

~~100647-5~~
105031-0

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**Métodos Intervalares para a
Resolução de Sistemas de
Equações Lineares**

por

Carlos Amaral Hölbig

Dissertação submetida como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação



Orientador
Prof.Dr.Dalcídio Moraes Cláudio

Co-orientador
Prof.Dr.Tiarajú Asmuz Diverio

Porto Alegre, fevereiro de 1996

UFRGS
INSTITUTO DE INFORMÁTICA
BIBLIOTECA

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Hölbig, Carlos Amaral

Métodos Intervalares para a Resolução de Sistemas de Equações Lineares / Carlos Amaral Hölbig. — Porto Alegre: CPGCC da UFRGS, 1996.

94 p.: il.

Dissertação (mestrado) — Universidade Federal do Rio Grande do Sul. Curso de Pós-Graduação em Ciência da Computação, Porto Alegre, 1996. Orientador: Claudio, Dalcidio Moraes; Co-orientador: Diverio, Tiarajú Asmuz.

1. Aritmética Intervalar. 2. Matemática Computacional. 3. Métodos Intervalares. 4. Sistemas de Equações Lineares. 5. Biblioteca Aplicativa Intervalar *selint.p.* I. Claudio, Dalcidio Moraes. II. Diverio, Tiarajú Asmuz. III. Título.

UFRGS INSTITUTO DE INFORMÁTICA BIBLIOTECA		
N.º CHAMADA 518.8(043) H723M		N.º REG.: 3739
ORIGEM: D		DATA: 02/07/96
DATA: 24/06/96		PREÇO: R\$ 30,00
FUNDO: II	FORN.: II	

matemática computacional - SBU
Análise: Intervalos
Aritmética intervalar
Equações lineares
ENPq 1.01.04.00-3

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Dr. Hélgio Casses Trindade

Pró-Reitor de Pós-Graduação: Prof. Dr. Claudio Scherer

Diretor do Instituto de Informática: Prof. Dr. Roberto Tom Price

Coordenador do CPGCC: Prof. Dr. Flávio Rech Wagner

Bibliotecária-Chefe do Instituto de Informática: Dra. Zita Prates de Oliveira

Agradecimentos

Aos funcionários e professores do Instituto de Informática e do Pós-Graduação em Ciência da Computação da UFRGS.

Aos orientadores e amigos Profs.Drs. Dalcídio e Tiaraju por todos os ensinamentos transmitidos, pela ajuda e amizade e principalmente pela confiança depositada em meu trabalho durante esses anos em que estou na UFRGS sob suas orientações.

Aos meus amigos que de uma maneira ou de outra me apoiaram e ajudaram durante a elaboração deste trabalho, em especial a: Úrsula, Fabiana, Simone, Marcelo, Cleber, Silvia, Débora e Alessandra. Ao Cristiano pela auxílio na implementação das bibliotecas e a Denise pelo auxílio na revisão deste trabalho. Aos membros do grupo de Matemática Computacional e aos colegas de mestrado pela amizade.

A minha família, que mesmo a distância, deu todo o apoio e carinho para que eu pudesse realizar este trabalho.

Muito Obrigado.

Sumário

Lista de Figuras.....	06
Lista de Fórmulas.....	07
Lista de Tabelas.....	09
Lista de Símbolos	10
Resumo	11
Abstract	13
1 Introdução	15
1.1 Revisão da Matrizes e da Álgebra Matricial Pontual	16
1.2 Revisão da Aritmética Matricial Intervalar	19
1.3 Revisão de Sistemas de Equações Lineares	21
1.4 Aplicabilidade de Sistemas de Equações Lineares	23
1.5 Descrição dos Capítulos	23
2 Métodos Intervalares.....	25
2.1 Introdução	25
2.2 Métodos Intervalares.....	25
2.2.1 Resolução de Sistemas de Equações de Ordem 1	26
2.2.1.1 Métodos Algébricos.....	27
2.2.1.2 Métodos de Newton Intervalar	28
2.2.2 Métodos Intervalares Baseados em Operações Algébricas.....	30
2.2.2.1 Métodos de Hansen	31
2.2.2.2 Método de Eliminação de Gauss	35
2.2.2.3 Cálculo da Matriz Inversa	38
2.2.3 Métodos Intervalares Baseados em Refinamento.....	39
2.2.3.1 Método para sistemas em geral	39
2.2.3.2 Método para sistemas com matrizes densas.....	41
2.2.3.3 Método para sistemas com matrizes bandas	42
2.2.3.4 Cálculo da matriz inversa	43
2.2.4 Métodos Intervalares Baseados em Iteração.....	44
2.2.4.1 Método de Jacobi Intervalar.....	45
2.2.4.2 Método de Gauss-Seidel Intervalar	45
2.2.4.3 Método de Relaxação Intervalar	45
2.2.4.4 Método de Relaxação Simétrico Intervalar.....	46
2.3 Conclusões	47
3 Biblioteca Aplicativa Intervalar - Módulo Sistemas	49
3.1 Proposta, Objetivos e Metodologia de Desenvolvimento	49
3.2 Estrutura da Biblioteca	50
3.2.1 Módulo <i>dirint</i> (métodos diretos).....	51
3.2.2 Módulo <i>refint</i> (métodos baseados em refinamento).....	52
3.2.3 Módulo <i>itrint</i> (métodos iterativos).....	52
3.2.4 Módulo <i>equalg</i> (sistemas de equações de ordem 1).....	52

3.3 Definição da Documentação.....	53
3.4 A Biblioteca <i>selint.p</i>	54
3.5 Bibliotecas utilizadas para comparações de resultados.....	56
4 Resolução de Problemas e Comparações.....	59
4.1 Exemplos selecionados.....	59
4.2 Comparação da solução: pontual \times intervalar.....	59
4.3 Comparação da solução intervalar: <i>selint.p</i> \times <i>libselint.a</i>	65
4.4 Conclusões.....	66
5 Conclusões.....	68
5.1 Resumo do Trabalho.....	68
5.2 Considerações Finais.....	70
5.3 Proposta para Novos Trabalhos.....	71
Anexo 1: Algoritmos dos Métodos Intervalares.....	73
1.1 Algoritmos dos Métodos Intervalares do Módulo <i>equal</i>	73
1.2 Algoritmos dos Métodos Intervalares do Módulo <i>dirint</i>	74
1.3 Algoritmos dos Métodos Intervalares do Módulo <i>refint</i>	77
1.4 Algoritmos dos Métodos Intervalares do Módulo <i>itrint</i>	80
Anexo 2: Exemplos Selecionados.....	81
Bibliografia.....	89

Lista de Figuras

Figura 1.1 - Definição do produto de duas matrizes	18
Figura 1.2 - Propriedades equivalentes de matrizes singulares e não singulares.....	19
Figura 1.3 - Classificação de um Sistema Linear quanto ao número de soluções.....	22
Figura 2.1 - Seqüência aninhada de intervalos.....	29
Figura 2.2 - Seqüência de intervalos gerado pelo Método de Newton pontual.....	30
Figura 2.3 - Conjunto solução X - área total	33
Figura 3.1 - Estrutura da Biblioteca <i>selint.p</i>	51
Figura 3.2 - Formato da documentação da biblioteca <i>selint.p</i>	54

Lista de Fórmulas

(1.1) Adição de duas matrizes.....	17
(1.2) Multiplicação de α pela matriz A.....	17
(1.3) Definição do elemento c_{ij} da multiplicação $A \times B$	18
(1.4) Definição da matriz inversa baseada em determinante.....	19
(1.5) Formulação de um sistema de equações lineares.....	21
(1.6) Sistema de equações lineares na forma $AX = B$	22
(2.1) Fórmula para a resolução da equação $A + X = B$	27
(2.2) Fórmula para a resolução da equação $AX + B = C$	27
(2.3) Fórmula para a resolução da equação $AX + BX = C$	27
(2.4) Fórmula para a resolução da equação $AX + B = CX + D$	28
(2.5) Função de iteração do Método de Newton.....	28
(2.6) Generalização do método de Newton Intervalar.....	28
(2.7) Operador Newtoniano.....	29
(2.8) Método de Newton Intervalar simplificado.....	29
(2.9) Resultado do exemplo que utiliza o Método de Newton.....	29
(2.10) Exemplo de sistema mal-condicionado.....	30
(2.11) Conjunto de equações lineares intervalares modificadas.....	32
(2.12) Conjunto de equações lineares intervalares modificadas - quadrante positivo.....	32
(2.13) Desenvolvimento do conjunto de equações lineares intervalares.....	32
(2.14) Exemplo de sistema de equações lineares com coeficientes reais intervalares.....	32
(2.15) Matriz P_r^l formada pelo Método de Hansen.....	33
(2.16) Matriz Q_r^l formada pelo Método de Hansen.....	34
(2.17) Vetor c_r^l formado pelo Método de Hansen.....	34
(2.18) Vetor d_r^l formado pelo Método de Hansen.....	34
(2.19) Substituições realizadas pelo Método de Hansen.....	34
(2.20) Conjuntos de sistemas de equações lineares.....	34
(2.21) Solução obtida pelo Método de Hansen.....	35
(2.22) Matriz obtida com a concatenação de A^l com B^l	36
(2.23) Fórmulas para obter nova matriz de coeficientes.....	36
(2.24) Nova matriz de coeficientes obtida pelo MEG - Intervalar.....	36
(2.25) Fórmulas para obter novos valores para A^l e B^l	36
(2.26) Matriz intervalar triangular superior.....	37
(2.27) Retrossubstituição no MEG - Intervalar.....	37
(2.28) Iteração de ponto-fixo obtida pelo método de Newton.....	39
(2.29) Iteração Newton-Like.....	40
(2.30) Modificação da iteração Newton-Like.....	40
(2.31) Iteração residual da equação $AY = D$	40
(2.32) Equação que contém a solução do sistema $AX = B$	41
(2.33) Equação que contém a solução do sistema $AX = B$ multiplicada por R.....	41
(2.34) Equação na forma de ponto-fixo para o erro Y.....	41
(2.35) Iteração intervalar para Y.....	41
(2.36) Iteração intervalar para Y.....	41

(2.37) Iteração intervalar para Y com produto escalar ótimo.....	42
(2.38) Iteração intervalar para Y com produto escalar ótimo.....	42
(2.39) Iteração utilizando a decomposição LU de A.....	43
(2.40) Extensão intervalar da iteração com decomposição LU de A.....	43
(2.41) Iteração do sistema triangular com matriz triangular L.....	43
(2.42) Iteração do sistema triangular com matriz triangular U.....	43
(2.43) Teste de inclusão da iteração.....	43
(2.44) Aproximação da matriz inversa de A.....	43
(2.45) Aproximação da matriz inversa de A com decomposição LU.....	44
(2.46) Aproximação LU de A sem pivotamento.....	44
(2.47) Formulação da iteração do Método de Jacobi.....	45
(2.48) Formulação da iteração do Método de Gauss-Seidel.....	45
(2.49) Condição de convergência do Método de Gauss-Seidel.....	45
(2.50) Formulação da iteração do Método de Relaxação.....	46
(2.51) Formulação da iteração do Método de Relaxação (notação vetorial).....	46
(2.52) Condição de convergência do Método de Relaxação.....	46
(2.53) Formulação da iteração do Método de Relaxação Simétrico.....	46
(2.54) Condição de convergência do Método de Relaxação Simétrico.....	47

Lista de Tabelas

Tabela 3.1 - Funções do módulo <i>dirint</i>	51
Tabela 3.2 - Funções do módulo <i>refint</i>	52
Tabela 3.3 - Funções do módulo <i>itrint</i>	52
Tabela 3.4 - Funções do módulo <i>equalg</i>	53
Tabela 4.1 - Soluções intervalares do exemplo 19	60
Tabela 4.2 - Análise dos resultados intervalares do exemplo 19	61
Tabela 4.3 - Análise dos resultados intervalares dos exemplos 1 a 7 e 9	61
Tabela 4.4 - Análise dos resultados intervalares dos exemplos 11, 12, 15, 18 e 20 a 23 ...	62
Tabela 4.5 - Análise dos resultados intervalares dos exemplos 8, 10, 13, 14, 16 e 17	63
Tabela 4.6 - Resultados obtidos pelo método <i>refbanda</i> e pelo método <i>de gauss s/pivot</i> . ..	64
Tabela 4.7 - Comparação referente ao tempo de processamento	64
Tabela 4.8 - Velocidade de processamento das operações de adição e subtração	65
Tabela 4.9 - Comparação entre as bibliotecas <i>selint.p</i> e <i>libselint.a</i> para o exemplo 1	66

Lista de Símbolos

#	Operação genérica	\approx	Aproximação
+	Operação de adição	ρ	Raio espectral
-	Operação de subtração	\diamond	Operações executadas exatamente
*	Operação de multiplicação	\subset	Inclusão no interior do intervalo
/	Operação de divisão	X^*	Solução única de X
:	Operação de divisão	∂x	Derivada parcial de x
Σ	Somatório	\tilde{X}	Aproximação da solução de X
$-\infty$	Menos infinito	$f(x)$	Função
$+\infty$	Mais infinito	$f'(x)$	Função derivada
		$F(X)$	Função intervalar
\leq	Menor ou igual	$F'(X)$	Função intervalar derivada
\geq	Maior ou igual	$m(X)$	Ponto médio de X
\subseteq	Contido ou igual	X^k	k iteração de X
\in	Relação de pertinência: pertence	$d(A)$	Diâmetro da matriz A
\forall	Quantificador Universal: Para todo	$q(A,B)$	Distância entre as matrizes A e B
\exists	Quantificador existencial: Existe		
\cap	Intersecção	I	Matriz identidade
\cup	União	A	Matriz/vetor pontual
		A_{ij}/A_i	Elemento da matriz/vetor A^i
$X=[x_1, x_2]$	Um intervalo cujos extremos são x_1 e x_2	a_{ij}/a_i	Elemento da matriz/vetor A
		A_c	Matriz centro de A
\emptyset	Conjunto vazio	$\det(A)$	Determinante da matriz A
F	Conjunto dos números em ponto-flutuante	$\text{Adj}(A)$	Matriz adjunta de A
R	Conjunto dos Números reais	AT	Matriz transposta de A
C	Conjunto dos Números complexos	AH	Matriz Hermitiana
IR	Conjunto dos intervalos reais	H_n	Matriz de Hilbert
IF	Conjunto dos intervalos em ponto-flutuante	MEG	Métodos de Eliminação de Gauss
IC	Conjunto dos intervalos complexos	X_k°	Interior de X
VC, MC	Conjuntos dos vetores e matrizes de C		
IVR, IMR	Conjuntos dos vetores e matrizes de R		
A^i	Matriz/vetor intervalar		
A^{-1}	Matriz inversa pontual		
$(A^i)^{-1}$	Matriz inversa intervalar		

Resumo

O estudo dos métodos intervalares é importante para a resolução de sistemas de equações lineares, pois os métodos intervalares produzem resultados dentro de limites confiáveis (do intervalo solução) e provam a existência ou não existência de soluções, portanto produzem resultados confiáveis, o que os métodos pontuais podem não proporcionar.

Outro aspecto a destacar é o campo de utilização de sistemas de equações lineares em problemas das engenharias e outras ciências, o que mostra a aplicabilidade desses métodos e por conseguinte a necessidade de elaboração de ferramentas que possibilitem a implementação desses métodos intervalares.

O objetivo deste trabalho não é a elaboração de novos métodos intervalares, mas sim o de realizar uma descrição e implementação de alguns dos métodos intervalares encontrados na bibliografia pesquisada. A versão intervalar dos métodos pontuais não é simples e o cálculo por métodos intervalares pode ser dispendioso, uma vez que se está tratando com vetores e matrizes de intervalos.

A implementação dos métodos intervalares só foi possível graças a existência de ferramentas, como o compilador Pascal-XSC, que incorpora as suas características aspectos importantes como a aritmética intervalar, a verificação automática do resultado, o produto escalar ótimo e a aritmética de alta exatidão.

Este trabalho é dividido em duas etapas. A primeira apresenta um estudo dos métodos intervalares para a resolução de sistemas de equações lineares. São caracterizadas as metodologias de desenvolvimento desses métodos. Metodologias estas, que foram divididas em três grupos de métodos: métodos intervalares baseados em operações algébricas intervalares ou métodos diretos, métodos intervalares baseados em refinamento ou métodos híbridos e métodos intervalares baseados em iterações. São definidas as características, os métodos que as compõe e a aplicabilidade desses métodos na resolução de sistemas de equações lineares.

A segunda etapa é caracterizada pela elaboração dos algoritmos referentes aos métodos intervalares estudados e sua respectiva implementação, dando origem a uma biblioteca aplicativa intervalar para a resolução de sistemas de equações lineares, *selint.p*, implementada no PC-486 e utilizando o compilador Pascal-XSC. Para este desenvolvimento foi realizado, previamente, um estudo sobre este compilador e sobre bibliotecas disponíveis que são utilizadas na implementação da biblioteca aplicativa intervalar.

A biblioteca *selint.p* é organizada em quatro módulos: o módulo *dirint* (referente aos métodos diretos); o módulo *refint* (referente aos métodos baseados em refinamento); o módulo *itrint* (referente aos métodos iterativos) e o módulo *equalg* (para sistemas de equações de ordem 1).

Por fim, através daquela biblioteca foram realizadas comparações entre os resultados obtidos (resultados pontuais, intervalares, seqüenciais e vetoriais) a fim de se realizar uma análise de desempenho quantitativa (exatidão) e uma comparação entre os resultados obtidos. Esses resultados serão comparados com os obtidos com a biblioteca *libselint.a*, biblioteca esta que está sendo desenvolvida para o ambiente do supercomputador Cray Y-MP do CESUP/UFRGS, como parte do projeto de Aritmética Vetorial Intervalar do Grupo de Matemática Computacional da UFRGS.

PALAVRAS-CHAVE: Matemática Intervalar, Intervalos, Sistemas de Equações Lineares (SELAS), Métodos Intervalares para Resolução de Sistemas Lineares, Biblioteca Aplicativa Intervalar.

Title: "Interval Methods for Resolution of Linear Equation Systems"

Abstract

The study of interval methods is important for resolution of linear equation systems, because such methods produce results into reliable bounds and prove the existence or not existence of solutions, therefore they produce reliable results that, the punctual methods can non present, save that there is an exhaustive analysis of errors.

Another aspect to emphasize is the field of utilization of linear equation systems in engineering problems and other sciences, in which is showed the applicability of that methods and, consequently, the necessity of tools elaboration that make possible the implementation of that interval methods.

