

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE MATEMÁTICA  
CURSO DE PÓS-GRADUAÇÃO EM MATEMÁTICA APLICADA

**UM ALGORITMO PARA O  
CÁLCULO DOS  
AUTOVALORES DA  
MATRIZ  $LTS_N$**

por

**Vânia Bolzan Denardi**

Dissertação submetida como requisito parcial  
para a obtenção do grau de  
Mestre em Matemática Aplicada

Prof. Dr. Rudnei Dias da Cunha  
Orientador

Porto Alegre, Junho de 1997.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Denardi, Vânia Bolzan

UM ALGORITMO PARA O CÁLCULO DOS AUTOVALORES DA MATRIZ  $LTS_N$  / Vânia Bolzan Denardi. – Porto Alegre: CPGMAP da UFRGS, 1997.

57 p.: il.

Dissertação (Mestrado)—Universidade Federal do Rio Grande do Sul. Curso de Pós-Graduação em Matemática Aplicada, Porto Alegre, 1997. da Cunha, Rudnei Dias, Orient.

I. Título.

16560

DISSERTACAO/MAT  
D391U  
1997

MAT  
1997/211008-4  
1997/09/22  
8027

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Profa. Wrana Panizzi

Pró-Reitor de Pesquisa e Pós-Graduação: Prof. José Carlos Ferraz Hennemann

Diretor do Instituto de Matemática: Prof. Aron Teiteibaum

Coordenador do CPGMAP: Profa. Maria Cristina Varriale

Bibliotecária-Chefe do Instituto de Matemática: Carlos Brandão Schwab

*Aos meus pais, meus  
irmãos e ao Gilvani.*

## AGRADECIMENTOS

Agradeço a todas as pessoas, que de uma maneira ou de outra, me ajudaram a concluir esse trabalho.

Em especial ao meu orientador, Prof. Dr. Rudnei Dias da Cunha, pela orientação paciente e incentivante, bem como o carinho e amizade compartilhados.

À Profa. Dra. Liliane Basso Barichello pelo auxílio e amizade.

À Profa. Dra. Maria Paula G. Fachin pelo auxílio prestado.

A todos os colegas da pós-graduação da matemática.

Aos professores do curso de Pós-Graduação em Matemática Aplicada da UFRGS, pelos conhecimentos transmitidos.

A Ana, secretária do CPGMAP, pela amizade e dedicação.

Aos meus pais, que com todo o amor, carinho e compreensão, me incentivaram e conduziram pela estrada da vida mostrando que sempre vale a pena lutar pelos nossos ideais.

Ao Gilvani, pelos momentos de compreensão, carinho, amor e paciência demonstrados ao longo de todos esses anos de convivência.

À minha tia Nellyr. Por mais que eu diga, sempre será insuficiente para mostrar o quanto sou grata.

A todos os familiares e amigos que sempre torceram pelo sucesso na realização deste trabalho.

A Deus, por tudo ...

# SUMÁRIO

LISTA DE FIGURAS . . . . .	iv
LISTA DE TABELAS . . . . .	vi
LISTA DE SÍMBOLOS . . . . .	viii
RESUMO . . . . .	x
ABSTRACT . . . . .	xi
1 INTRODUÇÃO . . . . .	1
2 MÉTODO $LTS_N$ . . . . .	3
2.1 Descrição do Método . . . . .	3
2.2 Inversão da matriz $\bar{A}_N(s)$ . . . . .	6
2.3 A Formulação $LTS_N$ . . . . .	7
3 REVISÃO DOS MÉTODOS BÁSICOS UTILIZADOS . . . . .	10
3.1 Método QR . . . . .	10
3.2 Método da bissecção para extração de zeros de funções . . . . .	12
3.2.1 Análise do Erro . . . . .	13
3.3 Método especial para matrizes tridiagonais simétricas . . . . .	15
4 CÁLCULO DOS AUTOVALORES DA MATRIZ $LTS_N$ . . . . .	18
4.1 Introdução . . . . .	18

4.2	Descrição do algoritmo para o cálculo dos autovalores . . . .	19
4.2.1	Normalização de $-A$ . . . . .	21
4.2.2	Cálculo dos coeficientes do polinômio característico da matriz $-A$ . . . . .	22
4.2.3	Cálculo dos autovalores do bloco $B$ . . . . .	27
4.2.4	Extração dos autovalores negativos da matriz $-A$ . . . . .	28
4.3	Análise do Custo Computacional . . . . .	29
4.4	Resultados Numéricos . . . . .	30
5	CONCLUSÃO . . . . .	38
	BIBLIOGRAFIA . . . . .	39
	ANEXO A-1 TABELAS DE AUTOVALORES . . . . .	41

## LISTA DE FIGURAS

Figura 4.1	$N = 4, L = 1$ , diâmetro máximo = 1.82823669129, diâmetro mínimo=0.67217010644191, maior número de iterações=54, menor número de iterações=52 . . . . .	32
Figura 4.2	$N = 4, L = 2$ , diâmetro máximo=1.979078335231, diâmetro mínimo=0.55575023120591, maior número de iterações=51, menor número de iterações=47 . . . . .	32
Figura 4.3	$N = 8, L = 1$ , diâmetro máximo=3.168641085296, diâmetro mínimo=0.560256072211, maior número de iterações=55, menor número de iterações=53 . . . . .	33
Figura 4.4	$N = 8, L = 1$ , diâmetro máximo=3.168641085296, diâmetro mínimo=0.560256072211, maior número de iterações=55, menor número de iterações=53 . . . . .	33
Figura 4.5	$N = 8, L = 2$ , diâmetro máximo=3.150995322157, diâmetro mínimo=0.49761814680651, maior número de iterações=55, menor número de iterações=52 . . . . .	34
Figura 4.6	$N = 8, L = 4$ , diâmetro máximo=3.137013080179, diâmetro mínimo=0.36951667091265, maior número de iterações=55, menor número de iterações=52 . . . . .	34
Figura 4.7	$N = 8, L = 7$ , diâmetro máximo=3.128890016497, diâmetro mínimo=0.27829454133904, maior número de iterações=55, menor número de iterações=52 . . . . .	35
Figura 4.8	$N = 16, L = 1$ , diâmetro máximo=6.556566175593, diâmetro mínimo=0.076846758074, maior número de iterações=56, menor número de iterações=49 . . . . .	35

Figura 4.9  $N = 16, L = 1$ , diâmetro máximo=6.556566175593, diâmetro mínimo=0.076846758074, maior número de iterações=56, menor número de iterações=49 . . . . . 36

Figura 4.10  $N = 18, L = 1$ , diâmetro máximo=6.47828957849, diâmetro mínimo=0.05800499089, maior número de iterações=56, menor número de iterações=51 . . . . . 37



## LISTA DE TABELAS

Tabela 4.1 $L = 1, C = 0.99, \sigma_t = 1.0$ . . . . .	22
Tabela A-1.1 $N = 4, L = 1$ . . . . .	42
Tabela A-1.2 $N = 4, L = 3$ . . . . .	42
Tabela A-1.3 $N = 8, L = 1$ . . . . .	42
Tabela A-1.4 $N = 8, L = 4$ . . . . .	42
Tabela A-1.5 $N = 16, L = 1$ . . . . .	43
Tabela A-1.6 $N = 16, L = 15$ . . . . .	43
Tabela A-1.7 $N = 32, L = 16$ . . . . .	44
Tabela A-1.8 $N = 64, L = 1$ . . . . .	45
Tabela A-1.8 $N = 64, L = 1$ . . . . .	46
Tabela A-1.9 $N = 128, L = 127$ . . . . .	47
Tabela A-1.9 $N = 128, L = 127$ . . . . .	48
Tabela A-1.9 $N = 128, L = 127$ . . . . .	49
Tabela A-1.9 $N = 128, L = 127$ . . . . .	50
Tabela A-1.10 $N = 256, L = 128$ . . . . .	51
Tabela A-1.10 $N = 256, L = 128$ . . . . .	52
Tabela A-1.10 $N = 256, L = 128$ . . . . .	53
Tabela A-1.10 $N = 256, L = 128$ . . . . .	54

Tabela A-1.10 $N = 256, L = 128$ . . . . .	55
Tabela A-1.10 $N = 256, L = 128$ . . . . .	56
Tabela A-1.10 $N = 256, L = 128$ . . . . .	57

## LISTA DE SÍMBOLOS

$A_N(s)$	Matriz do sistema gerado pela formulação $LTS_N$
$A_N^{-1}(s)$	Matriz inversa da matriz $A_N(s)$
$\bar{A}_N(s)$	Transformada de Laplace da matriz $A_N(s)$
$a_{ij}$	Elementos da matriz $A$ , onde $\bar{A}_N(s) = sI + A$
$(\bar{A}_N)^{-1}(s)$	Matriz inversa da matriz $\bar{A}_N(s)$
$C(x)$	Transformada inversa de Laplace da matriz $(\bar{A}_N)^{-1}(s)$
$f(m)$	Fluxo angular de partículas incidente na fronteira $x = 0$ , na direção $\mu_m$
$g(m)$	Fluxo angular de partículas incidente na fronteira $x = a$ , na direção $\mu_m$
$\psi(x, \mu)$	Fluxo angular de partículas
$\psi_m(x)$	Fluxo angular de partículas na direção $\mu_m$
$\bar{\psi}(s)$	Transformada de Laplace da função $\psi_m(x)$
$\vec{\bar{\psi}}(s)$	Vetor de componentes da função $\bar{\psi}(s)$
$\vec{S}(s)$	Vetor dos termos de fonte
$I$	Matriz identidade
$\kappa(A)$	Número de condição de $A$ , $\kappa(A) = \ A\ _p \ A\ _p^{-1}$
$x$	Variável espacial na geometria plana
$a(s)$	Determinante da matriz $\bar{A}_N(s)$
$a(k)$	Raízes de $a(s)$
$L$	Grau arbitrario de anisotropia
$N$	Número de pontos da quadratura de Gauss
$P_l(\mu_m)$	Polinômios de Legendre
$P_k$	Matrizes da inversão numerica da transformada de Laplace
$Q(x)$	Termo de fonte

$s$	Parâmetro complexo
$s_k$	Raizes do polinômio determinante de $A_N(s)$
$\bar{S}_m(s)$	Transformada de Laplace da função $S_m(x)$
$\vec{\bar{S}}_m(s)$	Vetor de componentes da função $\bar{S}_m(s)$
$\mathcal{L}^{-1}$	Transformada inversa de Laplace
$\mu$	Direção da partícula espalhada
$\mu'$	Direção da partícula incidente
$\mu_k$	Raizes do polinômio de Legendre de $N$ -ésimo grau
$\sigma_T$	Seção de choque total
$\sigma_{sl}$	Seção de choque diferencial de espalhamento
$\omega_m$	Pesos da quadratura de Gauss
:	Grupo de linhas da matriz
$\ A\ _1$	Norma-1 de uma matriz $A$ , $\ A\ _1 = \max_{i \leq j \leq N} \sum_{i=1}^N  a_{ij} $

## RESUMO

Apresentamos um novo algoritmo, baseado no algoritmo de inversão de matrizes de Leverrier-Fadeev, para extrair os autovalores e os coeficientes do polinômio característico da matriz  $(sI + A)$ , não-simétrica, que surge em conexão com o método  $LTS_N$  - o qual utiliza a transformada de Laplace para a solução da equação de ordenadas discretas  $S_N$ .

O algoritmo baseia-se em propriedades exibidas pela matriz, cuja estrutura e valores dos elementos fazem com que todos os seus autovalores sejam reais e simétricos em relação a zero. Evidências experimentais demonstram que, os autovalores do bloco superior esquerdo da matriz, de dimensão  $N/2$ , entrelaçam os autovalores negativos de  $-A$ .

O algoritmo foi implementado em FORTRAN 77, usando algumas rotinas do BLAS e do LAPACK, e estruturado de forma a explorar a estrutura da matriz, permitindo efetuar os cálculos necessários em um menor tempo e com um menor gasto de memória. No entanto, apesar de ganhos obtidos em comparação com o algoritmo usualmente utilizado, proposto por Barichello, nossos experimentos demonstram a instabilidade numérica do algoritmo de Leverrier-Fadeev.

## ABSTRACT

We present a new algorithm to compute the eigenvalues and the coefficients of the characteristic polynomial of a nonsymmetric matrix of the form  $(sI+A)$ , which arises in connection with the  $LTS_N$  method for the solution of the discrete ordinates equations  $S_N$ .

Our algorithm is a modification of the matrix inversion Leverrier-Fadeev algorithm, exploiting the pattern existent in the matrix  $-A$  and some properties exhibited by its eigenvalues, which have been determined experimentally. More specifically, its eigenvalues all lie on the real axis and are symmetrically distributed around zero. Also,  $-A$  has a block structure and the eigenvalues of the left-hand superior block interleave the negative eigenvalues of the matrix.

