \alpha} \text{sen}[\sqrt{\phi_j \cdot e^2/\alpha}] / \sqrt{\phi_j \cdot e^2/\alpha} + \cos \sqrt{\phi_j \cdot e^2/\alpha} \quad (51)$$

Comparando-se os resultados aqui apresentados com aqueles do item II.3.2, percebe-se que eles diferem no sinal. Para a aplicação do método de "NEWTON-RAPHSON" deve-se usar as derivadas obtidas neste item.

Porém, para evitar uma complicação lógica maior no programa, que usaria a derivada aqui obtida no método de "Newton-Raphson" e a derivada obtida no item II.3.2 para o restante do programa, alterou-se o sinal da equação do referido método, conforme indicado abaixo, e se usou apenas as derivadas obtidas no item II.3.2.

Método Original.

$$x_{\text{ novo }} = x - \frac{f(x)}{f'(x)}$$

Método Modificado

$$x_{\text{ novo }} = x + \frac{f(x)}{f'(x)}$$

III.2.2-Fórmulas de recorrência para o cálculo de dB/ds

Derivando-se a equação (45) usada no cálculo de $B(s)$, obtêm-se a seguinte forma de recorrência para sua derivada, que fornece os mesmos resultados obtidos a partir da equação que é a definição da derivada de um produto matricial:

$$\frac{dB_{Tj}}{ds} = \frac{dA_{Tj-1}}{ds} B_j + B_{Tj-1} \frac{dD_j}{ds} + A_{Tj-1} \frac{dB_j}{ds} + \frac{dB_{Tj-1}}{ds} D_j \quad (52)$$

para $j > 1$

$$\frac{dB_{Tj}}{ds} = \frac{dB_{T1}}{ds} \quad \text{e} \quad \frac{dA_{Tj}}{ds} = \frac{dA_{-1}}{ds} \quad (53)$$

para $j = 1$

sendo $\frac{dB_{Tj}}{ds}$ a derivada da função $B(s)$ para um elemento com j camadas

As derivadas das demais funções ($A(s)$, $C(s)$ e $D(s)$) podem ser obtidas seguindo-se o mesmo procedimento.

III.3-Unidade de ϕ_j

Os coeficientes A, B, C e D são fruto de funções trigonométricas, por exemplo: $A = \cos(\sqrt{(\phi_j \cdot e^2/\alpha)})$.

Da análise dimensional sabe-se que o argumento da função cosseno deve ser adimensional, ou seja:

$$[\sqrt{(\phi_j \cdot e^2/\alpha)}] = 1 \quad [\phi_j \cdot e^2/\alpha] = 1$$

$$[\phi_j] \cdot [e^2] \cdot [\alpha]^{-1} = 1$$

$$[e^2] = L^2 \quad ; [\alpha]^{-1} = T.L^{-2}$$

$$[\phi_j].L^2.T.L^{-2} = T^0.L^0$$

$$[\phi_j] = T^{-1}$$

Como neste trabalho foi usado o sistema internacional de unidades (SI), tem-se: $[\phi_j] = s^{-1}$.

Assim, como busca-se a resposta, em fluxo, em intervalos de tempo de uma hora, deve-se usar $\Delta = 3600$ s.

III.4-Resultados Gráficos

Os gráficos apresentados a seguir, foram feitos usando-se as seguintes propriedades para os materiais:

-concreto com argila expandida:

$$\rho = 1100 \text{ kg/m}^3, \quad c = 0,82 \text{ kJ/kg} \cdot \text{C}, \quad k = 0,40 \text{ W/m} \cdot \text{C}$$

-zinco:

$$\rho = 7000 \text{ kg/m}^3, \quad c = 0,39 \text{ kJ/kg} \cdot \text{C}, \quad k = 113 \text{ W/m} \cdot \text{C}$$

-areia seca 1,2 mm de diâmetro:

$$\rho = 1340 \text{ kg/m}^3, \quad c = 0,75 \text{ kJ/kg} \cdot \text{C}, \quad k = 0,27 \text{ W/m} \cdot \text{C}$$

-painel de borracha expandida:

$$\rho = 73 \text{ kg/m}^3, \quad c = 1,68 \text{ kJ/kg} \cdot \text{C}, \quad k = 0,03 \text{ W/m} \cdot \text{C}$$

Para se ter uma idéia da grandeza das raízes quando se considera calor específico em $J/(\text{kg} \cdot \text{C})$ no lugar de $\text{kJ}/(\text{kg} \cdot \text{C})$, como foi feito, deve-se dividir por 1000 a escala do eixo z.

Nas figuras 7 a 16 verificou-se a influência do material e da espessura das camadas no comportamento da função $B(z)$

Nas figuras 17 a 22 verificou-se a influência, em uma parede composta, da espessura da camada do material de

características isolantes (simulou-se espessuras de 1, 5 e 10 cm de borracha sobre uma camada de 5 cm de concreto).

Nas figuras 23 a 28 variou-se a espessura da camada de concreto (para 10 cm) e repetiu-se a simulação para as mesmas espessuras de borracha citadas acima.

Dos gráficos apresentados a seguir pode-se verificar claramente a influência da inércia térmica do elemento no posicionamento das raízes da função $B(z)$ ao longo do eixo z .

Pode-se notar, também, que com o aumento da inércia térmica, as raízes se aproximam muito rapidamente de zero. Além disto, verifica-se que as raízes representativas de elementos com baixa inércia térmica estão muito distanciadas entre si.

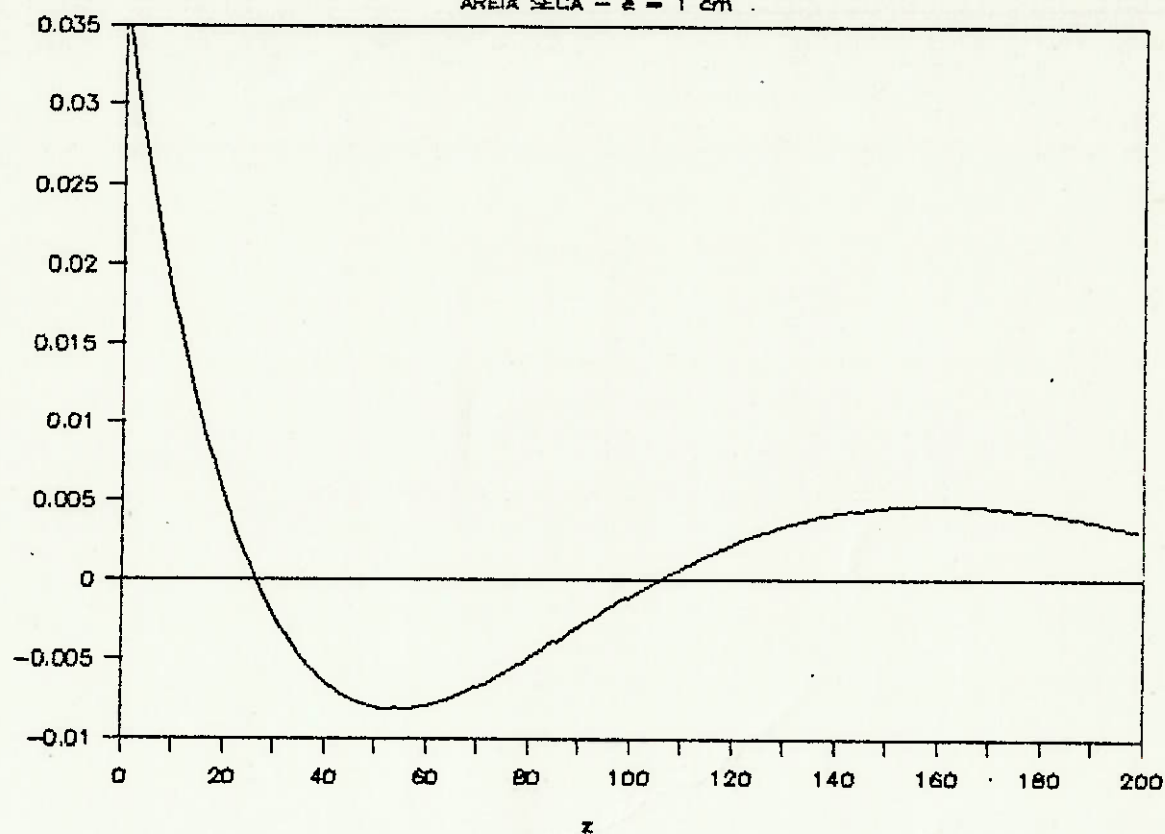
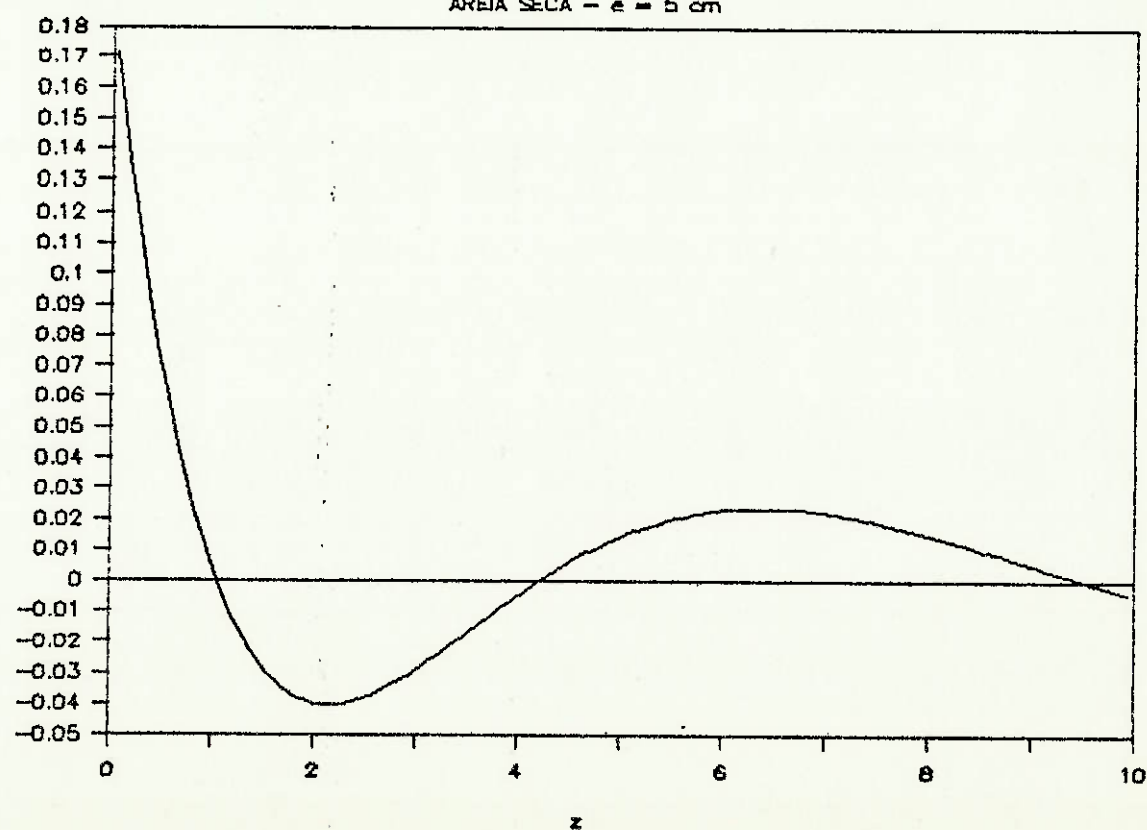
Figura 7: FUNCAO $B(z)$ — 1 CAMADAAREIA SECA — $e = 1$ cmfigura 8.: FUNCAO $B(z)$ — 1 CAMADAAREIA SECA — $e = 5$ cm