The goal of this work is not the elaboration of new interval methods, but to accomplish a description and implementation of some interval methods found in the searched bibliography. The interval version of punctual methods is not simple, and the calculus by interval methods can be expensive, respecting is treats of vectors and matrices of intervals.

The implementation of interval methods was only possible due to the existence of tools, as the Pascal-XSC compiler, which incorporates to their features, important aspects such as the interval arithmetic, the automatic verification of the result, the optimal scalar product and arithmetic of high accuracy.

This work is divided in two stages. The first presents a study of the interval methods for resolution of linear equation systems, in which are characterized the methodologies of development of that methods. These methodologies were divided in three method groups: interval methods based in interval algebraic operations or direct methods, interval methods based in refinement or hybrid methods, and interval methods based in iterations, in which are determined the features, the methods that compose them, and the applicability of those methods in the resolution of linear equation systems.

The second stage is characterized for the elaboration of the algorithms relating to the interval methods studied and their respective implementation, originating a interval applied library for resolution of linear equation systems, *selint.p*, implemented in PC-486 and making use of Pascal-XSC compiler. For this development was previously accomplished a study about compiler and available libraries that are used in the implementation of the interval applied library.

The library *selint.p* is organized in four modules: the *dirint* module (regarding to the direct methods); the *refint* module (regarding to the methods based in refinement); the *itrint* module (regarding to the iterative methods) and *equalg* module (for equation systems of order 1).

At last, through this library, comparisons were developed among the results obtained (punctual, interval, sequential and vectorial results) in order to be accomplished an analysis of quantitative performance (accuracy) and a comparison among the results obtained with *libselint.a* library, that is been developed for the Cray Y-MP supercomputer environment of CESUP/UFRGS, as part of the Interval Vectorial Arithmetic project of Group of Computational Mathematics of UFRGS.

KEYWORDS: Interval Mathematics, Intervals, Linear Equations Systems (SELAS), Intervals Methods for Resolution of Linear Systems, Interval Applied Library.

1 Introdução

A motivação para a elaboração deste trabalho, surgiu com estudos desenvolvidos junto ao Grupo de Matemática Computacional do CPGCC da UFRGS, a respeito de métodos intervalares e pontuais, linguagens para computação científica, aplicabilidade de sistemas lineares em vários ramos das engenharias e das ciências, análise sensitiva, proximidade de singularidade e instabilidade numérica (descritos em [HOL94a, HÖL95]). Através dessas pesquisas, pode-se notar que problemas de instabilidade, podem levar a resultados errôneos na resolução de sistemas de equações lineares. Devido a isto, foi feito um estudo mais aprofundado a respeito de metodologias de desenvolvimento de métodos intervalares, visando a compreensão de suas características, técnicas e aplicações em outras áreas. Pode-se observar que os métodos intervalares produzem resultados dentro de limites confiáveis e que provam a existência ou não existência de soluções, proporcionando, portanto, resultados confiáveis que métodos pontuais não proporcionam sem uma análise exaustiva de erros, o que determinou a importância desse estudo mais aprofundado.

Esta pesquisa enquadra-se dentro de um projeto de pesquisa do Grupo de Matemática Computacional do CPGCC da UFRGS, relacionada a aplicações da matemática intervalar, no sentido de proporcionar ferramentas que possibilitem aos usuários utilizá-las e disseminar a cultura do uso de intervalos na matemática aplicada, engenharias, física, química e demais ciências que dela necessitar. Por isso, estão sendo desenvolvidas bibliotecas aplicativos intervalares, contendo métodos intervalares, como para a resolução de sistemas de equações lineares, o que proporcionará condições que possibilitem o estudo e a demonstração da aplicabilidade de métodos intervalares.

A pesquisa trata, inicialmente, de métodos intervalares de resolução de sistemas de equações lineares encontrados e selecionados na literatura disponível. Foram identificadas características ou princípios comuns a esses métodos intervalares, no sentido de agrupá-los e de definir metodologias. Os métodos intervalares foram agrupados em três grupos: métodos intervalares baseados em operações algébricas intervalares ou métodos diretos, baseados em refinamento ou métodos híbridos e baseados em iterações. Com a definição desses grupos ou metodologias, os métodos que as compõem foram identificados, descritos, caracterizados e projetados os algoritmos.

Em seguida foi realizada a implementação desses métodos, dando origem a uma biblioteca aplicativo intervalar para a resolução de sistemas de equações lineares. Esta biblioteca foi desenvolvida em ambiente de computadores pessoais (tipo PC), denominada *selint.p*, implementada em um PC-486, utilizando o compilador Pascal-XSC. Essa implementação foi precedida de um estudo sobre esse compilador e sobre bibliotecas disponíveis que são utilizadas na implementação da biblioteca aplicativo intervalar. Finalmente, através dessa biblioteca, foram realizadas comparações entre os resultados obtidos (resultados pontuais, intervalares, seqüenciais e vetoriais) a fim de se realizar uma análise de desempenho qualitativa e uma comparação entre os resultados obtidos. Para essa

comparação foi utilizada a biblioteca intervalar *libselint.a*, biblioteca desenvolvida para o ambiente do supercomputador Cray Y-MP do CESUP/UFRGS, implementada em FORTRAN 90 e utilizando a biblioteca intervalar *libavi.a*. Esta biblioteca faz parte de um projeto do GMC/UFRGS, a respeito da aritmética vetorial intervalar, na qual a biblioteca intervalar *libavi.a* e a biblioteca intervalar *libselint.a* fazem parte desse projeto.

1.1 Revisão de Matrizes e Álgebra Matricial Pontual

Neste item é apresentado uma revisão de matrizes e álgebra matricial pontual. São descritos alguns tipos de matrizes e as operações matriciais. Maiores detalhes a respeito destes tópicos podem ser encontrados em [CLA89, DIV90 e HÖL95].

Entre os tipos de matrizes existentes pode-se destacar os seguintes tipos:

- a) Matriz retangular: uma matriz $m \times n$ é dita retangular quando $m \neq n$ (onde m é o número de linhas e n é o número de colunas);
- b) Matriz quadrada: uma matriz $m \times n$ é dita quadrada quando $m = n$, neste caso, diz-se que a matriz é de dimensão n , não sendo necessário dizer $n \times n$;
- c) Matriz coluna/linha: quando a matriz só possui uma coluna/linha, ou seja $n = 1$ ou $m = 1$. Também chamada de vetor;
- d) Matriz triangular: uma matriz triangular é um caso especial de matriz quadrada, na qual todos os elementos de um lado da diagonal principal são nulos. Se os elementos abaixo da diagonal forem nulos, a matriz é denominada triangular superior; se os elementos acima da diagonal forem nulos, a matriz é chamada de triangular inferior;
- e) Matriz diagonal: é uma matriz quadrada que só possui elementos não nulos na diagonal principal;
- f) Matriz tridiagonal: é uma matriz quadrada que possui os elementos não nulos na diagonal principal e na primeira diagonal acima e abaixo dela;
- g) Matriz simétrica: é uma matriz quadrada cujos valores são simétricos com relação à diagonal principal, ou seja $a_{ij} = a_{ji}$;
- h) Matriz densa: uma matriz onde a maioria dos seus elementos são não nulos (usualmente, maioria significa 2/3 dos elementos diferentes de zero);

i) Matriz esparsa: uma matriz onde cerca de dois terços dos elementos são nulos. Um caso especial de matriz esparsa é do tipo banda, onde os elementos não nulos se concentram em certas bandas;

j) Matriz conjugada: dada uma matriz complexa, define-se a matriz conjugada como a matriz composta pelo conjugado de cada elemento complexo a_{ij} ;

k) Matriz Hermitiana: é definida como sendo a transposta da matriz conjugada de A e anotada por A^H ;

l) Matriz de Hilbert de ordem n: é uma matriz onde todos os elementos são menores ou iguais a um. Uma matriz de Hilbert de ordem n é calculada da forma $H_n = (h_{ij}) = [1/(i+j-1)]$, para $i = 1, \dots, n$ e $j = 1, \dots, n$;

m) Matriz positiva definida: é uma matriz onde os elementos da diagonal principal são maiores que zero e que o determinante dessa matriz é diferente de zero.

Duas matrizes A e B são ditas iguais se possuírem o mesmo número de linhas e colunas e se $a_{ij} = b_{ij}$ para qualquer i e j. Se A e B forem ambas de ordem $m \times n$, a adição será definida pela equação (1.1).

$$A + B = a_{ij} + b_{ij} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \cdots & a_{1n} + b_{1n} \\ a_{21} + b_{21} & a_{22} + b_{22} & \cdots & a_{2n} + b_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} + b_{m1} & a_{m2} + b_{m2} & \cdots & a_{mn} + b_{mn} \end{bmatrix} \quad (1.1)$$

e a multiplicação por um número α através da equação (1.2).

$$\alpha A = \alpha (a_{ij}) = \begin{bmatrix} \alpha a_{11} & \alpha a_{12} & \cdots & \alpha a_{1n} \\ \alpha a_{21} & \alpha a_{22} & \cdots & \alpha a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha a_{m1} & \alpha a_{m2} & \cdots & \alpha a_{mn} \end{bmatrix} \quad (1.2)$$

Para matrizes A, B e C, possuindo o mesmo número de linhas e colunas, as seguintes propriedades podem ser obtidas a partir das definições anteriores, onde α , β são números quaisquer, reais ou complexos, e 0 é a matriz nula.

- | | |
|---|----------------------------|
| a) $A + (B + C) = (A + B) + C$ | (Propriedade Associativa) |
| b) $A + B = B + A$ | (Propriedade Comutativa) |
| c) $A + 0 = 0 + A = A$ | (Identidade sob adição) |
| d) $(\alpha + \beta)A = \alpha A + \beta A$, | (Propriedade Distributiva) |

$$e) \alpha(A + B) = \alpha A + \alpha B$$

O produto de duas matrizes A e B é definido somente quando o número de colunas de A é igual ao número de linhas de B. Se A for de ordem $m \times n$ e B for de ordem $n \times p$, o produto $C = AB$ será de ordem $m \times p$ como mostrado na fórmula (1.3), onde o elemento c_{ij} de AB é obtido multiplicando-se os elementos da i-ésima linha de A pelos correspondentes elementos da j-ésima coluna de B e soma-se, então, esses produtos, conforme figura 1.1, onde:

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj} = \sum_{k=1}^n a_{ik}b_{kj} \quad i = 1, \dots, m \quad j = 1, \dots, p \quad (1.3)$$

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{i1} & \dots & a_{in} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} b_{11} & \dots & b_{1j} & \dots & b_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ b_{n1} & \dots & b_{nj} & \dots & b_{np} \end{pmatrix} = \begin{pmatrix} c_{11} & \dots & c_{1p} \\ \dots & \dots & \dots \\ \dots & c_{ij} & \dots \\ \dots & \dots & \dots \\ c_{m1} & \dots & c_{mp} \end{pmatrix}$$

Figura 1.1 - Definição do produto de duas matrizes

Se o produto está definido, as propriedades seguintes são verificadas. A demonstração delas pode ser facilmente obtida pela definição do produto, da adição de matrizes com as propriedades associativas e comutativa dos reais.

- a) $A(BC) = (AB)C$
- b) $A(B + C) = AB + AC$
- c) $(A + B)C = AC + BC$

Deve-se salientar que o produto de matrizes não dispõe da propriedade comutativa, ou seja, $AB \neq BA$, pois o produto de matrizes é obtido pelo produto escalar de linhas da primeira matriz por colunas da segunda matriz.

Potências de matrizes são definidas como casos particulares de produtos de fatores iguais. Entretanto, cabe ressaltar que A^0 resulta na identidade, A^1 é a própria A, A^2 é $A \cdot A$ e, por fim, A^n é igual a um produto de n fatores iguais a A. Estas operações entre matrizes reais são estendíveis a matrizes de intervalos; para tanto, utilizam-se de matrizes intervalares operadas por operações intervalares, como será visto no item 1.2.

A transposta da matriz A é obtida trocando as linhas por colunas e colunas por linhas. Representa-se a transposta de A através do símbolo A^T . Se A for a matriz $m \times n$, então, A^T será a matriz $n \times m$ e numa matriz simétrica, $A = A^T$. Se A e B são duas matrizes quaisquer, para as quais é definido o produto AB, então $(AB)^T = B^T A^T$.

A divisão de matrizes não é uma operação legítima na álgebra matricial, entretanto, o conceito de matriz inversa guarda certa semelhança com a notação da divisão. A inversa de matriz quadrada A é definida como a matriz B , tal que o produto de A por B e vice-versa, produz a matriz identidade, ou seja, $A.B=B.A=I$ e escreve-se $B = A^{-1}$. Dada uma matriz quadrada A possui uma inversa se, e somente se, A não for singular. Caso a matriz inversa existir, ela será única.

Chama-se determinante da matriz quadrada A , ou simplesmente determinante de ordem n , a soma algébrica dos produtos dos elementos de uma matriz quadrada de n^2 elementos, obtidos permutando-se de todos os modos possíveis os índices superiores dos elementos da diagonal principal, fixados os inferiores e, admitindo-se que esses produtos sejam positivos e negativos, conforme os índices superiores de seus fatores formarem uma permutação de classe par ou ímpar, respectivamente. Se o $\det(A) = 0$, o sistema é dito singular.

Matrizes singulares	Matrizes não singulares
- O conjunto de equações não tem uma única solução.	- O conjunto de equações tem uma única solução.
- A eliminação de Gauss não pode evitar um zero na diagonal.	- A eliminação de Gauss prossegue sem um zero na diagonal.
- Não é inversível	- Possui matriz inversa
- As linha e colunas formam vetores linearmente dependentes	- As linhas e colunas formam vetores linearmente independentes.

Figura 1.2 - Propriedades equivalentes de matrizes singulares e não singulares

Freqüentemente o teste para determinar se a matriz possui inversa é baseado no conceito do determinante. A matriz A é inversível se, e somente se, o $\det(A) \neq 0$, então é possível expressar a solução de $AX = B$ em termos de determinante, pela regra de Cramer.

$$A^{-1} = \frac{1}{\det(A)} [Adj(A)]^T \quad (1.4)$$

Se A e B são matrizes quadradas, tais que $AB = BA = I$, diz-se que B é a inversa de A e escreve-se A^{-1} . A matriz inversa de A pode ser calculada pelo produto do inverso do determinante de A pela transposta da matriz adjunta de A , como descrito na fórmula (1.4).

1.2 Revisão da Aritmética Matricial Intervalar

Neste item é apresentada uma revisão da aritmética matricial intervalar, onde são revisadas as noções de igualdade, soma/subtração, multiplicação, inclusão entre matrizes e/ou vetores intervalares entre outros. Este item é importante para compreensão dos

métodos intervalares para a resolução de sistemas de equações lineares, especialmente aqueles baseados nas propriedades algébricas, conhecidos como métodos diretos. Para maiores detalhes ver [ALE83 e CLA89].

Os componentes das matrizes e vetores intervalares são representados por $A^I=(A_{ij})$ e $a^I=(A_i)$, respectivamente. Uma matriz intervalar, onde todos os componentes são intervalos degenerados pontuais é chamada de matriz pontual. O vetor pontual é definido de maneira similar.

Duas matrizes intervalares $m \times n$ ($A^I=(A_{ij})$ e $B^I=(B_{ik})$) são iguais, isto é, $A^I=B^I$, se e somente se houver igualdades entre todos os componentes correspondentes das matrizes, ou seja, $A^I=B^I \Leftrightarrow A_{ij}=B_{ij}$, $1 \leq i \leq m$, $1 \leq j \leq n$.

Sejam $A^I=(A_{ij})$ e $B^I=(B_{ij})$ duas matrizes intervalares $m \times n$, a inclusão é definida por $A^I \subseteq B^I \Leftrightarrow A_{ij} \subseteq B_{ij}$, $1 \leq i \leq m$, $1 \leq j \leq n$.

A soma/subtração de duas matrizes intervalares $m \times n$, $A^I=(A_{ij})$ e $B^I=(B_{ij})$, é dada por $A^I \pm B^I = (A_{ij} \pm B_{ij})$.

Seja $A^I=(A_{ij})$ uma matriz intervalar $m \times r$ e $B^I=(B_{ij})$ uma matriz intervalar $r \times n$, então $A^I B^I := \left(\sum_{v=1}^r A_{iv} B_{vj} \right)$ e, de uma maneira particular, tem-se $A^I=(A_{ij})$ e um vetor intervalar $u^I=(U_i)$, então $A^I u^I := \left(\sum_{v=1}^r A_{iv} U_v \right)$.

Seja $A^I=(A_{ij})$ e X um intervalo, então $XA^I=A^IX:=(XA_{ij})$.

Seja A^I e B^I matrizes intervalares e C um vetor pontual, então as seguintes relações são válidas:

- a) $\{A \pm B \mid A \in A^I, B \in B^I\} = A^I \pm B^I$,
- b) $\{AC \mid A \in A^I\} = A^I C$

O conjunto das matrizes intervalares é fechado com respeito às operações de soma/subtração. O conjunto das matrizes reais ou complexas é isomórfico às correspondentes matrizes pontuais. A seguir são apresentadas algumas propriedades para as operações acima.

Seja A^I, B^I, C^I matrizes intervalares e A, B, C, O, P matrizes pontuais, então:

- (a) $A^I + B^I = B^I + A^I$ (comutatividade da adição)
- (b) $A^I + (B^I + C^I) = (A^I + B^I) + C^I$ (associatividade da adição)
- (c) $A^I + O = O + A^I = A^I$, onde O é a matriz nula (elemento neutro da adição)
- (d) $A^I P = P A^I = A^I$, onde P é a matriz identidade (elemento neutro da multiplicação)

- (e) $\begin{cases} (A^I+B^I)C^I \subseteq A^I C^I + B^I C^I, \\ C^I(A^I+B^I) \subseteq C^I A^I + C^I B^I \end{cases}$ (subdistributividade)
- (f) $\begin{cases} (A^I+B^I)C = A^I C + B^I C, \\ C(A^I+B^I) = CA^I + CB^I \end{cases}$ (distributividade)
- (g) $\begin{cases} A^I(BC) \subseteq (A^I B)C, \\ (A^I B)C^I \subseteq A^I(B^I C^I) \text{ se } C^I = -C^I \\ A^I(B^I C) = (A^I B^I)C, \\ A^I(B^I C^I) = (A^I B^I)C^I \text{ se } B^I = -B^I \text{ e } C^I = -C^I \end{cases}$ (associatividade)
(da)
(multiplicação)

Sendo A^I, A^I, B^I, B^I matrizes intervalares e X^I, Y^I vetores intervalares. tem-se que, $A^I \subseteq B^I$, $A^I \subseteq B^I$ e $X^I \subseteq Y^I$.