The algorithm was designed to exploit these characteristics, computing only the negative eigenvalues of  $-A$  (due to their symmetrical distribution) by means of the well-know bisection method to obtain the zeros of the characteristic polynomial. Since the eigenvalues of the left-hand superior block of  $A$  interleave those of the matrix, it is possible to use intervals made of pairs of those eigenvalues which contain just a single eigenvalue of  $-A$ . Also, the structure of  $-A$  was used to develop optimized sections of code of the algorithm to reduce the number of operations required.

The whole algorithm was implemented in FORTRAN 77, making use of some of the BLAS and LAPACK routines. The results obtained although presenting a better performance than that used currently, due to Barichello, show that the algorithm is susceptible to the ill-conditioning of the matrix.

# 1 INTRODUÇÃO

A descrição matemática do transporte de partículas, como nêutrons, fótons e elétrons, através de um meio, é comumente chamada de Teoria de Transporte. Essa teoria é usada no tratamento de uma ampla variedade de fenômenos físicos, tais como: a descrição da difusão de nêutrons através dos elementos combustíveis de um reator nuclear, o transporte de radiação e o movimento de moléculas de gás, entre outros.

No entanto, a equação de transporte [8], apesar de poder representar uma ampla variedade de fenômenos, tem um tratamento matemático bastante difícil, uma vez que se trata de uma equação íntegro - diferencial, envolvendo sete variáveis independentes. Assim, na tentativa da obtenção de resultados numéricos precisos, duas linhas de modelagem tem sido implementadas: a probabilística, que busca soluções aproximadas para a forma exata da equação, e a determinística, que procura soluções exatas para formas aproximadas da equação de transporte. Neste último caso podemos citar o Método de Ordenadas Discretas ( $S_N$ ), [8] no qual a variável angular da equação é discretizada.

Segundo a abordagem determinística, recentemente foi proposto o método  $LTS_N$  [15], que determina uma solução analítica para as equações  $S_N$ , e que consiste na aplicação da Transformada de Laplace nessas equações, na variável espacial  $x$ , e inversão analítica da solução transformada do sistema algébrico, resultando numa expressão para o fluxo angular em termos dos valores na fronteira  $x = 0$  do domínio.

Ocorre que a aplicação da Transformada de Laplace gera um sistema simbólico que deve ser resolvido e, para resolvermos esse sistema precisamos calcular a inversa da matriz  $LTS_N$ . Foi determinada em [3] uma expressão analítica para o cálculo dessa inversa que tem se mostrado inviável computacionalmente à medida que  $N$  cresce. Além disso, nessa expressão se faz necessário o cálculo das

raízes do determinante da matriz simbólica, o que também dificulta o tratamento computacional.

No entanto, uma vez que o determinante a ser calculado é de uma matriz do tipo  $(sI + A)$ , que chamaremos matriz “ $LTS_N$ ” , as raízes equivalem aos autovalores da matriz  $-A$ . É importante salientar que a matriz  $LTS_N$  é completa e estruturada, conforme detalharemos nos capítulos seguintes.

Dessa forma, nesse trabalho, desenvolvemos um algoritmo para o cálculo dos autovalores da matriz “ $LTS_N$ ” explorando sua estrutura particular buscando a melhor maneira de se obter os autovalores da matriz  $-A$ .

No capítulo 2, apresentamos sucintamente a formulação  $LTS_N$  para o problema com anisotropia de grau arbitrário  $L$ .

No capítulo 3, é feita uma revisão dos métodos básicos utilizados para o cálculo dos autovalores da matriz  $LTS_N$ .

Finalmente, no capítulo 4, o cálculo dos autovalores é especificado, sendo fornecido detalhes sobre cada um dos passos que compõem este algoritmo. Resultados numéricos são também incluídos para testar o algoritmo desenvolvido.



## 2 MÉTODO $LTS_N$

### 2.1 Descrição do Método

Consideremos o seguinte problema de ordenadas discretas com anisotropia de grau arbitrário  $L$ , em uma placa homogênea,

$$\mu_m \frac{\partial \psi_m(x)}{\partial x} + \sigma_t \psi_m(x) = \frac{1}{2} \sum_{l=0}^L (2l+1) P_l(\mu_m) \sigma_{sl} \sum_{k=1}^N P_l(\mu_k) \psi_k \omega_k + Q(x) \quad (2.1.1)$$

com  $m = 1, \dots, N$ ;  $N$  par,  $0 \leq x \leq a$ ,  $L < N$ , sujeito às seguintes condições de contorno:

$$\psi_m(0) = f_m, \quad \mu > 0 \quad (2.1.1a)$$

e

$$\psi_m(a) = g_m, \quad \mu < 0 \quad (2.1.1b),$$

onde  $f_m$  e  $g_m$  são os fluxos incidentes na fronteira do domínio;  $\psi_m(x) = \psi(x, \mu_m)$  é o fluxo angular das partículas na direção de  $\mu_m$ ,  $\sigma_t$  é a seção de choque total;  $\sigma_{sl}$  ( $l = 0, \dots, L$ ) são as componentes de ordem  $l$  da seção de choque diferencial de espalhamento;  $Q(x)$  é o termo de fonte;  $P_l(\mu)$  são os Polinômios de Legendre;  $\mu_m$

são as raízes do polinômio de Legendre de  $N$ -ésimo grau e  $\omega_m$  são os respectivos pesos da quadratura de Gauss. Essas raízes são ordenadas da seguinte forma:

$$-1 < \mu_N < \dots < \mu_{\frac{N}{2}+1} < 0 < \mu_{\frac{N}{2}} < \dots < \mu_1 < 1. \quad (2.1.2)$$

Podemos reescrever a equação(2.1.1) como

$$\frac{\partial \psi_m(x)}{\partial x} + \frac{\sigma_t}{\mu_m} \psi_m(x) = \sum_{l=0}^L \frac{2l+1}{2} \frac{P_l(\mu_m)}{\mu_m} \sigma_{sl} \sum_{k=1}^N P_l(\mu_k) \psi_k \omega_k + S_m(x) \quad (2.1.3)$$

onde

$$S_m(x) = \frac{Q(x)}{\mu_m}. \quad (2.1.4)$$

Para a resolução do problema de ordenadas discretas (2.1.3), usamos o método  $LTS_N$ , que consiste em aplicar a Transformada de Laplace na variável espacial  $x$ , obtendo a equação transformada

$$\begin{aligned} & \left( s + \frac{\sigma_t}{\mu_m} - \sum_{l=0}^L \frac{2l+1}{2} \frac{P_l^2(\mu_m)}{\mu_m} \sigma_{sl} \omega_m \right) \bar{\psi}_m(s) \\ & - \sum_{l=0}^L \frac{2l+1}{2} \frac{P_l(\mu_m)}{\mu_m} \sigma_{sl} \sum_{\substack{k=1 \\ k \neq m}}^N P_l(\mu_k) \bar{\psi}_k \omega_k = \psi_m(0) + \bar{S}_m(s) \end{aligned} \quad (2.1.5)$$

A equação (2.1.5) representa um sistema de  $N$  equações lineares algébricas e  $N$  incógnitas que podem ser escritas na forma matricial como

$$\overline{A}_N(s)\overline{\psi}(s) = \psi(0) + \overline{S}(s) \quad (2.1.6)$$

onde a matriz  $\overline{A}_N(s)$  é de ordem  $N$ , dada por:

$$\overline{A}_N(s) = sI + A \quad (2.1.7)$$

e a matriz  $A$  é tal que seus elementos são descritos por:

$$a_{ij} = \begin{cases} \frac{\sigma_i}{\mu_i} - \sum_{l=0}^L \frac{2l+1}{2} \frac{P_l^2(\mu_i)}{\mu_i} \sigma_{sl} \omega_i, & \text{se } i = j \\ - \sum_{l=0}^L \frac{2l+1}{2} P_l(\mu_i) P_l(\mu_j) \frac{\omega_j}{\mu_i} \sigma_{sl}, & \text{se } i \neq j. \end{cases} \quad (2.1.8)$$

Usando ainda a seguinte notação,

$$\overline{\psi}(s) = \left( \overline{\psi}_1(s), \overline{\psi}_2(s), \dots, \overline{\psi}_N(s) \right)^T, \quad (2.1.9)$$

$$\psi(0) = \left( \psi_1(0), \psi_2(0), \dots, \psi_N(0) \right)^T \quad (2.1.10)$$

e

$$\overline{S}(s) = \left( \frac{\overline{Q}(s)}{\mu_1}, \frac{\overline{Q}(s)}{\mu_2}, \dots, \frac{\overline{Q}(s)}{\mu_N} \right)^T \quad (2.1.11)$$

e se conhecemos a inversa da matriz  $\overline{A}_N(s)$ , a solução do sistema (2.1.6) é dada por:

$$\bar{\psi}(s) = (\bar{A}_N)^{-1}(s)\psi(0) + (\bar{A}_N)^{-1}(s)\bar{S}(s). \quad (2.1.12)$$

Observamos que a principal dificuldade que se origina na aplicação dessa formulação é que para a solução do sistema (2.1.6) devemos determinar a matriz inversa da matriz  $\bar{A}_N(s)$ , que é dependente do parâmetro complexo  $s$ .

## 2.2 Inversão da matriz $\bar{A}_N(s)$

Foi determinado em [13], para grau arbitrário de anisotropia, uma expressão analítica para o cálculo da inversa da matriz  $\bar{A}_N(s)$ , dada por:

$$(\bar{A}_N)^{-1}(s) = \frac{s^{N-1}I + s^{N-2}B_1 + \cdots + sB_{N-2} + B_{N-1}}{a(s)} \quad (2.2.13)$$

onde

$$\begin{aligned} a(s) &= \det(sI + A) \\ &= s^N + a_1s^{N-1} + \cdots + a_{N-1}s + a_N \end{aligned} \quad (2.2.14)$$

é o polinômio característico da matriz  $-A$  e  $I$  denota a matriz unitária de ordem  $N$ .

Através do algoritmo de Leverrier-Fadeev [16], o qual esta sendo utilizado pela primeira vez para o método  $LTS_N$ , podemos determinar simultaneamente os coeficientes  $a_k$  do  $\det(sI + A)$  e as matrizes  $B_k$  através das seguintes equações:

$$a_1 = -\text{tr}A \quad (2.2.15)$$

$$B_1 = A + a_1 I \quad (2.2.16)$$

$$a_i = -\frac{\text{tr}(AB_{i-1})}{i}, \quad i = 2, \dots, n \quad (2.2.17)$$

$$B_i = AB_{i-1} + a_i I, \quad i = 2, \dots, n-1 \quad (2.2.18)$$

onde  $\text{tr}(A)$  denota o traço de  $A$ . Ficando assim perfeitamente determinada uma expressão analítica para a matriz  $(\bar{A}_N)^{-1}(s)$ .

## 2.3 A Formulação $LTS_N$

Como foi observado na seção 2.1, a principal dificuldade surgida na aplicação  $LTS_N$  é a solução do sistema (2.1.6) uma vez que os elementos da matriz são dependentes do parâmetro complexo  $s$ . Assim, uma vez obtida a expressão (2.2.13) para a matriz inversa  $(\bar{A}_N)^{-1}(s)$ , a solução de (2.1.6) pode ser escrita como:

$$\bar{\psi}(s) = \frac{s^{N-1}I + s^{N-2}B_1 + \dots + sB_{N-1}}{\det \bar{A}_N(s)} (\psi(0) + \bar{S}(s)) \quad (2.3.19)$$

onde  $\psi(0)$  e  $\bar{S}(s)$  são dados respectivamente em (2.1.10) e (2.1.11), as matrizes  $B_i$  ( $i = 1, \dots, N-1$ ) e o  $\det \bar{A}_N(s)$  estão definidos na seção 2.2. Agora, sendo  $\bar{\psi}(s)$

escrito em termos de quocientes de polinômio, a inversão da transformada pode ser feita analiticamente usando-se a técnica da expansão de Heaviside, obtendo-se:

$$\psi(x) = \mathcal{L}^{-1}\{(\overline{A}_N)^{-1}(s)\psi(0)\} + \mathcal{L}^{-1}\{(\overline{A}_N)^{-1}(s)S(s)\} \quad (2.3.20)$$

Agora sendo  $C(x) = \mathcal{L}^{-1}\{(\overline{A}_N)^{-1}(s)\}$  e  $S(x) = \mathcal{L}^{-1}\{S(s)\}$ , então:

$$\psi(x) = C(x)\psi(0) + C(x) * S(x) \quad (2.3.21)$$

onde

$$C(x) * S(x) = \int_{\tau=0}^x C(x-\tau)S(\tau)d\tau \quad (2.3.22)$$

e

$$C(x) = \sum_{k=1}^N B_k e^{s_k x} \quad (2.3.23)$$

com  $s_k$  sendo as  $N$  raízes do determinante da matriz  $\overline{A}_N(s)$ , também autovalores de  $-A$ , e  $B_k$  sendo  $N$  matrizes coeficientes, provenientes da inversão da transformada de Laplace.