Figura 9: FUNCAO $B(z)$ — 1 CAMADA

AREIA SECA — $a = 10$ cm

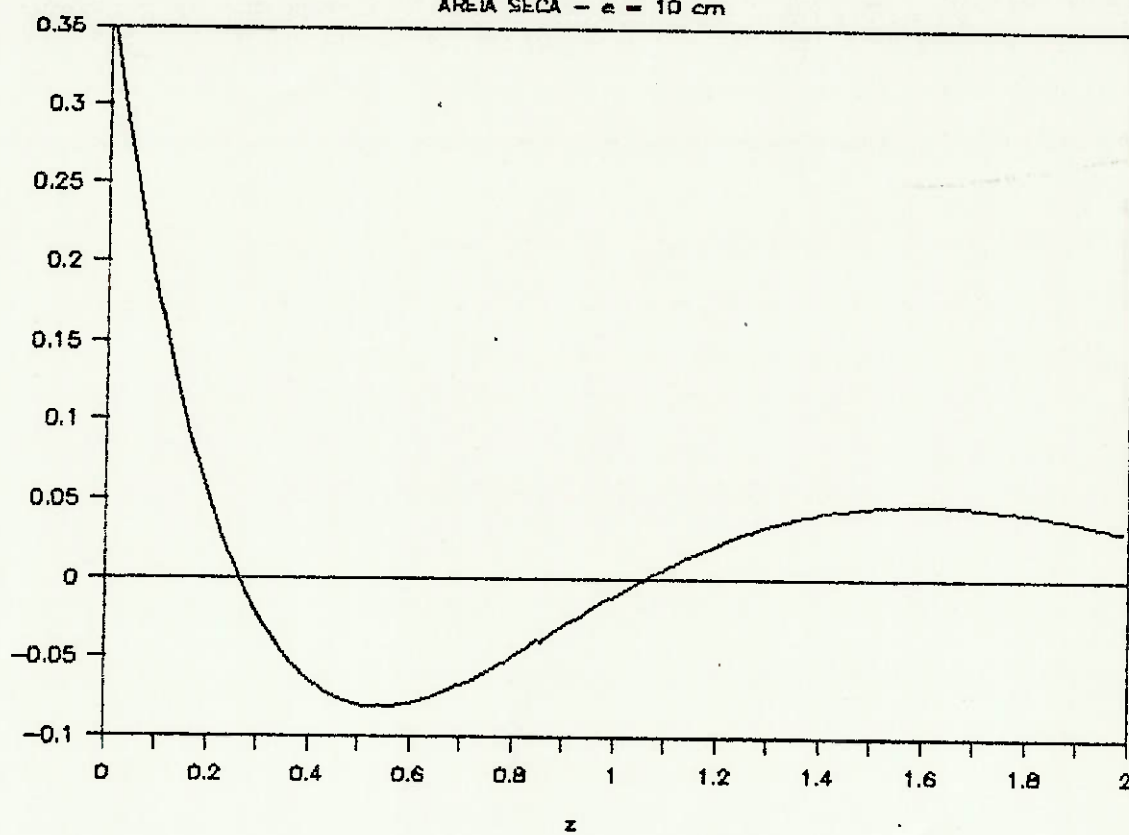


Figura 10 FUNCAO $B(z)$ — 1 CAMADA

AREIA SECA — $a = 10$ cm

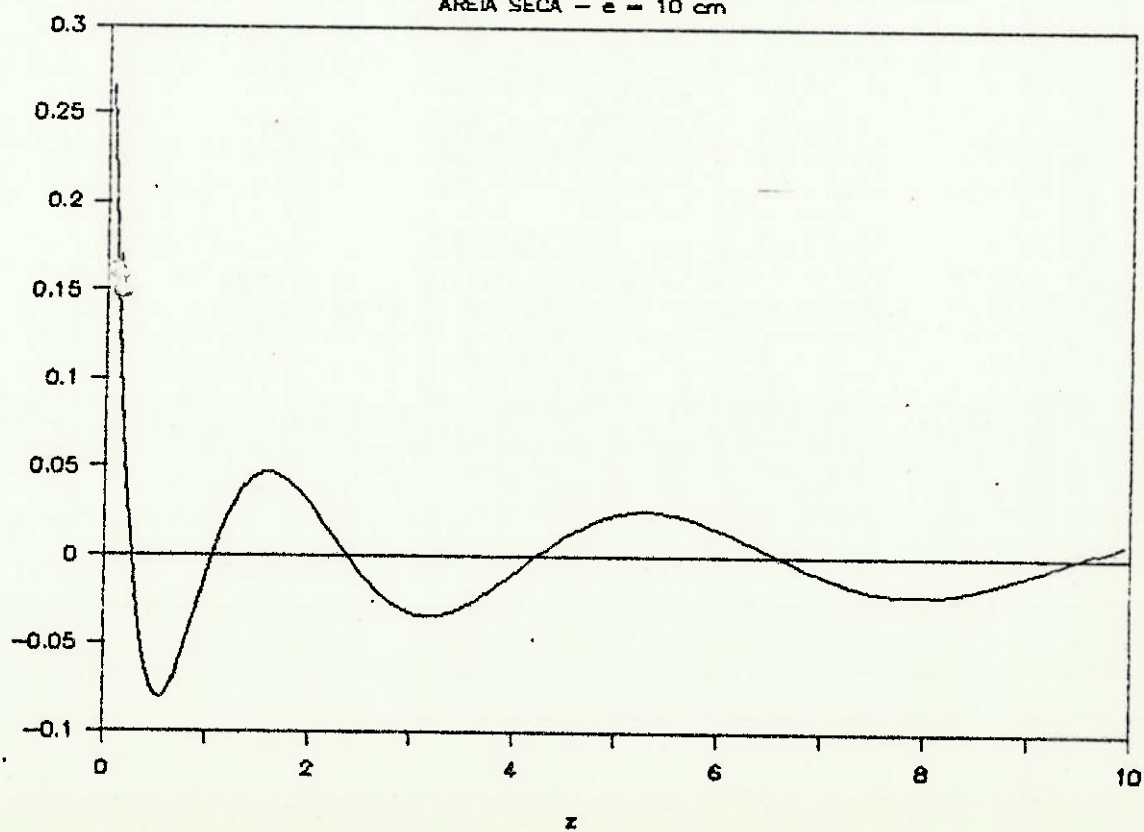


Figura 11: FUNCAO $B(z)$ — 1 CAMADA

ZINCO — $a = 15$ cm

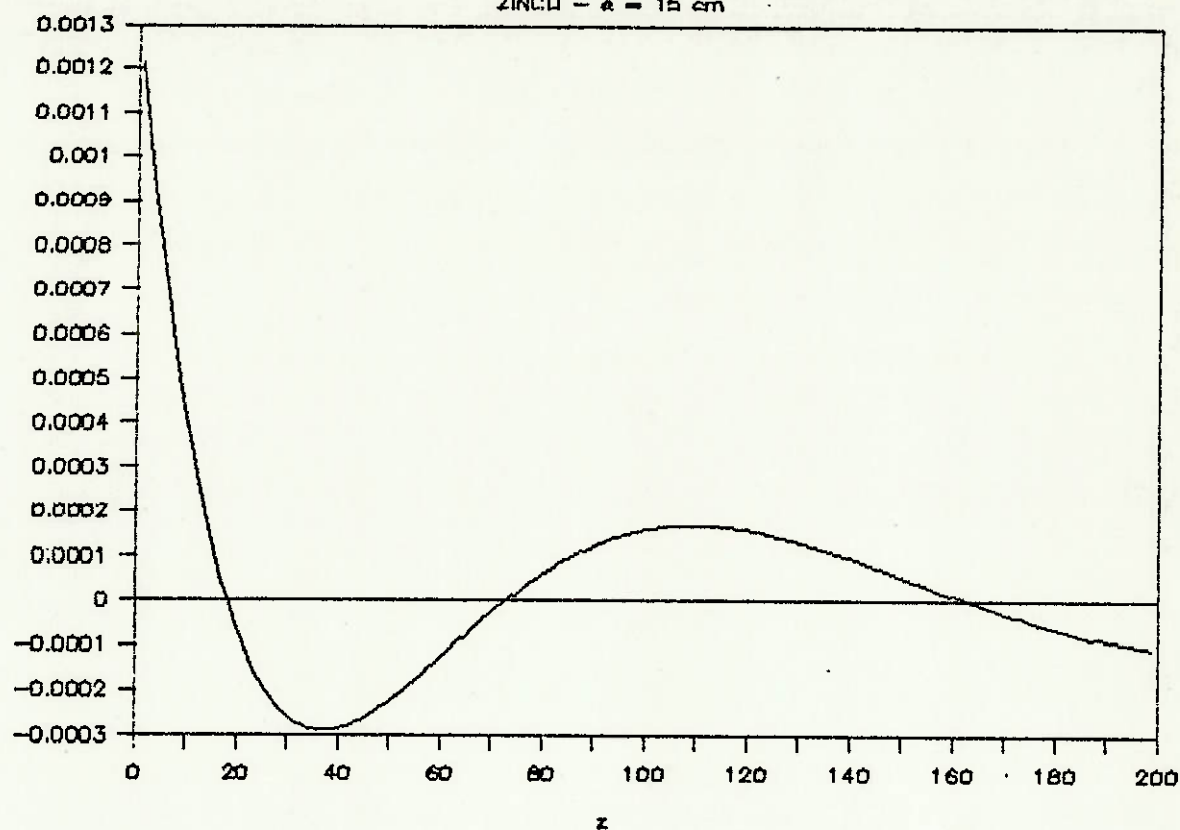


Figura 12: FUNCAO $B(z)$ — 1 CAMADA

ZINCO — $a = 25$ cm

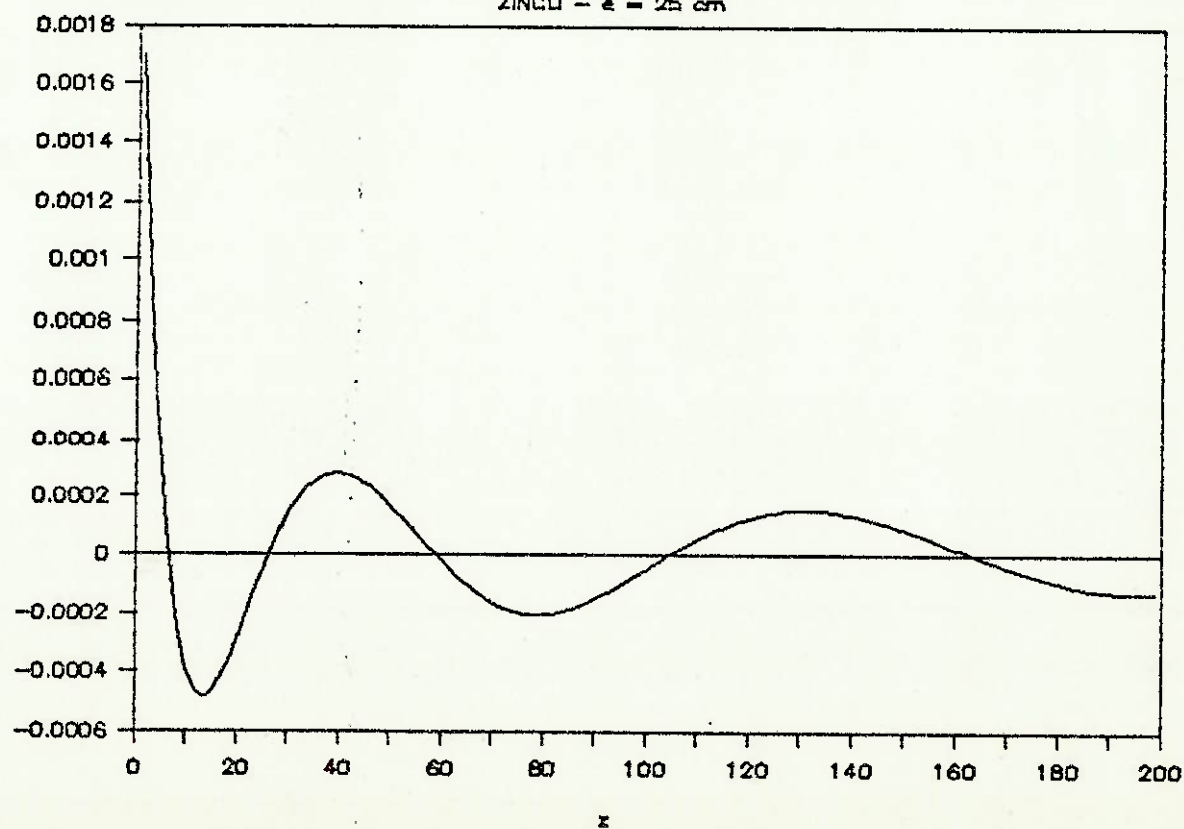


Figura 13: FUNCAO $B(z)$ — 2 CAMADAS

CONCRETO ($e=1$ cm) + BORRACHA ($e=1$ cm)