Se $A^I = (A_{ij})$ uma matriz intervalar, então a matriz diâmetro é definida por $d(A^I) := (d(A_{ij}))$, sendo $d(A^I)$ uma matriz real não negativa.

A matriz valor absoluto é definida por $|A^I| := (|A_{ij}|)$, sendo também $|A^I|$ uma matriz real não negativa.

Sendo $A^I = (A_{ij})$ e $B^I = (B_{ij})$, a matriz distância ou a distância entre A^I e B^I é dada por $q(A^I, B^I) := (q(A_{ij}, B_{ij}))$.

Provas das operações apresentadas neste item poderão ser encontrados em [ALE83].

1.3 Revisão de Sistemas de Equações Lineares

Um sistema de equações lineares pode ser escrito na forma:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n = b_3 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{mn}x_n = b_n \end{cases} \quad (1.5)$$

onde n é o número de incógnitas do sistema, também denominado de ordem do sistema.

Todo sistema de equações pode ser reescrito na forma de uma matriz de coeficientes A multiplicada pelo vetor das incógnitas X , resultando no vetor dos termos independentes B , ou seja, $AX = B$, onde:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \dots \\ b_n \end{bmatrix} \quad (1.6)$$

Um problema típico de resolução de equações é: dado um sistema de equações lineares de ordem $n \times n$, deseja-se determinar os valores das incógnitas que multiplicados por A produzem o vetor B .

Uma solução para um sistema linear é um conjunto de valores das n incógnitas que satisfaz a todas as equações. Um sistema linear pode ser classificado (ver figura 1.3) quanto ao número de soluções em: **compatível**, quando apresenta solução e; **incompatível** caso contrário.

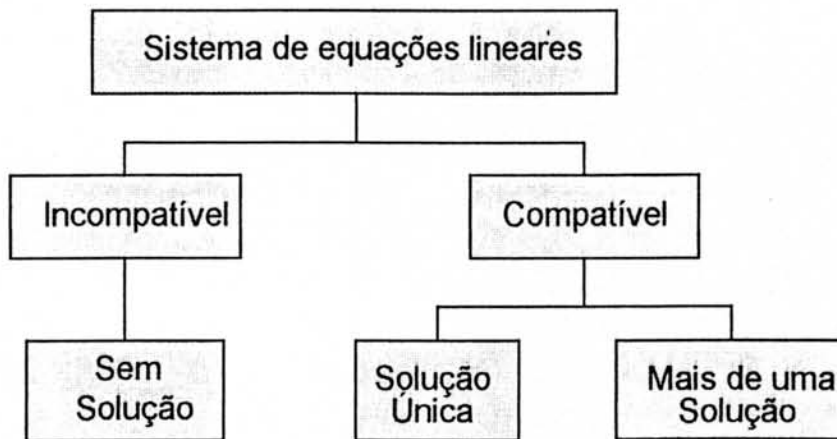


Figura 1.3 - Classificação de um Sistema Linear quanto ao número de soluções

O sistema é dito **homogêneo**, quando o vetor dos termos independentes é nulo, ou seja, $b_i = 0$, para todo $i=1,2, \dots, n$. Todo sistema homogêneo é compatível, pois admite, pelo menos, a solução trivial, que tem como vetor solução X , o vetor nulo.

Os sistemas compatíveis podem ainda ser classificados em: **determinados**, quando apresenta uma única solução e **indeterminados**, caso contrário. Ou seja, se o conjunto solução é único, o sistema é determinado ou não singular. Se existirem uma infinidade de soluções, o sistema é indeterminado.

Uma das grandes dificuldades na resolução de sistemas de equações é o problema da instabilidade, ocasionado pela natureza do problema ou pelas limitações e erros de aproximações, que estão ligados aos algoritmos e máquinas usadas para resolver o problema. A instabilidade pode resultar em situações indesejáveis. A influência que a propagação dos erros nos dados de entrada exerce sobre o resultado é descrita pela noção de condição do problema ou condicionamento.

Um sistema é dito estável ou bem condicionado quando se introduz pequenas variações em seus dados de entrada e, com isso, são produzidas pequenas variações em seu resultado final. Um sistema é dito instável ou mal condicionado, quando se introduz pequenas variações em seus dados de entrada e, por consequência, são produzidas grandes variações no resultado. Para maiores detalhes ver [HÖL94, HÖL94a e HÖL94b].

1.4 Aplicabilidade de Sistemas de Equações Lineares

Uma variedade de problemas de engenharia pode ser resolvido através de sistemas lineares, tais como: a determinação de potenciais em certas redes elétricas, o cálculo do *stress* numa armação de construção ou de uma estrutura de ponte, o cálculo do padrão de escoamento num sistema hidráulico com ramos interconectados ou o cálculo das estimativas das concentrações de reagentes sujeitos a simultâneas reações químicas entre outras.

Alguns exemplos de problemas que envolvem a resolução de sistemas de equações lineares podem ser encontrados em [HÖL96], que ilustram a aplicação de sistemas lineares na solução destes problemas, os quais podem ser modelados e resolvidos por essa técnica.

1.5 Descrição dos Capítulos

Neste capítulo foram apresentados conceitos básicos que são abordados durante o restante do trabalho. Primeiramente, foi apresentada a motivação, os aspectos teóricos e práticos que originaram e possibilitaram a realização deste trabalho. A seguir, foi apresentada uma revisão sobre matrizes, álgebra matricial pontual e intervalar, sistemas de equações lineares e sua aplicabilidade.

No capítulo dois são apresentados os métodos intervalares para a resolução de sistemas de equações lineares. Esses métodos são divididos em três grupos e são apresentadas suas características e seus respectivos algoritmos. Ao final do capítulo é apresentada uma análise desses métodos do ponto de vista de sua implementação e resultados gerados.

No capítulo três é proposta a biblioteca aplicativa intervalar para a resolução de sistemas de equações lineares, *selint.p*, desenvolvida para ambiente PC e utilizando o compilador Pascal-XSC. É descrita a estrutura da biblioteca *selint.p*, bem como suas características, módulos, padrão de documentação e o manual de utilização.

O capítulo quatro é de extrema importância para este trabalho. Neste capítulo são apresentados os resultados obtidos com a utilização da biblioteca proposta. Foram feitas comparações quanto a qualidade dos resultados obtidos entre os diferentes métodos (resultado pontual \times intervalar). Foram feitos testes comparativos entre os resultados intervalares obtidos com a biblioteca aplicativa intervalar *selint.p* com os resultados intervalares da biblioteca aplicativa intervalar *libselint.a* que vem sendo desenvolvida no ambiente do supercomputador Cray Y-MP (em FORTRAN 90) pelo Grupo de Matemática Computacional da UFRGS. Através desses testes, pretende-se realizar uma análise do desempenho qualitativo da biblioteca aplicativa intervalar *selint.p*, em termos de exatidão e confiabilidade dos resultados.

O capítulo cinco é destinado as conclusões do trabalho. É apresentado, primeiramente, um resumo do mesmo, depois são apresentadas as conclusões obtidas com este trabalho e, por fim, são apresentadas propostas para continuidade deste trabalho.

O anexo 1 deste trabalho apresentam os algoritmos dos métodos intervalares que foram apresentados no capítulo 2 (Métodos Intervalares), algoritmos que foram utilizados na implementação da biblioteca intervalar *selint.p*. O anexo 2 apresenta uma lista de sistemas de equações lineares que foram resolvidos e seus resultados utilizados para realizar a análise de desempenho quantitativo da biblioteca *selint.p*.

2 Métodos Intervalares

2.1 Introdução

A matemática intervalar tem sido utilizada em diversas áreas, para resolver sistemas de equações lineares e não lineares, equações diferenciais ordinárias e parciais, equações integrais e problemas de otimização, como pode ser encontrado em [HAM93, KRÄ96]. Para cada uma dessas classes de problemas numéricos, o emprego da matemática intervalar tem sido acompanhada pelo desenvolvimento de novas técnicas, as quais vão além da mera substituição dos coeficientes reais por intervalos e do uso de operações intervalares.

A extensão ingênua, por assim dizer, de métodos pontuais em métodos intervalares tem mostrado ser ineficiente, pois não há convergência, como foi observado, por exemplo, na extensão ingênua do método de Newton para o cálculo de raízes reais (como demonstrado em [DIV94]).

As técnicas para o cálculo numérico com verificação automática do resultado utilizam o produto escalar ótimo e a aritmética intervalar como ferramentas essenciais, além da aritmética de ponto-flutuante simples. A aritmética intervalar permite o cálculo de extremos seguros para as soluções de um problema. Obtém-se alta exatidão por meio do produto escalar ótimo. A combinação destas, proporciona um grande avanço na análise numérica.

Uma mera aplicação da aritmética intervalar levará a extremos confiáveis. Entretanto, estes podem ser tão grandes que eles são realmente inúteis. Este efeito já era observado há quase vinte anos atrás por analistas numéricos, que muitas vezes criaram rejeições a essa técnica [HÖL95].

2.2 Métodos Intervalares

Os sistemas de equações lineares podem envolver matrizes de coeficientes densas ou esparsas. Observa-se, em geral, que os métodos diretos pontuais para a solução dos sistemas lineares são mais adequados a sistemas cuja a matriz de coeficientes é densa, enquanto que sistemas cuja matriz de coeficientes é esparsa, do tipo banda ou tridiagonal, são resolvidos mais eficientemente por métodos iterativos. Existe uma grande variedade de métodos, cujo desempenho é melhor para determinado tipo de sistema, cuja matriz possui determinada propriedade, como por exemplo simétrica e definida positiva. Entretanto, devido a instabilidade (ou estabilidade) do problema, a erros de arredondamentos, principalmente resultantes do produto escalar presente nas operações de multiplicação de

vetores, de matrizes e de matrizes por vetores; e ao efeito acumulativo, resultados diferentes da solução exata são produzidos.

TEMPO

A extensão intervalar destes métodos pontuais não é muito simples e o cálculo da solução por métodos intervalares pode ser dispendioso, uma vez que se está tratando com vetores e matrizes de intervalos. Uma alternativa econômica e bastante utilizada é calcular uma aproximação pontual e, a partir desta melhorá-la através de métodos intervalares. Nesta idéia baseiam-se os métodos híbridos, que constituem uma das metodologias identificadas dos métodos intervalares.

Entre os métodos intervalares levantados, foram identificadas características que permitiram a reunião dos métodos em três classes ou grupos de métodos. O primeiro grupo consiste de métodos baseados em operações algébricas matriciais intervalares, são a versão intervalar de métodos diretos. O segundo grupo é baseado em refinamentos, são métodos híbridos, onde uma aproximação da solução exata é calculada por algum método pontual. Então a solução é refinada através do uso da matriz inversa real ou intervalar, produzindo uma solução intervalar contendo a solução exata, desta forma se tem limites confiáveis de forma não dispendiosa. A busca de uma solução exata pode ser feita através do refinamento do erro de uma aproximação ou do erro da matriz inversa. Por fim, têm-se os métodos intervalares baseados em iterações, são versões intervalares dos métodos iterativos pontuais, onde o conceito de iteração está baseado em inclusões.

Através desta divisão em grupos, os métodos intervalares serão caracterizados e também foram elaborados seus respectivos algoritmos (algoritmos descritos no Anexo 1), os quais serão implementados dando origem a biblioteca aplicativa intervalar para a resolução de sistemas de equações lineares *selint.p* (ver capítulo 3 Biblioteca Aplicativa Intervalar - Módulo Sistemas).

2.2.1 Resolução de Sistemas de Equações Lineares de Ordem 1

Uma equação algébrica, para fins deste trabalho, será considerada como um caso particular de sistema de equações, onde a ordem do sistema é um, ou seja, um sistema com uma única equação e com uma única variável. A resolução de equações por métodos pontuais pode ser feita por vários tipos de métodos, entre estes se pode destacar os métodos de quebra, os métodos iterativos e os métodos híbridos (como classificados por Diverio em [DIV86]).

A resolução de equações por métodos de quebra, como o conhecido método da Bisseção, necessita de um *intervalo* inicial que contenha um zero da equação. Este intervalo é "quebrado" por alguma regra (como por exemplo, pela metade), de forma a diminuir o *intervalo*, até que se esteja suficientemente próximo do zero da equação. Por fim, tem-se um *intervalo* que contém a raiz cujos extremos diferem apenas nas últimas casas da

mantissa. Observa-se que o conceito de *intervalo* é utilizado, mas sem a utilização de propriedades e operações intervalares.

Os métodos híbridos, como o método híbrido C (definido por Cláudio, D.M em [CLA89] e, também descrito em [DIV90]), também utiliza a idéia de *intervalo* que contém uma raiz, o qual é refinado através de operações compostas sobre os extremos que envolvem retas tangentes e secantes interseccionando o eixo x. Mais uma vez é utilizado o conceito de *intervalo*, sem operações intervalares.

2.2.1.1 Métodos Algébricos

Neste item são apresentados métodos algébricos para a resolução de um sistema de equações lineares de ordem 1. São apresentados quatro métodos de resolução, de acordo com o tipo de equação a ser resolvida. Para maiores detalhes, demonstrações e exemplos ver [KOR94].

Sejam os intervalos $A = [a1, a2]$, $B = [b1, b2]$, $C = [c1, c2]$, $D = [d1, d2]$ e $X = [x1, x2]$ intervalos reais, então as seguintes equações intervalares possuem o zero contido no intervalo solução respectivo:

a) Equação $A + X = B$

$$\text{Intervalo solução: } X := [(b1 - a2), (b2 - a1)] \quad (2.1)$$

b) Equação $AX + B = C$

Intervalo solução: $X := [\min(x), \max(x)]$, onde

$$x := \left\{ \frac{(c1 - b2)}{a2}, \frac{(c1 - b2)}{a1}, \frac{(c2 - b1)}{a2}, \frac{(c2 - b1)}{a1} \right\} \quad (2.2)$$

c) Equação $AX + BX = C$

Intervalo solução: $X = [\min(x), \max(x)]$, onde

$$x := \left\{ \frac{c1}{(a1 + b1)}, \frac{c2}{(a2 + b2)}, \frac{c1}{(a2 + b2)}, \frac{c2}{(a1 + b1)} \right\} \quad (2.3)$$

d) Equação $AX + B = CX + D$

Intervalo solução: $X = [\min(x), \max(x)]$, onde

$$x := \left\{ \frac{(d1 - b2)}{(a2 - c1)}, \frac{(d1 - b2)}{(a1 - c2)}, \frac{(d2 - b1)}{(a2 - c1)}, \frac{(d2 - b1)}{(a1 - c2)} \right\} \quad (2.4)$$

2.2.1.2 Método de Newton Intervalar

O método iterativo mais conhecido é, sem dúvida o método de Newton-Raphson, também conhecido como método da tangente. É um método iterativo. Algebricamente, o método parte de um valor inicial, que aplicado a uma função de iteração, gera a nova aproximação. Pelo aplicar sucessivo dos valores produzidos pela função de iteração, tem-se uma seqüência de valores que, muitas vezes, é convergente. Quando converge, o limite é uma aproximação de um dos zeros da equação. A função de iteração do método de Newton é dada pela fórmula (2.5), onde se tem que o valor atual é produzido pelo valor anterior, menos o quociente do valor da função no ponto anterior pela derivada da função no ponto anterior.

$$x^{k+1} := x^k - f(x^k) / f'(x^k) \quad (2.5)$$

Aparentemente, o método de Newton não se utiliza do conceito de intervalo. Mas, como método iterativo que é, a convergência é determinada pela verificação de certas propriedades em uma vizinhança, a qual pode ser vista como um intervalo. O método de Newton é estendido a raízes múltiplas e complexas e, pode também, ser estendido a cálculos intervalares.

Uma generalização ingênua do método de Newton para o uso da aritmética intervalar seria converter os valores reais por intervalos e as avaliações da função e da derivada, por extensões intervalares da mesma. Ou seja, $X^0 \subseteq X$ e para f e f' sejam F e F' avaliações intervalares com $0 \notin F'(X^0)$. Seja α um zero de $F(X)$, α é única no intervalo.

$$X^{k+1} := X^k - F(X^k) / F'(X^k) \quad (2.6)$$

Este método é divergente para todo intervalo inicial $X^0 \subseteq X$, com $\alpha \in X$, $X^0 \neq \alpha$, pois se for verificado o tamanho do intervalo, pelo diâmetro, nota-se que o intervalo cresce a cada iteração, ou seja $d(X^{k+1}) \geq d(X^k)$, logo a seqüência de intervalos $\{ X^k \}$ é divergente.

Para ilustrar esta passagem, a seguir é considerado uma das versões do método de Newton Intervalar, conhecido como método de Newton Intervalar Simplificado [DIV91].

Sejam X e X^0 intervalos, se $f \in F(X)$ e $X^0 \subseteq X$. α é raiz de f determinada em X , e se α é única; M é a avaliação intervalar F' em X^0 . Define-se o operador Newtoniano pela fórmula 2.7, onde N_i e M são intervalos.

$$N_i(X) = X - f(X)/M, \text{ onde } 0 \notin M. \quad (2.7)$$

O método de Newton Intervalar simplificado é dado por:

$$X^{k+1} := N_i(X^k) \cap X^k \text{ para } k=1,2,\dots,n \quad (2.8)$$

Observa-se a propriedade que se $x^0 \in X^0$ e $N_i(x^0) \cap X^0 = \emptyset$, então não existe raiz real de f em X^0 . Observa-se, ainda, que as variações na definição do operador Newtoniano é que definem as versões do método de Newton intervalar. Na verdade, pode-se mesmo permitir que 0 esteja contido em $F'(X^0)$ se for usada a aritmética intervalar estendida como é mostrado no exemplo a seguir. Considere $f(x) = -2.001 + 3x - x^3$ sendo $f'(x) = 3(1 - x^2)$. Pode-se usar a extensão intervalar $F'(X) = 3(1 - X^2)$; Se for feito $X^0 = [-3,3]$, obtém-se $F'(X^0) = [-24,3]$ e, em aritmética intervalar estendida, obtém-se:

$$F'(X^0)^{-1} = [-\infty, -1/24] \cup [1/3, +\infty] \quad (2.9)$$

Assim o intervalo X^0 é mapeado aqui para a união de dois intervalos ilimitados e disjuntos. Deste ponto segue a produção de uma (ou mais) seqüências de intervalos usando a fórmula (2.8) começando com outro dos intervalos na união (2.9). Neste exemplo, se o segundo intervalo for escolhido na união (2.9), a seqüência de intervalos gerados por (2.8) terminará em um número finito de passos com o intervalo vazio desde que não exista zeros de $f(x)$ acima de $1/3$. Inversamente, desde que um zero de $f(x)$ esteja em $N_i(X)$ sempre que está em X , pode-se dizer de uma intersecção vazia $X \cap N(X)$ que não haja zero em X . Se escolhe-se o primeiro intervalo em (2.9), então encontra-se, usando (2.8), uma seqüência aninhada de intervalos.