É importante observar que nenhuma aproximação numérica foi feita para a solução da aproximação  $S_N$ .

No entanto, a solução (2.3.21) ainda não está completamente definida, pois conhecemos somente as primeiras  $N/2$  componentes do vetor  $\psi(0)$ . Entretanto,

aplicando-se as condições de contorno em  $x = a$  dados por (2.1.1b), a solução do sistema linear de ordem  $(N/2 \times N/2)$  fornecerá os valores das incógnitas em  $x = 0$ , ou seja, os valores  $\psi_m(0)$ ,  $m = N/2 + 1, \dots, N$ . Dessa forma a solução do problema (2.1.1) fica perfeitamente determinada.

### 3 REVISÃO DOS MÉTODOS BÁSICOS UTILIZADOS

O algoritmo apresentado neste trabalho utiliza, em conjunto com o algoritmo de Leverrier-Fadeev (Seção 2.2), o método  $QR$ , para extração dos autovalores do bloco superior esquerdo da matriz  $-A$ , e o método da bissecção para o cálculo de raízes de funções, os quais são brevemente recapitulados a seguir.

#### 3.1 Método QR

O método  $QR$  é um método de formulação simples que calcula todos os autovalores de uma matriz  $A$  e, se desejado, todos os autovetores. Para isso, fazendo  $A = A_0$ , fatoramos essa matriz em:

$$A_0 = Q_0 R_0$$

multiplicando os fatores em ordem inversa, obtemos:

$$A_1 = R_0 Q_0$$

onde, essa nova matriz é similar à matriz  $A_0$  pois,

$$Q_0^{-1} A_0 Q_0 = Q_0^{-1} (Q_0 R_0) Q_0 = A_1$$

Logo, esse processo não altera os autovalores da matriz original.



Dessa forma, o método  $QR$  é definido pelas seguintes relações:

$$A_k = Q_k R_k \quad (3.1.1)$$

e

$$A_{k+1} = R_k Q_k \quad (3.1.2)$$

onde  $A_k$  aproxima-se a uma forma triangular e suas entradas diagonais aproximam seus autovalores, que são também os autovalores da matriz original  $A_0$ .

Na prática, não é vantajoso aplicarmos o método  $QR$  diretamente em uma matriz cheia, pois cada iteração envolveria  $O(n^3)$  operações.

Então, antes de aplicarmos o método  $QR$  reduzimos a matriz à forma de Hessenberg, o que reduz para  $O(n^2)$  operações em cada iteração.

Considerando que o número de iterações para encontrar cada autovalor seja pequeno ( $\ll n$ ), o número total de operações para calcular os  $n$  autovalores é  $O(n^3)$ .

Porém, para acelerar a convergência do método  $QR$  utiliza-se translações da matriz original. Assim, as equações (3.1.1) e (3.1.2) devem ser trocadas, respectivamente, por:

$$A_k - \alpha_k I = Q_k R_k \quad (3.1.3)$$

e

$$A_{k+1} = R_k Q_k + \alpha_k I \quad (3.1.4)$$

onde  $A_k$  e  $A_{k+1}$  são semelhantes, pois:

$$Q_k^{-1} A_k Q_k = Q_k^{-1} (Q_k R_k + \alpha_k I) Q_k = A_K$$

Usualmente, após 3 ou 4 iterações, a matriz  $A_k$ , para  $n = 4$ , assume a forma:

$$A_k = \left( \begin{array}{cccc|c} * & * & * & & * \\ * & * & * & & * \\ & * & * & & * \\ - & - & - & + & - \\ & & \epsilon & & \lambda'_1 \end{array} \right) = \left( \begin{array}{ccc|c} B & & & C \\ - & + & - & \\ 0 & & & D \end{array} \right) \quad (3.1.5)$$

Quando  $\epsilon$  for considerado pequeno o suficiente,  $\lambda'_1$  é assumido como sendo uma boa aproximação para o verdadeiro autovalor  $\lambda_1$  de  $A_0$ .

Para achar o próximo autovalor, o algoritmo continua com a matriz  $B$  da equação (3.1.5) cujos elementos diagonais devem ser reduzidos pelo primeiro passo  $QR$ , e outros dois passos são suficientes para achar  $\lambda_2$ . Esse é um procedimento sistemático para achar todos os autovalores de  $A_0$ . Assim, o método  $QR$  está completamente descrito, pois não é de nosso interesse o cálculo dos autovetores.

## 3.2 Método da bissecção para extração de zeros de funções

Dada uma função  $f(c)$  contínua no intervalo  $[a_0, b_0]$  tal que  $f(a_0)f(b_0) < 0$ , o método da bissecção localiza uma das raízes contidas em  $[a_0, b_0]$  através da bipartição sucessiva desse intervalo.

Dividindo  $[a_0, b_0]$  ao meio, obtém-se o ponto médio  $c_0$ , havendo, pois, dois subintervalos,  $[a_0, c_0]$  e  $[c_0, b_0]$ , a serem considerados.

Se  $f(c_0) = 0$ , então a raiz  $\xi$  é o ponto médio; caso contrário, a raiz estará no subintervalo onde a função tem sinais opostos nos pontos extremos, ou seja, se  $f(a_0)f(c_0) < 0$ , então,  $\xi \in [a_0, c_0] = [a_1, b_1]$ ; senão  $\xi \in [c_0, b_0] = [a_1, b_1]$ .

O novo intervalo  $[a_1, b_1]$  que contém  $\xi$  é dividido ao meio e obtém-se o ponto  $c_1$ . O processo se repete até que se obtenha uma aproximação para a raiz exata  $\xi$ , sujeita a uma tolerância  $\epsilon$  especificada.

O método da bissecção tem a garantia de sempre localizar uma das raízes contidas no intervalo  $[a_0, b_0]$  (desde que  $f(a_0)f(b_0) < 0$  seja satisfeita). Apesar de apresentar uma taxa de convergência menor do que outros métodos, tais como o de Newton e suas variantes, ele exige um menor esforço computacional do que estes, pelo fato de necessitar apenas uma avaliação da função por iteração. No caso particular em que existe apenas uma raiz no intervalo considerado, é o método mais indicado.

### 3.2.1 Análise do Erro

Seja  $[a_0, b_0], [a_1, b_1], \dots, [a_n, b_n]$  a seqüência de intervalos geradas pelo método da bissecção; note que:

$$a_0 \leq a_1 \leq a_2 \leq \dots \leq b_0 \tag{3.2.6}$$

$$b_0 \geq b_1 \geq b_2 \geq \dots \geq a_0 \tag{3.2.7}$$

$$b_{n+1} - a_{n+1} = \frac{1}{2}(b_n - a_n) \tag{3.2.8}$$

a seqüência  $a_n$  é crescente e superiormente limitada, logo ela converge; da mesma forma,  $b_n$  converge. Aplicando a equação (3.2.8) repetidamente, temos:

$$b_n - a_n = 2^{-n}(b_0 - a_0) \quad (3.2.9)$$

e, aplicando limites a ambos os termos desta equação, temos

$$\lim_{n \rightarrow \infty} b_n - \lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} 2^{-n}(b_0 - a_0) = 0 \quad (3.2.10)$$

se chamamos  $r = \lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n$ , então tomando os limites na relação  $0 \geq f(a_n)f(b_n)$ , temos  $0 \geq f^2(r)$ , logo  $f(r) = 0$ .

Suponha que deseja-se parar o processo de bissecção na  $n$ -ésima iteração; então, é sabido que a raiz encontra-se no intervalo  $[a_n, b_n]$ , e a melhor estimativa para a raiz é o ponto médio do intervalo

$$c_n = \frac{a_n + b_n}{2} \quad (3.2.11)$$

então, o erro é limitado por:

$$|r - c_n| \leq \frac{1}{2}|b_n - a_n| \leq 2^{-(n+1)}(b_0 - a_0) \quad (3.2.12)$$

assim, provamos o seguinte teorema:

**Teorema 3.1.** *Se a seqüência  $[a_0, b_0], [a_1, b_1], \dots, [a_n, b_n], \dots$ , denota os intervalos gerados no método da bissecção, então os limites  $\lim_{n \rightarrow \infty} a_n$  e  $\lim_{n \rightarrow \infty} b_n$  existem, são iguais, e representam um zero de  $f$ . Se  $r = \lim_{n \rightarrow \infty} c_n$  e  $c_n = \frac{1}{2}(a_n + b_n)$  então*

$$|r - c_n| \leq 2^{-(n+1)}(b_0 - a_0)$$

Dado um intervalo  $[a_0, b_0]$ , suponha que se deseja avaliar quantas iterações são necessárias para se alcançar uma exatidão relativa de  $\epsilon$ , isto é:

$$\frac{|r - c_n|}{|r|} \leq \epsilon \quad (3.2.13)$$

pela descrição acima,  $r \geq a_0$ ; assim, se satisfizermos  $|r - c_n|/|r| \leq \epsilon$ , alcançaremos aquela exatidão. Pelo resultado do teorema acima, determinamos  $n$  por:

$$2^{-(n+1)} \frac{|b_0 - a_0|}{|a_0|} \leq \epsilon \quad (3.2.14)$$

Para exemplificar, utilizamos em nosso algoritmo o método da bissecção com  $\epsilon = 1.1102230246252 \times 10^{-16}$ , isto é, o épsilon da máquina, em precisão dupla (64 *bits*), padrão *IEEE* – 754; em média, 50 iterações são necessárias para obter-se tal tolerância, por raiz calculada.

Note que o custo computacional do algoritmo da bissecção corresponde ao número de iterações  $n$ , conforme (3.2.14), multiplicado pelo custo de uma avaliação da função a cada iteração.

### 3.3 Método especial para matrizes tridiagonais simétricas

Apesar da matriz  $LTS_N$  não ser simétrica nem tampouco tridiagonal, o algoritmo que desenvolvemos inspira-se no método descrito nessa seção, o qual incluímos para melhor entendimento.

Explorando a rica estrutura matemática das matrizes tridiagonais simétricas foi desenvolvido em ( ver [9, p.437]), um algoritmo alternativo para calcular seus autovalores. Esse algoritmo procede da seguinte maneira:

Se  $T_r$  denota a principal submatriz de

$$T = \begin{pmatrix} a_1 & b_1 & & \dots & 0 \\ b_1 & a_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots & b_{n-1} \\ 0 & \dots & & b_{n-1} & a_n \end{pmatrix} \quad (3.3.15)$$

temos que

$$\begin{aligned} p_r(x) &= \det(T_r - xI) \\ &= (a_r - x)p_{r-1}(x) - b_{r-1}^2 p_{r-2}(x) \end{aligned} \quad (3.3.16)$$

para  $r = 2 : n$  e  $p_0(x) = 1$ .

Como o cálculo de  $p_n(x)$  envolve  $O(n)$  operações, podemos achar suas raízes usando o método da bissecção.

No entanto, algumas vezes é necessário computar o  $k$ -ésimo maior autovalor de  $T$ , para algum valor de  $k$  prescrito. Isso pode ser feito eficientemente usando a idéia de bissecção e o seguinte resultado:

**Teorema 3.2.** *Seqüência de Sturm - Se a matriz tridiagonal(3.3.15) é irredutível, então os autovalores de  $T_{r-1}$  separam estritamente os autovalores de  $T_r$ :*

$$\lambda_r(T_r) < \lambda_{r-1}(T_{r-1}) < \lambda_{r-1}(T_r) < \dots < \lambda_2(T_r) < \lambda_1(T_{r-1}) < \lambda_1(T_r)$$

Portanto, se  $a(\lambda)$  denota o número de mudanças no sinal da seqüência

$$p_0(\lambda), p_1(\lambda), \dots, p_n(\lambda)$$

então  $a(\lambda)$  é o número de autovalores de  $T$  que são menores do que  $\lambda$ , onde os polinômios  $p_r(x)$  são definidos por (3.3.16) e  $p_r(\lambda)$  tem o sinal oposto de  $p_{r-1}(\lambda)$  se  $p_r(\lambda) = 0$ .

Logo, pelas propriedades da seqüência de Sturm, computamos  $\lambda_k(T)$  através do método da bissecção, que produz uma seqüência de subintervalos que sempre contém um autovalor de  $T$ .

Durante a execução do algoritmo da bissecção, informações sobre a localização de outros autovalores são obtidas. Seguindo sistematicamente essas informações é possível computar o subconjunto contínuo de  $\lambda(t)$ , isto é,  $\lambda_k(T), \lambda_{k+1}(T), \dots, \lambda_{k+j}(T)$ .