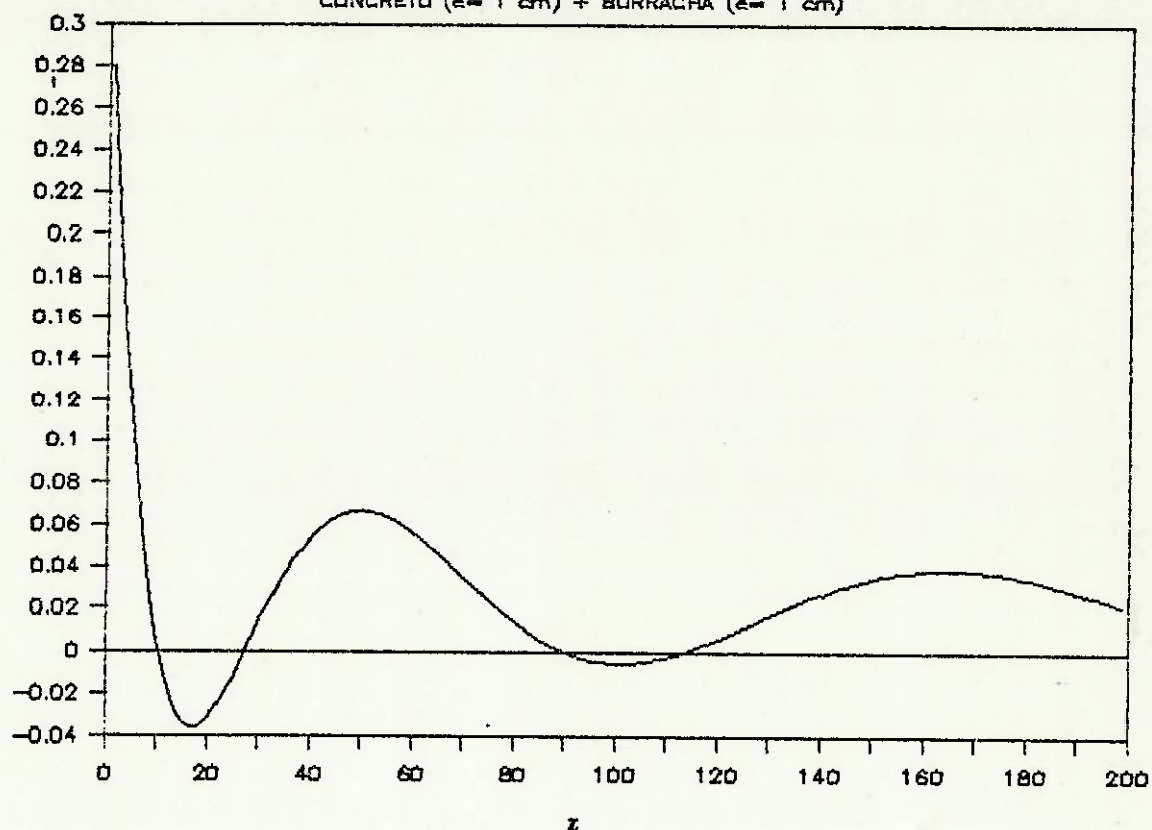


Figura 14: FUNCAO $B(z)$ — 1 CAMADA

CONCRETO C/ ARGILA EXPANDIDA — $e=1$ cm

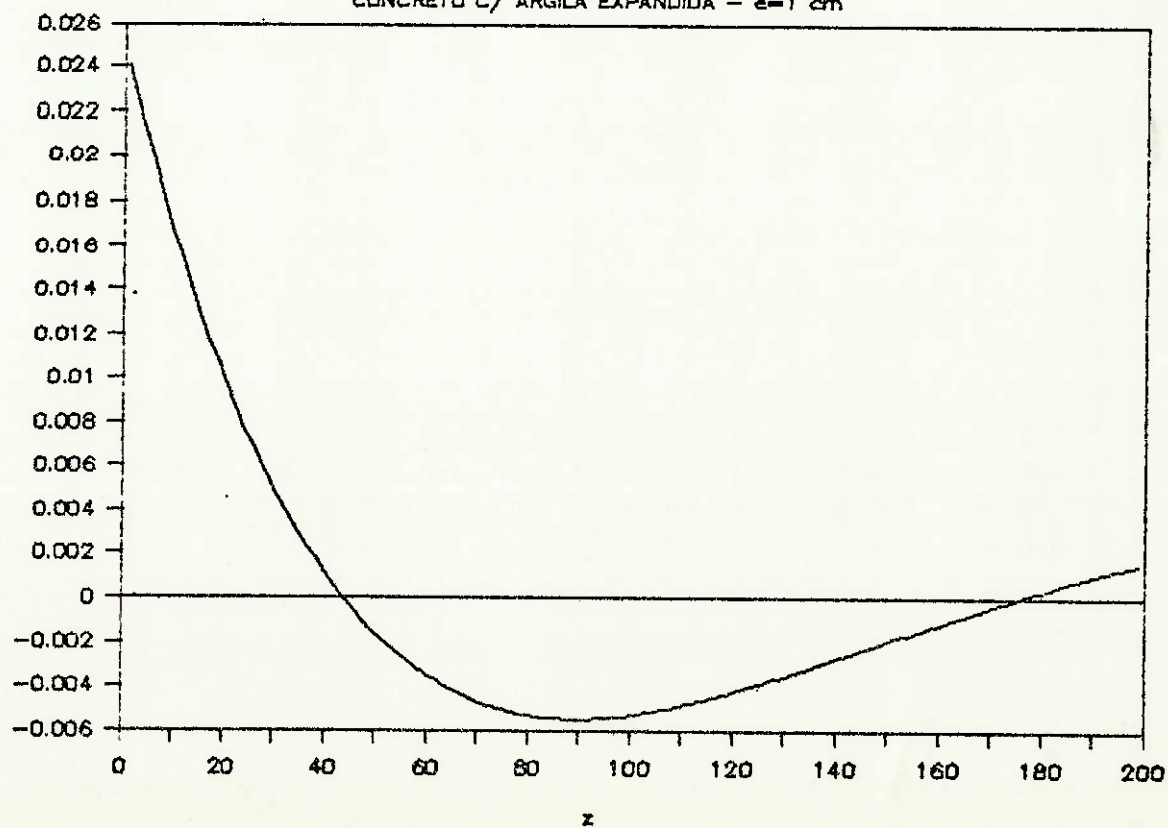


Figura 15: FUNCAO $B(z)$ — 1 CAMADA
CONCRETO C/ ARGILA EXPANDIDA — $e=5$ cm

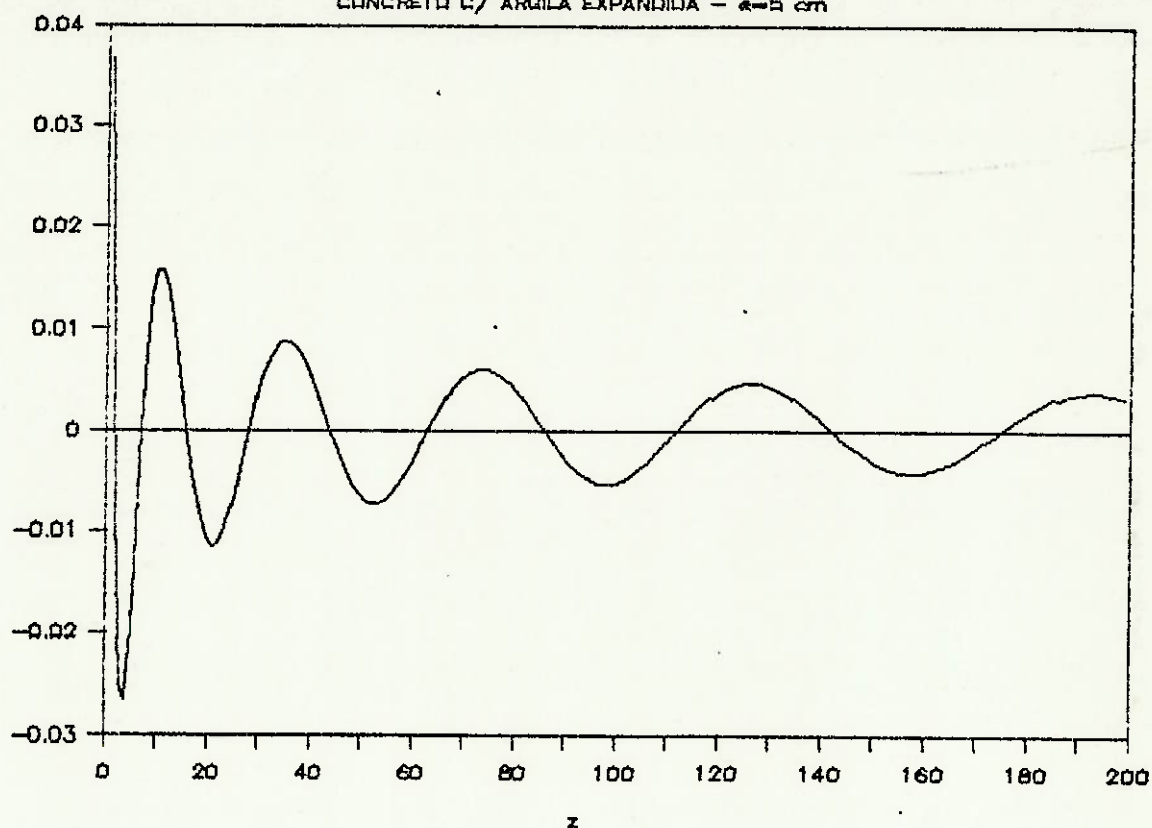


Figura 16: FUNCAO $B(z)$ — 1 CAMADA
CONCRETO C/ ARGILA EXPANDIDA — $e=10$ cm

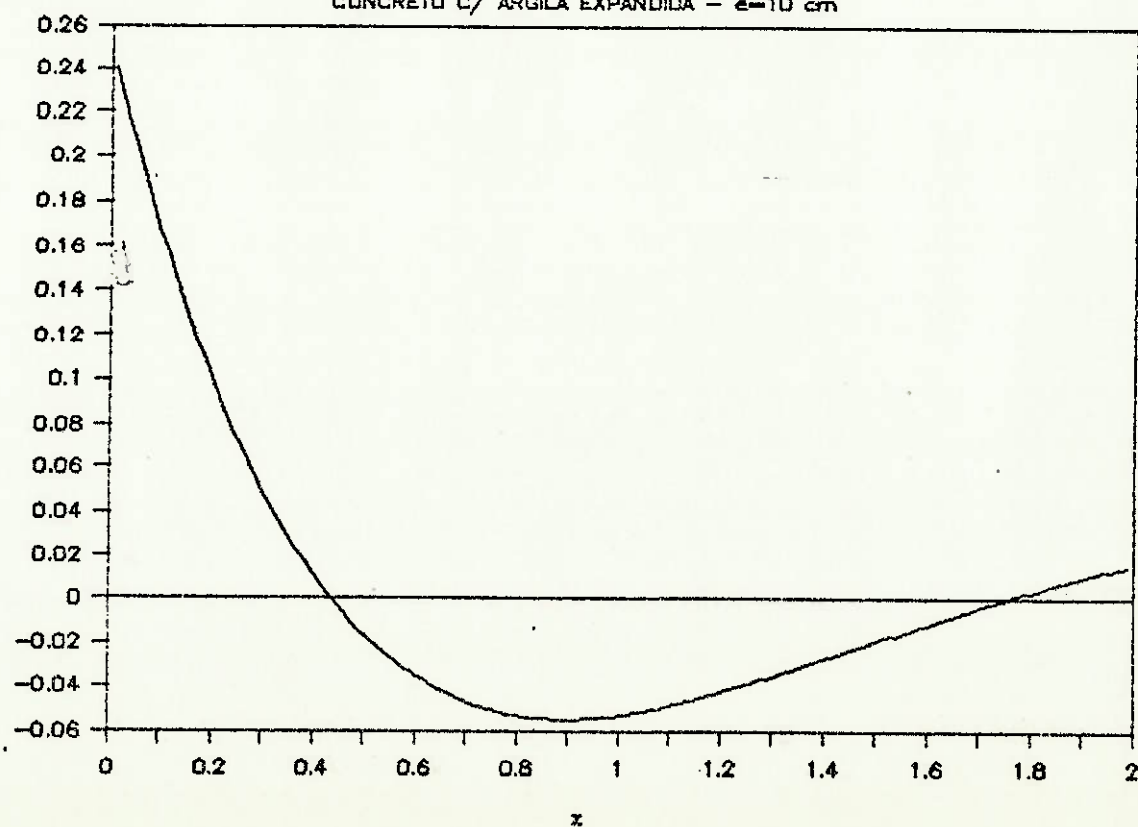


Figura 17: FUNCAO $B(z)$ — 2 CAMADAS
CONCRETO ($e=5$ cm) + BORRACHA ($e=1$ cm)

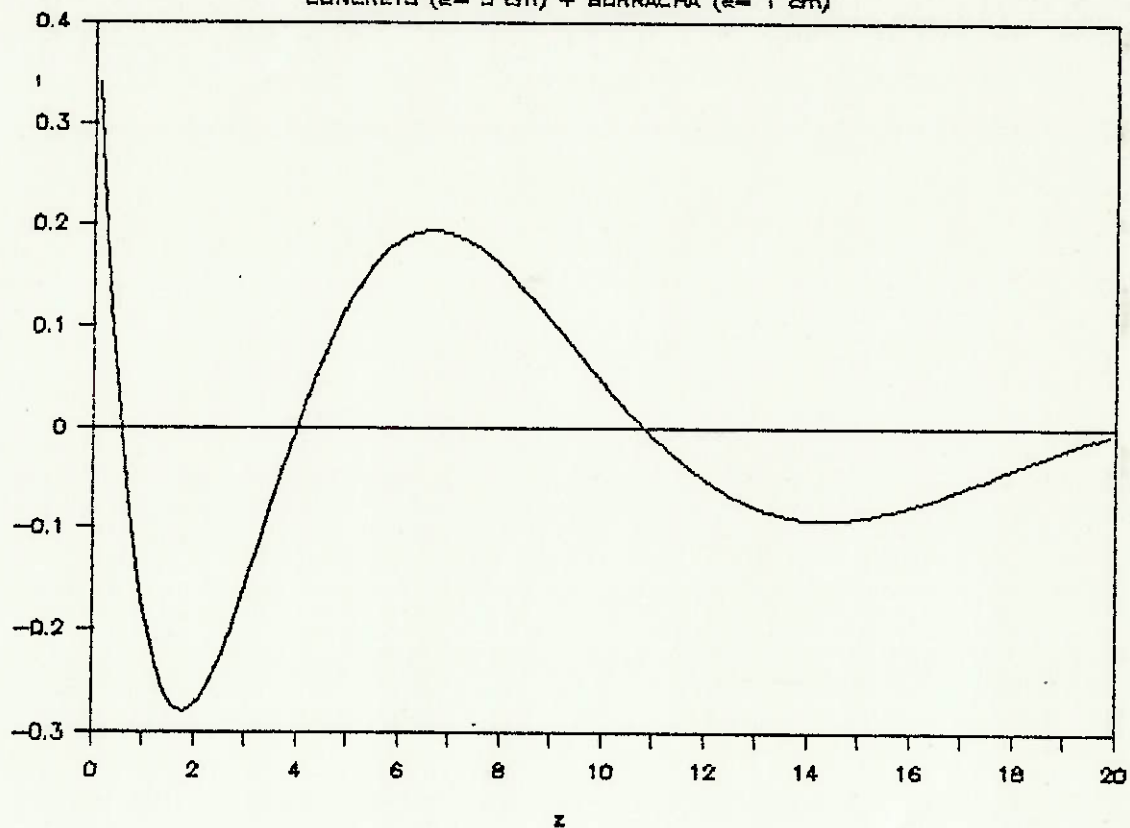


Figura 18: FUNCAO $B(z)$ — 2 CAMADAS
CONCRETO ($e=5$ cm) + BORRACHA ($e=1$ cm)

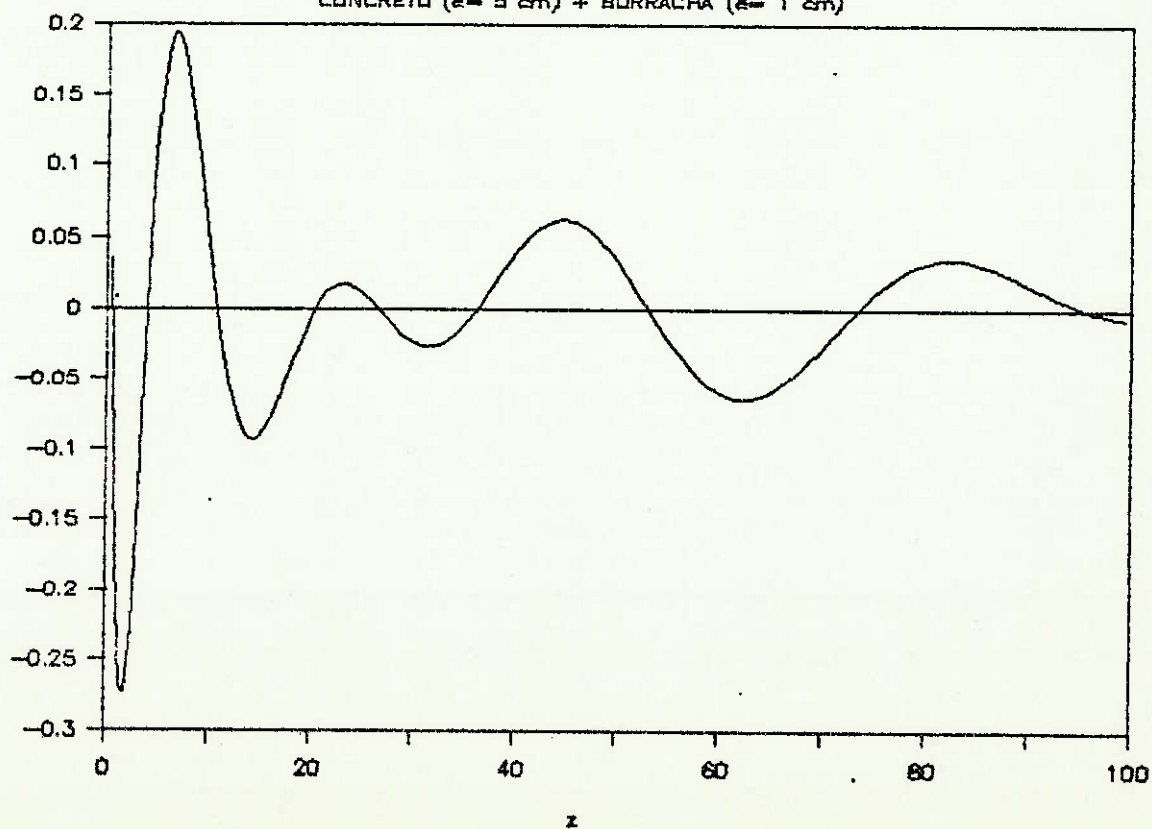


FIGURA 19 FUNCAO $B(z)$ - 2 CAMADAS
CONCRETO ($e=5$ cm) + BORRACHA ($e=5$ cm)

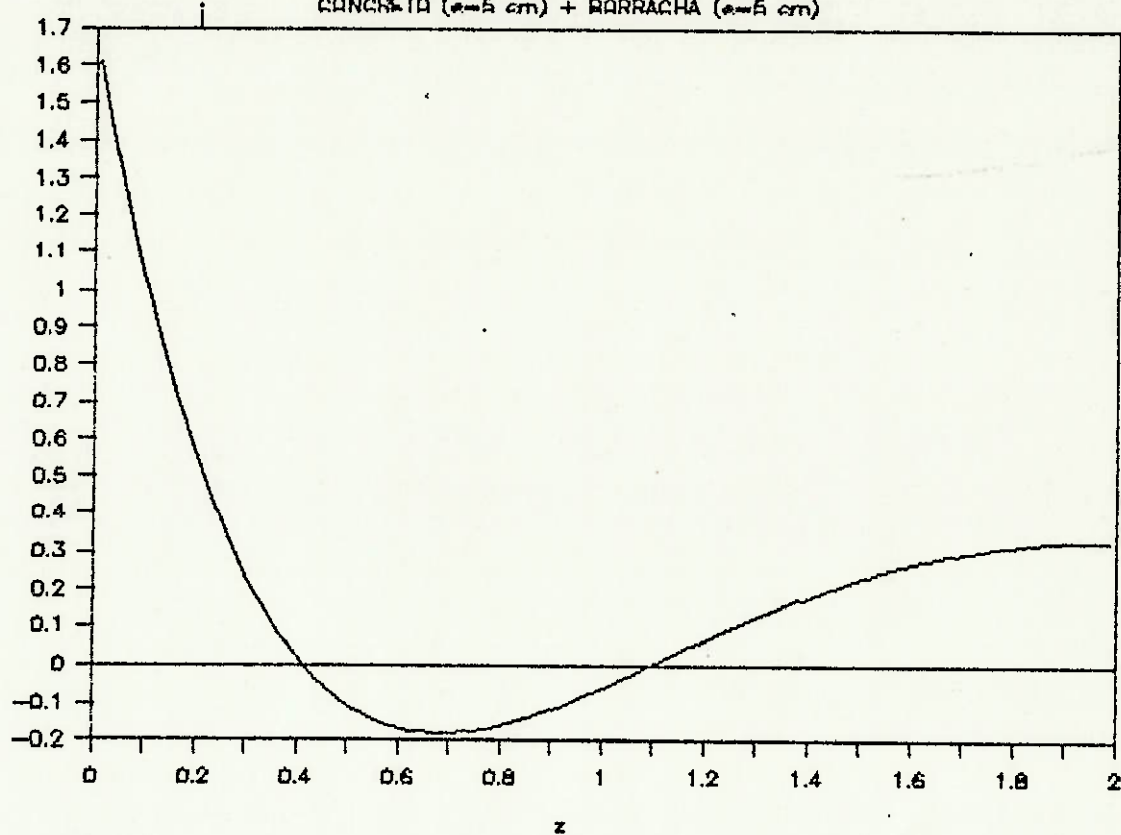


Figura 20: FUNCAO $B(z)$ - 2 CAMADAS
CONCRETO ($e=5$ cm) + BORRACHA ($e=5$ cm)

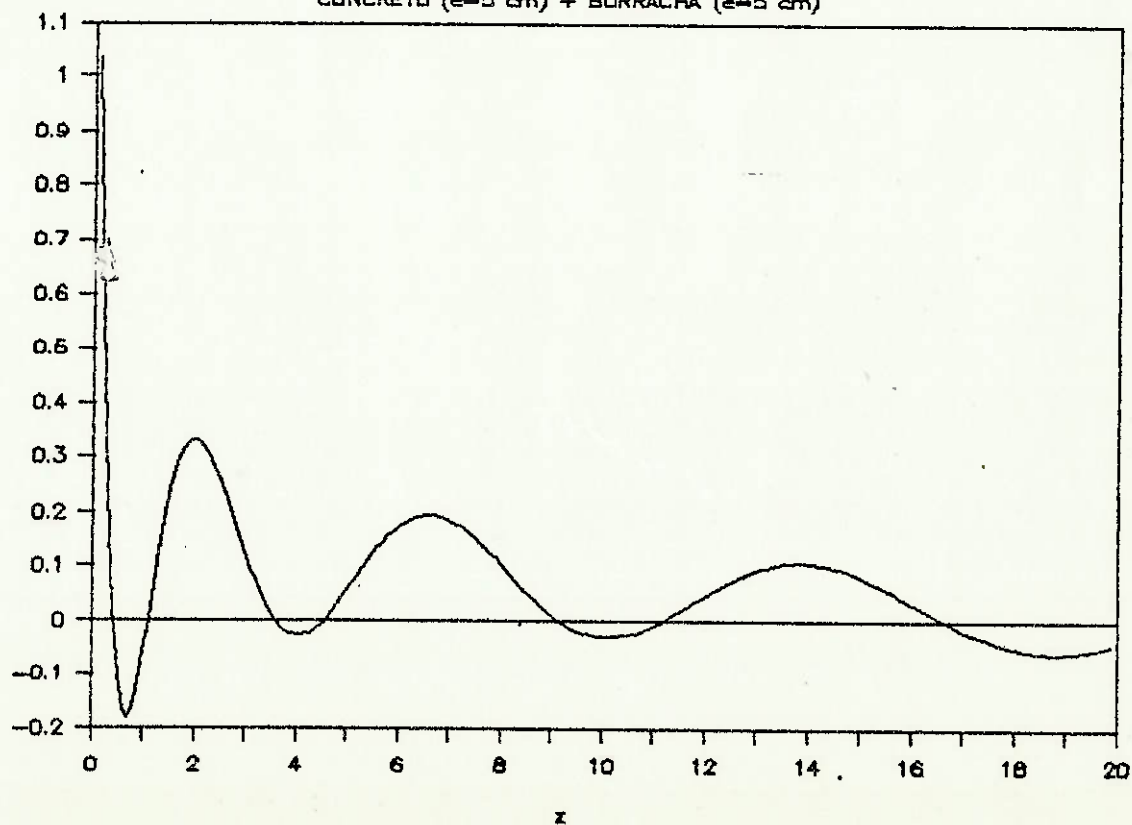


Figura 21: FUNCAO $B(z)$ — 2 CAMADAS
CONCRETO ($e=5$ cm) + BORRACHA ($e=10$ cm)