$X^0 = [-3,3]$
$X^1 = [-3,-0.083375]$
$X^2 = [-3,-1.66526]$
$X^3 = [-2.17875,-1.63830]$
$X^4 = [-2.17875,-2.06189]$
$X^5 = [-2.0162,-2.0003]$
$X^6 = [-2.00024,-2.00006]$
$X^7 = [-2.000112,-2.000110]$

Figura 2.1 - Seqüência aninhada de intervalos

Durante os cálculos, percebe-se que $N(X^4) \subseteq X^4$, com $0 \notin F'(X^4)$. Segue que $f(x)$ tem um zero em X^k para todos k nesta seqüência. Se $0 \in F'(X)$ e $N_i(X) \subseteq X$ com $N_i(X) =$

$m(X) - f(m(X)) / F'(X)$, então $f(x)$ tem um zero em X . Em contraste ao comportamento da seqüência encontrada acima, o método de Newton comum, $x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)}$ gera uma seqüência muito errônea para o exemplo, a menos que x^0 seja menor que -1 . Por exemplo, com $x^0 = 0$, o método de Newton gera a seqüência:

$x^1 = 0.667$
$x^2 = 0.84518716$
$x^3 = 0.92592529$
$x^4 = 0.965774$
$x^5 = 0.98794069$
$x^6 = 1.0078932$
$x^7 = 0.98291958$
$x^8 = 1.0013261$
$x^9 = 0.87506664$
$x^{10} = 0.94034361$

Figura 2.2 - Seqüência de intervalos gerado pelo Método de Newton pontual

2.2.2 Métodos Intervalares Baseados em Operações Algébricas

Os métodos intervalares baseados em propriedades e operações algébricas, tendem a ser uma extensão dos métodos diretos pontuais de resolução de sistemas. Eles são estendidos a métodos intervalares pelo uso de operações e argumentos intervalares. Um exemplo é dado a seguir, onde o sistema de ordem dois é mal condicionado. Este exemplo foi resolvido em [FER94] e é transcrito para cá, com o objetivo de ilustrar o uso de intervalos para gerar a solução do sistema.

Exemplo: Considere o seguinte sistema mal-condicionado abaixo,

$$\begin{cases} 2.000x + 3.001y = 1.000 \\ 0.6667x + 1.000y = 0.3333 \end{cases} \quad (2.10)$$

Se os cálculos forem feitos usando aritmética intervalar arredondada com n dígitos decimais, para $n=4, 5, 6, 7, 8$ e 9 , pode-se substituir o termo x na segunda equação, obtendo-se $(1.000 - (0.6667 / 2.000)(3.001))y = 0.3333 - (0.6667 / 2.000)(1.000)$. Os seguintes resultados foram obtidos para y .

- para $n=4$: $0.6667/2.000 \in [0.3333, 0.3334]$
 $(0.6667/2.000)(3.001) \in [1.000, 1.001]$,
 $0.3333 - (0.6667/2.0000)(1.000) \in [-0.0010, 0]$,
 $(1.000 - (0.6667/2.000)(3.001)) \in [-0.0010, 0]$,
 $y \in [0, +\infty)$ (sem limite em y)
- para $n=5$: $0.667/2.000 \in [0.33335, 0.33335]$,
 $(0.6667/2.000)(3.001) \in [1.0003, 1.0004]$,
 $0.3333 - (0.6667/2.000)(1.000) \in [-0.00005, -0.00005]$,
 $(1.000 - (0.6667/2.000)(3.001)) \in [-0.00040, -0.00030]$,
 $y \in [0.12500, 0.16667]$;
- para $n=6$: $y \in [0.128205, 0.131579]$;
- para $n=7$: $y \in [0.1302083, 0.1305484]$;
- para $n=8$: $y \in [0.13041210, 0.13044613]$;
- para $n=9$: $y \in [0.130429111, 0.130429112]$.

O repentino aumento na exatidão que ocorre entre os $n=8$ e $n=9$ é explicado pelo fato que, para $n>8$ o erro de arredondamento restante é gerado na divisão final por y . Neste exemplo, o tamanho do intervalo calculado diminui na ordem de 10^{-1} , a medida que n cresce para $n>5$.

2.2.2.1 Métodos de Hansen

O método de Hansen descrito em [ALE83, HAN65, HAN67, HAN69] se utiliza de propriedades dos intervalos para calcular a solução. Ele estuda o comportamento de inequações em cada quadrante para produzir a solução composta, que pode sofrer o *efeito envolvente*, ou seja, obter o menor intervalo retangular que contenha a solução,

Considere o sistema de equações $AX = B$ onde $B = (b_i)$ é um vetor real de ordem n e $A = (a_{ij})$ é uma matriz real não singular de ordem n . O vetor solução X é o vetor $A^{-1}B$. Suponha, contudo, que A e B são sujeitos a erros. Suponha que se sabe somente que $a_{ij} \in A_{ij} = [a1_{ij}, a2_{ij}]$ e $b_i \in B_i = [b1_i, b2_i]$ para $i, j=1, \dots, n$. Denote $A^I = (A_{ij})$ e $B^I = (B_i)$, deseja-se saber o conjunto de soluções $X = \{X: AX = B, A \in A^I, B \in B^I\}$ para a equação $A^I X = B^I$.

Hansen e Smith¹ discutiram métodos para computar um vetor X^I contendo X . Um vetor $X^I = [x1, x2]$ define uma região em um espaço n -dimensional limitado pelos planos

¹ em [HAN65, HAN67]

$X_i = x1_i$ e $X_i = x2_i$ ($i = 1, \dots, n$). Em geral, o conjunto X não é uma região limitada pelos planos paralelos aos eixos coordenados e então o menor vetor intervalar X^I não é igual a X . (Pelo menor vetor intervalar, entende-se o vetor na qual os elementos são todos os menores possíveis).

Contudo o menor X^I é importante. Será mostrado como obter ambos os limites (superiores e inferiores) do menor $X^I \supset X$. Assume-se que pelo menos um elemento de A^I ou B^I é um intervalo de tamanho diferente de 0.

Agora considera-se o conjunto X . Pode-se reescrever $A^I X = B^I$ por

$$\sum_{j=1}^n A_{ij} X_j = B_i \quad (i = 1, \dots, n). \quad (2.11)$$

Suponha que alguns pontos fixos $x \in X$ localizam-se no quadrante positivo. Para este X , (2.11) pode ser escrito como

$$\sum_{j=1}^n a1_{ij} X_j, \sum_{j=1}^n a2_{ij} X_j = B_i \quad (i = 1, \dots, n) \quad (2.12)$$

Existe $A \in A^I$ e $B \in B^I$ tal que $AX=B$, para este x fixo, contanto que os intervalos à esquerda e à direita de (2.12) cruzem-se para cada $i=1, \dots, n$. Isto é, $x \in X$ (se $x_j \geq 0$ para $j=1, \dots, n$) sob condição que:

$$\sum_{j=1}^n a1_{ij} x_j \leq b2_i, \sum_{j=1}^n a2_{ij} x_j \geq b1_i \quad (i = 1, \dots, n). \quad (2.13)$$

Seja $x_j \leq 0$ para algum valor de j , então $A_{ij} X_j = [a2_{ij} x_j, a1_{ij} x_j]$.

De acordo com o conjunto de equações lineares do sistema envolvido no cálculo, obtém-se a solução para cada quadrante e, com isso, constrói-se, através da reunião das áreas delimitadas em cada quadrante, uma figura geométrica que representa a solução intervalar do sistema. O vetor intervalar X^I será limitado pelos pontos de máximo e mínimo da região resultante. A seguir será apresentado um exemplo.

Considere as equações

$$\begin{cases} [2,3]x + [0,1]y = [0,120] \\ [1,2]x + [2,3]y = [60,240] \end{cases} \quad (2.14)$$

Mesmo no caso de duas dimensões, o conjunto X^I não é particularmente fácil de se representar. Obviamente, em dimensões maiores, X^I é difícil de representar em geral. Pode-se, contudo, facilmente representar o menor paralelepípedo contendo X^I tendo lados paralelos aos eixos coordenados. No exemplo acima, este paralelepípedo é limitado pelos planos $x = -120, x = 90, y = -60, y = 240$. Pode-se então representá-lo por

$$\text{Solução pelo método de Hansen: } X^I = \begin{bmatrix} [-120, 90] \\ [-60, 240] \end{bmatrix}$$

Representação Gráfica:

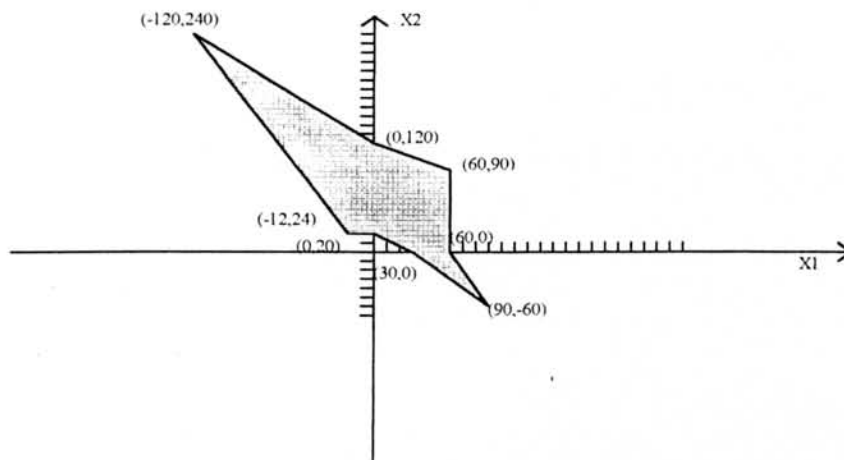


Figura 2.3 - Conjunto solução X^I - área total

O Método de Hansen para o caso geral é apresentado a seguir (método descrito em [HAN69]). Tendo um sistema de equações lineares $AX = B$, onde $A \in A^I, B \in B^I$, calcula-se o X^I (vetor que contém o vetor solução X). Através deste cálculo formam-se dois sistemas de equações lineares, os quais serão utilizados para achar o vetor X^I final. Estes sistemas são formados da seguinte maneira:

Dado $A_{ij} = [a1_{ij}, a2_{ij}]$ forma-se as matrizes P^I e Q^I . Os elementos destas duas matrizes são determinadas de acordo com (2.15) e (2.16)

$$P_{ij} = \begin{cases} a1_{ij} & \text{se } \frac{\partial x_r}{\partial a_{ij}} \geq 0 \quad \forall A \in A^I, b \in B^I \\ a2_{ij} & \text{se } \frac{\partial x_r}{\partial a_{ij}} \leq 0 \quad \forall A \in A^I, b \in B^I \\ A_{ij} & \text{outra condição} \end{cases} \quad (2.15)$$

$$Q_{ij} = \begin{cases} a_{2ij} \text{ se } \frac{\partial x_r}{\partial a_{ij}} \geq 0 \quad \forall A \in A^I, b \in B^I \\ a_{1ij} \text{ se } \frac{\partial x_r}{\partial a_{ij}} \leq 0 \quad \forall A \in A^I, b \in B^I \\ A_{ij} \text{ outra condiç \~{a}o} \end{cases} \quad (2.16)$$

Dado que $B_i = [b_{1i}, b_{2i}]$ forma-se os vetores C^I e D^I . Os elementos destes dois vetores são determinados de acordo com (2.17) e (2.18).

$$C_i = \begin{cases} b_{1ij} \text{ se } \frac{\partial x_r}{\partial b_i} \geq 0 \quad \forall A \in A^I, b \in B^I \\ b_{2ij} \text{ se } \frac{\partial x_r}{\partial b_i} \leq 0 \quad \forall A \in A^I, b \in B^I \\ B_i \text{ outra condiç \~{a}o} \end{cases} \quad (2.17)$$

$$D_i = \begin{cases} b_{2ij} \text{ se } \frac{\partial x_r}{\partial b_i} \geq 0 \quad \forall A \in A^I, b \in B^I \\ b_{1ij} \text{ se } \frac{\partial x_r}{\partial b_i} \leq 0 \quad \forall A \in A^I, b \in B^I \\ B_i \text{ outra condiç \~{a}o} \end{cases} \quad (2.18)$$

Dado que I = matriz identidade e $W = A^{-1}$, utiliza-se os elementos destas duas matrizes para substituir os elementos que utilizam derivadas parciais. Com isto tem-se

$$\frac{\partial x_r}{\partial a_{ij}} = -w_{ri} x_j \quad \text{e} \quad \frac{\partial x_r}{\partial b_i} = w_{ri} \quad (2.19)$$

Obtidos estes dois conjuntos de sistema de equações lineares, resolve-se o conjunto de equações definido em (2.20), utilizando os métodos apresentados abaixo ou outro da preferência do usuário.

$$\begin{aligned} P^I Y^I &= C^I \\ Q^I Z^I &= D^I \end{aligned} \quad (2.20)$$

Com a resolução destes dois sistemas pode-se determinar o X^I , que é formado da seguinte maneira:

$$X^I = [|\text{limite inferior do intervalo da solução de } Y^I|, |\text{limite superior do intervalo da solução de } Z^I|] \quad (2.21)$$

Existem várias versões de métodos que são utilizadas para calcular o vetor intervalar da solução. Estes métodos foram apresentados por Hansen, em [HAN67], e são descritos a seguir. Deve-se salientar que estes métodos poderão ser utilizados separadamente para obter-se a solução de um sistema de equações lineares ou poderão ser utilizados como parte do Método de Hansen para o caso geral (método descrito acima).

- **Método Hansen1:** este método realiza os passos de uma rotina de resolução de equações lineares usando a aritmética intervalar. Neste método é usada a fatorização triangular (descrita em [HAN65]), embora qualquer "bom" método seja suficiente. De acordo com Hansen esses métodos tornam-se inadequados quando usados diretamente. Este método é usado como parte de outros métodos, mas somente depois que o problema tenha sido modificado de tal maneira que as dificuldades inerentes forem minimizadas.
- **Método Hansen2:** Dadas as matrizes intervalares A^I e B^I , calcula-se a matriz centro de A^I (A_c) e a matriz centro de B^I (B_c). Resolve-se $A_c X = B_c$ usando aritmética com arredondamento e obtém-se uma solução aproximada Y . Usando aritmética intervalar, calcula-se $B^I - A^I Y$ e resolve-se $A^I Z^I = B^I - A^I Y$, obtendo Z^I . A solução da equação original está contida em X^I , $X^I = Z^I + Y$.
- **Método Hansen3:** Dadas as matrizes intervalares A^I e B^I , calcula-se a matriz inversa pontual de A_c (matriz centro de A^I). Em seguida a matriz inversa intervalar de A^I . Por fim, resolve-se $X^I = (A^I)^{-1} B^I$, e com isso X^I contém a solução do sistema.
- **Método Hansen4:** Dadas as matrizes intervalares A^I e B^I , calcula-se a matriz centro de A^I (A_c) e a matriz inversa pontual de A_c . Usando a aritmética intervalar encontra-se $(A_c)^{-1} \cdot A^I$ e $(A_c)^{-1} \cdot B^I$. Finalmente resolve-se $(A_c)^{-1} \cdot A^I X^I = (A_c)^{-1} \cdot B^I$, obtendo com isso a solução do sistema.
- **Método Hansen5:** Este método combina o método Hansen2 e Hansen4. Encontra-se Y e $(A_c)^{-1}$. Usando a aritmética intervalar, $(A_c)^{-1} \cdot A^I$ e $(A_c)^{-1} (B^I - A^I Y)$. Resolve-se $(A_c)^{-1} \cdot A^I Z^I = (A_c)^{-1} (B^I - A^I Y)$, obtendo Z^I . Então $X^I = Y + Z^I$.

2.2.2.2 Método de Eliminação de Gauss

A estrutura do método de Eliminação de Gauss Pontual, onde a matriz A é concatenada com o vetor dos termos independentes, para ser triangularizada, através de operações elementares é, em linhas gerais, mantida na passagem para a sua versão intervalar. As mudanças que ocorreram no algoritmo de implementação pontual para o intervalar

deram-se apenas no aspecto que correspondia as matrizes/vetores que são utilizados, passando de matrizes/vetores reais para matrizes/vetores reais intervalares.