Conforme será descrito no capítulo a seguir, o algoritmo por nós desenvolvido para calcular os autovalores da matriz  $LTS_N$  foi inspirado nas idéias expostas nesta seção.

## 4 CÁLCULO DOS AUTOVALORES DA MATRIZ $LTS_N$

### 4.1 Introdução

Na formulação do método  $LTS_N$  (seção 2.3) são requisitadas as raízes do determinante da matriz  $(sI + A)$ , que equivalem aos autovalores da matriz  $-A$ .

Com o objetivo de investigar a melhor maneira de se obter os autovalores da matriz  $-A$ , foi desenvolvido um algoritmo que calcula esses autovalores através do cálculo dos autovalores do bloco superior esquerdo, de ordem  $N/2$ , procedendo ao cálculo das raízes negativas do polinômio característico, via bissecção, da matriz  $-A$ , usando os autovalores do bloco como intervalos de estimação das raízes. Porém, convém salientar que esse procedimento foi desenvolvido baseado na estrutura particular da matriz  $-A$  não sendo aplicável para outras matrizes. Esse procedimento será apresentado na seção 4.2.

Ainda na seção 4.3, serão apresentados resultados numéricos obtidos com respeito a determinação das raízes propriamente dita, bem como o custo computacional do cálculo das mesmas.



## 4.2 Descrição do algoritmo para o cálculo dos autovalores

Para desenvolvermos o cálculo dos autovalores da matriz  $-A$ , devemos em primeiro lugar observar a estrutura particular dessa matriz.

A matriz  $-A$  apresenta uma estrutura de blocos cujos elementos estão dispostos da seguinte maneira:

$$-A = \begin{pmatrix} B & | & C \\ - & + & - \\ D & | & E \end{pmatrix} = \begin{pmatrix} a_{11} & \dots & a_{1m} & | & a_{1,m+1} & \dots & a_{1n} \\ a_{21} & \dots & a_{2m} & | & a_{2,m+1} & \dots & a_{2n} \\ \vdots & \ddots & \vdots & | & \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mm} & | & a_{m,m+1} & \dots & a_{mn} \\ - & - & - & + & - & - & - \\ -a_{mn} & \dots & -a_{m,m+1} & | & -a_{mm} & \dots & -a_{m1} \\ \vdots & \ddots & \vdots & | & \vdots & \ddots & \vdots \\ -a_{2n} & \dots & -a_{2,m+1} & | & -a_{2m} & \dots & -a_{21} \\ -a_{1n} & \dots & -a_{1,m+1} & | & -a_{1m} & \dots & -a_{11} \end{pmatrix} \quad (4.2.1)$$

onde  $m = N/2$

Infelizmente, a matriz  $-A$  é não-simétrica e, como é sabido, poucos resultados existem a respeito da distribuição dos autovalores de tais matrizes. Particularmente, pode-se apenas afirmar que muito provavelmente existirão autovalores complexos.

No entanto, experimentalmente (utilizando rotinas do LAPACK [1]) foi possível determinar que, para o problema em questão, os autovalores de  $-A$  são reais e simétricos em relação a zero (o qual não pertence ao conjunto de autovalores

de  $-A$ ), o que confirma os resultados conhecidos em teoria de transporte para esse tipo de problema.

Fatos interessantes também surgem quando determinamos o polinômio característico dessa matriz; ocorre que os coeficientes do polinômio característico relacionados aos termos de ordem ímpar são nulos e o termo de maior grau é 1. Assim, a equação característica da matriz  $-A$  é uma soma de potências de ordem par, o que nos garante que seus autovalores são simétricos em relação ao zero. Além disso, como a ordem da matriz é par, temos um número par de autovalores, o que elimina a possibilidade de termos o “zero” como um autovalor. Sendo assim, basta calcularmos apenas  $N/2$  autovalores.

Por outro lado, a própria estrutura da matriz nos levou a investigarmos uma maneira alternativa de se obter os autovalores de  $-A$ , talvez mais rápido do que, por exemplo, utilizarmos diretamente o método  $QR$  sobre a matriz.

Procedemos então à determinação experimental dos autovalores dos blocos da matriz ( novamente utilizando as rotinas do LAPACK [1]); particularmente, escolhemos os blocos de ordem  $N/2$  ao longo da diagonal. Os resultados de alguns destes experimentos, para diferentes valores de  $N$  e graus de anisotropia  $L$ , encontram-se tabelados no Apêndice 1; com eles nos foi possível observar que os autovalores do bloco superior esquerdo entrelaçam os autovalores negativos de  $-A$ .

Se então tivermos disponíveis os coeficientes do polinômio característico de  $-A$ , podemos desenvolver um algoritmo que extrai as raízes negativas do polinômio utilizando o método da bissecção, uma vez que dois autovalores sucessivos do bloco superior esquerdo de  $-A$  formarão um intervalo onde encontram-se exatamente apenas um dos autovalores negativos de  $-A$ .

O algoritmo *LTSNEIG*, baseado nestas idéias, é exposto a seguir:

**Algoritmo 4.2.1. LTSNEIG**

1. *normaliza a matriz  $-A$ , usando uma norma adequada ( p. ex.,  $\|\cdot\|_\infty$ )*
2. *calcula os coeficientes  $a_k$ , executando o algoritmo POLCAR*
3. *calcula os autovalores do bloco  $B$*
4. *extraí os autovalores de  $-A$ , determinando as raízes negativas do polinômio característico via bissecção*
5. *multiplica os autovalores obtidos em 4 pelo valor da norma utilizada em 1*

Detalharemos cada um dos passos que compõem esse algoritmo.

#### 4.2.1 Normalização de $-A$

Uma breve inspeção de Leverrier-Fadeev, descrito pelas equações (2.2.15)-(2.2.18), nos permite identificar a existência de produtos de matrizes na forma  $AB_{i-1}$ . Essa operação é potencialmente uma fonte de erros de ponto-flutuante, particularmente de “overflow”.

Além disso, se investigarmos o número de condição  $\kappa(-A)$ , veremos que a matriz é extremamente mal - condicionada, conforme mostra a seguinte tabela ( obtida para um problema típico):

Tabela 4.1  $L = 1, C = 0.99, \sigma_t = 1.0$ 

$N$	$\ A\ _1$	$\ A^{-1}\ _1$	$\kappa(A)$
4	3.4716	60.1721	208.8921
8	6.8935	70.9534	489.1173
16	12.7393	75.3185	959.5020
18	14.1345	76.2216	1077.3505
32	23.6312	77.0713	1821.2868
64	44.6946	77.8155	3477.9298
128	86.1258	78.1512	6730.8345

De forma a minimizar tais problemas, normalizamos a matriz  $-A$  calculando uma norma ( p. ex.  $\|A\|_1$ ) de forma apropriada. Essa normalização não altera o problema a ser resolvido, pois valemo-nos de uma propriedade conhecida dos autovalores: se  $B = \alpha A$ , onde  $A$  e  $B$  são matrizes,  $\lambda$  é autovalor de  $A$  e  $\mu$  é autovalor de  $B$ , então  $\mu = \alpha\lambda$ .

#### 4.2.2 Cálculo dos coeficientes do polinômio característico da matriz $-A$

Os coeficientes do polinômio característico são calculados através de uma modificação no algoritmo de Leverrier-Fadeev, descrito na seção (2.2). Esse algoritmo modificado calcula apenas os coeficientes relacionados aos termos de ordem par, pois como mencionamos anteriormente os coeficientes relacionados aos termos de ordem ímpar são nulos. O algoritmo, chamado de POLCAR, é detalhado a seguir.

**Algoritmo 4.2.2. POLCAR**

```
1.  $Z \leftarrow A \times A$ 
2.  $C \leftarrow Z$ 
3.  $a_1 \leftarrow 0$ 
4.  $k \leftarrow 2$ 
   for  $i = 1, 2, \dots, \frac{n}{2} - 1$ 
5.  $a_k \leftarrow -tr(c)/k$ 
6.  $a_{k+1} \leftarrow 0$ 
7.  $Q \leftarrow Z \times C$ 
8.  $C \leftarrow Q + a_k Z$ 
9.  $k \leftarrow k + 2$ 
   endfor
10.  $a_k \leftarrow -tr(c)/k$ 
```

Nesse algoritmo há duas multiplicações matriz-matriz - passos 1 e 7 - os quais são calculadas através dos algoritmos PROD1 e PROD2, descritos a seguir.

**Algoritmo 4.2.3. *PROD1***

```
for  $j = 1, 2, \dots, N/2$ 
1.  $Z_j \leftarrow a_{11}A_1$ 
   for  $k = 2, 3, \dots, N/2$ 
2.  $Z_j \leftarrow a_{kj}A_k + Z_j$ 
   endfor
   for  $k = N/2, N/2 - 1, \dots, 1$ 
   for  $i = 1, 2, \dots, N$ 
3.  $V_{n+1-i} = -a_{ik}$ 
   endfor
4.  $Z_j \leftarrow a_{n+1-k,j}V + Z_j$ 
   endfor
endfor
```

Essa multiplicação é baseada na estrutura da matriz (4.1). Considerando o caso  $N = 4$ , temos:

$$A_1 = \left( \begin{array}{cc|cc} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ - & - & + & - \\ -a_{24} & -a_{23} & -a_{22} & -a_{21} \\ -a_{14} & -a_{13} & -a_{12} & -a_{11} \end{array} \right) \quad (4.2.2)$$

Então:

$$Z = \begin{pmatrix} a_{11} & a_{12} & | & a_{13} & a_{14} \\ a_{21} & a_{22} & | & a_{23} & a_{24} \\ & - & + & - & \\ -a_{24} & -a_{23} & | & -a_{22} & -a_{21} \\ -a_{14} & -a_{13} & | & -a_{12} & -a_{11} \end{pmatrix} \times \begin{pmatrix} a_{11} & a_{12} & | & a_{13} & a_{14} \\ a_{21} & a_{22} & | & a_{23} & a_{24} \\ & - & + & - & \\ -a_{24} & -a_{23} & | & -a_{22} & -a_{21} \\ -a_{14} & -a_{13} & | & -a_{12} & -a_{11} \end{pmatrix} \quad (4.2.3)$$

expandindo o produto acima temos:

$$Z = \begin{pmatrix} a_{11}a_{11} + a_{12}a_{21} + a_{13}(-a_{24}) + a_{14}(-a_{14}) & a_{11}a_{12} + a_{12}a_{22} + a_{13}(-a_{23}) + a_{14}(-a_{13}) \\ a_{21}a_{11} + a_{22}a_{21} + a_{23}(-a_{24}) + a_{24}(-a_{14}) & a_{21}a_{12} + a_{22}a_{22} + a_{23}(-a_{23}) + a_{24}(-a_{13}) \\ (-a_{24})a_{11} + (-a_{23})a_{21} + (-a_{22})(-a_{24}) + (-a_{21})(-a_{14}) & (-a_{24})a_{12} + (-a_{23})a_{22} + (-a_{22})(-a_{23}) + (-a_{21})(-a_{13}) \\ (-a_{14})a_{11} + (-a_{13})a_{21} + (-a_{12})(-a_{24}) + (-a_{11})(-a_{14}) & (-a_{14})a_{12} + (-a_{13})a_{22} + (-a_{12})(-a_{23}) + (-a_{11})(-a_{13}) \\ a_{11}a_{13} + a_{12}a_{23} + a_{13}(-a_{22}) + a_{14}(-a_{12}) & a_{11}a_{14} + a_{12}a_{24} + a_{13}(-a_{21}) + a_{14}(-a_{11}) \\ a_{21}a_{13} + a_{22}a_{23} + a_{23}(-a_{22}) + a_{24}(-a_{12}) & a_{21}a_{14} + a_{22}a_{24} + a_{23}(-a_{21}) + a_{24}(-a_{11}) \\ (-a_{24})a_{13} + (-a_{23})a_{23} + (-a_{22})(-a_{22}) + (-a_{21})(-a_{12}) & (-a_{24})a_{14} + (-a_{23})a_{24} + (-a_{22})(-a_{21}) + (-a_{21})(-a_{11}) \\ (-a_{14})a_{13} + (-a_{13})a_{23} + (-a_{12})(-a_{22}) + (-a_{11})(-a_{12}) & (-a_{14})a_{14} + (-a_{13})a_{24} + (-a_{12})(-a_{21}) + (-a_{11})(-a_{11}) \end{pmatrix} \quad (4.2.4)$$

por inspeção, vemos que as colunas  $N/2 + 1, N/2 + 2, \dots, N$ , são as mesmas que as colunas  $1, 2, \dots, N/2$ , apenas com seus elementos tomados na ordem inversa de linhas e colunas. Assim sendo, podemos calcular apenas metade das colunas da matriz  $Z$ ,

$$Z = \begin{pmatrix} x_{11}(x_{11}) + x_{21}(x_{12}) + x_{31}(-x_{42}) + x_{41}(-x_{41}) & x_{12}(x_{11}) + x_{22}(x_{12}) + x_{32}(-x_{42}) + x_{42}(-x_{41}) \\ x_{11}(x_{21}) + x_{21}(x_{22}) + x_{31}(-x_{32}) + x_{41}(-x_{31}) & x_{12}(x_{21}) + x_{22}(x_{22}) + x_{32}(-x_{32}) + x_{42}(-x_{31}) \\ x_{11}(x_{31}) + x_{21}(x_{32}) + x_{31}(-x_{22}) + x_{41}(-x_{21}) & x_{12}(x_{31}) + x_{22}(x_{32}) + x_{32}(-x_{22}) + x_{42}(-x_{21}) \\ x_{11}(x_{41}) + x_{21}(x_{42}) + x_{31}(-x_{12}) + x_{41}(-x_{11}) & x_{12}(x_{41}) + x_{22}(x_{42}) + x_{32}(-x_{12}) + x_{42}(-x_{11}) \end{pmatrix} \quad (4.2.5)$$

onde os dois primeiros termos de cada elemento são calculados pelas expressões (1) e