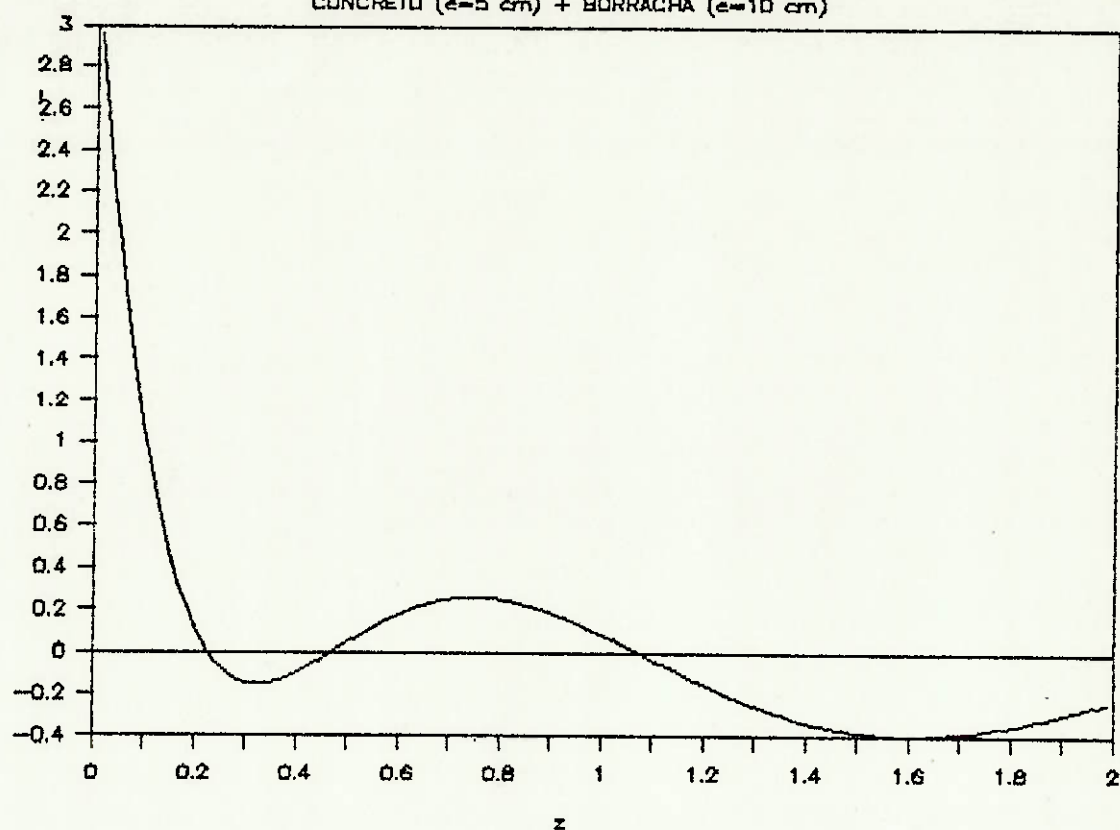


Figura 22: FUNCAO $B(z)$ — 2 CAMADAS
CONCRETO ($e=5$ cm) + BORRACHA ($e=10$ cm)

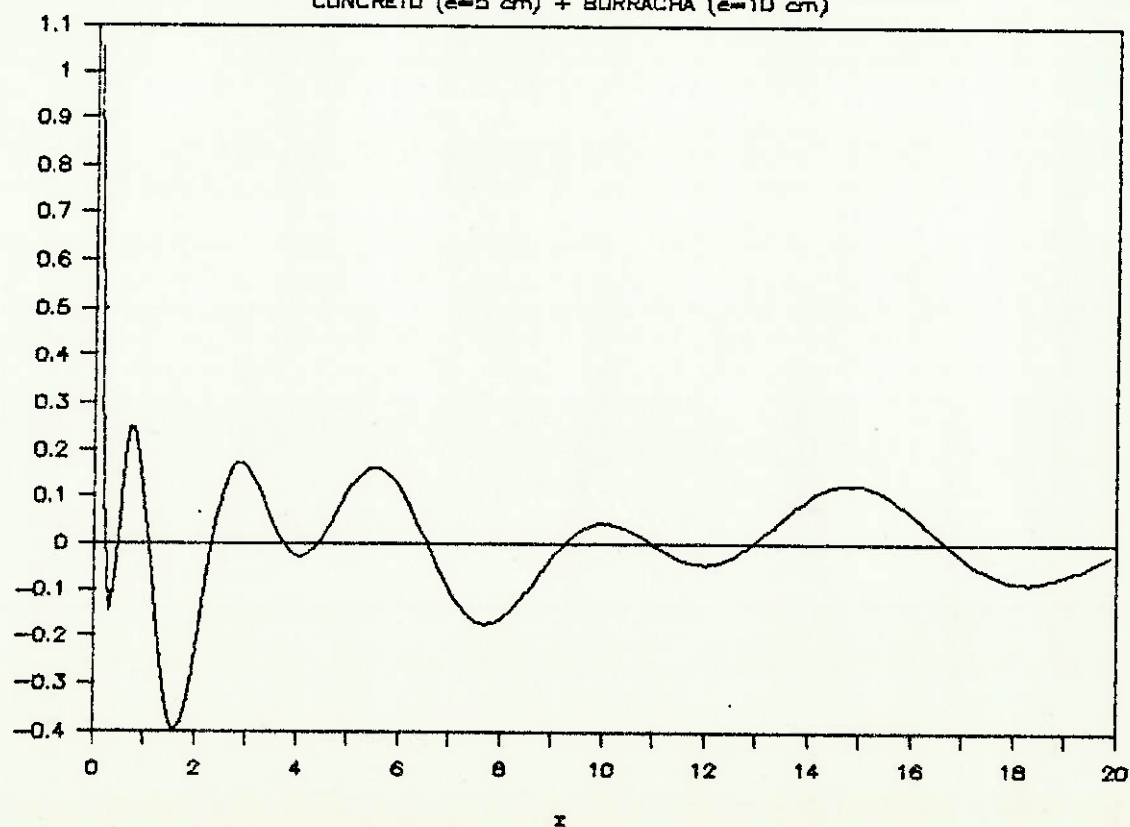


Figura 23: FUNCAO $B(z)$ — 2 CAMADAS
CONCRETO ($e=10$ cm) + BORRACHA ($e=1$ cm)

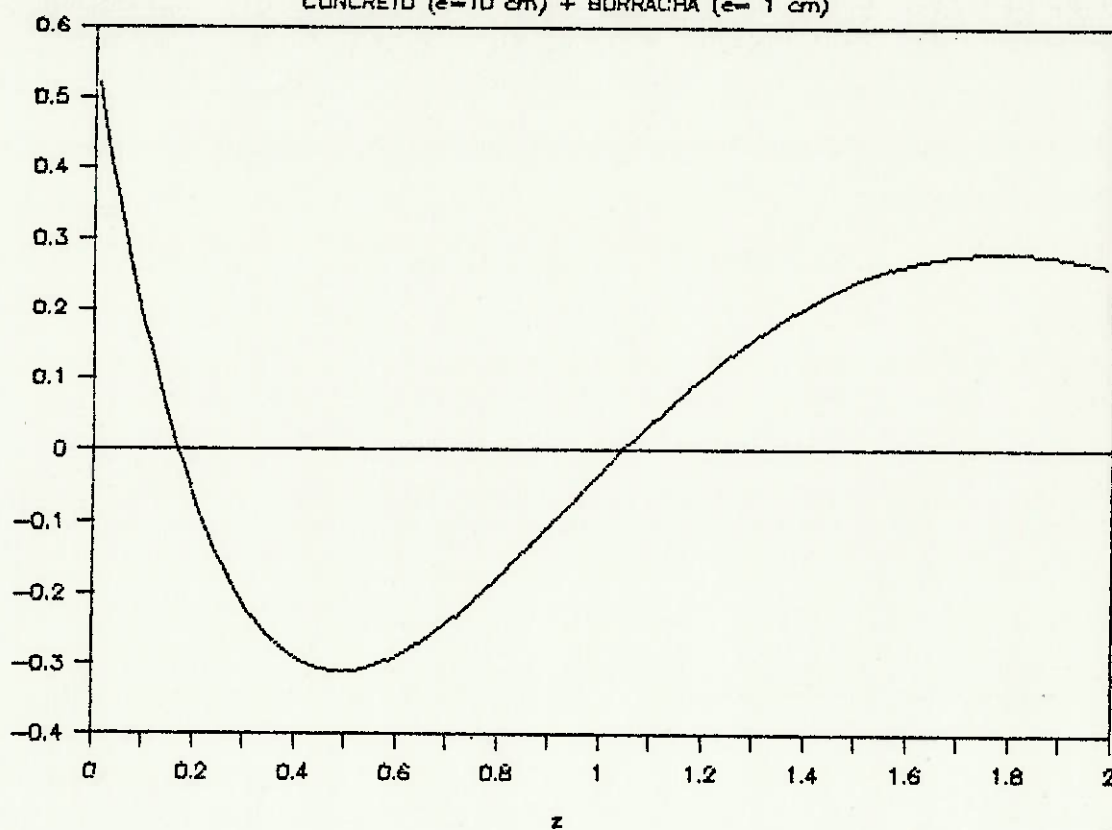


Figura 24: FUNCAO $B(z)$ — 2 CAMADAS
CONCRETO ($e=10$ cm) + BORRACHA ($e=1$ cm)

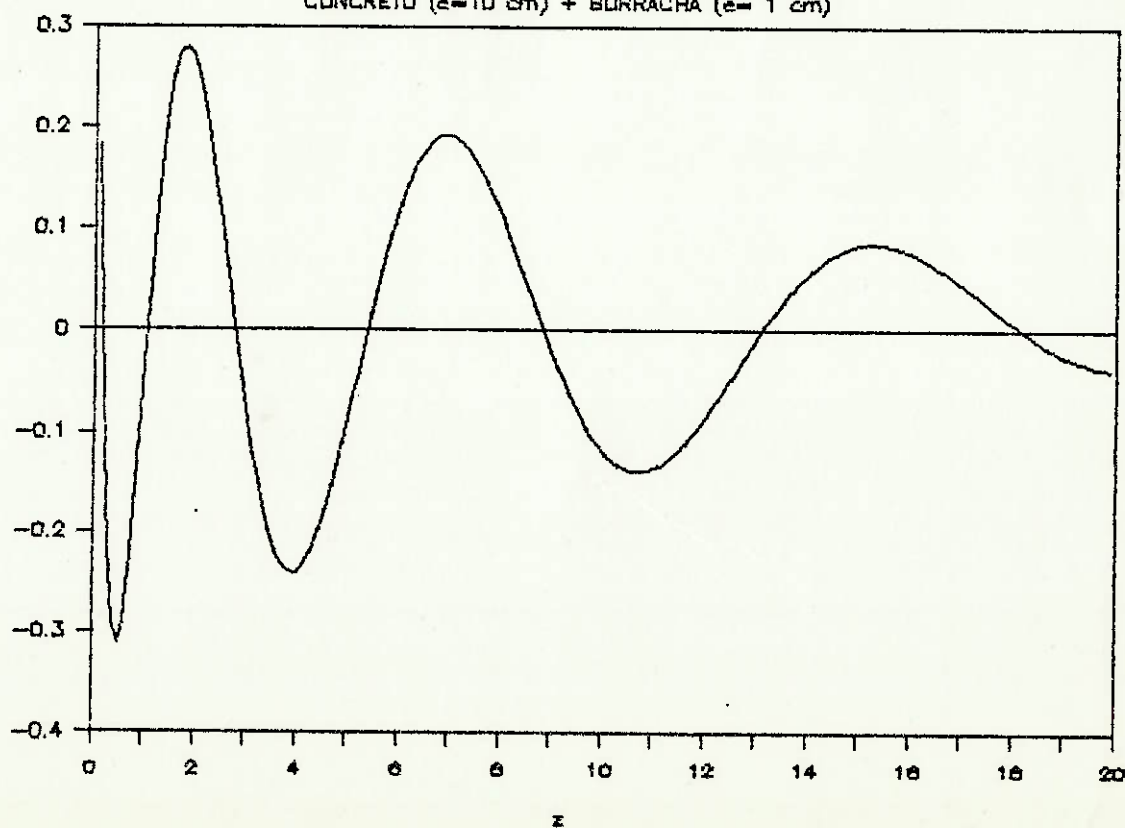


Figura 25: FUNCAO $B(z)$ — 2 CAMADAS
CONCRETO ($e=10$ cm) + BORRACHA ($e=5$ cm)

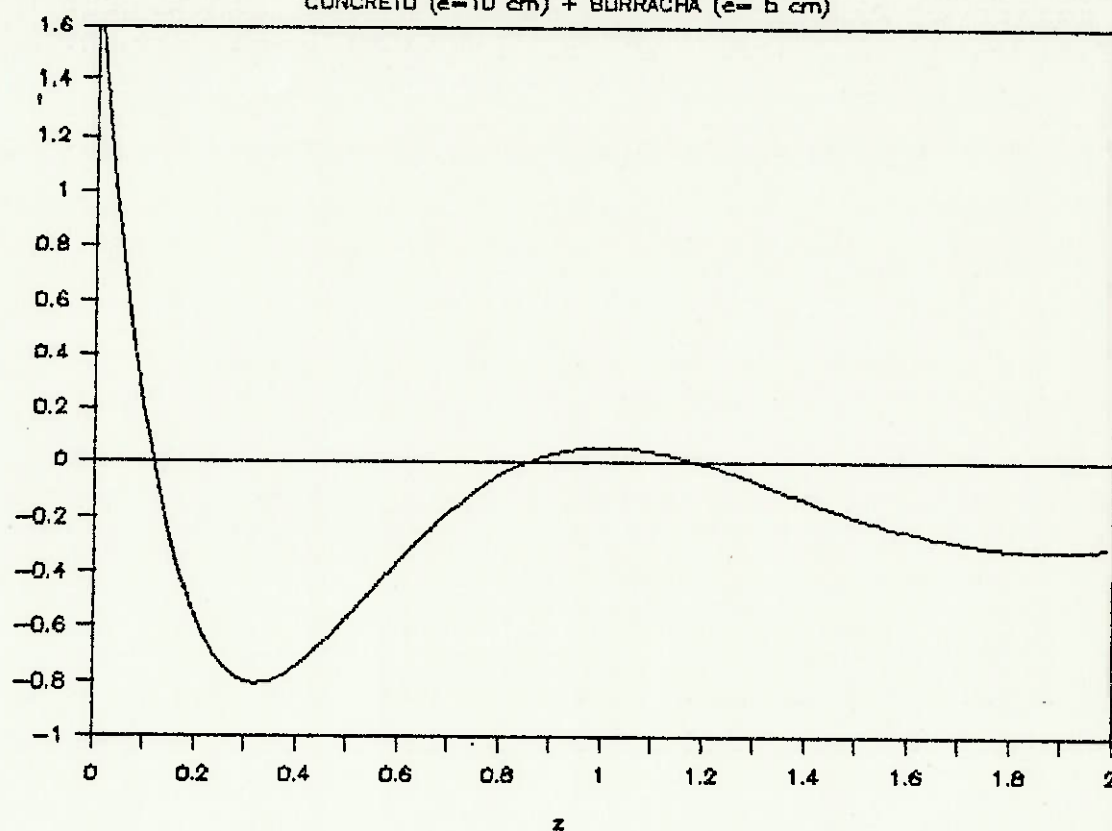


Figura 26: FUNCAO $B(z)$ — 2 CAMADAS
CONCRETO ($e=10$ cm) + BORRACHA ($e=5$ cm)