Seendo A^I uma matriz intervalar e B^I um vetor intervalar, assume-se que existe a inversa A^{-1} para todo $A \in A^I$. Deseja-se encontrar o conjunto $\{X | AX=B, A \in A^I \text{ e } B \in B^I\}$. Este vetor intervalar solução pode ser determinado pelo algoritmo de Gauss para sistemas de equações lineares com coeficientes intervalares. A estrutura do método de eliminação de Gauss é mantida, ou seja, a matriz A^I é concatenada ao vetor dos termos independentes B^I produzindo uma matriz de ordem $n \times n+1$, como em (2.22)

$$\begin{array}{cccc} A_{11} & \cdots & A_{1n} & B_1 \\ \vdots & \vdots & \vdots & \vdots \\ A_{n1} & \cdots & A_{nn} & B_n \end{array} \quad (2.22)$$

Aplicando as fórmulas descritas em (2.23) sobre a matriz descrita em (2.22), assumindo que $0 \notin A_{11}$, resulta na nova matriz de coeficientes descrita em (2.24).

$$\begin{array}{ll} A'_{ij} = A_{ij}, & 1 \leq j \leq n, \\ B'_i = B_i, & \\ A'_{ij} = A_{ij} - A_{ij}(A_{i1}/A_{11}), & 2 \leq i, j \leq n, \\ B'_i = B_i - B_1(A_{i1}/A_{11}), & 2 \leq i \leq n, \\ A'_{i1} = 0, & 2 \leq i \leq n \end{array} \quad (2.23)$$

$$\begin{array}{cccc} A'_{11} & A'_{12} & \cdots & A'_{1n} & B'_1 \\ 0 & A'_{22} & \cdots & A'_{2n} & B'_2 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & A'_{n2} & \cdots & A'_{nn} & B'_n \end{array} \quad (2.24)$$

Então mostra-se que $\{X | AX=B, A \in A^I \text{ e } B \in B^I\} \subseteq \{Y | A'Y=B', A' \in A^I \text{ e } B' \in B^I\}$ é válido. Assume-se $A \in A^I$ e $B \in B^I$ e que $AX=B$. A matriz $A'=(A'_{ij})$ e o vetor $B'=(B'_i)$, são calculados a partir de (2.25). De forma que os sistemas sejam semelhantes, ou seja, os sistemas $A'Y=B'$ e $AX=B$ têm a mesma solução.

$$\begin{array}{ll} A'_{ij} = A_{ij}, & 1 \leq j \leq n, \\ B'_i = B_i, & \\ A'_{ij} = A_{ij} - A_{ij}(A_{i1}/A_{11}), & 2 \leq i, j \leq n, \\ B'_i = B_i - B_1(A_{i1}/A_{11}), & 2 \leq i \leq n, \\ A'_{i1} = 0, & 2 \leq i \leq n \end{array} \quad (2.25)$$

Se este passo é aplicado $n-1$ vezes, então a matriz concatenada de ordem $n \times n+1$ descrita em (2.22) é transformada em uma matriz intervalar triangular superior, como ilustrado em (2.26), cujo sistema linear é semelhante ao inicial e, para qual é válida a relação de inclusão: $\{X|AX = B, A \in A^I, b \in B^I\} \subseteq \{\tilde{X}|\tilde{A}\tilde{X} = \tilde{b}, \tilde{A} \in \tilde{A}^I, \tilde{b} \in \tilde{B}^I\}$.

$$\begin{array}{ccccccc} \tilde{A}_{11} & \tilde{A}_{12} & \cdots & \tilde{A}_{1n} & \tilde{B}_1 & & \\ & \tilde{A}_{22} & & & & & \\ & & \ddots & \vdots & \vdots & & \\ & & & \tilde{A}_{nn} & \tilde{B}_n & & \end{array} \quad (2.26)$$

Usando as fórmulas (2.27), que correspondem a retrossubstituição, obtém-se o vetor intervalar $X^I = (X_i)$ satisfazendo $\{X | AX = B, A \in A^I, b \in B^I\} \subseteq X^I$.

$$X_i = \left(\tilde{B}_i - \sum_{j=i+1}^n \tilde{A}_{ij} X_j \right) / \tilde{A}_{ii}, \quad 1 \leq i \leq n-1, \quad X_n = \tilde{B}_n / \tilde{A}_{nn}, \quad (2.27)$$

Se $A^I = (A_{ij})$ é uma matriz intervalar não singular, então o algoritmo de Gauss é viável para cada vetor intervalar B^I . Poderá então ser necessário trocar colunas durante o processo de eliminação. Isto é equivalente a multiplicação da matriz A com uma permutação da matriz no lado esquerdo do processo de eliminação. As fórmulas do algoritmo de Eliminação de Gauss intervalar, para serem aplicadas, exigem que o zero não pertença aos elementos intervalares da diagonal principal, ou seja $0 \notin A_{ii}$, para todo i , variando $1 \leq i \leq n-1$, em todos os $n-1$ passos da triangularização. Se para algum elemento intervalar da diagonal principal (A_{ii}), o zero estiver contido ($0 \in A_{ii}$) então, analogamente ao método pontual, trocas de linhas ou colunas serão necessários, para retirar da diagonal principal o elemento intervalar que contém o zero. Isto é sempre possível se for assumido inicialmente que A^{-1} exista para todo $A \in A^I$. Se não for possível realizar este passo é porque todos os membros da coluna contém zero e, então isto implicará que a matriz original A^I contém uma matriz singular A , ou seja não tem inversa, o que contradiz a suposição inicial.

No Método de Gauss Intervalar a troca de linhas do sistema de equações lineares intervalar possui uma grande importância, pois há uma modificação no intervalo solução, embora esta solução sempre contenha todas as soluções reais dos sistemas reais envolvidos no sistema intervalar. O ideal neste caso seria ter um pivô na diagonal principal da matriz intervalar A . Este pivô poderia ser determinado como o maior dos pontos médios de cada coeficiente intervalar, calculados em cada coluna da matriz A .

A seguir será apresentado um exemplo na qual a troca de linhas produziu um melhor resultado (exemplo este retirado de [KOR94], página 114).

$$\text{Sistema original: } \begin{cases} [10,15]x + [1,3]y = [19,27] \\ [2,7]x + [12,14]y = [15,16] \end{cases}$$

$$\text{Vetor Intervalar } X^I: \begin{pmatrix} [-11.087, 19.4062] \\ [-0.394, 7.144] \end{pmatrix}$$

$$\text{Sistema com troca de linhas: } \begin{cases} [2,7]x + [12,14]y = [15,16] \\ [10,15]x + [1,3]y = [19,27] \end{cases}$$

$$\text{Vetor Intervalar } X^I: \begin{pmatrix} X \\ Y \end{pmatrix} = \begin{pmatrix} [-2.026, 5.2427] \\ [-0.3939, 1.36090] \end{pmatrix}$$

2.2.2.3 Cálculo da Matriz Inversa

Este cálculo foi apresentado por Hansen em [HAN65]. O cálculo da matriz inversa é realizado da seguinte maneira:

- a) A^I é a matriz intervalar a ser invertida;
- b) Calcula-se a matriz pontual centro de A (A_c);
- c) Utilizando a matriz centro de A^I , calcula-se a sua aproximação da inversa pontual (A^{-1});
- d) Utilizando a matriz A^I e a aproximação da inversa da matriz centro de A^I , calcula-se a matriz inversa intervalar de A^I . Este cálculo é realizado em 4 passos:

- **Passo 1)** Calcula-se o erro existente entre a multiplicação da matriz A^I pela aproximação da inversa da matriz centro de A^I menos a matriz identidade. Este cálculo é realizado da seguinte maneira: $E^I = I - A^I * B$, onde I é a matriz identidade e B é a aproximação da inversa da matriz centro de A^I ;
- **Passo 2)** Em seguida são realizadas operações que destinam-se ao cálculo da matriz P^I , matriz esta que será utilizada para inflacionar a matriz inversa intervalar de A^I . Deve-se salientar que os valores de P^I são obtidos através da matriz erro E^I e seus valores são menores que o menor valor encontrado na matriz E^I ;
- **Passo 3)** Após o passo 2 a matriz S^I é calculada, matriz esta que é obtida pela seguinte multiplicação: $S^I = I + E^I * (I + E^I * (I + E^I))$. Esta matriz será utilizada no passo seguinte para o cálculo da matriz inversa intervalar;

- **Passo 4)** Ao final é calculada a matriz inversa intervalar de A^I através da multiplicação da aproximação da inversa do centro de A^I pela matriz S^I calculada no passo anterior;

2.2.3 Métodos Intervalares Baseados em Refinamento

Antes de descrever os métodos intervalares deste grupo, será caracterizado o conceito de refinamento. O refinamento é uma técnica que possibilita que se tenha uma medida de exatidão da resposta e uma medida para se avaliar o sistema quanto ao seu condicionamento. A medida de exatidão da resposta é obtida comparando duas aproximações sucessivas; já a análise do condicionamento é feita através da evolução da medida de exatidão das aproximações sucessivas. Se elas não crescerem, pode ser por causa do mal condicionamento da matriz.

A técnica dos refinamentos compreende dois passos: obtenção de uma primeira aproximação X^1 da solução exata X e refinamento sucessivo da aproximação da solução X^k , gerando nova aproximação X^{k+1} , até que se obtenha X como limite da seqüência (X^k) , sob certas condições de convergência. A geração das aproximações é baseada no cálculo do valor do erro contido na solução aproximada. Este erro pode ser somado a aproximação gerando nova aproximação, ou então, ser refinado, uma vez que no cálculo deste também existem erros de arredondamentos embutidos. Isto se constitui um processo iterativo de refino da solução, onde cada iteração é chamada de refinamento, gerando uma nova aproximação, a qual poderá convergir ou não para a solução exata. Isto é determinado pela comparação de aproximações sucessivas, como por exemplo pelo números de algarismos significativos corretos.

O erro ao invés de ser somado pode ser refinado, melhorando sua qualidade. Aplica-se o refinamento para determinar o valor exato do erro na solução. Isto é muito utilizado nos métodos intervalares. Na verdade, a técnica de refinamento é uma das bases para o desenvolvimento de métodos intervalares. Neste tipo de métodos intervalares a solução inicial X^1 é determinada por algum método eficiente pontual. A solução é então transformada em um intervalo inflacionado, para então ser refinado.

2.2.3.1 Método para Sistemas em Geral

Seja $AX = B$ um sistema real de equações com $A \in \mathbb{R}^{n \times n}$ e $B, X \in \mathbb{R}^n$. Encontrar a solução do sistema $AX = B$ é equivalente a encontrar um zero de $f(X) = AX - B$. Portanto, o método de Newton dá o seguinte esquema de iteração de ponto fixo

$$X^{k+1} = X^k - A^{-1}(AX^k - B), \quad k=0,1,\dots \quad (2.28)$$

Aqui, X^0 é algum valor inicial arbitrário. Em geral, a inversa de A não é conhecida. Assim em vez de (2.28), usa-se a iteração *Newton-like*

$$X^{k+1} = X^k - R(AX^k - B), \quad k = 0, 1, \dots, \quad (2.29)$$

onde $R \approx A^{-1}$ é uma aproximação da inversa de A .

Deixe-nos agora substituir as iterações reais X^k por vetores intervalares $X^k \in \mathbb{IR}^n$. Se lá existe um índice k com $X^{k+1} \subseteq X^k$, então, pelo teorema do ponto fixo de Brouwer ([TAR94]), a fórmula 2.29 tem pelo menos um ponto fixo $X \in X^k$. Suponhamos que R seja regular. Então este ponto fixo é também solução de $AX = B$. Portanto, se nós considerarmos o diâmetro de X^{k+1} , nós obtemos $d(X^{k+1}) = d(X^k) + d(R(AX^k - B)) \geq d(X^k)$. Assim, em geral a relação de subconjunto não será satisfeita. Por esta razão, modifica-se o segundo membro de (2.29) para

$$X^{k+1} = RB + (I - RA) X^k, \quad k=0, 1, \dots, \quad (2.30)$$

onde I denota a matriz identidade $n \times n$. Rump em [RUM80] provou para a fórmula (2.30) que se lá existe um índice k com $X^{k+1} \Subset X^k$, então as matrizes R e A são regulares, e existe uma solução única X do sistema $AX = B$ com $X \in X^{k+1}$. E mais ainda, este resultado é válido para qualquer matriz R . Note que o operador \Subset denota a inclusão no interior do intervalo.

É bem conhecido o princípio numérico no qual uma solução aproximada \tilde{X} de $AX^I = B$ pode ser melhorada resolvendo o sistema $AY = D$, onde $D = B - A\tilde{X}$ é o resíduo de $A\tilde{X}$. Como $Y^I = A^{-1}(B - A\tilde{X}) = X^I - \tilde{X}$, a solução exata de $AX^I = B$ é dada por $X^I = \tilde{X} + Y^I$.

$$Y^{k+1} = R(B - A\tilde{X}) + (I - RA)Y^k, \quad k = 0, 1, \dots \quad (2.31)$$

De acordo com os resultados mencionados acima, a equação residual $AY = D$ tem uma única solução $Y \in Y^{k+1}$ caso consiga-se encontrar com sucesso um índice k satisfazendo $Y^{k+1} \Subset Y^k$ para o esquema de iteração intervalar correspondente. Mais ainda, desde que $Y^I = X^I - \tilde{X} \in Y^{k+1}$, nós então temos uma inclusão verificada da única solução de $AX^I = B$ dada por $\tilde{X} + Y^{k+1}$.

Estes resultados continuam válidos caso substitua-se as expressões exatas para Z e C em (2.31) por extensões intervalares. Portanto, para evitar efeitos de sobrestimação, é muito bem recomendado avaliar $B - A\tilde{X}$ e $I - RA$ sem qualquer arredondamento intermediário.

2.2.3.2 Método para Sistemas com Matrizes Densas

O objetivo dos métodos baseados em refinamento para sistemas com matrizes densas é que os mesmos propiciem a elaboração de algoritmos que verifiquem a existência de uma solução e calculem uma inclusão para a solução do sistema $AX = B$, sendo A uma matriz $n \times n$.

Começando com o problema da resolução de um sistema de equações lineares $AX = B$, assume-se primeiro que tem-se uma aproximação da solução \tilde{X} e uma aproximação da inversa R de uma matriz quadrada A . Melhor que calcular uma inclusão da solução diretamente, pode-se incluir o erro da aproximação da solução, o que significará uma exatidão maior. O erro $Y = X - \tilde{X}$ da solução verdadeira x satisfaz a equação

$$AY = B - A\tilde{X} \quad (2.32)$$

que poderá ser multiplicada por R e reescrita na forma

$$Y = R(B - A\tilde{X}) + (I - RA)Y \quad (2.33)$$

ou com $f(Y) := R(B - A\tilde{X}) + (I - RA)Y$

$$Y = f(Y) \quad (2.34)$$

Esta é uma equação na forma de ponto-fixo para o erro Y . Se R é aproximação suficientemente boa de A^{-1} então uma iteração baseada em (2.34) pode ser convergente desde que $I - RA$ tenha um raio espectral pequeno.

Com isso, de (2.34) tem-se a seguinte iteração, onde se usa a aritmética intervalar e intervalos Y^k para Y :

$$Y^{k+1} = R \diamond (B - A\tilde{X}) + \diamond (I - RA)Y^k \quad (2.35)$$

ou

$$Y^{k+1} = F(Y^k) \quad (2.36)$$

onde F é uma extensão intervalar de f .

Aqui \diamond significa que as operações realizadas devem ser executadas de forma exata e o resultado é arredondado por uma inclusão intervalar (vetor ou matriz). Visto que o cálculo

do defeito $B - A\tilde{X}$ e da iteração matricial $I - RA$ poderá causar séria perda de exatidão, essas operações deverão ser calculadas usando o produto escalar ótimo. Com $Z = R \diamond (B - A\tilde{X})$ e $C = \diamond (I - RA)$ agora (2.35) pode ser escrito como:

$$Y^{k+1} = Z + CY^k \quad (2.37)$$

Para a existência da solução de (2.32) e de $AX^1 = B$, usa-se o teorema de ponto-fixo de Brouwer ([TAR94]), o qual implica que um índice de iteração $k+1$ tem uma propriedade de inclusão da forma

$$Y^{k+1} = F(Y^k) \subset Y_k^\circ \quad (2.38)$$

onde Y_k° significa o interior de Y^k . Se este teste de inclusão (2.38) for válido, então a função de iteração f mapeia Y^k dentro dele mesmo e do teorema de Ponto-fixo de Brouwer tem-se que f tem um ponto fixo Y^* o qual está contido em Y^k e também em Y^{k+1} . se Y^k é mapeado em seu interior, tem-se, na verdade, que este ponto fixo também é único, isto é, (2.32) tem uma única solução Y^* e $AX = B$ também tem uma única solução $X^* = \tilde{X} + Y^*$. Não será descrito neste item mas, o programa implementado também realiza os seguintes cálculos: o cálculo de uma inclusão para a solução do sistema $AX = B$ no caso sobre-determinado, isto é, para uma matriz A de ordem $m \times n$, onde $m > n$; o cálculo de uma inclusão para a solução do sistema $AX = B$ no caso sub-determinado, isto é, para uma matriz A de ordem $m \times n$, onde $m < n$; o cálculo de uma inclusão para a inversa A^{-1} de A ; o cálculo de uma inclusão da pseudo inversa A^+ de A no caso sobre-determinado, isto é, para uma matriz A de ordem $m \times n$, onde $m > n$; o cálculo de uma inclusão da pseudo inversa A^+ de A no caso sub-determinado, isto é, para uma matriz A , de ordem $m \times n$, onde $m < n$.

2.2.3.3 Método para Sistemas com Matrizes Bandas

A mesma abordagem pode ser utilizada para sistemas esparsos, onde, por exemplo, a matriz dos coeficientes é do tipo banda (ou tridiagonal). Entretanto, como a inversa de uma matriz banda é, em geral, uma matriz densa, o algoritmo proposto para sistemas densos consumiria muito tempo, tornando-se ineficiente. Para sistemas com matrizes bandas pode-se aplicar a decomposição LU sem pivotamento para a matriz A (ver item 2.2.3.4) e gerar uma iteração intervalar similar a (2.35). Aqui não é necessário o cálculo da aproximação da matriz inversa R de A e a iteração poderá ser realizada pela resolução de dois sistemas com matrizes triangulares bandas (L e U).

A definição matemática para grandes sistemas lineares com matrizes bandas é exatamente a mesma que foi abordada para sistemas com matrizes densas.

Utilizando a decomposição LU de A tem-se uma iteração análoga a (2.35), que é dada por

$$Y^{k+1} = (LU)^{-1} (B - A\tilde{X}) + (I - (LU)^{-1}A)Y^k \quad (2.39)$$

ou, multiplicando com LU e realizando a extensão intervalar

$$LU \cdot Y^{k+1} = \diamond (B - A\tilde{X}) + \diamond (LU - A)Y^k \quad (2.40)$$

Com isso, tem-se que resolver dois sistemas triangulares, com matrizes bandas, L e U, para calcular Y^{k+1} , isto é, para realizar um passo da iteração (2.40). Em cada iteração é calculada primeiramente uma inclusão para a solução de

$$LZ^{k+1} = \diamond (B - A\tilde{X}) + \diamond (LU - A)Y^k \quad (2.41)$$

e então de Y^{k+1}

$$UY^{k+1} = Z^{k+1} \quad (2.42)$$

Aqui também o teste de inclusão é dado por

$$Y^{k+1} = F(Y^k) \subset Y_k^\circ \quad (2.43)$$

e é verificado da mesma maneira como no caso de sistemas com matrizes densas.

2.2.3.4 Cálculo da Matriz Inversa

O cálculo de uma boa aproximação da matriz inversa de A, juntamente com a aproximação da solução X é de extrema importância para a rápida convergência da iteração do método para o cálculo da solução do sistema $AX = B$.