(2) do algoritmo PROD1 e os outros dois termos de cada elemento pelas expressões (3) e (4). Por inspeção vemos que os elementos são:

$$z(1 : 4, 1) = x(1, 1)x(1 : 4, 1) + x(2, 1)x(1 : 4, 2) - x(3, 1)x(4 : 1, 2) - x(4, 1)x(4 : 1, 1) \quad (4.2.6)$$

e

$$z(1 : 4, 2) = x(1, 2)x(1 : 4, 1) + x(2, 2)x(1 : 4, 2) - x(3, 2)x(4 : 1, 2) - x(4, 2)x(4 : 1, 1) \quad (4.2.7)$$

onde a notação  $1 : 4$  indica um grupo de linhas da matriz  $z$ .

Assim o algoritmo PROD1 é implementado como uma seqüência de  $\_AXPYs$ , uma rotina das BLAS( Basic Linear Algebra Subroutines)[4], [5], [6] e [7]. O algoritmo PROD2, foi desenvolvido de maneira semelhante, para calcular o produto  $Q = Z \times C$ , onde  $Z = (-A) \times (-A)$  e  $C = -A + a_k I$ .



**Algoritmo 4.2.4. PROD2**

```
for j = 1, 2, ..., N/2
1.  $Q_j \leftarrow c_{11}Z_1$ 
   for k = 2, 3, ..., N/2
2.  $Q_j \leftarrow c_{kj}Z_k + Q_j$ 
   endfor
   for k = N/2, N/2 - 1, ..., 1
     for i = 1, 2, ..., N
3.  $V_{n+1-i} = z_{ik}$ 
     endfor
4.  $Q_j \leftarrow c_{n+1-k,j}V + Q_j$ 
   endfor
endfor
```

#### 4.2.3 Cálculo dos autovalores do bloco $B$

Os autovalores do bloco  $B$ , de ordem  $N/2$ , são calculados de forma clássica, utilizando transformações de Householder para uma redução à forma de Hessenberg superior, seguido da aplicação do método  $QR$  sobre a matriz reduzida. Salientamos que, como o bloco  $B$  é uma matriz completa, não - simétrica e sem uma estrutura particular, não é possível utilizar qualquer método especial.

Para efetuar o cálculo dos autovalores de  $B$ , utilizamos as rotinas DGEHRD e DHSEQR da biblioteca LAPACK [1].

#### 4.2.4 Extração dos autovalores negativos da matriz $-A$

Para extrair as raízes negativas da matriz  $LTS_N$ , usamos o método da bissecção. Os autovalores  $\gamma_i$  do bloco  $B$ , calculados no passo anterior, são utilizados como intervalos para a determinação das raízes negativas do polinômio característico de  $-A$ , como segue:

$$\lambda_i \in J_i = [\gamma_i, \gamma_{i+1}], \quad 1 \leq i \leq N/2 - 1 \quad (4.2.8)$$

$$\lambda_i \in J_i = [\gamma_i, 0], \quad i = N/2 \quad (4.2.9)$$

note  $\gamma_1 < \gamma_2 < \dots < \gamma_{N/2-1} < \gamma_{N/2} < 0$ .

Visando reduzir o número de operações de ponto flutuante, calculamos o polinômio usando a forma de Horner, a qual tendo em vista a existência apenas de potências de ordem par no polinômio, é expressa da seguinte forma:

$$a(s) = (((a_2s^2 + a_4)s^2 + \dots)s^2 + a_{n-2})s^2 + a_n \quad (4.2.10)$$

### 4.3 Análise do Custo Computacional

A análise do custo computacional do algoritmo desenvolvido para calcular as raízes da matriz  $LTS_N$  é feita em cada passo, considerando-se basicamente o número de multiplicações em ponto flutuante.

1. Normalização de  $-A$ :  $\frac{N^2}{2}$  ( devido à estrutura da matriz)

2. Cálculo dos coeficientes do polinômio característico

$$\text{PROD1} : \frac{N^3 - N^2}{2}$$

$$\text{PROD2} : \frac{N^3 - N^2}{2}$$

$$\text{POLCAR} : \frac{N^4 + 2N^2 - 2N}{4}$$

3. Autovalores do bloco B (Redução para Hessenberg+QR):  $10\frac{N^3}{8}$

4. Autovalores negativos ( método da bissecção ) : conforme apresentado no final da seção 3.2, o algoritmo utilizado apresenta um custo da ordem de  $nO_f$ , onde  $n$  é obtido através da equação(3.2.14), e  $O_f$  é o custo da avaliação da função. Como a função é um polinômio de grau  $N$ , (contando apenas com as potências de ordem par) e, conforme seção 4.2.3, ele é avaliado usando a forma de Horner, então  $(N/2-1)$  produtos são efetuados para calcular a expressão e um produto é efetuado para se calcular  $s^2$ , totalizando  $N/2$  multiplicações em ponto flutuante.

Como o número de iterações  $n$  para cada  $i$ -ésimo intervalo de busca  $J_i$  de um zero pode ser diferente, obtemos a seguinte expressão (ver [12]) para o custo computacional dessa seção do algoritmo:

$$\sum_{i=1}^{N/2} \left( 1 - \log_2 \epsilon \frac{|j_{0i}|}{|j_{1i} - j_{0i}|} \right) \frac{N}{2}$$

onde  $j_{0i}$  e  $j_{1i}$  indicam os limites inferior e superior do intervalo  $J_i$ .

5. Multiplicação dos autovalores pelo valor da norma de  $-A$ :  $N/2$

Portanto, o custo total desse algoritmo é de:

$$\frac{N^2}{2} + \frac{N^4 + 2N^2 - 2N}{4} + 10\frac{N^3}{8} + \sum_{i=1}^{N/2} \left( 1 - \log_2 \epsilon \frac{|j_{0i}|}{|j_{1i} - j_{0i}|} \right) \frac{N}{2} + \frac{N}{2} \quad (4.3.11)$$

ao passo que se aplicarmos diretamente o método  $QR$  na matriz  $LTS_N$ , reduzida à forma de Hessenberg, o custo computacional seria de  $10n^3$ . Sendo assim, o algoritmo desenvolvido seria melhor que o método  $QR$  apenas para  $N \leq 33$  ( se considerarmos uma média de 50 iterações para determinar cada raiz).

O custo computacional deste algoritmo é inferior ao usualmente utilizado para o cálculo dos coeficientes do polinômio característico, o qual foi proposto por Barichello. Naquele algoritmo, baseado no cálculo de matrizes adjuntas, o custo é proporcional a  $N^4$ ; dessa forma, nosso algoritmo representa uma redução razoável no custo, obtido principalmente através das rotinas otimizadas para o cálculo dos produtos entre matrizes ( ver secção 4.2.2).

Porém, cabe salientar que a determinação destas raízes está ligada a um problema muito maior, a inversão analítica da matriz  $\bar{A}_N(s)$ , e para essa inversão se faz necessário o cálculo dos coeficientes do polinômio característico.

## 4.4 Resultados Numéricos

O cálculo das raízes da matriz  $\bar{A}_N(s)$ , segundo o procedimento acima descrito, foi implementado na linguagem FORTRAN 77, e testado em estações de trabalho Sun SPARC1 localizados no Instituto de Matemática/UFRGS.

Calculamos os autovalores da matriz  $\bar{A}_N(s)$  para diferentes valores de  $N$  (4, 8, 16, 18, 32, 68, 128, 256). Para ilustrar o problema de instabilidade numérica encontrada, apresentamos a seguir os resultados para  $N = 4, 8, 16$  e 18, com grau de anisotropia arbitrário [11] e utilizando valores das seções de choque obtidos de

Barichello [2]. Após, comparamos esses resultados com os autovalores calculados pelo LAPACK. Incluímos, também, alguns gráficos para testar o algoritmo desenvolvido, nos quais podemos observar que para  $N = 4$  ocorre uma intercalação entre os autovalores do bloco  $B$  e os autovalores da matriz  $\bar{A}_N(s)$  e uma concordância entre os resultados obtidos pelo algoritmo desenvolvido e pelo LAPACK, sendo que essa concordância permanece para os casos  $N = 8$  e  $N = 16$ , também testados. Com isso, se confirma a validade do algoritmo desenvolvido para esses casos.

Nos gráficos seguintes,  $\times$  representa os autovalores do bloco  $B$ ,  $\circ$  os autovalores da matriz  $\bar{A}_N(s)$  calculados pelo algoritmo desenvolvido e, no caso da figura 4.10,  $*$  representa os autovalores calculados pelo LAPACK. O diâmetro máximo é a medida do maior intervalo.

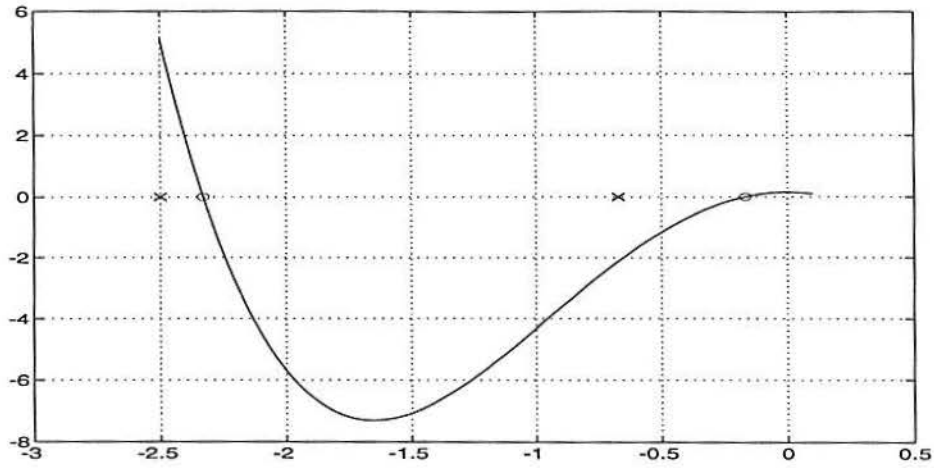


Figura 4.1  $N = 4$ ,  $L = 1$ , diâmetro máximo = 1.82823669129, diâmetro mínimo=0.67217010644191, maior número de iterações=54, menor número de iterações=52

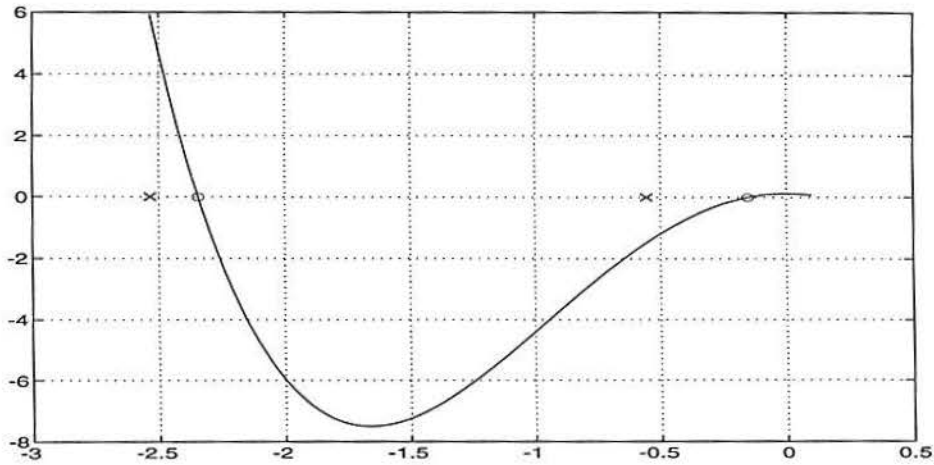


Figura 4.2  $N = 4$ ,  $L = 2$ , diâmetro máximo=1.979078335231, diâmetro mínimo=0.55575023120591, maior número de iterações=51, menor número de iterações=47