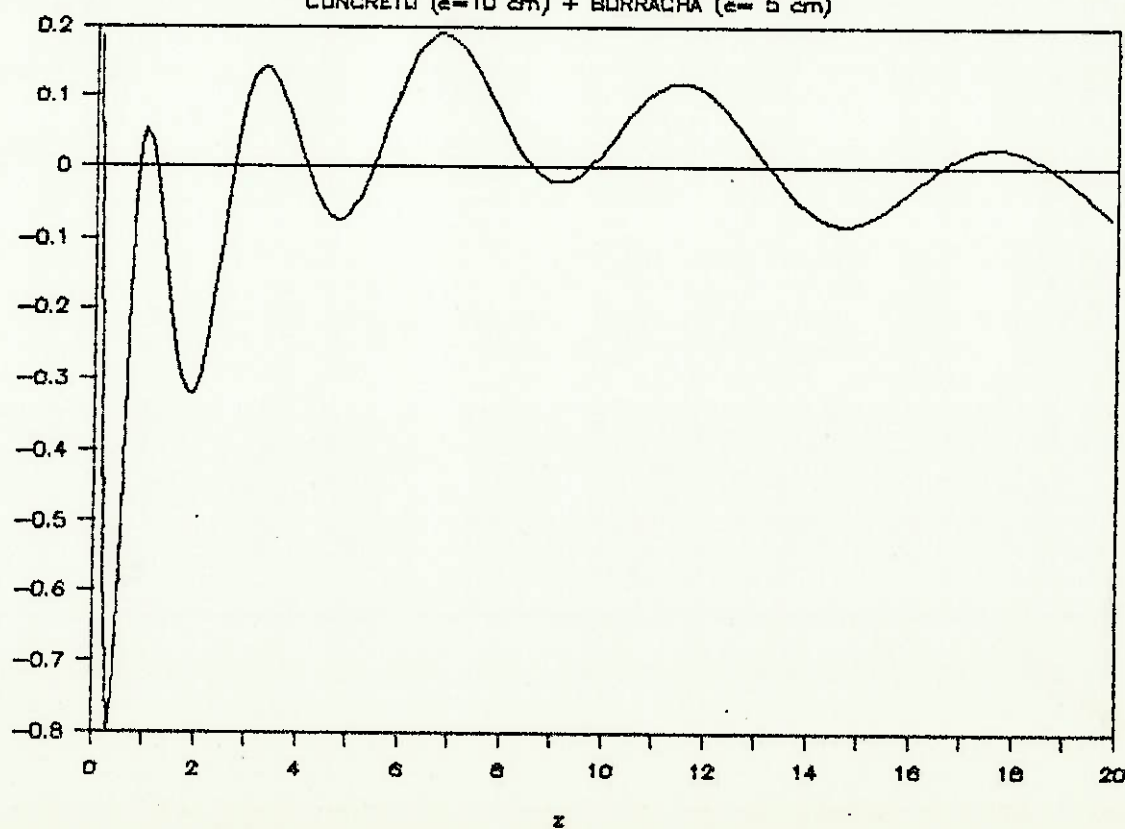


Figura 27: FUNCAO $B(z)$ — 2 CAMADAS
CONCRETO ($e=10$ cm) + BORRACHA ($e=10$ cm)

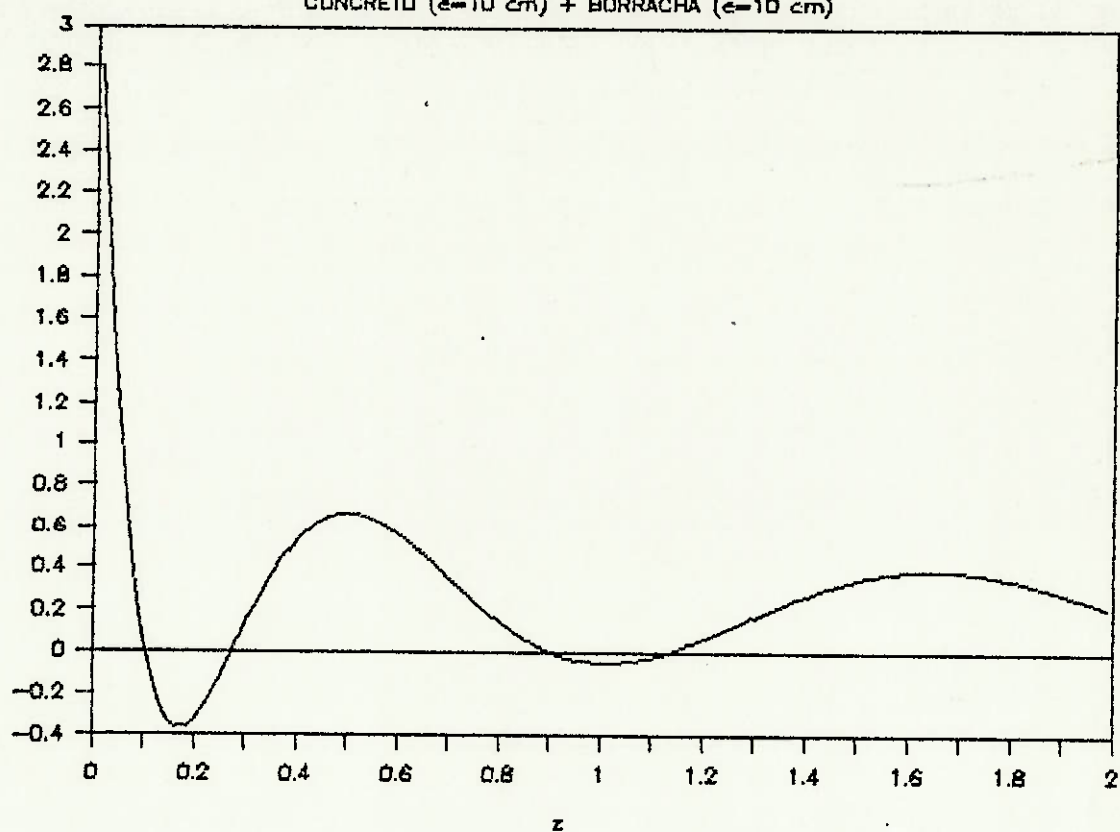
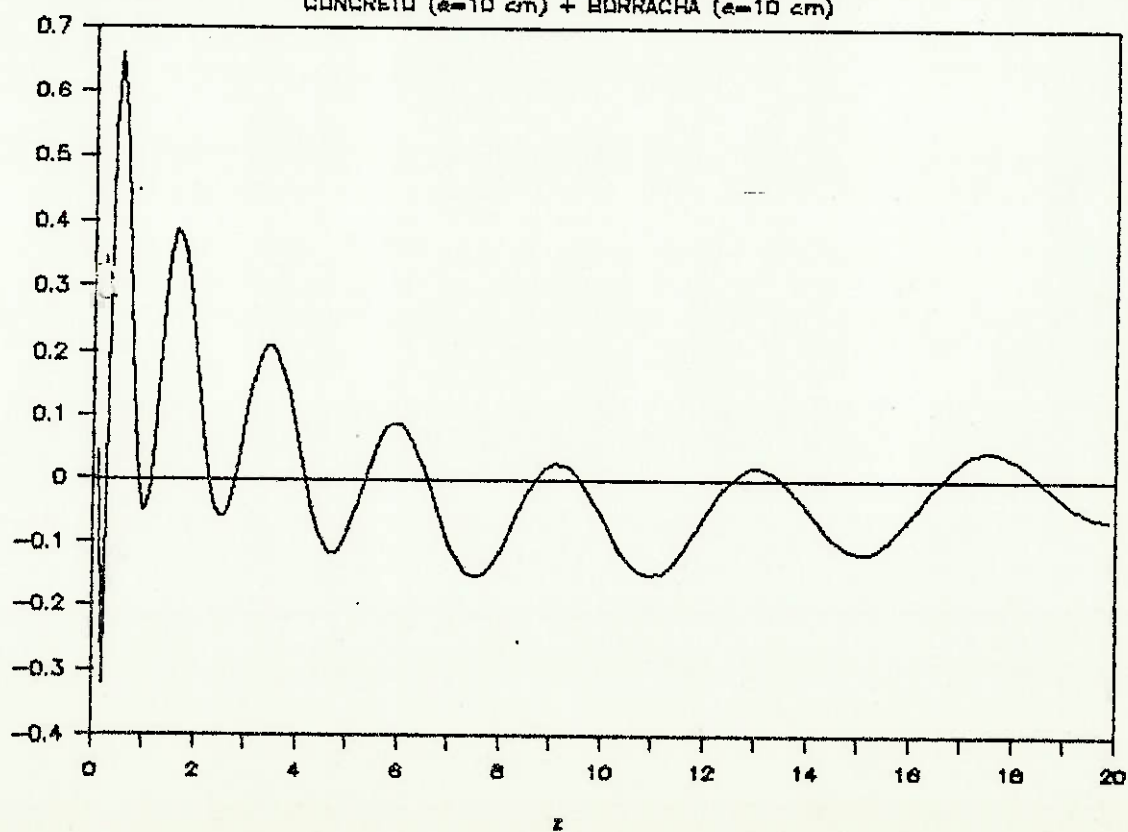


Figura 28: FUNCAO $B(z)$ — 2 CAMADAS
CONCRETO ($e=10$ cm) + BORRACHA ($e=10$ cm)



IV-Descrição do programa "FATORES"

Serão apresentados aqui os fluxogramas das principais rotinas de cálculo usadas no programa, sendo deixadas de lado as rotinas de entrada e saída, por não apresentarem complexidade ou interesse maior para este trabalho.

Observando-se a listagem do programa, pode-se perceber que as rotinas mais usadas foram colocadas nas linhas iniciais. Isto se deve ao fato de que este procedimento acelera a velocidade de execução do programa, já que quando determinada rotina é acionada, o computador começa a procura-la, de maneira sequencial, a partir da primeira linha de codificação. Assim, caso as rotinas mais acessadas se encontrassem no final do programa, o tempo gasto para suas localizações seria muito maior.

Será detalhado aqui apenas o módulo principal do programa FATORES por ser este que controla as demais rotinas. Este módulo apresenta a seguinte sequência de cálculos:

- Inicialização das variáveis;
- Chamada da rotina de entrada de dados;
- Cálculo dos dois primeiros termos das equações (34) e (35);
- Determinação da primeira raiz da equação $B(s) = 0$ pelo método de Newton-Raphson;

- Execução de um procedimento iterativo para a obtenção das demais raízes, até que a variação porcentual provocada por estas nos primeiros fatores de resposta seja menor que 0,01%;
- Determinação dos demais fatores de resposta usando as equações (35) e (36) até que seja atingida a razão comum.
- Chamada da rotina de impressão dos dados.

Na página seguinte é apresentada, a título de exemplo, uma tela usada para a entrada de dados no programa. Observando-a verifica-se que a mesma possibilita a correção de qualquer dado que por ventura tenha sido digitado erroneamente, antes da execução do programa.

A saída datela apresentada a seguir encontra-se no capítulo V.

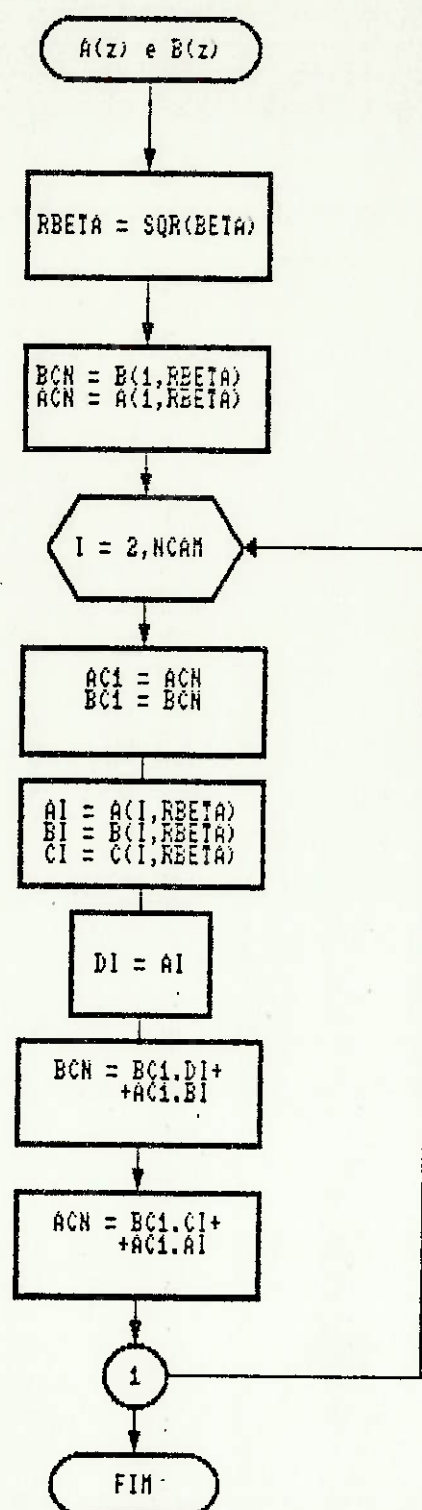
***** PROGRAMA PARA CALCULO DOS FATORES DE RESPOSTA TERMICA *****

Entre com o numero de camadas do elemento ? 3

Propriedades termicas dos materiais :

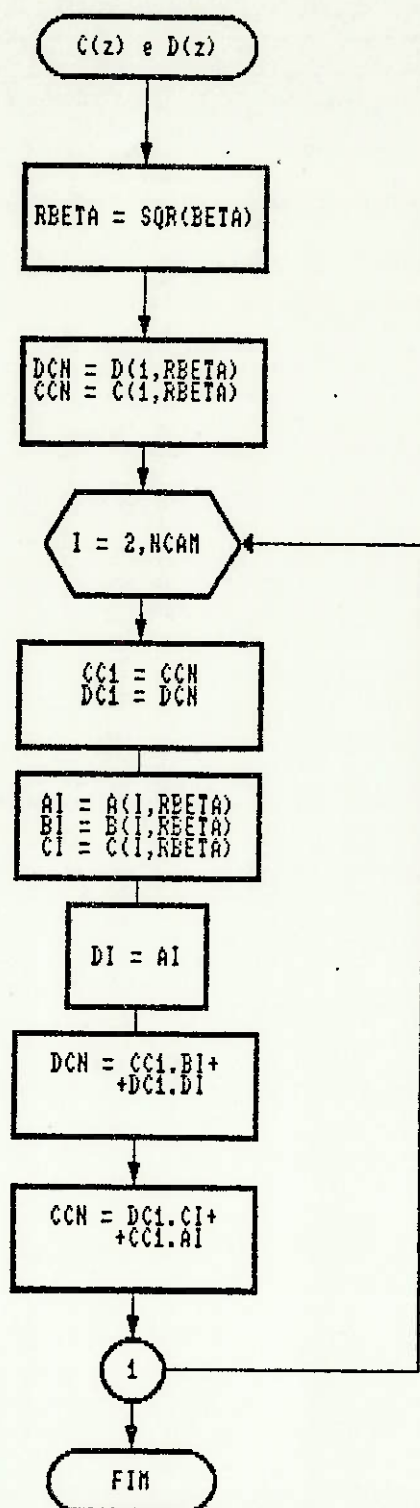
	L (m)	K (W/mK)	D (kg/m ³)	C (J/kgK)	Descricao
1 :	? .02	? 1	? 1790	? 780	? ARGAMASSA
2 :	? .2	? .8	? 1750	? 840	? ALVENARIA 1 TIJOLO
3 :	? .02	? 1	? 1750	? 780	? ARGAMASS

DESEJA CORRIGIR ? (S/N)



As funcoes $A(i)$, $B(i)$, $C(i)$ e $D(i)$ sao descritas no item II.3.2 e podem ser calculadas pelas equacoes (24), (25), (26) e (27), respectivamente.
As formulas usadas para o calculo de ACN e BCN sao apresentadas no item III.1.

Figura 29: Fluxograma da rotina que calcula $A(z)$ e $B(z)$



As funcoes $A(i)$, $B(i)$, $C(i)$ e $D(i)$ sao descritas no item II.3.2 e podem ser calculadas pelas equacoes (24), (25), (26) e (27), respectivamente.
As formulas usadas para o calculo de CCN e DCN sao apresentadas no item III.1.