Para iniciar, será calculada uma aproximação da inversa $R \approx A^{-1}$. Assumindo que tem-se uma aproximação da inversa R, mesmo quando A é mal condicionada, a matriz RA é geralmente melhor condicionada do que A. Agora a simples relação

$$A^{-1} = (RA)^{-1}R \quad (2.44)$$

sugere o cálculo de outra aproximação da inversa S de RA e tem-se com isso que o produto S e R é uma melhor aproximação de A^{-1} . Com isso pode-se calcular uma aproximação da inversa $R_1 + R_2$ de tamanho duplo pelos seguintes passos:

- a) Calcule uma aproximação da inversa R de A com Gauss-Jordan;
- b) Calcule RA;

- c) Calcule uma aproximação da inversa S de RA com Gauss-Jordan;
- d) Calcule SR num acumulador longo e armazene essa soma em duas matrizes de ponto-flutuante $R1$ e $R2$.

Agora tem-se uma aproximação da inversa $R = R1 + R2$ de tamanho duplo e então pode-se executar o algoritmo descrito a seguir, utilizando o produto escalar ótimo para o cálculo do R . Essa estratégia pode ser resumida em dois passos:

- a) Parte I - Calcule uma aproximação da inversa R de A de tamanho simples e execute o algoritmo de inclusão. Se falhar vá para Parte II;
- b) Parte II - utilize a aproximação da inversa apresentada por Rump ([RUM80]) e execute o algoritmo de inclusão com a aproximação da inversa de A de tamanho duplo $R = R1 + R2$.

No caso de sistemas esparsos, onde, por exemplo, a matriz dos coeficientes é do tipo banda (ou tridiagonal) uma outra abordagem para o cálculo da aproximação da matriz inversa R da matriz A é a utilização da técnica de decomposição. Decompõe-se a matriz A em duas matrizes triangulares, uma inferior (L) e outra superior (U), ou seja, $A=LU \in \mathbf{MR}$. Têm-se então que a inversa R é igual a:

$$R = (LU)^{-1} = U^{-1} \cdot L^{-1} \approx A^{-1} \quad (2.45)$$

onde

$$LU \approx A \quad (2.46)$$

é uma aproximação da decomposição LU de A sem pivotamento. Mesmo sem o uso de pivotamento, L e U são matrizes bandas e são triangular inferior e superior, respectivamente.

2.2.4 Métodos Intervalares Baseados em Iteração

Os métodos intervalares iterativos são a extensão dos métodos pontuais iterativos. Para a aplicação dos métodos iterativos é necessário que o sistema de equações a ser resolvido esteja na forma $X^I = A^I X^I + B^I$ onde A^I é a matriz intervalar de ordem $n \times n$ que contém os coeficientes intervalares e B^I é o vetor intervalar de ordem n que contém os termos intervalares independentes (X^I é o vetor intervalar a ser encontrado).

2.2.4.1 Método de Jacobi Intervalar

O método de Jacobi Intervalar é um método análogo ao método correspondente para sistemas de equações pontuais. A iteração deste método é determinada conforme (2.47).

$$X^{k+1} = AX^k + B, \quad k \geq 0, \quad (2.47)$$

converge para um único ponto fixo X^* da equação $X^I = A^I X^I + B^I$, para todo X^0 se $\rho(|A^I|) < 1$.

Se A^I uma matriz intervalar para a qual $\rho(|A^I|) < 1$, então para o ponto fixo X^* da equação $X^I = A^I X^I + B$, a relação $\{Y^I = (I^I - A^I)^{-1} \cdot B^I \mid A^I \in A, B^I \in B\} \subseteq \{X^I \mid X^I \in X^*\}$ é verdadeira.

Se $A^I = (A_{ij}) \in M^{n \times n}(I(\mathbb{R}))$, e se a desigualdade $i(A_{ij}) \geq 0$ se verifica para $A_{ij} = [i(A_{ij}), s(A_{ij})]$, então X^* é ótimo no seguinte sentido: Não há nenhum vetor intervalar $X^I \in V_n(I(\mathbb{R}))$ que satisfaz $X \in X^*$, $X \neq X^*$, e para a qual $\{Y^I = (I^I - A^I)^{-1} \cdot B^I \mid A^I \in A, B^I \in B\} \subseteq \{X^I \mid X^I \in X^*\}$ é verdadeira.

2.2.4.2 Método de Gauss-Seidel Intervalar

O método análogo ao método de Gauss-Seidel pontual é obtido decompondo a matriz intervalar em $A^I = L^I + D^I + U^I$, onde L^I é a matriz triangular estritamente inferior, e U uma matriz triangular estritamente superior e D^I é a matriz da diagonal de A^I . O método de Gauss-Seidel Intervalar é então definido pela iteração

$$X^{k+1} = LX^{k+1} + (D + U)X^k + B, \quad k \geq 0. \quad (2.48)$$

A convergência do método de Gauss-Seidel Intervalar é considerada a seguir.

A iteração acima com um vetor inicial arbitrário X^0 converge para um único ponto fixo X^* se, e somente se

$$\rho((I - |L^I|)^{-1} (|D^I| + |U^I|)) < 1 \quad (2.49)$$

2.2.4.4 Método de Relaxação Intervalar

Existem muitos outros métodos para resolução de sistemas lineares pontuais de equações da forma $X^I = A^I X^I + B^I$. A maioria destes métodos podem ser facilmente

transformados para métodos iterativos usando-se vetores intervalares. Como exemplo é considerado o método de Relaxação Intervalar para o caso de passos simples.

A matriz A^I é decomposta em $A^I = L^I + D^I + U^I$ como no método de Gauss-Seidel Intervalar onde L^I é uma matriz triangular estritamente inferior, U^I é uma matriz triangular estritamente superior e D^I uma matriz diagonal. Com isso a iteração do método é dada por (2.50).

$$X_i^{k+1} = \sum_{j=1}^{i-1} A_{ij} X_j^{k+1} + \sum_{j=i}^n A_{ij} X_j^k + B_i, \quad (2.50)$$

$$X_i^{k+1} = (1-\omega) X_i^k + \omega X_i^{k+1}, \quad 1 \leq i \leq n, \quad k \geq 0$$

iniciando com um vetor intervalar arbitrário X^0 . Usando a notação de vetor isto pode ser escrito

$$X^{k+1} = (1-\omega) X^k + \omega \{L X^{k+1} + (D+U) X^k + B\}, \quad k \geq 0, \quad (2.51)$$

onde $\omega > 0$ é um parâmetro utilizado para se obter aceleração em relação ao método de Gauss-Seidel. Como nos casos do método de Jacobi Intervalar e o método de Gauss-Seidel Intervalar deve ser mostrado que

$$\rho((I-\omega|L^I|)^{-1} \{ |1-\omega|I + \omega(|D^I| + |U^I|) \}) < 1 \quad (2.52)$$

é uma condição necessária e suficiente para a convergência do método para um único ponto fixo para um vetor inicial arbitrário. Se $\rho(|A^I|) < 1$, então a condição acima é satisfeita para todos ω , para os quais $0 < \omega < 2/(1+\rho(|A^I|))$ (ver [ALE83]).

Se o método de Jacobi Intervalar converge e se ω satisfaz a condição da desigualdade acima, então o método de relaxação também converge para um ponto fixo X^* satisfazendo a equação $X^* = (1-\omega)X^* + \omega(AX^* + B)$.

2.2.4.5 Método de Relaxação Simétrico Intervalar

As afirmações do método de relaxação intervalar são válidas para o método de relaxação simétrico intervalar, o qual sua formulação é dado por (2.53).

$$X^{k+1/2} = (1-\omega) X^k + \omega \{L X^{k+1/2} + U X^k + B\}, \quad (2.53)$$

$$X^{k+1} = (1-\omega) X^{k+1/2} + \omega \{L X^{k+1/2} + U X^{k+1} + B\}, \quad k \geq 0$$

Condições necessárias e suficientes para a convergência deste método para um vetor intervalar inicial arbitrário para um único ponto fixo são dadas pela condição

$$\rho((I-\omega|U|)^{-1} (|1-\omega|I' + \omega|L|)(I-\omega|L|)^{-1} (|1-\omega|I' + \omega|U|)) < 1 \quad (2.54)$$

Se, além disso, $\rho(|A''|) < 1$, então este é o caso se $0 < \omega < 2/(1 + (|A''|))$. A prova pode ser realizada da mesma maneira como para o método de relaxação.

2.3 Conclusões

Neste capítulo foi apresentada uma descrição de alguns métodos intervalares encontrados na bibliografia pesquisada. Estes métodos foram agrupados em três metodologias de desenvolvimento e classificados como métodos diretos, iterativos e baseados em refinamento.

Os métodos intervalares diretos descritos foram os métodos de Hansen e o método de Eliminação de Gauss Intervalar com e sem pivotamento. Os métodos de Hansen, basicamente, possuem a mesma característica, ou seja, para a resolução do sistema $AX = B$. Esses métodos calculam a matriz inversa de A e a multiplicam pelo vetor B , obtendo com isso a solução do sistema. O Método de Eliminação de Gauss Intervalar mantém, de uma maneira geral, as mesmas características de sua versão pontual. Deve-se destacar que esse método não consegue resolver um sistema de equações lineares que contiver o zero pertencente ao intervalo localizado na diagonal principal da matriz A . Notou-se que, em alguns casos, o método de Eliminação Gauss Intervalar com pivotamento resolve um sistema de equações lineares onde o método sem pivotamento não obteve sucesso.

Os métodos intervalares baseados em refinamento são métodos que, primeiramente, obtêm uma aproximação pontual da solução do sistema e, posteriormente, através do refinamento dessa solução, obtêm a solução intervalar do sistema. Nesse grupo de métodos foram descritos três métodos: um método para a resolução de sistemas em geral; outro para a resolução de sistemas densos e um para a resolução de sistemas com matrizes bandas.

Com relação aos métodos iterativos, encontrou-se algumas dificuldades quanto a escolha da aproximação inicial da solução do sistema e quanto a convergência dos métodos implementados. Estas dificuldades deverão ser estudadas com maiores detalhes, em trabalhos futuros, a fim de poder eliminar ou amenizar esses problemas e com isso possa-se melhorar os resultados obtidos pelos métodos intervalares iterativos.

Como o objetivo deste trabalho é o estudo de alguns métodos intervalares e sua respectiva implementação, o método de Hansen para o caso geral, o método de Relaxação e de Relaxação Simétrico foram apenas descritos e sua implementação será realizada em uma

próxima etapa deste trabalho. Devido a estrutura modular da biblioteca *selint.p*, futuras implementações são facilitadas, devido ao fato de que basta incluir esses novos métodos implementados no seu respectivo módulo de seu grupo de métodos.

Detalhes a respeito dos resultados obtidos pelos métodos intervalares implementados na biblioteca *selint.p* e a comparação desses resultados com outras bibliotecas são apresentados no capítulo 4.

3 Biblioteca Aplicativa Intervalar - Módulo Sistemas

Neste capítulo é proposta a biblioteca aplicativa intervalar para a resolução de sistemas de equações lineares *selim.p*. Esta biblioteca foi desenvolvida no ambiente de computadores pessoais (tipo PC), utilizando o compilador Pascal-XSC.

Serão discutidos, neste capítulo, os objetivos e a metodologia de desenvolvimento dessa biblioteca, a sua estrutura e os módulos que a compõe, a definição de sua documentação e uma breve descrição do ambiente computacional na qual essa biblioteca foi desenvolvida.

3.1 Proposta, Objetivos e Metodologia de Desenvolvimento

Um dos objetivos deste trabalho é a elaboração de ferramentas computacionais, que possibilitem a resolução de sistemas de equações lineares, utilizando a aritmética intervalar. O desenvolvimento dessa biblioteca se deu através de estudos realizados junto ao Grupo de Matemática Computacional da UFRGS, estudos que iniciaram primeiramente fazendo-se uma pesquisa a respeito de métodos pontuais para a resolução de sistemas de equações lineares, seguido por um estudo a respeito da extensão intervalar desses métodos, estudando-se, com isso, suas características e metodologias de desenvolvimento.

Os principais objetivos com o desenvolvimento das bibliotecas aplicativas intervalares são os seguintes: utilização dessas bibliotecas na difusão do estudo de métodos intervalares junto aos meios acadêmicos e a pessoas que estejam interessadas a respeito desses métodos; utilização das bibliotecas na resolução de problemas computacionais nas mais diversas áreas de pesquisa. Exemplos de problemas que envolvem a resolução de sistemas de equações lineares e que podem utilizar essas bibliotecas são apresentadas em [HÖL96].

O desenvolvimento da biblioteca aplicativa intervalar, para a resolução de sistemas de equações lineares, foi realizado em quatro etapas. A primeira consistiu do estudo e caracterização de alguns dos métodos intervalares existentes na bibliografia estudada. Após essa caracterização, foram elaborados os respectivos algoritmos desses métodos.

Na segunda etapa foi realizado um estudo a respeito do ambiente computacional na qual foi desenvolvida a biblioteca *selim.p*, bem como, um estudo a respeito da linguagem que foi utilizada para a implementação da biblioteca. Após esses estudos, foi implementada a biblioteca *selim.p* em Pascal-XSC.

Na terceira etapa foram pesquisados problemas práticos que necessitam da resolução de sistemas de equações lineares e através dessa pesquisa, foram selecionados exemplos que

serão resolvidos e seus resultados analisados a fim de se validar essa biblioteca. Esses problemas foram reunidos e descritos em [HÖL96].

A última etapa consiste da determinação das limitações da biblioteca implementada e a influência que essas limitações provocarão nos exemplos resolvidos. Através da análise dessas limitações serão elaboradas propostas de melhoramentos da biblioteca aplicativa intervalar desenvolvida neste trabalho.

3.2 Estrutura da Biblioteca

Neste item são feitas algumas considerações sobre a estruturação da biblioteca aplicativa intervalar *selint.p*. São feitas algumas considerações sobre o conjunto de rotinas que compõem essa biblioteca, rotinas baseadas no estudo de métodos intervalares descritos no capítulo dois (Métodos Intervalares). Essas rotinas abordam métodos intervalares de solução de sistemas lineares e nesse estudo, considera-se uma equação algébrica como um sistema linear de ordem um, ou seja, com uma só equação.

Através do estudo dos métodos intervalares e de seus algoritmos relatados no capítulo dois e no anexo 1, respectivamente, visualizou-se a necessidade de definir bibliotecas aplicativas que tratassem de capítulos da matemática numérica. A primeira destas bibliotecas científicas aplicativas é a que trata da resolução de sistemas de equações lineares, denominada *selint.p* (*sel* de sistemas de equações lineares e *int* de intervalar). A criação de bibliotecas científicas independentes, justifica-se no fato de que estas devem ser incluídas nos programas aplicativos dos usuários.

Através desse estudo foram identificadas três tipos de metodologias de desenvolvimento ou abordagens dos métodos intervalares. Essas abordagens geraram, cada uma, um módulo da biblioteca (a estrutura dessa biblioteca é apresentada pela figura 3.1). Estes módulos devem ser:

- a) Módulo *dirint*, inclui os métodos baseados em operações algébricas intervalares e propriedades intervalares, também são conhecidos como métodos diretos intervalares;
- b) Módulo *refint*, inclui os métodos baseados em inclusões ou refinamentos intervalares da solução e do erro. Também podem ser chamados de métodos híbridos, uma vez que se pode utilizar a solução inicial calculada por métodos pontuais ou a matriz inversa pontual;
- c) Módulo *itrint*, inclui os métodos iterativos intervalares, conhecidos também como métodos de relaxação, eles também se baseiam em inclusões monotônicas.

Além desses três módulos, foi definido um quarto módulo, *equalg*, que deve conter rotinas do caso particular de sistema linear de ordem um, ou seja, resolução algébrica de equações por métodos intervalares, como as versões intervalares do método de Newton. Como existe uma grande variedade de versões intervalares para o método de Newton, como pode ser visto em [DIV91], foi escolhida apenas uma versão para se validar o uso da matemática intervalar, o método de Newton Intervalar Simplificado.

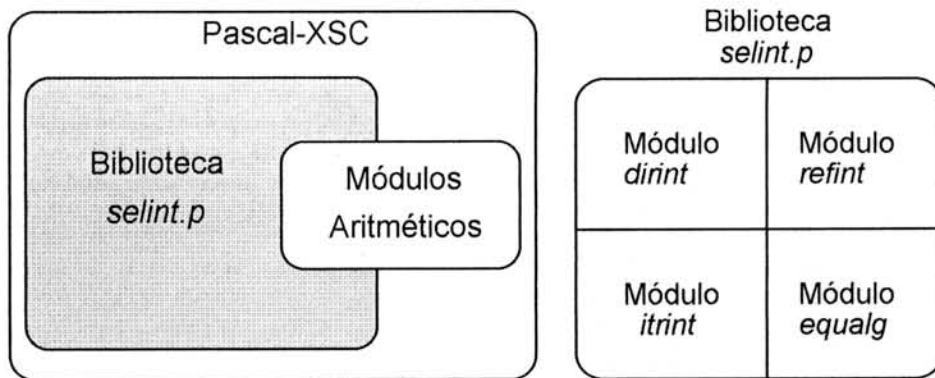


Figura 3.1 - Estrutura da biblioteca *selint.p*

Para o uso da biblioteca *selint.p*, é necessário a inclusão dos módulos aritméticos avançados do Pascal-XSC, visto que utiliza-se operações intervalares que não estão contidas na biblioteca *selint.p*. A seguir, são caracterizados os módulos que compõem essa biblioteca, bem como alguns de seus respectivos métodos. Esses métodos foram descritos no capítulo dois e seus respectivos algoritmos são apresentados no anexo A-1 deste trabalho.

3.2.1 Módulo *dirint* (métodos diretos)

O módulo *dirint* é constituído de rotinas que implementam o método de Eliminação de Gauss intervalar com e sem pivotamento, os métodos de Hansen e o cálculo da matriz inversa. Esses métodos são baseados em propriedades e operações algébricas intervalares.

Tabela 3.1 - Funções do módulo *dirint*

Função	Descrição
<i>hansen2(A,B)</i>	Método de Hansen2
<i>hansen3(A,B)</i>	Método de Hansen3
<i>hansen4(A,B)</i>	Método de Hansen4
<i>hansen5(A,B)</i>	Método de Hansen5
<i>gaussd(A,B)</i>	Método de Eliminação de Gauss sem pivotamento
<i>gaussp(A,B)</i>	Método de Eliminação de Gauss com pivotamento
<i>imatinv(A,AR)</i>	Cálculo da Matriz Inversa Intervalar por Hansen
<i>matinv(A,R,erro)</i>	Cálculo da Matriz Inversa Pontual
<i>minv(A,erro)</i>	Cálculo da matriz inversa pontual

3.2.2 Módulo *refint* (métodos baseados em refinamento)

O módulo *refint* é constituído de rotinas que implementam os métodos baseados em refinamento. Alguns destes métodos necessitam de uma solução aproximada inicial produzida por métodos pontuais ou da inversa pontual da matriz dos coeficientes. Adotou-se para o cálculo da inversa pontual, o método de Gauss-Jordan e, para o cálculo de estimativa inicial, o método de Eliminação de Gauss sem pivotamento. Essa escolha foi arbitrária, poderia ser qualquer outro método, por isso, eles foram implementados em subrotinas.