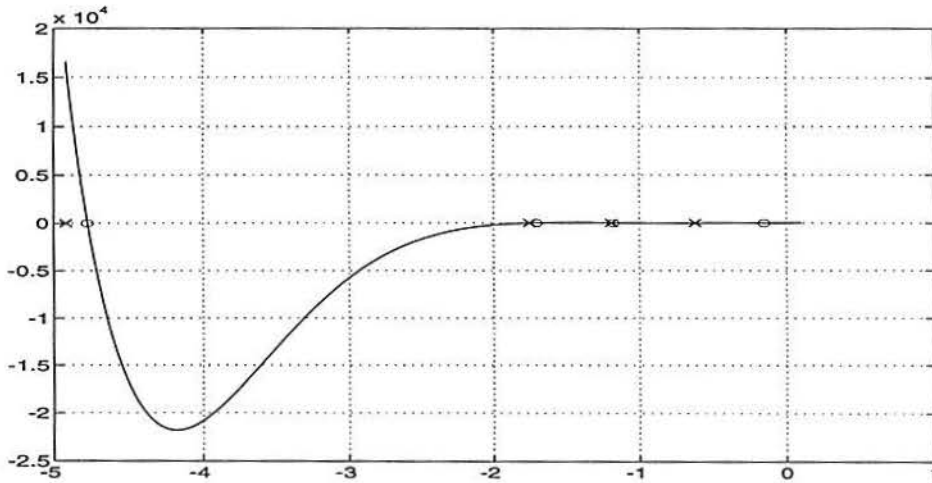


Figura 4.3  $N = 8$ ,  $L = 1$ , diâmetro máximo=3.168641085296, diâmetro mínimo=0.560256072211, maior número de iterações=55, menor número de iterações=53

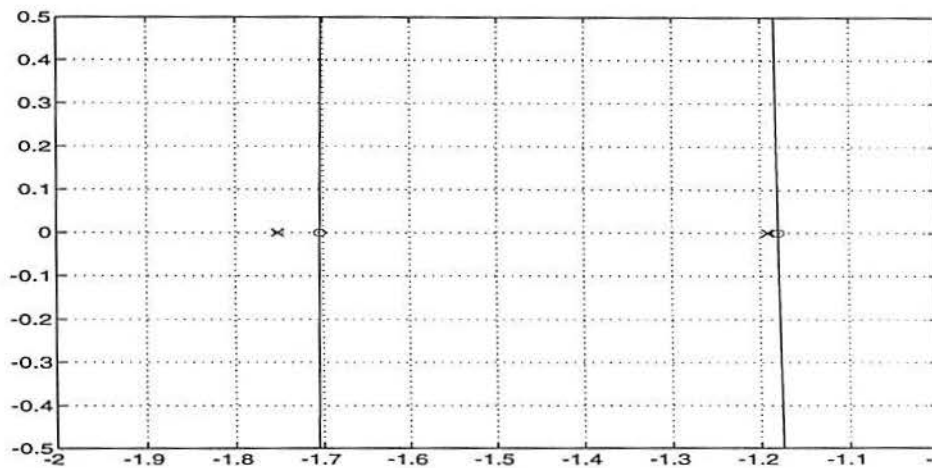


Figura 4.4  $N = 8$ ,  $L = 1$ , diâmetro máximo=3.168641085296, diâmetro mínimo=0.560256072211, maior número de iterações=55, menor número de iterações=53

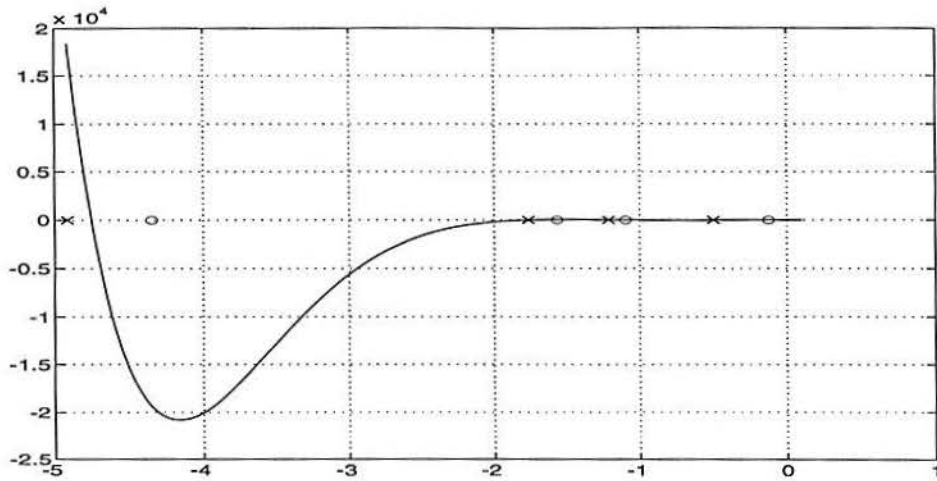


Figura 4.5  $N = 8$ ,  $L = 2$ , diâmetro máximo=3.150995322157, diâmetro mínimo=0.49761814680651, maior número de iterações=55, menor número de iterações=52

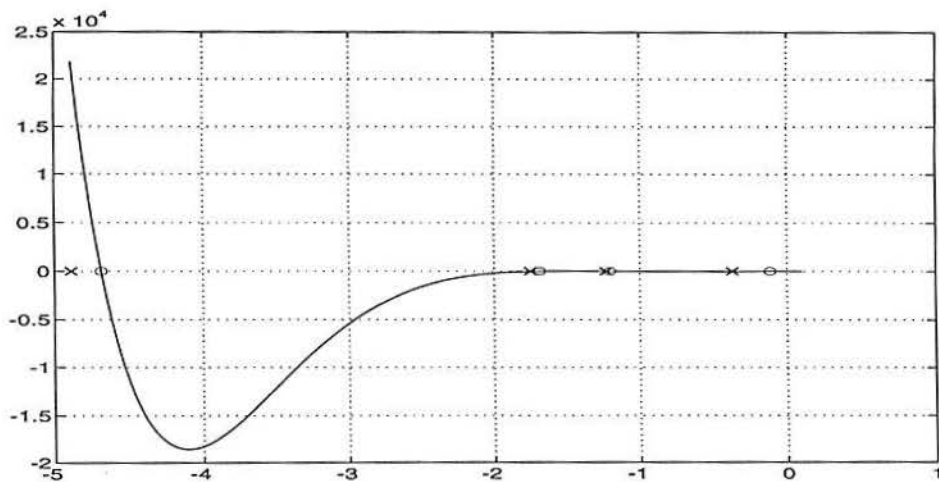


Figura 4.6  $N = 8$ ,  $L = 4$ , diâmetro máximo=3.137013080179, diâmetro mínimo=0.36951667091265, maior número de iterações=55, menor número de iterações=52



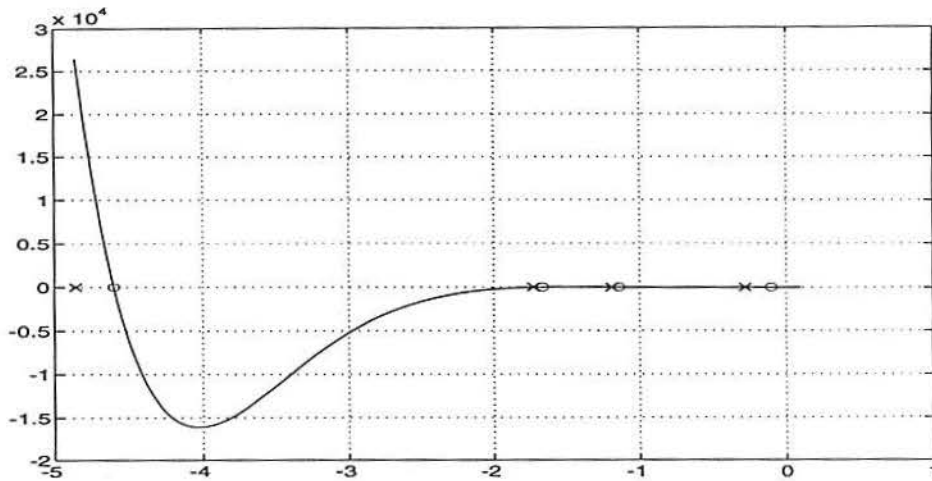


Figura 4.7  $N = 8$ ,  $L = 7$ , diâmetro máximo=3.128890016497, diâmetro mínimo=0.27829454133904, maior número de iterações=55, menor número de iterações=52

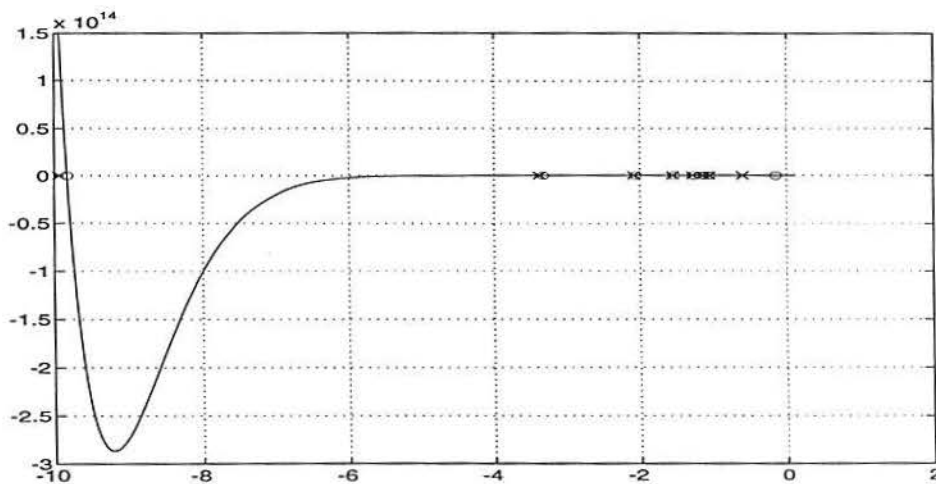


Figura 4.8  $N = 16$ ,  $L = 1$ , diâmetro máximo=6.556566175593, diâmetro mínimo=0.076846758074, maior número de iterações=56, menor número de iterações=49

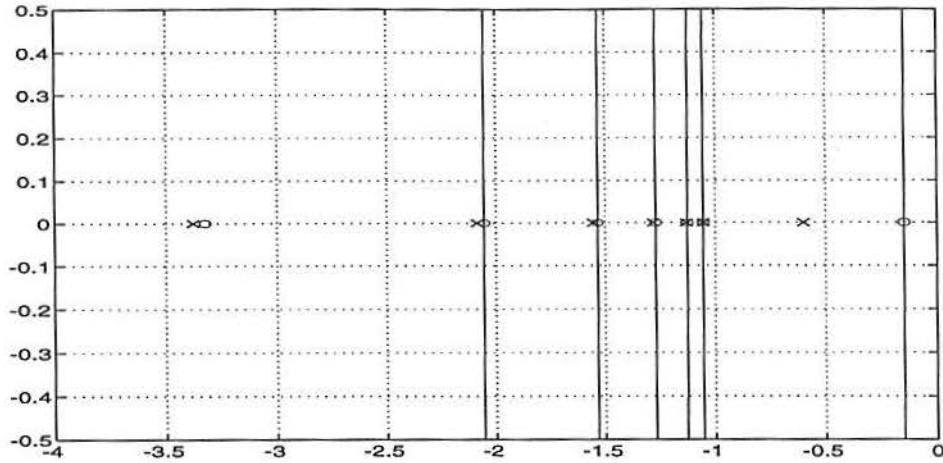


Figura 4.9  $N = 16$ ,  $L = 1$ , diâmetro máximo=6.556566175593, diâmetro mínimo=0.076846758074, maior número de iterações=56, menor número de iterações=49

No entanto, à medida que  $N$  cresce, os coeficientes do polinômio característico crescem muito causando uma oscilação em um dos extremos. Isso pode ser observado na figura 4.10, na qual a intercalação entre os autovalores do bloco  $B$  e os autovalores da matriz  $\bar{A}_N(s)$  não mais acontece, nem tampouco a concordância entre os resultados obtidos pelo algoritmo desenvolvido e pelo LAPACK. Essa diferença acentua-se para valores de  $N > 18$ .