Figura 30: Fluxograma da rotina que calcula $C(z)$ e $D(z)$

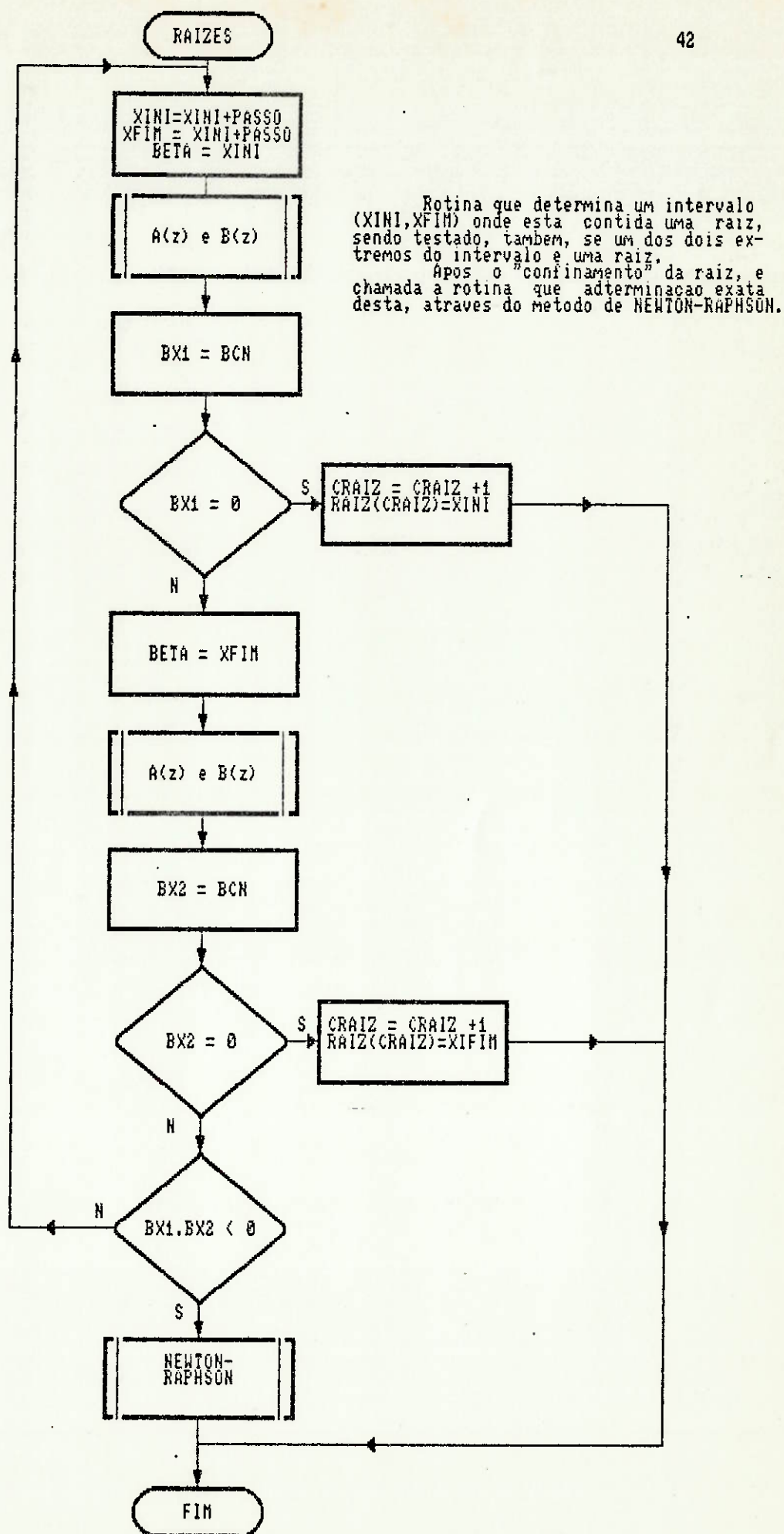
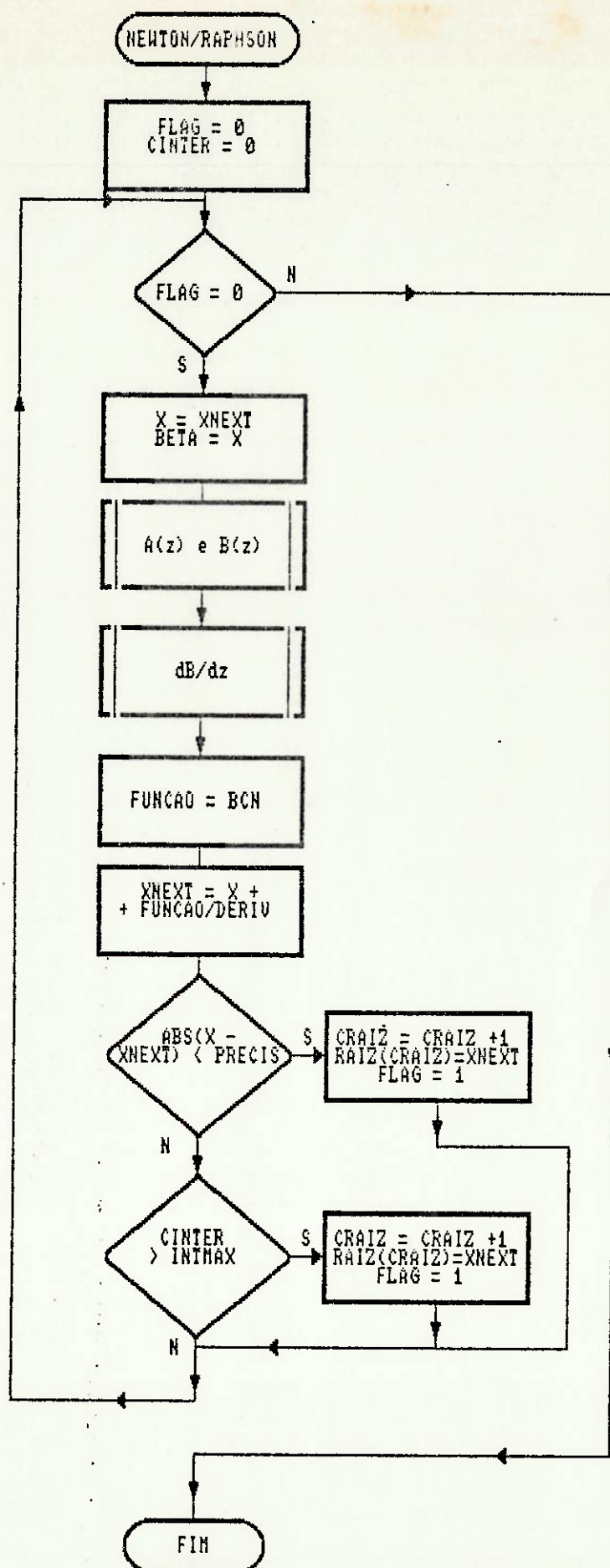
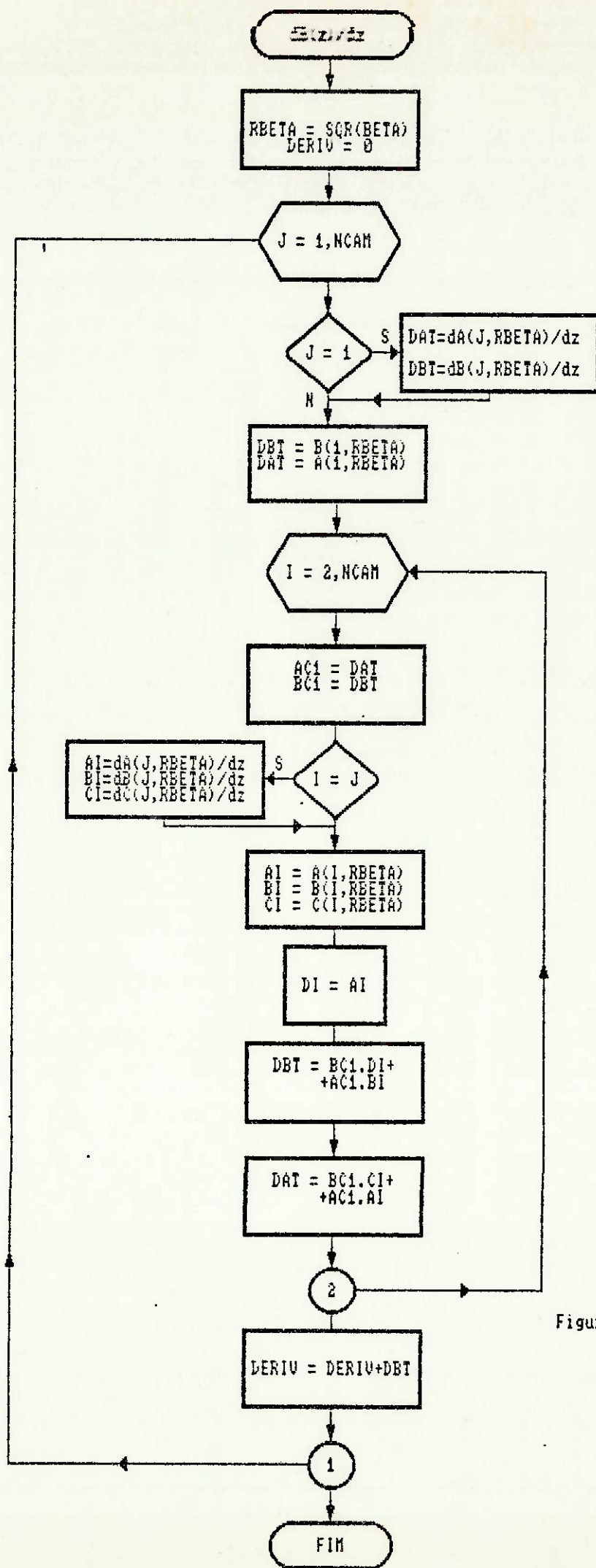


Figura 31: Fluxograma da rotina RAIZES



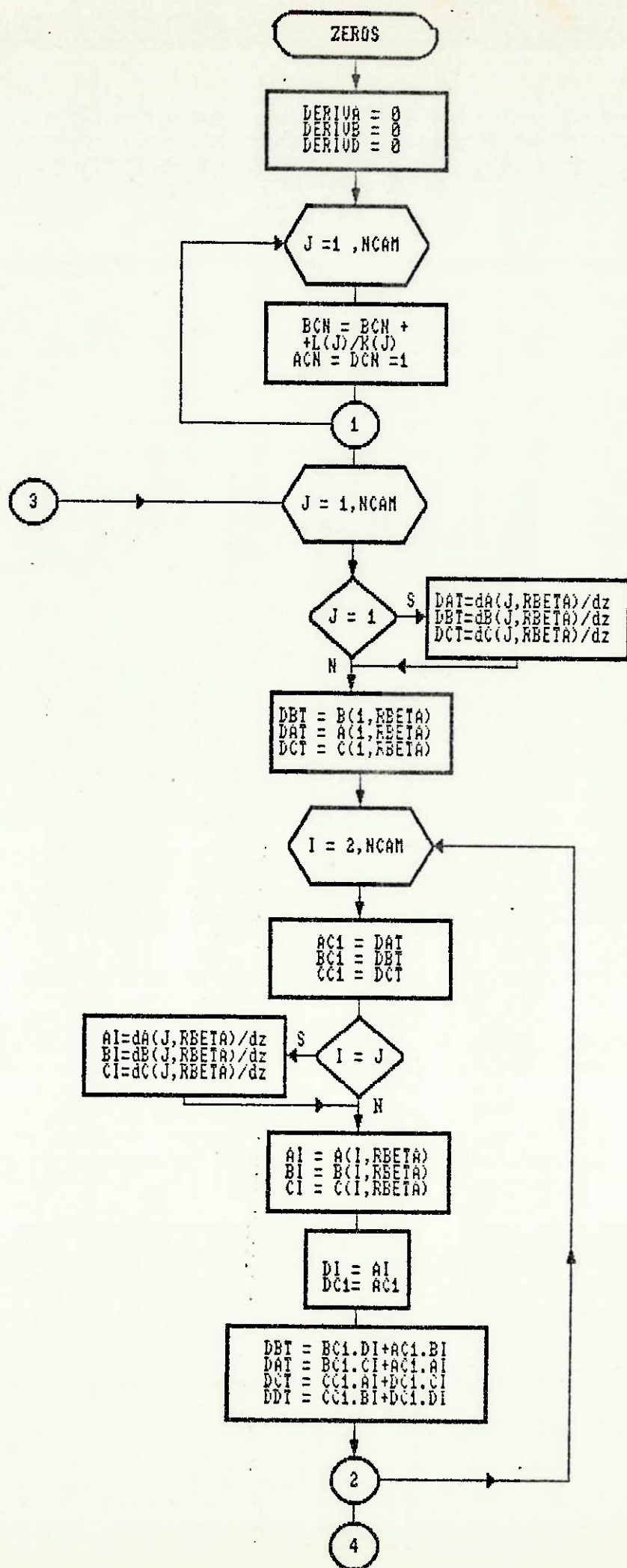
Rotina para o calculo das raizes da equacao $B(z) = 0$ pelo metodo de NEWTON-RAPHSON modificado, segundo o exposto no item III.2.1. O processamento e interrompido quando e atingida a precisao desejada ou o numero maximo de iteracoes.

Figura 32: Fluxograma da rotina NEWTON-RAPHSON



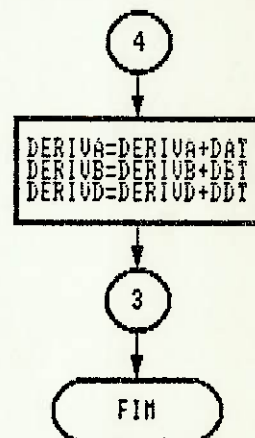
As funções $A(i)$, $B(i)$, $C(i)$ e $D(i)$ são descritas no item 11.3.2 e podem ser calculadas pelas equações (24), (25), (26) e (27), respectivamente.
As funções usadas para calcular as derivadas dos coeficientes citados acima, são (28) a (31).

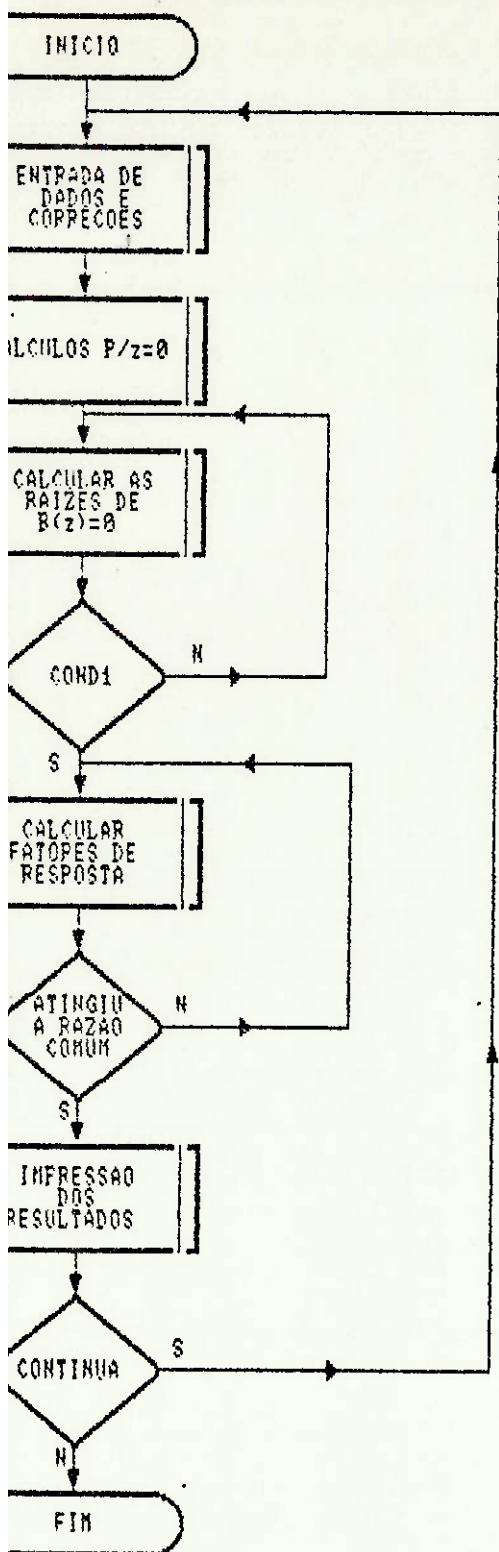
Figura 33: Fluxograma da rotina que calcula dB/dz p/ $z = BETA$



As funções $A(i)$, $B(i)$, $C(i)$ e $D(i)$ são descritas no item 11.3.2 e podem ser calculadas pelas equações (24), (25), (26) e (27), respectivamente.

As funções usadas para calcular as derivadas dos coeficientes citados acima, são apresentadas no mesmo item.





COND1: Condicao que determina a continuidade ou não do calculo das raizes da funcao $B(z) = 0$ em funcao da influencia da ultima raiz calculada no valor dos primeiros fatores de resposta $(X0, Y0, Z0)$.

Figura 35: Fluxograma simplificado do programa FAI0RES

V- Resultados

São apresentados nas páginas a seguir resultados de processamentos realizados com o programa FATORES.

Comparando-se os resultados obtidos nestes processamentos com aqueles obtidos quando se usa o programa listado no anexo B, encontra-se diferenças de menos de 1%, valor este considerado muito bom se forem levados em conta os erros computacionais característicos de microcomputadores.

Outro aspecto a ser considerado é o menor número de cálculos realizados pelo programa FATORES se comparado com o mesmo programa listado no anexo B, ou seja, enquanto o programa do anexo B calcula 40 raízes para a equação $B(z) = 0$, sem se importar com a influência desta raízes nos fatores de resposta, o programa FATORES calcula apenas o número de raízes necessário para que a variação porcentual do primeiro fator de resposta (já que este é o elemento mais fortemente afetado pelas raízes) seja menor que 0,01%. Este procedimento acelera em muito a velocidade de processamento; por exemplo a cobertura apresentada na página 48 consumiu apenas 3 segundos de processamento.