Tabela 3.2 - Funções do módulo *refint*

Função	Descrição
<i>refgeral</i> (A,B,X,erro)	Método para sistemas em geral
<i>refdensa</i> (A,B,X,erro)	Método para sistemas com matrizes densas
<i>refbanda</i> (A,B,X,l,k)	Método para sistemas com matrizes bandas
<i>inv</i> (A,Y,erro)	Cálculo da matriz inversa intervalar

3.2.3 Módulo *itrint* (métodos iterativos)

O módulo *itrint* é constituído de métodos iterativos intervalares como métodos de relaxação simples e simétrico, método de Gauss-Seidel Intervalar e o método de Jacobi. A descrição desses métodos é apresentada no capítulo dois. Os métodos iterativos intervalares também são mais utilizados no caso de sistemas lineares, onde a matriz dos coeficientes é esparsa.

Tabela 3.3 - Funções do módulo *itrint*

Função	Descrição
<i>jacobi</i> (A,B,w,iter)	Método de Jacobi
<i>gauss</i> (A,B,w,iter)	Método de Gauss-Seidel

3.2.4 Módulo *equalg* (sistemas de equações de ordem 1)

Esse módulo contém as rotinas que implementam a versão intervalar do Método de Newton para resolução de equações algébricas. Esta versão foi descrita em detalhes em [DIV91]. Esse módulo contém, também, as rotinas que resolvem sistemas de equações de ordem 1, diretamente.

Tabela 3.4 - Funções do módulo equalg

Função	Descrição
newton(X.X ^{li})	Método de Newton Intervalar simplificado
equal1(A,B)	Método direto para a equação $AX = B$
equal2(A,B,C)	Método direto para a equação $AX + B = C$
equal3(A,B,C)	Método direto para a equação $AX + BX = C$
equal4(A,B,C,D)	Método direto para a equação $AX + B = CX + D$

3.3 Definição da Documentação

Uma boa documentação deve ser clara, concisa, fácil de entender, explanando como funciona e como usar. Isso é útil ao usuário para decidir sobre o uso ou não da rotina ou programa e, também, para diagnosticar as dificuldades provenientes do uso. Deve ser capaz de satisfazer a usuários que se utilizam esporadicamente da documentação para resolver pequenas dificuldades.

A documentação da biblioteca aplicativa intervalar será constituída do manual de utilização da *selint.p*. Ele é constituído por duas partes, a primeira contém informações sobre o uso e vantagens de utilização da biblioteca e, a outra parte contém a documentação das rotinas que compõem essa biblioteca. A primeira parte visa atender um usuário que deseja maiores informações da biblioteca, enquanto que a segunda parte visa atender ao uso esporádico.

Na documentação das rotinas da biblioteca *selint.p* será usado o mesmo formato usado na documentação dos módulos aritméticos do Pascal-XSC. A documentação da biblioteca *selint.p* seguirá o padrão encontrado nos módulos aritméticos I_ARI, C_ARI, CI_ARI, MV_ARI, MVI_ARI, MVC_ARI, e MVCI_ARI. Este padrão estabelece que a biblioteca seja dividida em módulos, que são compostos por rotinas de um mesmo grupo.

Para cada rotina é apresentada uma documentação específica, denominada de documentação detalhada da rotina. O formato de documentação detalhada das rotinas é o descrito na figura 3.2. As documentações específicas são organizadas na ordem alfabética. Para os nomes das rotinas foram adotados mnemônicos que lembram a função real ou matemática.

NOME	Contém o nome da rotina
CABEÇALHO	Descrição da função com seus parâmetros de entrada e saída.
TIPOS	Descreve os tipos de dados envolvida na rotina, tanto dados de entrada como de saída
DESCRIÇÃO	Descreve a rotina com detalhes
ARQUIVOS	Lista os módulos aritméticos que cada rotina usa
EXEMPLOS	Contém exemplos de uso
MENSAGENS	Descreve informações, diagnósticos e mensagens e erro que podem aparecer.
VER TAMBÉM	Lista outras rotinas que tenham ligação com a rotina que está sendo descrita.

Figura 3.2 - Formato da documentação da biblioteca *selint.p*

3.4 A Biblioteca *selint.p*

Neste item é descrito o ambiente computacional mínimo necessário para a execução da biblioteca *selint.p*, biblioteca esta desenvolvida em Pascal-XSC. Cabe ressaltar que o Pascal-XSC é um pré-processador da linguagem C, existindo versões de Pascal-XSC para diferentes compiladores C e para diferentes plataformas. O compilador C utilizado na instalação disponível no Instituto de Informática da UFRGS é do C-GNU, também conhecido como C DJGPP.

Observa-se ainda que o Compilador Pascal-XSC é distribuído pela empresa alemã Numerik Software GmbH¹ e foi cedido para o Instituto de Informática da UFRGS para fins de estudo e pesquisa. Já o compilador C GNU é de domínio público. Ele está disponível na Rede, podendo ser importado através do utilitário *ftp*.

O compilador Pascal-XSC está disponível para várias plataformas, como pode ser visto em [HÖL93 e HÖL95]. Em todas elas os cálculos produzem os mesmos resultados, portanto a aritmética que se tornou disponível por esta linguagem é independente da plataforma.

A versão disponível no Instituto de Informática da UFRGS é para equipamento do tipo IBM-PC 486 ou IBM-PC 386 com co-processador aritmético 80387. São necessários 640 KB de memória convencional e, pelo menos, 4 *Megabytes* de memória estendida. Para um melhor desempenho em termos de velocidade de processamento recomenda-se que o equipamento tenha 8 *Megabytes* de memória.

São necessários, ainda, um disco rígido de pelo menos 10 *Megabytes* de espaço disponível, *drive* de 1.2 ou 1.44 *Megabytes*, um monitor monocromático VGA. O sistema

¹ Numerik Software GmbH, POBox 2232 D-76492 Baden-Baden Germany FAX 0049721694418

operacional necessário para a instalação é o MS-DOS (ou compatível) versão 5.0 ou posterior.

O sistema Pascal-XSC é composto pelos programas de configuração, pelo gerenciador, pelo compilador, pelos módulos padrões e por uma biblioteca de rotinas que compõem o sistema de execução de tempo real. O compilador não contém um gerador de código para uma máquina específica, ele produz um código C, conforme padrão C ANSI. Na verdade ele é um pré-processador que converte para C. O código C é que, quando compilado, gera o código objeto para a máquina específica.

PASCAL-XSC é uma linguagem de programação de propósito geral que proporciona condições especiais à implementação de algoritmos numéricos sofisticados, que verificam matematicamente os resultados. PASCAL-XSC é uma extensão da linguagem de programação PASCAL para computação científica. Seu nome vem do inglês, **PASCAL eXtension for Scientific Computation**. Ele contém as características do Pascal Padrão; o conceito de operador universal (operador definido pelo usuário); funções e operadores com tipo de resultado arbitrário; *overloading* de rotinas, funções e operadores; *overloading* de atribuição de operadores; *overloading* de rotinas de entrada e saída (*read* e *write* genéricos); conceito de módulos; *arrays* dinâmicos; acesso a *subarrays*; conceito de *string*; arredondamento controlado; produto escalar ótimo (exato); tipo-padrão *dotprecision* (um formato de ponto-fixa abrangendo todo o intervalo do produto de valores em ponto-flutuante); aritmética especial para tipos-padrões de complexos e intervalos; aritmética de alta exatidão para todos os padrões; alta exatidão para funções elementares e avaliação exata de expressões (*#-expressions*). O novo sistema PASCAL-XSC tem as vantagens de ser portátil a várias plataformas, estando disponível para computadores pessoais (PC), estações de trabalho do tipo Sun e Hp, *mainframes* e supercomputadores. A portabilidade é garantida pelo uso de um compilador que traduz para a linguagem ANSI-C.

Este sistema proporciona uma completa simulação da aritmética de ponto-flutuante definida pelo padrão binário IEEE-754. Graças a isto, programas em PASCAL-XSC produzem resultados idênticos em todas as plataformas.

Pelo uso dos módulos matemáticos do PASCAL-XSC, algoritmos numéricos que providenciam alta exatidão e verificação automática de resultados podem ser facilmente programados. Além disso, o compilador PASCAL-XSC simplifica o projeto de programas para as Engenharias e para a Computação Científica, graças à estrutura modular dos programas, à possibilidade de definição de operadores, ao *overloading* de funções, rotinas e operadores, às funções e operadores com tipos arbitrários de dados e aos *arrays* dinâmicos. Outras características presentes nos módulos da aritmética padrão para os tipos adicionais de dados numéricos incluem operadores e funções elementares com alta exatidão e com avaliação exata de expressões.

Os programas escritos em PASCAL-XSC são de fácil leitura, mesmo se existirem as operações com tipos de dados dos espaços matemáticos avançados, onde os operadores

usados para as operações obedecem à notação matemática convencional. Existe ainda, uma grande quantidade de problemas numéricos que podem ser resolvidos pelas bibliotecas de rotinas com verificação automática do resultado. O PASCAL-XSC possui grandes facilidades para o desenvolvimento de tais rotinas.

Uma completa descrição da linguagem PASCAL-XSC e dos módulos aritméticos, assim como uma grande variedade de exemplos são dados em [KLA91, KLA92, HÖL93, HAM93, KRÄ96 e HÖL95].

3.5 Bibliotecas utilizadas para comparações de resultados

Neste item é descrito o ambiente computacional, linguagem e bibliotecas na qual foi desenvolvida a biblioteca *libselint.a*, biblioteca que foi utilizada para realizar comparações de resultados com a biblioteca *selint.p*. Serão descritos, resumidamente, o Supercomputador Cray, a linguagem FORTRAN 90 e a biblioteca *libavi.a*. Para maiores detalhes ver [DIV95].

O Supercomputador Cray encontra-se no Centro Nacional de Supercomputação (CESUP/RS). O CESUP instalou e opera o primeiro supercomputador de uso geral da América do Sul. Ele é da família Y-MP com dois processadores vetoriais com as seguintes características:

- velocidade total máxima de 660 MFlops;
- palavra de 64 bits;
- memória RAM de 256 Mbytes e 16 Gbytes de disco;
- sistema operacional UNICOS, compatível com UNIX System V.

Atualmente as aplicações² no CRAY Y-MP estão concentradas nas áreas de: Geofísica; Química computacional; Dinâmica de Fluidos; Análise estrutural; Matemática/algoritmos; Física do estado sólido; Física (campos/ partículas/ ótica/ astrofísica); Física de Plasma; Computação de Alto Desempenho; Redes Neurais; Engenharia Química e Ciências do Ambiente.

O processamento escalar e vetorial no CrayY-MP2E são efetuados sobre dados. Processamento escalar ocorre seqüencialmente e usa um operando ou um par de operandos para produzir um único resultado. Processamento escalar é efetuado usando registradores escalares. A principal vantagem do processamento vetorial sobre o escalar é a eliminação do

² Segundo fontes do próprio CESUP, nos anais do Seminário de Supercomputação Aplicada, Supercomp 94,

tempo de carga (*startup*) de todas operações. O tempo de carga para operações vetoriais é suficientemente pequeno para que o processamento vetorial seja mais eficiente do que o processamento escalar para vetores contendo poucos elementos.

No Cray do CESUP/RS, estão disponíveis os compiladores FORTRAN (77 e 90), C e Pascal com subrotinas científicas e matemáticas altamente otimizadas, incluindo Boeing CS, BLAS3, LINPACK, EISPACK, FFT e operações matriciais. Dispõe, também, dos seguintes pacotes de aplicação: MPGS: sistema de visualização de 2 e 3 dimensões, UNICHEM: sistema interativo de química quântica, MOPAC II: analisador de química quântica, GAMESS: analisador atômico e molecular, SPEEDUP: pacote de simulação de processos e plantas químicas, ANSYS: pacote de análise de estruturas, calor e campos, MCS/NASTRAN: análise estrutural e de transferência de calor, MCS/EMAS: análise de campos eletromagnéticos, MCS/DYTRAN: análise dinâmica de sólidos-fluidos, MCS/XL: pacote de visualização de dados, ADAMS: simulação de estruturas mecânicas articuladas, SPICE2 e 3: simulador elétrico, PISCES-2B: simulação de dispositivos semicondutores, PVM/HeNCE: sistema de programação paralela por troca de mensagens, REDUCE: sistema de computação algébrica e FLOTRAN: pacote de análise de fluxo de fluidos por elementos finitos.

O FORTRAN ainda é a principal linguagem de programação utilizada na área científica. O FORTRAN vem evoluindo desde o compilador FORTRAN IV, para FORTRAN 77, que incluiu processamento vetorial e paralelo e, enfim, para o FORTRAN 8x, finalmente conhecido como FORTRAN 90. Portanto, o FORTRAN 90 é uma extensão do FORTRAN padrão. O FORTRAN 90 contém todo o FORTRAN 77, ou seja, todos os programas escritos em FORTRAN 77 ainda são compilados, podendo haver algumas diferenças na forma como serão executados, mas em geral serão compatíveis. O FORTRAN 90 foi escrito pensando no usuário, do ponto de vista do programador.

Entre as principais características existentes no FORTRAN 77, podem-se destacar a sintaxe de vetores, *arrays* automáticos, subprogramas recursivos, linha de inclusão (*include*) e os comandos estruturados *do-end* e *do-while*. O FORTRAN 90, além destas características introduz: nova forma para programas fonte, números complexos com dupla precisão, atribuição de ponteiros, *arrays* alocáveis, estruturas, módulos, rotinas internas, especificações de interface, várias novas funções intrínsecas e novas rotinas de entrada e saída.

No desenvolvimento da biblioteca *libselint.a* também foi utilizada a biblioteca com aritmética vetorial intervalar *libavi.a*. A seguir é feita uma breve descrição a respeito desta biblioteca.

A biblioteca de rotinas intervalares *libavi.a* foi projetada para viabilizar o uso da matemática intervalar, em supercomputadores, na resolução de problemas que necessitem alta exatidão. As soluções calculadas com o auxílio desta biblioteca são intervalos que contêm a solução exata, portanto se tem a solução com limites confiáveis.

A biblioteca de rotinas intervalares é composta de quatro módulos. O módulo *básico* inclui o arquivo que contém a definição de intervalos reais e complexos. Neste módulo são implementados todas as operações entre intervalos reais. Estas operações servem de base a todos os demais módulos.

O módulo *mvi* inclui o módulo *básico* e é incluído no módulo de aplicações *aplic*. Este módulo implementa todas as operações entre vetores de intervalos, matrizes de intervalos e rotinas de diferentes tipos de dados com vetores e matrizes de intervalos. O módulo *aplic* inclui o módulo *mvi* e o *básico*, todas as rotinas dos dois módulos estão disponíveis.

O módulo dos intervalos complexos *ci* contém o módulo *básico*. Nele estão as rotinas que manipulam intervalos complexos de forma análoga a intervalos reais. Os tipos vetores e matrizes de intervalos complexos foram definidos, mas não foram desenvolvidas as rotinas que os manipulam, mas podem ser facilmente implementados a partir do módulo *mvi* e *ci*. Elas são a extensão do módulo *mvi* para complexos.

A *libavia* foi desenvolvida em FORTRAN 90 da Cray. Ela totaliza cerca de 290 rotinas que manipulam intervalos reais, vetores e matrizes de intervalos reais, intervalos complexos e algumas rotinas intervalares estendidas da álgebra linear que manipulam matrizes e vetores, incluídas na biblioteca BLAS.

4 Resolução de Problemas e Comparações

Neste capítulo são realizadas comparações e a análise dos resultados obtidos pela biblioteca *selint.p*. São comparados os resultados pontuais com os intervalares, os resultados obtidos pela *selint.p* com os obtidos pela biblioteca *libselint.a* (biblioteca desenvolvida no supercomputador Cray Y-MP do CESUP/UFRGS). É realizada, também, uma comparação entre os resultados obtidos entre os diferentes métodos intervalares implementados na *selint.p*. Nesse capítulo vai se referenciar os métodos intervalares baseados em refinamento pelo nome que foi utilizado em sua implementação, ou seja, método para o caso geral (*refgeral*), método para sistemas com matrizes densas (*refdensa*) e método para sistemas com matrizes bandas (*refbanda*).

4.1 Exemplos selecionados

Foram selecionados para este trabalho, exemplos de sistemas de equações lineares encontrados na bibliografia pesquisada. Estes sistemas são, em quase toda sua totalidade, de pequeno porte, pois a biblioteca *selint.p* foi desenvolvida para computadores do tipo PC o que dificulta a resolução de sistemas de grande porte.

Foram escolhidos 23 (vinte e três) sistemas de equações lineares para serem solucionados, e seus resultados comparados. Em sua grande maioria (dezessete ao todo) os elementos da matriz A e do vetor B do sistema $AX = B$ são números reais e com base nesta característica é realizada uma comparação do resultado pontual versus o resultado intervalar. Nos sistemas na qual os elementos da matriz A e do vetor B são intervalos reais (seis ao todo) a comparação é realizada entre os resultados obtidos pela *selint.p* com os obtidos pela *libselint.a*. Todos os sistemas selecionados são apresentados no anexo 2 e numerados de 1 a 23. Esta numeração será utilizada como referência para as comparações no restante deste capítulo.

4.2 Comparação da solução: pontual \times intervalar

Para realizar a comparação da solução pontual dos sistemas selecionados com a solução intervalar obtida pelos métodos da *selint.p* são analisados dois aspectos:

- a) Se o resultado intervalar contém o resultado pontual;
- b) Se contém, qual é a qualidade do resultado intervalar obtido.

Essas comparações são apresentadas através de tabelas que descrevem os dois aspectos apresentados acima. O método *refbanda*, por ser específico a sistemas com matrizes bandas, não será comparado com os outros métodos com relação aos exemplos apresentados no anexo 2. No final deste item é realizada uma comparação com um sistema com matrizes bandas e, então, o método *refbanda* será comparado com outros três métodos, comparação que abrange o resultado obtido e o tempo de processamento que cada método levou para solucionar o sistema com matrizes bandas.