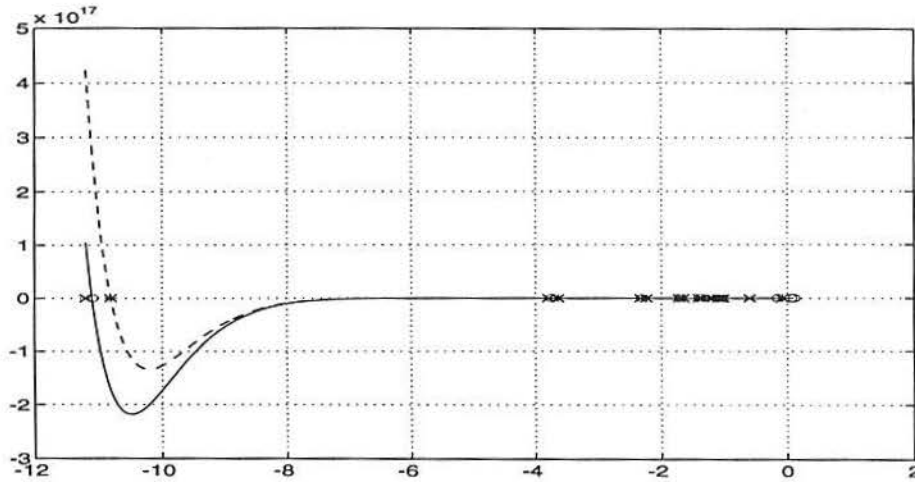


Figura 4.10  $N = 18$ ,  $L = 1$ , diâmetro máximo=6.47828957849, diâmetro mínimo=0.05800499089, maior número de iterações=56, menor número de iterações=51

Ainda na figura 4.10, podemos observar o polinômio característico calculado pelo algoritmo desenvolvido ( representado pelo traço contínuo) e o verdadeiro polinômio característico ( representado pelo traço pontilhado) o qual foi calculado pela seguinte expressão:

$$p(x) = (x - \lambda_1)(x - \lambda_2) \cdots (x - \lambda_N) \quad (4.4.12)$$

onde

$$\lambda_i, \quad i = 1, \dots, N \quad (4.4.13)$$

são os verdadeiros autovalores da matriz  $\bar{A}_N(s)$  calculados pelo LAPACK.

## 5 CONCLUSÃO

Neste trabalho, desenvolvemos um novo algoritmo para o cálculo dos autovalores da matriz que surge em conexão com o método  $LTS_N$ .

Baseando-nos no algoritmo de Leverrier-Fadeev, para obtenção dos coeficientes, foi possível desenvolver um algoritmo que explora as propriedades da matriz, através da quais constatamos a intercalação dos autovalores negativos de  $-A$  pelos autovalores de seu bloco superior esquerdo e nos possibilitando a escrever rotinas com um menor custo computacional. Este algoritmo foi desenvolvido tendo em vista também a necessidade de se obter os coeficientes do polinômio característico de uma matriz da forma  $sI + A$ , pois os mesmos são utilizados para o processo de inversão desta matriz.

O algoritmo de Leverrier-Fadeev foi, aqui, testado pela primeira vez no método  $LTS_N$  e implementado com algumas modificações, considerando a estrutura particular da matriz  $-A$ .

No entanto, devido a instabilidade numérica encontradas no algoritmo de Leverrier-Fadeev, para as matrizes em questão, o algoritmo desenvolvido só deve ser utilizado para matrizes de ordem igual ou inferior a 16, nos quais obtemos melhores resultados que o método  $QR$ .

## BIBLIOGRAFIA

- [1] ANDERSON, E. and BAI, Z. and BISCHOF, C. and DEMMEL, J. and DONGARRA, J. and DU CROZ, J. and GREENBAUM, A. and HAMMARLING, S. and McKENNEY, A. and OSTROUCHOV, S. and SORENSEN, D., LAPACK Users'Guide, Second Edition. SIAM. Philadelphia, (1995).
- [2] BARICHELLO, L. B., Formulação Analítica para Solução do Problema de Ordenada Discreta Unidimensional. Tese de Doutorado pelo Programa de Pós-Graduação em Engenharia Mecânica(PROMECA) - UFRGS, Porto Alegre, (1992).
- [3] BARICHELLO, L. B. e VILHENA, M.T., A General Approach to One - Group One Dimensional Transport Equation. *Kernntechnik*, v.58, n.3, p. 182-184, (1993).
- [4] DONGARRA, J.J. and DU CROZ, J. and HAMMARLING, S. and HANSON, R.J., An extended set of FORTRAN Basic Linear Algebra Subprograms. *ACM Transactions on Mathematical Software*, v.14, n.1, p. 1-17, (1988).
- [5] DONGARRA, J.J. and DU CROZ, J. and HAMMARLING, S. and HANSON, R.J., Algorihm 656: An extended set of Basic Linear Algebra Subprograms: model implementation and test programs.*ACM Transactions on Mathematical Software*, v.14, n.1, p. 18-32, (1988).
- [6] DONGARRA, J.J. and DU CROZ, J. and HAMMARLING, S. and DUFF, I.S., A set of Level 3 Basic Linear Algebra Subprograms. *ACM Transactions on Mathematical Software*, v.16, n.1, p. 1-17, (1990).
- [7] DONGARRA, J.J. and DU CROZ, J. and HAMMARLING, S. and DUFF, I.S., Algorithm 679: A set of Level 3 Basic Linear Algebra Subprograms:

- model implementation and test programs. **ACM Transactions on Mathematical Software**, v.16, n.1, p. 18-28, (1990).
- [8] DUDERSTADT, J. e MARTIN, W., **Transport Theory**. New York, John Wiley & Sons, London, (1979).
- [9] GOLUB, G. H. and VAN LOAN, C. F., **Matrix Computations**. The Johns Hopkins University Press, USA, (1988).
- [10] HAGER, W. W., **Applied Numerical Linear Algebra**. Prentice Hall International, Inc., USA, (1988).
- [11] KAPER, H. G., SHUTLIS, J. K and VENINGA, J. G., **Numerical evolution of the slab albedo problem solution in one-speed anisotropic transport theory**. *Journal of Computational Physics*, v.6, p. 288-313, (1970).
- [12] KINCAID, D. R. and CHERREY, E. W., **Numerical Analysis**. Brooks Cole, Pacific Grove, (1991).
- [13] Oliveira, J.V.P.de, *Formulação  $LTS_N$  para o problema de ordenada discreta com anisotropia..Dissertação Mestrado (CPGMAp)- UFRGS.Porto Alegre,UFRGS,(1993).*
- [14] STRANG, G., **LINEAR ALGEBRA AND ITS APPLICATIONS**. Harcourt Brace Jovanovich, Inc., USA, (1988).
- [15] VILHENA, M. T. and BARICHELLO, L. B., **A New Analytical Approach to solve the neutron transport equation**. *Kerntechnik*, v.56, n.5, p. 334-336, (1991).
- [16] WANG, G. and LIN, Y., **A New Extension of Leverrier's Algorithm**. **LINEAR ALGEBRA AND ITS APPLICATIONS**,pp. 227-238, (1993).

## ANEXO A-1 TABELAS DE AUTOVALORES

Nessa seção, listamos alguns dos inúmeros problemas testados, conforme relatado na seção 4.2 .

As tabelas mostradas a seguir contêm os autovalores  $\gamma_i$  do bloco superior esquerdo  $B$  da matriz  $A_N$  e os autovalores negativos  $\lambda_i, i = 1, 2, \dots, N/2$  dessa matriz. A formatação das tabelas foi feita de forma a evidenciar a intercalação dos autovalores  $\lambda_i$  pelos autovalores  $\gamma_i$ .

Os valores tabulados foram obtidos através do método  $QR$  utilizando as rotinas LAPACK, DGEHRD e DHSEQR, ambas aplicadas sobre o bloco  $B$  e sobre a matriz  $A_N$ .

Tabela A-1.1  $N = 4, L = 1$ 

$i$	$\gamma_i$	$\lambda_i$
1	-0.2124104405D+01	-0.1979939479D+01
2	-0.5710108792D+00	-0.1412078631D+00

Tabela A-1.2  $N = 4, L = 3$ 

$i$	$\gamma_i$	$\lambda_i$
1	-0.1909523536D+01	-0.1749577312D+01
2	-0.3669977826D+00	-0.1098145158D+00

Tabela A-1.3  $N = 8, L = 1$ 

$i$	$\gamma_i$	$\lambda_i$
1	-0.4603553991D+01	-0.4467686039D+01
2	-0.1638977604D+01	-0.1594013883D+01
3	-0.1114802768D+01	-0.1103682290D+01
4	-0.5787197711D+00	-0.1412078482D+00

Tabela A-1.4  $N = 8, L = 4$ 

$i$	$\gamma_i$	$\lambda_i$
1	-0.4281546636D+01	-0.4104456825D+01
2	-0.1533966672D+01	-0.1476864002D+01
3	-0.1077888220D+01	-0.1056005884D+01
4	-0.3236443635D+00	-0.1004342293D+00



Tabela A-1.5  $N = 16, L = 1$ 

$i$	$\gamma_i$	$\lambda_i$
1	-0.9636673094D+01	-0.9536398956D+01
2	-0.3276819008D+01	-0.3225643530D+01
3	-0.2023794716D+01	-0.1993416391D+01
4	-0.1507568103D+01	-0.1489485111D+01
5	-0.1241962695D+01	-0.1232003026D+01
6	-0.1095865800D+01	-0.1091348205D+01
7	-0.1021324621D+01	-0.1020106430D+01
8	-0.5810096055D+00	-0.1412078482D+00

Tabela A-1.6  $N = 16, L = 15$ 

$i$	$\gamma_i$	$\lambda_i$
1	-0.8505822135D+01	-0.8238631739D+01
2	-0.2887104682D+01	-0.2794827340D+01
3	-0.1781506789D+01	-0.1738145461D+01
4	-0.1336182024D+01	-0.1313016428D+01
5	-0.1119529302D+01	-0.1106382505D+01
6	-0.1021510328D+01	-0.1014774498D+01
7	-0.7178493293D+00	-0.6597169810D+00
8	-0.1372385154D+00	-0.6083781653D-01

Tabela A-1.7  $N = 32, L = 16$ 

$i$	$\gamma_i$	$\lambda_i$
1	-0.1856362681D+02	-0.1834582324D+02
2	-0.6241180628D+01	-0.6138259752D+01
3	-0.3772407187D+01	-0.3710785606D+01
4	-0.2721371454D+01	-0.2680828510D+01
5	-0.2145326231D+01	-0.2117148166D+01
6	-0.1786244922D+01	-0.1765914713D+01
7	-0.1544710909D+01	-0.1529644174D+01
8	-0.1374374843D+01	-0.1362994463D+01
9	-0.1250847332D+01	-0.1242134401D+01
10	-0.1160240716D+01	-0.1153491043D+01
11	-0.1094278181D+01	-0.1088968159D+01
12	-0.1047949186D+01	-0.1043677405D+01
13	-0.1018224827D+01	-0.1014725722D+01
14	-0.1003131259D+01	-0.1000265571D+01
15	-0.6934562022D+00	-0.6341042889D+00
16	-0.1316573156D+00	-0.5922453055D-01

Tabela A-1.8  $N = 64, L = 1$ 

$i$	$\gamma_i$	$\lambda_i$
1	-0.4011935113D+02	-0.4007735784D+02
2	-0.1339913466D+02	-0.1337019035D+02
3	-0.8058168813D+01	-0.8035422789D+01
4	-0.5772685072D+01	-0.5753888619D+01
5	-0.4506089989D+01	-0.4490138143D+01
6	-0.3702817757D+01	-0.3689059812D+01
7	-0.3149148866D+01	-0.3137158867D+01
8	-0.2745334371D+01	-0.2734813139D+01
9	-0.2438563069D+01	-0.2429289526D+01
10	-0.2198261572D+01	-0.2190065922D+01
11	-0.2005507176D+01	-0.1998254955D+01
12	-0.1847963817D+01	-0.1841545854D+01
13	-0.1717248496D+01	-0.1711574270D+01
14	-0.1607468058D+01	-0.1602461146D+01
15	-0.1514361419D+01	-0.1509956259D+01
16	-0.1434773846D+01	-0.1430913372D+01
17	-0.1366322453D+01	-0.1362956321D+01
18	-0.1307176453D+01	-0.1304259678D+01

Tabela A-1.8  $N = 64, L = 1$ 

$i$	$\gamma_i$	$\lambda_i$
19	-0.1255908758D+01	-0.1253400647D+01
20	-0.1211393342D+01	-0.1209256640D+01
21	-0.1172732749D+01	-0.1170932951D+01
22	-0.1139205966D+01	-0.1137710755D+01
23	-0.1110230347D+01	-0.1109009130D+01
24	-0.1085333433D+01	-0.1084356958D+01
25	-0.1064131888D+01	-0.1063371935D+01
26	-0.1046315659D+01	-0.1045744784D+01
27	-0.1031636073D+01	-0.1031227399D+01
28	-0.1019897021D+01	-0.1019624064D+01
29	-0.1010948810D+01	-0.1010785305D+01
30	-0.1004684868D+01	-0.1004604528D+01
31	-0.1001043644D+01	-0.1001019563D+01
32	-0.5817792778D+00	-0.1412078482D+00