*** PROGRAMA PARA CALCULO DOS FATORES DE RESPOSTA TERMICA ****

Numero de camadas do elemento : 2

Propriedades termicas dos materiais :

	L (m)	K (W/mC)	D (kg/m ³)	C (J/kgC)	Descricao
1 :	0.10	1.900	2250	840	LAJE MACICA-CONCRET
2 :	0.01	0.060	100	840	CARPETE

X (W/m ² K)	Y (W/m ² K)	Z (W/m ² K)
+35.1986	+1.4931	+7.1688
-23.3065	+2.9625	-0.9101
-4.1638	+1.0094	-0.2448
-1.2671	+0.3073	-0.0745
-0.3857	+0.0935	-0.0227

Razao Comum (CR): 0.3044

*** PROGRAMA PARA CALCULO DOS FATORES DE RESPOSTA TERMICA ****

Numero de camadas do elemento : 3

Propriedades termicas dos materiais :

	L (m)	K (W/mC)	D (kg/m ³)	C (J/kgC)	Descricao
1 :	0.02	1.000	1790	780	ARGAMASSA
2 :	0.20	0.800	1750	840	ALVENARIA 1 TIJOLO
3 :	0.02	1.000	1750	780	ARGAMASSA

X (W/m ² K)	Y (W/m ² K)	Z (W/m ² K)
+21.3716	+0.0023	+21.2334
-12.8533	+0.1858	-12.7252
-2.0041	+0.6251	-1.9985
-1.0258	+0.6986	-1.0240
-0.6390	+0.5607	-0.6381
-0.4282	+0.4094	-0.4277
-0.2948	+0.2902	-0.2944
-0.2048	+0.2037	-0.2046
-0.1428	+0.1425	-0.1426
-0.0997	+0.0996	-0.0996
-0.0696	+0.0695	-0.0695
-0.0486	+0.0486	-0.0485

Razao Comum (CR): 0.6983

VI- Bibliografia

- AKUTSU, M. Aplicação do Método dos Fatores de Resposta para a Determinação da Resposta Térmica de Edificações. São Paulo, 1983. p irreg, (Dissertação de Mestrado apresentada à Escola Politécnica da USP).
- KUSUDA, T. Thermal response factors for multi-layer structures of various heat conduction systems. ASHRAE TRANSACTIONS, Part 1, p 250-69, 1969.
- MITALAS, G. P. & ARSENAULT, J. G. Fortran IV program to calculate heat flux response factors for a multi layer slab. Ottawa, NRCC/DBR, 1967. 20 p. (DBR Computer Program of The Division of Building Research) xerox.
- PEAVY, B. A. Determination and verification of thermal response factors for thermal conduction applications. Washington, NBS, 1978. 30p. (NBS Internal Report 77-1405).

ANEXO - A: Listagem do Programa "FATORES"

```

M RAIZ(50),X(50),Y(50),Z(50),F(10),G(10),L(10),K(10),D(10),C(10),DESC$(10)
SSO = .00005
RP = .0001
IMAX = 15
ECIS = .00000001
PACO$ =SPACE$(50)
LTA = 3600
VTI$ = "S"
RECFR = .0001
OTO 770 : REM CHAMADA DO PROGRAMA PRINCIPAL
EM ***** subrotina que calcula B(z) e A(z)
BETA = SQR(BETA)
CN = SIN (RBETA*F(1))/(RBETA*G(1)) : ACN = COS (RBETA*F(1))
FOR I = 2 TO NCAM
  AC1 = ACN:BC1 = BCN
  P1 =RBETA*F(I) :P2 = RBETA*G(I)
  BI = SIN(P1)/P2
  AI = COS(P1)
  CI = -SIN(P1)*P2
  DI = AI
  BCN = BC1*DI+AC1*BI
  ACN = BC1*CI+AC1*AI
NEXT I
RETURN
REM ***** subrotina que calcula dB/dz p/ z = beta
BETA = SQR(BETA)
DERIV = 0
FOR J = 1 TO NCAM
  P1 =RBETA*F(1) :P2 = RBETA*G(1)
  IF J<>1 THEN DBT = SIN(P1)/P2 :DAT = COS (P1) ELSE DAT =F(1)*SIN(P1)/2/RB
  DBT = (SIN(P1)/RBETA-COS(P1)*F(1))/BETA/2/G(1)
  FOR I = 2 TO NCAM
    BC1 = DBT : AC1 = DAT
    P1 =RBETA*F(I) :P2 = RBETA*G(I)
    IF I<>J THEN BI = SIN(P1)/P2 :AI = COS (P1) :CI = - SIN(P1)*P2 ELSE AI
  )*SIN(P1)/2/RBETA :BI = (SIN(P1)/RBETA-F(I)*COS(P1))/BETA/2/G(I) : CI= G(I)
  IN(P1)/RBETA+COS (P1)*F(I))
    DI = AI
    DBT = BC1*DI+AC1*BI
    DAT = BC1*CI+AC1*AI
  NEXT I
  DERIV = DERIV + DBT
EXT J
RETURN
EM ***** subrotina que calcula C(z) e D(z)
BETA = SQR(BETA)
CN = -RBETA*G(1)*SIN (RBETA*F(1)) : DCN = COS (RBETA*F(1))
FOR I = 2 TO NCAM
  DC1 = DCN:CC1 = CCN
  P1 = RBETA*F(I) :P2 = RBETA*G(I)
  BI = SIN(P1)/P2
  AI = COS(P1)
  CI = -SIN(P1)*P2
  DI = AI
  CCN = CC1*AI+DC1*CI
  DCN = CC1*BI+DC1*DI
EXT I
RETURN
EM ***** SUBROTINA NEWTON-RAPHSON *****
CINTER = 0 : FLAG = 0
WHILE FLAG = 0
  CINTER = CINTER + 1

```

```

X = XNEXT
BETA = X
GOSUB 40 : REM subrotina que calcula A,B
GOSUB 180 : REM subrotina que calcula dB/dz p/ z=beta
FUNCAO = BCN
IF DERIV = 0 THEN CRAIZ = CRAIZ + 1 : RAIZ(CRAIZ) = X : FLAG = 1
XNEXT = X + FUNCAO/DERIV
IF ABS(X-XNEXT) <= PRECIS THEN CRAIZ = CRAIZ + 1 : RAIZ(CRAIZ) = XNEXT : FL
1
IF CINTER > INTMAX THEN CRAIZ = CRAIZ + 1 : RAIZ(CRAIZ) = XNEXT : FLAG = 1
WEND
RETURN
REM ***** SUBROTINA QUE DETERMINA UM INTERVALO PARA A RAIZ *****
XINI = XINI + PASSO
XFIM = XINI + PASSO
BETA = XINI
GOSUB 40 : REM subrotina que calcula A(z) e B(z)
BX1 = BCN
IF BX1 = 0 THEN CRAIZ = CRAIZ + 1 : RAIZ(CRAIZ) = XINI: XINI=XFIM : PRINT "R
",CRAIZ," = ",XINI : RETURN
BETA = XFIM
GOSUB 40 : REM subrotina que calcula A(z) e B(z)
BX2 = BCN
IF BX2 = 0 THEN CRAIZ = CRAIZ + 1 : RAIZ(CRAIZ) = XFIM:XINI=XFIM : PRINT "RA
",CRAIZ," = ",XFIM : RETURN
IF BX1*BX2 > 0 THEN GOTO 650 ELSE XNEXT = XINI : GOSUB 490 : REM subrotina
"TON-RAPHSON"
RETURN
REM ***** PROGRAMA PRINCIPAL QUE CALCULA OS FRS *****
***
WHILE CONTI$ = "S" OR CONTI$ = "s"
FOR I = 1 TO 50
X(I) = 0
Y(I) = 0
Z(I) = 0
RAIZ(I) = 0
NEXT I
CRAIZ = 0
GOSUB 1510 : REM ROTINA QUE FAZ A ENTRADA DE DADOS
GOSUB 2050 : REM subrotina que calcula A,B,C,D e suas derivadas p/ z = 0
XNEXT = .0000001
XAUX = XO : YAUX = YO : ZAUX = ZO
GOSUB 490 : REM rotina que determina a primeira raiz por NEWTON-RAPHSON
BETA = RAIZ(1)
GOSUB 180 : REM rotina que calcula dB/dz p/ z = beta
GOSUB 40 : REM rotina que calcula A (z) p/ z = beta
YAA = 1*EXP(-BETA*DELTA)/(DERIV*BETA^2*DELTA)
ZAA = ACN*EXP(-BETA*DELTA)/(DERIV*BETA^2*DELTA)
GOSUB 350 : REM rotina que calcula D (z) p/ z = beta
XAA = DCN*EXP(-BETA*DELTA)/(DERIV*BETA^2*DELTA)
X1 = XAA + XO
Y1 = YAA + YO
Z1 = ZAA + ZO
WHILE ABS((XAUX-X1)/X1) > VARP OR ABS ((YAUX-Y1)/Y1) > VARP OR ABS ((ZAUX-Z
) > VARP
XINI = XNEXT
XAUX = X1 : YAUX = Y1 : ZAUX = Z1
GOSUB 640 : REM rotina que calcula as raizes da equacao
BETA = RAIZ(CRAIZ)
GOSUB 180 : REM rotina que calcula dB/dz p/ z = beta
GOSUB 40 : REM rotina que calcula A (z) p/ z = beta
SOMAY = 1*EXP(-BETA*DELTA)/(DERIV*BETA^2*DELTA)
SOMAZ = ACN*EXP(-BETA*DELTA)/(DERIV*BETA^2*DELTA)
GOSUB 350 : REM rotina que calcula D (z) p/ z = beta
SOMAX = DCN*EXP(-BETA*DELTA)/(DERIV*BETA^2*DELTA)
X1 = X1 + SOMAX

```



```

Y1 = Y1 + SOMAY
Z1 = Z1 + SOMAZ
WEND
X(1) = X1 : Y(1) = Y1 : Z(1) = Z1
SOMAX=0 : SOMAY=0 : SOMAZ=0
FOR IND = 1 TO CRAIZ
  BETA = RAIZ(IND)
  GOSUB 180 : REM subrotina que calcula dB/dz p/ z = beta
  GOSUB 40 : REM rotina que calcula A (z) p/ z = beta
  SOMAY = SOMAY + 1*EXP(-2*BETA*DELTA)/(DERIV*BETA^2*DELTA)*(1-2*EXP(BETA*
A))
  SOMAZ = SOMAZ + ACN*EXP(-2*BETA*DELTA)/(DERIV*BETA^2*DELTA)*(1-2*EXP(BET
LTA))
  GOSUB 350 : REM rotina que calcula D (z) p/ z = beta
  SOMAX = SOMAX + DCN*EXP(-2*BETA*DELTA)/(DERIV*BETA^2*DELTA)*(1-2*EXP(BET
LTA))
NEXT IND
X(2) = XXO + SOMAX
Y(2) = YYO + SOMAY
Z(2) = ZZO + SOMAZ
CRX = X(2)/X(1) : CRY = Y(2)/Y(1) : CRZ = Z(2)/Z(1)
INDFR = 2
WHILE ABS(CRX - CRY) > PRECFR OR ABS (CRY - CRZ) > PRECFR OR ABS(CRZ - CR
PRECFR
  INDFR = INDFR +1
  FOR IND = 1 TO CRAIZ
    BETA = RAIZ(IND)
    GOSUB 40 : REM rotina que calcula A (z) p/ z = beta
    GOSUB 180 : REM subrotina que calcula dB/dz p/ z = beta
    Z (INDFR) = Z(INDFR) + ACN*EXP(-(INDFR)*BETA*DELTA)/(DERIV*BETA^2*DELT
1-EXP(BETA*DELTA))^2)
    Y (INDFR) = Y(INDFR) + 1*EXP(-(INDFR)*BETA*DELTA)/(DERIV*BETA^2*DELTA)
-EXP(BETA*DELTA))^2)
    GOSUB 350 : REM rotina que calcula D (z) p/ z = beta
    X (INDFR) = X(INDFR) + DCN*EXP(-(INDFR)*BETA*DELTA)/(DERIV*BETA^2*DELT
1-EXP(BETA*DELTA))^2)
  NEXT IND
  CRX = X(INDFR)/X(INDFR-1)
  CRY = Y(INDFR)/Y(INDFR-1)
  CRZ = Z(INDFR)/Z(INDFR-1)
WEND
LOCATE 22,20 : PRINT ESPACO$
LOCATE 18,20 : INPUT "DESEJA SAIDA NA IMPRESSORA ? (S/N) ", IMPR$
IF IMPR$ = "S" OR IMPR$ = "s" THEN GOSUB 1970 : GOTO 1430
IF IMPR$ = "n" OR IMPR$ = "N" THEN GOSUB 2400 ELSE BEEP : GOTO 1411
REM GOSUB 1970 : REM ROTINA QUE FAZ A IMPRESSAO DOS RESULTADOS
CLS
LOCATE 10,30 : PRINT "DESEJA FAZER OUTRO CALCULO ? (S/N) "
CONTI$ = INKEY$ : IF CONTI$ = "" THEN 1450
IF CONTI$ = "S" OR CONTI$ = "s" THEN GOTO 1480
IF CONTI$ = "n" OR CONTI$ = "N" THEN GOTO 1480 ELSE BEEP : GOTO 1440
WEND
CLS
END
REM ***** subrotina que faz a entrada de dados
CLS
LOCATE 1,3 : PRINT "***** PROGRAMA PARA CALCULO DOS FATORES DE RESPOSTA
ICA *****"
LOCATE 3,10 : INPUT "Entre com o numero de camadas do elemento ";NCAMX$
IF NCAMX$ <> "" THEN NCAM = VAL(NCAMX$)
LOCATE 5,13 : PRINT "Propriedades termicas dos materiais : "
LOCATE 6,13 : PRINT "L (m) K (W/mK) D (kg/m^3) C (J/kgK) Descricao"
FOR I = 1 TO NCAM
  LOCATE I+6,1 : PRINT "Camada ";I;": " : LOCATE I+6,13 : INPUT LX$
  LOCATE I+6,21 : INPUT KX$
  LOCATE I+6,32 : INPUT DX$
  LOCATE I+6,43 : INPUT CX$

```

```

LOCATE 1+6,45 : INPUT CX$
LOCATE 1+6,57 : INPUT DESCX$
IF KX$ <> "" THEN K(I) = VAL(KX$)
IF LX$ <> "" THEN L(I) = VAL(LX$)
IF DX$ <> "" THEN D(I) = VAL(DX$)
IF CX$ <> "" THEN C(I) = VAL(CX$)
IF DESCX$ <> "" THEN DESC$(I) = DESCX$
NEXT I
LOCATE 1+8,25 : PRINT "DESEJA CORRIGIR ? (S/N) "
ENT$ = INKEY$ : IF ENT$ = "" THEN 1710
IF ENT$ = "S" OR ENT$ = "s" THEN LOCATE 1+8,23 : PRINT ESPACO$ : GOTO 1540
IF ENT$ = "n" OR ENT$ = "N" THEN GOTO 1740 ELSE BEEP : GOTO 1700
FOR I = 1 TO NCAM
  RALFA = SQR (K(I)/(D(I)*C(I)))
  F(I) = L(I)/RALFA
  G(I) = K(I)/RALFA
NEXT I
LOCATE 22,25 : COLOR 31: PRINT " A G U A R D E - C A L C U L A N D O " : CO
7,0
RETURN
REM ***** subrotina que faz a IMPRESSAO DO CABECALHO DA FOLHA
LPRINT CHR$(140)
LPRINT : LPRINT : LPRINT
LPRINT "***** PROGRAMA PARA CALCULO DOS FATORES DE RESPOSTA TERMICA **
*****" : LPRINT : LPRINT
LPRINT "
          Numero de camadas do elemento :";NCAM
LPRINT : LPRINT
LPRINT "
          Propriedades termicas dos materiais : "
LPRINT "
          L (m)          K (W/mC)          D (kg/m^3)          C (J/kgC)          Des
10"
FOR I = 1 TO NCAM
LPRINT "Camada ";I;": " USING " #.##          #.###          #####          #####
&";L(I),K(I),D(I),C(I),DESC$(I)
NEXT I
LPRINT : LPRINT : LPRINT
LPRINT "
          n          X (W/m^2 K)          Y (W/m^2 K)          Z (W/m^2 K)"
JA = 1
RETURN
REM ***** SUBROTINA QUE IMPRIME OS VALORES CALCULADOS
JA = 16
FOR II=1 TO INDFR
  IF JA > 15 THEN GOSUB 1810
  LPRINT USING " ##          +##.####          +##.####          +##.####";II-
I),Y(II),Z(II)
  JA = JA + 1
NEXT II
LPRINT : LPRINT : LPRINT "
          Razao Comum (CR): " USING "###.###";CRX
RETURN
REM ***** SUBROTINA QUE CALCULA OS VALORES DE A,B,D E SUAS DERIVADAS
= 0
ACN = 1 : DCN = 1 : CCN = 0
BCN = 0
FOR I = 1 TO NCAM : BCN = BCN + L(I)/K(I) : NEXT I
DERIVA = 0
DERIVB = 0
DERIVD = 0
FOR J = 1 TO NCAM
  P1 = L(1)^2/(2*(K(1)/(D(1)*C(1)))) : P2 = L(1)/K(1)
  IF J<>1 THEN DBT = P2 : DAT = 1 : DCT = 0 ELSE DAT = P1 : DBT = P1*P2/3 :
1/P2*2
  DDT = DAT
  FOR I = 2 TO NCAM
    BC1 = DBT : AC1 = DAT : CC1 = DCT : DC1 = DDT
    P1 = L(I)^2/(2*(K(I)/(D(I)*C(I)))) : P2 = L(I)/K(I)
    IF I=J THEN BI = P1*P2/3 : AI = P1 : CI = P1/P2*2 ELSE AI = 1 : BI = P2
0

```



```

DI = AI
DBT = BC1*DI+AC1*BI
DAT = BC1*CI+AC1*AI
DDT = CC1*BI+DC1*DI
DCT = CC1*AI+DC1*CI
NEXT I
DERIVB = DERIVB + DBT
DERIVA = DERIVA + DAT
DERIVD = DERIVD + DDT
NEXT J
XO = DCN/BCN + (DERIVD/(BCN*DELTA) - DCN*DERIVB/(BCN^2*DELTA))
YO = 1/BCN - 1*DERIVB/(BCN^2*DELTA)
ZO = ACN/BCN + (DERIVA/(BCN*DELTA) - ACN*DERIVB/(BCN^2*DELTA))
XXO = -(DERIVD/(BCN*DELTA) - DCN*DERIVB/(BCN^2*DELTA))
YYO = 1*DERIVB/(BCN^2*DELTA)
ZZO = -(DERIVA/(BCN*DELTA) - ACN*DERIVB/(BCN^2*DELTA))
RETURN
REM ***** SUBROTINA QUE IMPRIME OS VALORES CALCULADOS
JA = 16
FOR II=1 TO INDFR
  IF JA > 15 THEN GOSUB 2600
  PRINT USING " ##          +##.####          +##.####          +##.####"; II-1
  ,Y(II),Z(II)
  JA = JA +1
NEXT II
PRINT :PRINT :PRINT "          Razao Comum (CR): " USING "#.####";CRX
PRINT :INPUT "          PRESSIONE [ENTER] P/ CONTINUAR ",LLL
RETURN
REM ***** subrotina que faz a IMPRESSAO DO CABECALHO DA FOLHA
CLS
PRINT "***** PROGRAMA PARA CALCULO DOS FATORES DE RESPOSTA TERMICA ***
*****" :PRINT :PRINT
PRINT "          Numero de camadas do elemento :";NCAM
PRINT :PRINT
PRINT "          Propriedades termicas dos materiais : "
PRINT "          L (m)          K (W/mC)          D (kg/m^3)          C (J/kgC)          Desc
"
FOR I = 1 TO NCAM
PRINT "Camada ";I;": " USING " #.##          #.###          ####          ####
";L(I),K(I),D(I),C(I),DESC$(I)
NEXT I
PRINT : PRINT : PRINT
PRINT "          n          X (W/m^2 K)          Y (W/m^2 K)          Z (W/m^2 K)"
JA = 1
RETURN

```

ANEXO - B: Listagem do Programa "RESFX"

[illegible]

```

00002190
00002200
00002210
00002220
00002230
00002240
00002250
00002260
00002270
00002280
00002290
00002300
00002310
00002320
00002330
00002340
00002350
00002360
00002370
00002380
00002390
00002400
00002410
00002420
00002430
00002440
00002450
00002460
00002470
00002480
00002490
00002500
00002510
00002520
00002530
00002540
00002550
00002560
00002570
00002580
00002590
00002600
00002610
00002620
00002630
00002640
00002650
00002660
00002670
00002680
00002690
00002700
00002710
00002720
00002730
00002740
00002750
00002760
00002770
00002780

200 FORMAT ( ' FOLHA DE DADOS 6: N/L,K,P,C,R' )
210 FORMAT ( ' FOLHA DE DADOS 7: DESCRICAO DE CADA CAMADA' )
220 FORMAT (2H )
230 FORMAT (7H1 IRF=110)
240 FORMAT (77H0 CAMADA L(I) K(I) D(I) C(I)
*ES(I) DESCRICAO )
250 FORMAT ( ' NUM.
2 DAS CAMADAS' )
260 FORMAT (116,1F11.3,1F10.3,1F10.2,1F10.3,1F8.2,2X,6A4)
270 FORMAT (59H0 CONDUCTANCIA TERMICA
2 U=1F7.3)
280 FORMAT (49H0
290 FORMAT (50H0
300 FORMAT (770H
2 Z)
310 FORMAT (1117,1F23.4,2F15.4)
320 FORMAT (6A4)
330 FORMAT (44H0
340 FORMAT (50H0 COMPOSICAO DA PAREDE
350 FORMAT (1F27.3,1F10.2,1F10.3,10X,6A4)
360 FORMAT (50H0 J
370 FORMAT (1124,1F21.5) F
380 FORMAT ( )

END
SUBROUTINE RESPTK (K,L,G,AG,KG,X,Y,Z,NL,DT,NR,CR,U)
*****
DIMENSION K(10),L(10),G(10),X(50),Y(50),Z(50),AP(10),BP(10),
2 CP(10),DP(10),A(10),B(10),C(10),D(10),Zr1(3),Zr2(3),RAP(3),
3 RGT(50),RA(3,50),ZRK(3,50),RX(50),RY(50)
REAL K,L,KG
IRF=IRF+1
IF(NL.EQ.0) GO TO 500
PI=4.*ATAN(1.)
ZN=1.
5 M3=3
10 CALL ABCD2 (O,K,L,G,AX,BX,CX,DX,NL)
KG(1)=DX
R2(2)=1.
R2(3)=AX
U=1./5X
DO 20 I=1,NL
PX=0
CALL ABCD2 (PX,K(1),L(1),G(1),AP(1),BP(1),CP(1),DP(1))
20 CALL ABCD2 (PX,K(1),L(1),G(1),A(1),B(1),C(1),D(1),1)
IF (NL.LT.2) GO TO 30
CALL SERV1 (A,B,C,D,AP,BP,CP,DP,APP,BPP,CPP,DPP,NL)
GO TO 40
25 ZN=ZN+1.
IF(ZN.GT.4.)GO TO 510
GO TO 5
30 APP=AP(1)
BPP=BP(1)
CPP=CP(1)
DPP=DP(1)

```



```

C1=RAP(I)/BX/DT
C2=RB(I)*3PP/BX/BX/DT
ZR2(I)=-C1+C2
50 ZR1(I)=-ZR2(I)+RB(I)/BX
   ROOTS OF 8(P)=0.
   NMAX=40
   TESTMX=40.
   PX=0.001
   CPG=G.GG1/DT
   CLX=0.0001
   N=0
60 DL=DPO
   CALL ABCD2 (PX,K,L,G,AX,BX,CX,DX,NL)
70 PXP=PX+DL
   CALL ABCD2 (PXP,K,L,G,AXP,BXP,CXP,DXP,NL)
   IF (3X*BXP) 90,110,80
30 PX=PXP
   BX=EXP
   TESTX=PX*DT
   IF (TESTX-TESTMX) 70,170,170
90 IF (CL-CLX) 140,140,100
100 CL=DL/2.
   GO TO 70
C
110 IF (BX) 130,120,130
120 RXX=PX
   GO TO 150
C
130 RXX=PXP
   GO TO 150
C
140 AB=ABS(BX/BXP)
   RXX=(PX+AB*PXP)/(1.+AB)
150 N=N+1
   ROOT(N)=RXX
   IF (N.GT.1)DPO=(ROOT(N)-ROOT(N-1))/ZN
   NRT=N
   PX=RXX+CLX
   TESTX=PXX*DT
   IF (TESTX-TESTMX) 160,160,170
160 IF (N.LT.NMAX) GO TO 60
   WRITE(6,/) PXP,NRT,BX,BXP,PX
170 CONTINUE
   IF (ROOT(NRT)-100.) 190,180,180
180 NRT=NRT-1
190 DG 250 JJ=1,NRT
   PX=RJOT(JJ)
   DG 200 J=1,NL
   CALL ABCD2 (PX,K(J),L(J),G(J),A(J),B(J),C(J),D(J),1)
200 CALL ABCD2 (PX,K(J),L(J),G(J),AP(J),BP(J),CP(J),DP(J))
   CALL ABCD2 (PX,K,L,G,AX,BX,CX,DX,NL)
   IF (NL.LT.2) GO TO 210
   CALL DERVT (A,B,C,D,AP,BP,CP,DP,APP,BPP,CPP,DPP,NL)
   GO TO 220
210 APP=AP(1)
   BPP=BP(1)
   CPP=CP(1)
   DPP=DP(1)
220 PY=BPPP*PX*PX*DT
   RA(1,JJ)=DX/PY

```



```

00003460
00003470
00003480
00003490
00003500
00003510
00003520
00003530
00003540
00003550
00003560
00003570
00003580
00003590
00003600
00003610
00003620
00003630
00003640
00003650
00003660
00003670
00003680
00003690
00003700
00003710
00003720
00003730
00003740
00003750
00003760
00003770
00003780
00003790
00003800
00003810
00003820
00003830
00003840
00003850
00003860
00003870
00003880
00003890
00003900
00003910
00003920
00003930
00003940
00003950
00003960
00003970
00003980
00003990
00004000
00004010
00004020
00004030
00004040
00004050
00004060

```

```

IF (PZ-20.) 240,240,230
230 RX(JJ)=0.
    RY(JJ)=25.E16
    GO TO 250
240 RX(JJ)=EXP(-PZ)
    RY(JJ)=(1.-EXP(PZ))*2
250 CONTINUE
    DO 260 JJ=1,NRT
    DO 260 M=1,M3
    ZR1(M)=RA(M,JJ)*RX(JJ)+ZR1(M)
260 ZR2(M)=RA(M,JJ)*(RX(JJ)*RX(JJ)-2.*RX(JJ))+ZR2(M)
    II=1
    III=2
    IF (ZR1(2).LT.0) ZR1(2)=0.
    DO 270 M=1,M3
    ZRK(M,1)=ZR1(M)
270 ZRK(M,2)=ZR2(M)
    NT=48
    DO 310 N=3,NT
    NR=N
    DO 280 M=1,M3
    ZRK(M,N)=0.
    DO 290 M=1,M3
    DO 290 JJ=1,NRT
    IF (RX(JJ).LE.0.) GO TO 288
    APOWER=ALOG10(RX(JJ))
    NDERFL=ABS(APOWER*N)
    IF (NDERFL.LE.37) GO TO 285
288 PZ=0.
    GO TO 290
285 PZ=(RX(JJ))*N
290 ZRK(M,N)=ZRK(M,N)+PZ*RY(JJ)*RA(M,JJ)
    IF (ZRK(1,N-1)*ZRK(1,N-2)) 300,320,300
300 IF (N.LT.5) GO TO 310
    DO 311 NB=1,3
    TEST1=ZRK(NB,N)/ZRK(NB,N-1)
    TEST2=ZRK(NB,N-1)/ZRK(NB,N-2)
    TEST3=ABS(TEST1-TEST2)
    IF (TEST3-0.001) 309,309,310
309 IF (NB.GT.2) GO TO 320
311 CONTINUE
310 CONTINUE
    DO 330 N=1,NR
    X(N)=ZRK(1,N)
    Y(N)=ZRK(2,N)
    IF (Y(N).LT.-1.0E-04) GO TO 25
330 Z(N)=ZRK(3,N)
    IF (X(3)) 350,340,350
340 CR=0.
350 RETURN
    CRX=RX(1)
    TEST=ABS(CRX-TEST2)
    IF (TEST.GT.0.1) GO TO 510
    CN=CRX
    RETURN
510 WRITE(6,520) IRF,CR,CRX
520 FORMAT(' ERRO NA SUBROTINA RESPTK NO IRF =',I5,' CR=',F10.5,'
    *X=',F10.5)
    STOP
    END

```

```

00004750
00004760
00004770
00004780
00004790
00004800
00004810
00004820
00004830
00004840
00004850
00004860
00004870
00004880
00004890
00004900
00004910
00004920
00004930
00004940
00004950
00004960
00004970
00004980
00004990
00005000
00005010
00005020
00005030
00005040
00005050
00005060
00005070
00005080
00005090
00005100
00005110
00005120
00005130
00005140
00005150
00005160
00005170
00005180
00005190
00005200
00005210
00005220
00005230
00005240
00005250
00005260
00005270
00005280
00005290
00005300
00005310
00005320
00005330
00005340

```

40 CONTINUE
X=L*L*0.5/G
AP=X
SP=X*L/K/3
CP=K/L*X*2.
DP=X
GG TO 50
50 RETURN
END
SUBROUTINE DERVT (A,B,C,D,AP,BP,CP,DP,APP,BPP,CPP,DPP,N)
DIMENSION A(N),B(N),C(N),D(N),AP(N),BP(N),CP(N),DP(N),ATT(10),DTT(10)
DO 30 I=1,N
DO 20 J=1,N
IF (I.EQ.J) GO TO 10
AT(J)=A(I)
BT(J)=B(J)
CT(J)=C(J)
DT(J)=D(J)
GO TO 20
10 AT(J)=AP(J)
BT(J)=BP(J)
CT(J)=CP(J)
DT(J)=DP(J)
20 CONTINUE
30 CALL MULT (AT,BT,CT,DT,ATT(I),BT(I),CT(I),DTT(I),N)
APP=ATT(I)
BPP=BT(I)
CPP=CT(I)
DPP=DT(I)
DO 40 I=2,N
APP=APP+ATT(I)
BPP=BPP+BT(I)
CPP=CPP+CT(I)
DPP=DPP+DT(I)
40 RETURN
END
SUBROUTINE MULT (A,B,C,D,AT,BT,CT,DT,N)

DIMENSION A(N),B(N),C(N),D(N)
ATT=A(1)
BT=B(1)
CT=C(1)
DT=D(1)
IF (N.LT.2) GO TO 20
DO 10 J=2,N
AT=ATT+A(J)+BT*C(J)
BT=ATT+B(J)+BT*D(J)
CT=CT+A(J)+DT*C(J)
DT=CT+B(J)+DT*D(J)
ATT=AT

```

C      GO TO 50
C      40 CONTINUE
C      X=L*L*0.5/G
C      AP=X
C      SP=X*L/K/3
C      CP=K/L*X*2.
C      DP=X
C      GO TO 50
C      50 RETURN
C      END
C      SUBROUTINE DERVT (A,B,C,D,AP,BP,CP,DP,APP,BPP,CPP,DPP,N)
C
C      DIMENSION A(N),B(N),C(N),D(N),AP(N),BP(N),CP(N),DP(N),AT(10),
C      2 BT(10),CT(10),DT(10),ATT(10),BIT(10),CIT(10),DIT(10)
C      DO 30 I=1,N
C      DO 20 J=1,N
C      IF (I.EQ.J) GO TO 10
C      AT(J)=A(I)
C      BT(J)=B(J)
C      CT(J)=C(J)
C      DT(J)=D(J)
C      GO TO 20
C      10 AT(J)=AP(J)
C      BT(J)=BP(J)
C      CT(J)=CP(J)
C      DT(J)=DP(J)
C      20 CONTINUE
C      30 CALL MULT (AT,BT,CT,DT,ATT(I),BIT(I),CIT(I),DIT(I),N)
C      APP=ATT(I)
C      BPP=BIT(I)
C      CPP=CIT(I)
C      DPP=DIT(I)
C      GO 40 I=2,N
C      APP=APP+ATT(I)
C      BPP=BPP+BIT(I)
C      CPP=CPP+CIT(I)
C      DPP=DPP+DIT(I)
C      40 RETURN
C      END
C      SUBROUTINE MULT (A,B,C,D,AT,BT,CT,DT,N)
C      *****
C      DIMENSION A(N),B(N),C(N),D(N)
C      ATT=A(1)
C      BIT=B(1)
C      CIT=C(1)
C      DIT=D(1)
C      IF (N.LT.2) GO TO 20
C      DO 10 J=2,N
C      AT=ATT*A(J)+BT*B(J)+
C      BT=ATT*B(J)+BIT*B(J)+
C      CT=CIT*A(J)+DIT*C(J)+
C      DT=CTT*B(J)+DIT*D(J)
C      ATT=

```


C 20 AT=ATT
BI=3TT
CT=CTT
DT=DTT
30 RETURN
C
END

00005400
00005410
00005420
00005430
00005440
00005450
00005460
00005470