Tabela A-1.9  $N = 128, L = 127$ 

$i$	$\gamma_i$	$\lambda_i$
1	-0.7594642674D+02	-0.7547054818D+02
2	-0.2541417595D+02	-0.2516251168D+02
3	-0.1526858524D+02	-0.1510430525D+02
4	-0.1091272446D+02	-0.1079607954D+02
5	-0.8491434889D+01	-0.8404519915D+01
6	-0.6951123236D+01	-0.6884178538D+01
7	-0.5885850408D+01	-0.5832961470D+01
8	-0.5105888960D+01	-0.5063226080D+01
9	-0.4510663326D+01	-0.4475633237D+01
10	-0.4041882719D+01	-0.4012671802D+01
11	-0.3663432457D+01	-0.3638740551D+01
12	-0.3351746098D+01	-0.3330622146D+01
13	-0.3090793476D+01	-0.3072529602D+01
14	-0.2869293309D+01	-0.2853354322D+01
15	-0.2679077556D+01	-0.2665052881D+01
16	-0.2514088000D+01	-0.2501658288D+01
17	-0.2369737201D+01	-0.2358650779D+01
18	-0.2242488306D+01	-0.2232544352D+01

Tabela A-1.9  $N = 128, L = 127$ 

$i$	$\gamma_i$	$\lambda_i$
19	-0.2129571007D+01	-0.2120607396D+01
20	-0.2028784864D+01	-0.2020669282D+01
21	-0.1938360188D+01	-0.1930983565D+01
22	-0.1856857765D+01	-0.1850129363D+01
23	-0.1783095337D+01	-0.1776939043D+01
24	-0.1716092885D+01	-0.1710444362D+01
25	-0.1655031315D+01	-0.1649835813D+01
26	-0.1599220899D+01	-0.1594431528D+01
27	-0.1548076859D+01	-0.1543653235D+01
28	-0.1501100294D+01	-0.1497007455D+01
29	-0.1457863134D+01	-0.1454070667D+01
30	-0.1417996155D+01	-0.1414477492D+01
31	-0.1381179365D+01	-0.1377911202D+01
32	-0.1347134223D+01	-0.1344096048D+01
33	-0.1315617302D+01	-0.1312791002D+01
34	-0.1286415094D+01	-0.1283784628D+01
35	-0.1259339726D+01	-0.1256890849D+01
36	-0.1234225400D+01	-0.1231945438D+01
37	-0.1210925434D+01	-0.1208803090D+01

Tabela A-1.9  $N = 128, L = 127$ 

$i$	$\gamma_i$	$\lambda_i$
38	-0.1189309778D+01	-0.1187334970D+01
39	-0.1169262927D+01	-0.1167426649D+01
40	-0.1150682158D+01	-0.1148976368D+01
41	-0.1133476037D+01	-0.1131893563D+01
42	-0.1117563148D+01	-0.1116097611D+01
43	-0.1102871009D+01	-0.1101516762D+01
44	-0.1089335144D+01	-0.1088087228D+01
45	-0.1076898280D+01	-0.1075752391D+01
46	-0.1065509668D+01	-0.1064462118D+01
47	-0.1055124495D+01	-0.1054172168D+01
48	-0.1045703421D+01	-0.1044843693D+01
49	-0.1037212258D+01	-0.1036442866D+01
50	-0.1029621845D+01	-0.1028940706D+01
51	-0.1022908174D+01	-0.1022313221D+01
52	-0.1017052778D+01	-0.1016541973D+01
53	-0.1012043330D+01	-0.1011615158D+01
54	-0.1007874441D+01	-0.1007529258D+01
55	-0.1004549188D+01	-0.1004291578D+01
56	-0.1002084132D+01	-0.1001924486D+01

Tabela A-1.9  $N = 128, L = 127$

$i$	$\gamma_i$	$\lambda_i$
57	-0.1000527197D+01	-0.1000471152D+01
58	-0.9975627454D+00	-0.9944347585D+00
59	-0.9191860059D+00	-0.9064761231D+00
60	-0.7597563835D+00	-0.7400583708D+00
61	-0.5507764208D+00	-0.5271716453D+00
62	-0.3315410480D+00	-0.3084500201D+00
63	-0.1458382127D+00	-0.1287108514D+00
64	-0.3183102019D-01	-0.2572337925D-01



Tabela A-1.10  $N = 256, L = 128$ 

$i$	$\gamma_i$	$\lambda_i$
1	-0.1572524536D+03	-0.1569319255D+03
2	-0.5251096511D+02	-0.5231342111D+02
3	-0.3153551751D+02	-0.3139138818D+02
4	-0.2253845958D+02	-0.2242599459D+02
5	-0.1753724889D+02	-0.1744614809D+02
6	-0.1435352730D+02	-0.1427791461D+02
7	-0.1214902485D+02	-0.1208516581D+02
8	-0.1053236133D+02	-0.1047770860D+02
9	-0.9296243002D+01	-0.9248970279D+01
10	-0.8320607302D+01	-0.8279356226D+01
11	-0.7531100279D+01	-0.7494831776D+01
12	-0.6879211986D+01	-0.6847113534D+01
13	-0.6331943935D+01	-0.6303368873D+01
14	-0.5866067203D+01	-0.5840493426D+01
15	-0.5464755598D+01	-0.5441756514D+01
16	-0.5115521568D+01	-0.5094745120D+01
17	-0.4808902357D+01	-0.4790055502D+01
18	-0.4537596396D+01	-0.4520433408D+01

Tabela A-1.10  $N = 256, L = 128$ 

$i$	$\gamma_i$	$\lambda_i$
19	-0.4295879637D+01	-0.4280193126D+01
20	-0.4079201380D+01	-0.4064815300D+01
21	-0.3883898317D+01	-0.3870662451D+01
22	-0.3706988327D+01	-0.3694773889D+01
23	-0.3546019226D+01	-0.3534715319D+01
24	-0.3398956102D+01	-0.3388466841D+01
25	-0.3264096220D+01	-0.3254338378D+01
26	-0.3140003942D+01	-0.3130905008D+01
27	-0.3025460356D+01	-0.3016956936D+01
28	-0.2919423902D+01	-0.2911460378D+01
29	-0.2820999271D+01	-0.2813526683D+01
30	-0.2729412626D+01	-0.2722387737D+01
31	-0.2643991696D+01	-0.2637376200D+01
32	-0.2564149660D+01	-0.2557909513D+01
33	-0.2489371998D+01	-0.2483476856D+01
34	-0.2419205705D+01	-0.2413628438D+01
35	-0.2353250372D+01	-0.2347966657D+01
36	-0.2291150778D+01	-0.2286138745D+01
37	-0.2232590699D+01	-0.2227830634D+01

Tabela A-1.10  $N = 256, L = 128$ 

$i$	$\gamma_i$	$\lambda_i$
38	-0.2177287705D+01	-0.2172761789D+01
39	-0.2124988771D+01	-0.2120680858D+01
40	-0.2075466554D+01	-0.2071361980D+01
41	-0.2028516220D+01	-0.2024601633D+01
42	-0.1983952729D+01	-0.1980215950D+01
43	-0.1941608511D+01	-0.1938038401D+01
44	-0.1901331451D+01	-0.1897917807D+01
45	-0.1862983154D+01	-0.1859716610D+01
46	-0.1826437439D+01	-0.1823309381D+01
47	-0.1791579023D+01	-0.1788581516D+01
48	-0.1758302374D+01	-0.1755428094D+01
49	-0.1726510706D+01	-0.1723752885D+01
50	-0.1696115096D+01	-0.1693467468D+01
51	-0.1667033704D+01	-0.1664490461D+01
52	-0.1639191089D+01	-0.1636746838D+01
53	-0.1612517596D+01	-0.1610167323D+01
54	-0.1586948818D+01	-0.1584687858D+01
55	-0.1562425115D+01	-0.1560249119D+01
56	-0.1538891184D+01	-0.1536796095D+01

Tabela A-1.10  $N = 256, L = 128$ 

$i$	$\gamma_i$	$\lambda_i$
57	-0.1516295676D+01	-0.1514277704D+01
58	-0.1494590855D+01	-0.1492646455D+01
59	-0.1473732286D+01	-0.1471858140D+01
60	-0.1453678561D+01	-0.1451871561D+01
61	-0.1434391050D+01	-0.1432648280D+01
62	-0.1415833674D+01	-0.1414152397D+01
63	-0.1397972704D+01	-0.1396350347D+01
64	-0.1380776573D+01	-0.1379210719D+01
65	-0.1364215716D+01	-0.1362704087D+01
66	-0.1348262411D+01	-0.1346802865D+01
67	-0.1332890645D+01	-0.1331481161D+01
68	-0.1318075991D+01	-0.1316714663D+01
69	-0.1303795486D+01	-0.1302480517D+01
70	-0.1290027536D+01	-0.1288757225D+01
71	-0.1276751811D+01	-0.1275524554D+01
72	-0.1263949165D+01	-0.1262763445D+01
73	-0.1251601552D+01	-0.1250455933D+01
74	-0.1239691952D+01	-0.1238585077D+01
75	-0.1228204308D+01	-0.1227134889D+01

Tabela A-1.10  $N = 256, L = 128$ 

$i$	$\gamma_i$	$\lambda_i$
76	-0.1217123456D+01	-0.1216090276D+01
77	-0.1206435077D+01	-0.1205436981D+01
78	-0.1196125636D+01	-0.1195161530D+01
79	-0.1186182339D+01	-0.1185251186D+01
80	-0.1176593084D+01	-0.1175693902D+01
81	-0.1167346424D+01	-0.1166478281D+01
82	-0.1158431526D+01	-0.1157593537D+01
83	-0.1149838133D+01	-0.1149029462D+01
84	-0.1141556537D+01	-0.1140776390D+01
85	-0.1133577545D+01	-0.1132825170D+01
86	-0.1125892449D+01	-0.1125167136D+01
87	-0.1118493005D+01	-0.1117794080D+01
88	-0.1111371403D+01	-0.1110698231D+01
89	-0.1104520248D+01	-0.1103872231D+01
90	-0.1097932539D+01	-0.1097309113D+01
91	-0.1091601646D+01	-0.1091002282D+01
92	-0.1085521294D+01	-0.1084945498D+01
93	-0.1079685548D+01	-0.1079132858D+01
94	-0.1074088793D+01	-0.1073558780D+01

Tabela A-1.10  $N = 256, L = 128$ 

$i$	$\gamma_i$	$\lambda_i$
95	-0.1068725721D+01	-0.1068217989D+01
96	-0.1063591318D+01	-0.1063105502D+01
97	-0.1058680852D+01	-0.1058216618D+01
98	-0.1053989860D+01	-0.1053546901D+01
99	-0.1049514137D+01	-0.1049092174D+01
100	-0.1045249729D+01	-0.1044848508D+01
101	-0.1041192925D+01	-0.1040812213D+01
102	-0.1037340251D+01	-0.1036979830D+01
103	-0.1033688465D+01	-0.1033348130D+01
104	-0.1030234555D+01	-0.1029914108D+01
105	-0.1026975738D+01	-0.1026674983D+01
106	-0.1023909461D+01	-0.1023628201D+01
107	-0.1021033403D+01	-0.1020771442D+01
108	-0.1018345480D+01	-0.1018102622D+01
109	-0.1015843848D+01	-0.1015619910D+01
110	-0.1013526911D+01	-0.1013321736D+01
111	-0.1011393327D+01	-0.1011206806D+01
112	-0.1009442018D+01	-0.1009274119D+01
113	-0.1007672189D+01	-0.1007522985D+01

Tabela A-1.10  $N = 256, L = 128$ 

$i$	$\gamma_i$	$\lambda_i$
114	-0.1006083354D+01	-0.1005953047D+01
115	-0.1004675379D+01	-0.1004564303D+01
116	-0.1003448565D+01	-0.1003357132D+01
117	-0.1002403761D+01	-0.1002332309D+01
118	-0.1001542533D+01	-0.1001491011D+01
119	-0.1000867355D+01	-0.1000834789D+01
120	-0.1000381747D+01	-0.1000365549D+01
121	-0.1000090356D+01	-0.1000085776D+01
122	-0.9968737239D+00	-0.9934485133D+00
123	-0.9165215365D+00	-0.9036558444D+00
124	-0.7564224738D+00	-0.7366587065D+00
125	-0.5477558629D+00	-0.5241741312D+00
126	-0.3294897016D+00	-0.3064911863D+00
127	-0.1449202569D+00	-0.1279084146D+00
128	-0.3169621215D-01	-0.2565259359D-01