Para melhor ilustrar essas comparações, será apresentado o sistema 19 (ver anexo A-2) com sua solução pontual, e a solução obtida com a utilização da biblioteca *selint.p*.

Exemplo 19: Sendo A uma matriz real quadrada $n \times n$ e B um vetor real de ordem n, o sistema $AX = B$ é dado a seguir:

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & -1 & 1 \\ 3 & 2 & -1 \end{pmatrix} X = \begin{pmatrix} 6 \\ 3 \\ 4 \end{pmatrix}, \text{ e a solução pontual encontrada é } X = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

De acordo com os testes realizados, as soluções e o diâmetro dos resultados obtidos pelos métodos implementados na *selint.p* são apresentados na tabela 4.1.

Tabela 4.1 - Soluções intervalares do exemplo 19

Método Intervalar	Resultado obtido	Diâmetro do resultado
GAUSS s/pivot.	[9.999999999999998E-001, 1.000000000000001E+000] [1.999999999999999E+000, 2.000000000000001E+000] [2.999999999999999E+000, 3.000000000000001E+000]	3.108624468950438E-015 8.881784197001252E-016 1.776356839400250E-015
GAUSS c/pivot.	[9.999999999999998E-001, 1.000000000000001E+000] [1.999999999999998E+000, 2.000000000000002E+000] [2.999999999999998E+000, 3.000000000000002E+000]	3.219646771412954E-015 3.330669073875470E-015 2.664535259100376E-015
HANSEN2	[9.999999999999998E-001, 1.000000000000001E+000] [1.999999999999999E+000, 2.000000000000001E+000] [2.999999999999999E+000, 3.000000000000001E+000]	3.330669073875470E-016 6.661338147750939E-016 1.332267629550188E-015
HANSEN3	[9.999999999999999E-001, 1.000000000000001E+000] [1.999999999999998E+000, 2.000000000000002E+000] [2.999999999999999E+000, 3.000000000000001E+000]	1.776356839400250E-015 3.330669073875470E-015 3.996802888650564E-015
HANSEN4	[-1.2E+002, 1.3E+002] [-2.1E+002, 2.0E+002] [-2.3E+002, 2.2E+002]	2.384581180330887E+002 3.974301967218145E+002 4.371732163939959E+002
HANSEN5	[1.000000000000000E+000, 1.000000000000001E+000] [1.999999999999999E+000, 2.000000000000003E+000] [3.000000000000000E+000, 3.000000000000001E+000]	1.776356839400250E-015 2.886579864025407E-015 3.552713678800501E-015
REF.GERAL	[1.000000000000000E+000, 1.000000000000000E+000] [2.000000000000000E+000, 2.000000000000000E+000] [3.000000000000000E+000, 3.000000000000000E+000]	0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000
REF.DENSA	[1.000000000000000E+000, 1.000000000000000E+000] [2.000000000000000E+000, 2.000000000000000E+000] [3.000000000000000E+000, 3.000000000000000E+000]	0.000000000000000E+000 0.000000000000000E+000 0.000000000000000E+000

Tabela 4.4 - Análise dos resultados intervalares dos exemplos 11, 12, 15, 18 e 20 a 23

Método Intervalar	contém a solução pontual							
	11	12	15	18	20	21	22	23
GAUSS s/pivot.	sim	-	sim	sim	sim	sim	sim	sim
GAUSS c/pivot.	sim	-	sim	sim	sim	sim	sim	sim
HANSEN2	sim	sim	sim	sim	sim	sim	sim	sim
HANSEN3	sim	sim	sim	sim	sim	sim	sim	sim
HANSEN4	sim	sim	sim	sim	sim	sim	sim	sim
HANSEN5	sim	sim	sim	sim	sim	sim	sim	sim
REF.GERAL	sim	sim	sim	sim	sim	sim	sim	sim
REF.DENSA	sim	sim	sim	sim	sim	sim	sim	sim
JACOBI	sim	sim	sim	sim	sim	sim	sim	sim
GAUSS-SEIDEL	sim	sim	sim	sim	sim	sim	sim	sim

As tabelas 4.3 e 4.4 de análise de resultados permitem que se tire as seguintes conclusões:

a) Que os métodos implementados geraram resultados que continham a solução pontual do sistema;

b) Que, em alguns sistemas, os métodos de eliminação de Gauss sem e com pivotamento falharam, pois, durante a sua execução ocorreram zeros pertencentes ao intervalo da diagonal da matriz A;

c) De uma maneira geral, os resultados gerados foram considerados bons (com diâmetro pequeno) e os métodos baseados em refinamento, em alguns dos exemplos testados, apresentaram resultados considerados exatos. Com relação a esse fato, foi realizado quatro testes para certificar a exatidão desses resultados. Com o resultado obtido no sistema $AX = B$, realizou-se as seguintes operações:

- $A \times X - B$, $A \times$ (a matriz ponto médio de X) - B;
- $A \times$ (extremo inferior de X) - B;
- $A \times$ (extremo superior de X) - B,

em todos esses cálculos o resultado obtido foi zero, o que nos leva a formular a conjectura de que os resultados gerados pelos métodos baseados em refinamento, nesses exemplos, foram os resultados exatos;

d) Alguns dos métodos implementados geraram resultados intervalares com diâmetro muito grandes, o que torna o resultado inútil para a correta avaliação dos sistemas de equações lineares.

Para a comparação dos resultados intervalares, referentes aos exemplos cujos sistemas são formados por matrizes/vetores de intervalos reais, é analisado se o resultado contém a solução intervalar do sistema (nessa comparação os resultados obtidos foram

comparados com os resultados encontrados na bibliografia pesquisada, por isso, neste trabalho, não se questionará a exatidão dos resultados desses sistemas). Estas comparações são apresentadas na tabela 4.5.

Tabela 4.5 - Análise dos resultados intervalares dos exemplos 8, 10, 13, 14, 16 e 17

Método Intervalar	contém a solução pontual					
	8	10	13	14	16	17
GAUSS s/pivot.	sim	sim	sim	-	sim	sim
GAUSS c/pivot.	sim	sim	sim	-	sim	sim
HANSEN2	sim	sim	sim	sim	sim	sim
HANSEN3	sim	sim	sim	sim	sim	sim
HANSEN4	sim	sim	sim	sim	sim	sim
HANSEN5	sim	sim	sim	sim	sim	sim
REF.GERAL	não	sim	sim	sim	não	sim
REF.DENSA	sim	sim	não	sim	não	sim

Pode-se notar que através da análise da tabela 4.5, a maioria dos métodos produzem resultados que contém a solução do sistema. Algumas diferenças nos resultados foram notadas em relação aos métodos baseados em refinamento, o que ocasionou resultados diferentes dos apresentados pelos exemplos. Essas diferenças deverão ser analisadas com maior cuidado em trabalhos futuros, a fim de que se tire melhores conclusões a respeito dos resultados obtidos em sistemas cujas matriz A e vetor B sejam do tipo intervalo.

Embora as comparações não sejam o objetivo deste trabalho, em termos de velocidade de processamento, vai se apresentar os tempos obtidos com a utilização de um PC-486 de 40 Mhz, tempos obtidos por quatro métodos intervalares para a resolução de um sistema de equações lineares com matrizes bandas. O objetivo dessa comparação é mostrar a diferença de tempo de processamento entre o método *refbanda* (específico para sistemas com matrizes bandas) e três outros métodos selecionados (*hansen2*, *refgeral* e *refdensa*). Deve-se salientar que os resultados obtidos por esses três últimos métodos diferem apenas na 15ª casa decimal com relação ao resultado pelo método *refbanda*. O que difere é o tempo de processamento desses três métodos com relação ao método *refbanda*. A seguir é apresentado o sistema que foi utilizado para essa comparação, sistema de ordem igual a 51.

$$\begin{pmatrix} 1 & & & & & & \\ 0 & 1 & & & & & \\ 1 & -1.98 & 1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & & 1 & -1.98 & 1 & \\ & & & & & & \end{pmatrix} X = \begin{pmatrix} 1 \\ 0.99 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Na tabela 4.6, são apresentados os resultados obtidos pelo método *refbanda* na resolução desse sistema, e na tabela 4.7, são apresentados os tempos de processamento obtidos pelos quatro métodos testados.

Tabela 4.6 - Resultados obtidos pelo método *refbanda* e pelo método de *gauss s/pivot.*

	<i>refbanda</i>	<i>gauss s/pivot.</i>
1:	[1.000000000000000E+000, 1.000000000000000E+000]	1.000
2:	[9.899999999999999E-001, 9.900000000000002E-001]	0.990
3:	[9.601999999999999E-001, 9.602000000000003E-001]	0.960
4:	[9.111959999999998E-001, 9.111960000000004E-001]	0.911
5:	[8.439680799999997E-001, 8.439680800000004E-001]	0.844
6:	[7.598607983999997E-001, 7.598607984000004E-001]	0.760
7:	[6.605563008319996E-001, 6.605563008320004E-001]	0.661
8:	[5.480406772473596E-001, 5.480406772473604E-001]	0.548
9:	[4.24564240117772E-001, 4.24564240117774E-001]	0.425
10:	[2.92596518185829E-001, 2.92596518185831E-001]	0.293
41:	[8.129440470900618E-001, 8.129440470900625E-001]	0.813
42:	[8.869640273620158E-001, 8.869640273620163E-001]	0.887
43:	[9.432447270867294E-001, 9.432447270867298E-001]	0.943
44:	[9.806605322697085E-001, 9.806605322697088E-001]	0.981
45:	[9.984631268072933E-001, 9.984631268072935E-001]	0.998
46:	[9.962964588087321E-001, 9.962964588087323E-001]	0.996
47:	[9.742038616339963E-001, 9.742038616339967E-001]	0.974
48:	[9.326271872265805E-001, 9.326271872265809E-001]	0.933
49:	[8.723979690746331E-001, 8.723979690746337E-001]	0.872
50:	[7.947207915411930E-001, 7.947207915411937E-001]	0.795
51:	[7.011491981769291E-001, 7.011491981769298E-001]	0.701

Tabela 4.7 - Comparação referente ao tempo de processamento

Método	Tempo de processamento (em segundos)
<i>refbanda</i>	0,989
<i>refgeral</i>	7,635
<i>refdensa</i>	14,007
<i>hansen2</i>	40,539
<i>gauss s/pivot.pontual</i>	1,208

Pode-se observar, através dessas tabelas, que, embora os resultados obtidos pelos métodos sejam praticamente os mesmos, o método *refbanda* apresenta o menor tempo de processamento com relação aos outros, devido ao fato que ele foi desenvolvido especificamente para sistemas com matrizes bandas. Já o método direto *hansen2*, foi o que apresentou o maior tempo de processamento, devido a quantidade de operações algébricas que são realizadas. Em comparação com o método de Gauss sem pivotamento pontual, o método intervalar para matrizes bandas mostrou-se ainda melhor, embora trabalhe com intervalos, o que acarreta um aumento no tempo de processamento e, também, o método pontual possui uma exatidão de três dígitos, enquanto que o método *refbanda* de 16 dígitos.

Tabela 4.8 - Velocidade de processamento das operações de adição e subtração

Operação	Valores por operação em segundos	
	intervalos reais	reais
Adição	0,0000604	0,0000385
Multiplificação	0,0001620	0,0000330

Para realizar essa comparação, também foram realizados testes referentes a velocidade de processamento das operações de adição e multiplicação para números em reais e para intervalos reais. Nessa comparação, constatou-se que as operações de adição em intervalos é cerca de 56% mais demorada que em reais e, que as operações de multiplicação em intervalos, são quase quatro vezes mais demoradas do que em reais (cerca de 390%). Essas comparações são apresentadas na tabela 4.8.

4.3 Comparação da solução intervalar: *selint.p* × *libselint.a*

Neste item é realizada a comparação entre os resultados obtidos pela biblioteca *selint.p* com os resultados obtidos pela biblioteca *libselint.a*. Deve-se salientar que esta comparação é feita somente em termos de qualidade do resultado, pois uma comparação que leve em conta o tempo de processamento não é importante para essa etapa do trabalho, devido ao fato da diferença, em termos de velocidade de processamento, que existe entre os equipamentos em que foram implementadas as duas bibliotecas.

A comparação quantitativa dos resultados será importante para avaliar, em futuros trabalhos, a qualidade dos resultados produzidos pela biblioteca *libselint.a*, pois a mesma não conta ainda com todas as características que estão disponíveis na *selint.p*, como por exemplo, o produto escalar ótimo e a aritmética de alta exatidão (características disponíveis no Pascal-XSC, ferramenta na qual foi desenvolvida a biblioteca *selint.p*).

As comparações entre os resultados obtidos entre essas duas bibliotecas foram realizadas em termos de exatidão dos resultados. Observou-se que, de uma maneira geral, os resultados obtidos pelas duas bibliotecas são aproximados, mas com o detalhe que os resultados da *selint.p* são mais exatos que os da *libselint.a*, pois os resultados da *selint.p* possuem 16 dígitos de exatidão, enquanto os da *libselint.a* possuem 14 dígitos de exatidão. Conforme pode ser visto na tabela 4.9, que apresenta uma comparação entre os resultados obtidos referentes ao exemplo 1, pelo Método de Eliminação de Gauss sem pivotamento nas duas bibliotecas.

Tabela 4.9 - Comparação entre as bibliotecas *selint.p* e *libselint.a* para o exemplo 1

<i>libselint.a</i>	<i>selint.p</i>
Por hansen2	
[3.919999999997E+00, 3.920000000003E+00]	[3.919999999999E+00, 3.920000000002E+00]
[5.119999999995E+00, 5.120000000004E+00]	[5.119999999999E+00, 5.120000000001E+00]
[2.319999999997E+00, 2.320000000003E+00]	[2.319999999999E+00, 2.320000000001E+00]
Por hansen3	
[3.919999999994E+00, 3.920000000007E+00]	[3.919999999997E+00, 3.920000000005E+00]
[5.119999999996E+00, 5.120000000003E+00]	[5.119999999997E+00, 5.120000000004E+00]
[2.319999999998E+00, 2.320000000002E+00]	[2.319999999997E+00, 2.320000000003E+00]
Por hansen5	
[3.919999999995E+00, 3.920000000005E+00]	[3.9199999999956E+00, 3.9200000000036E+00]
[5.119999999994E+00, 5.120000000006E+00]	[5.11999999999872E+00, 5.1199999999946E+00]
[2.319999999997E+00, 2.320000000003E+00]	[2.31999999999616E+00, 2.3200000000008E+00]
Por gauss sem pivotamento	
[3.919999999999E+00, 3.920000000001E+00]	[3.919999999998E+00, 3.920000000002E+00]
[5.119999999999E+00, 5.120000000001E+00]	[5.119999999998E+00, 5.120000000002E+00]
[2.319999999999E+00, 2.320000000001E+00]	[2.319999999999E+00, 2.320000000001E+00]

4.4 Conclusões

Os resultados obtidos pelos métodos intervalares implementados mostraram-se, em sua grande maioria, bons resultados quando comparados com os resultados pontuais dos sistemas testados.

Nos métodos diretos, o único método que não apresentou bons resultados, foi o método de *hansen4*, pois produziu intervalos muito grandes como resultados, tornando esses resultados inúteis para os propósitos da biblioteca *selint.p*.

Observou-se que, em alguns dos exemplos testados, os resultados dos métodos iterativos foram divergentes. Em alguns exemplos, também, observou-se que os Métodos de Eliminação de Gauss com e sem pivotamento falharam na resolução do sistema, devido a existência de zeros pertencentes a algum intervalo localizado na diagonal principal da matriz *A* e, às vezes, nem com a troca de linhas e colunas o problema pôde ser solucionado.

Os métodos baseados em refinamento, em geral, foram considerados como os métodos que produziram os melhores resultados e, em alguns sistemas, produziram um resultado que é considerado aritmeticamente exato, conforme foi apresentado no item 4.2.

Outro fato a destacar foi a comparação, em termos de velocidade de processamento, realizada com um sistema com matrizes bandas. Essa análise demonstrou que o método específico para esse tipo de sistema (*refbanda*), resolve o mesmo em um tempo muito menor do que em relação aos métodos comparados (*hansen2*, *refgeral*, *refdensa* e *gauss/pivot.pontual*).

A comparação da biblioteca implementada neste trabalho com a biblioteca *libselint.a*, demonstrou que os resultados obtidos em ambas as bibliotecas foram aproximados, mas os obtidos pela biblioteca *selint.p* possuem uma maior exatidão, devido as características disponíveis no compilador Pascal-XSC, características que ainda não estão todas disponíveis na biblioteca *libselint.a*. Um ponto negativo da biblioteca *selint.p*, é o fato que a resolução de sistemas de grande porte pode tornar-se inviável devido a capacidade de processamento do equipamento no qual ela foi implementada, fato que não ocorre com a *libselint.a*.

5 Conclusões

Neste capítulo é apresentado uma visão geral do trabalho realizado, destacando seu desenvolvimento e os resultados que foram obtidos com sua elaboração. São apresentadas as conclusões finais do trabalho, conclusões referentes aos métodos intervalares implementados e aos resultados que por eles foram obtidos. Ao final deste capítulo são apresentadas algumas limitações do trabalho e propostas para trabalhos futuros.

5.1 Resumo do Trabalho

O trabalho sobre métodos intervalares para a resolução de sistemas de equações lineares originou-se através de estudos desenvolvidos junto ao grupo de Matemática Computacional da UFRGS. Foram estudados, primeiramente, os tipos de sistemas de equações lineares e os métodos numéricos pontuais utilizados para sua resolução. Neste estudo foi abordado ainda aspectos referentes a instabilidade numérica e a proximidade da singularidade em sistemas de equações lineares. Estes estudos estão descritos em [HÖL94, HÖL94a e HÖL94b].

Em seguida foram realizados estudos referentes a versão intervalar dos métodos pontuais e também sobre a aplicabilidade de sistemas de equações lineares ([HOL95]), o que serviu de base para a realização deste trabalho.

Este trabalho foi realizado em duas etapas. A primeira consistiu na identificação e caracterização das metodologias de desenvolvimento dos métodos intervalares. Estas metodologias ou grupos de métodos foram classificados como: métodos intervalares baseados em operações algébricas ou métodos diretos; métodos intervalares baseados em iteração e métodos intervalares baseados em refinamento ou métodos híbridos. Com a definição destas metodologias os métodos intervalares pesquisados foram agrupados nestes três grupos.

Deve-se salientar que não é objetivo deste trabalho a elaboração de novos métodos intervalares, mas sim a caracterização de alguns desses métodos visando com isso a elaboração de ferramentas que possibilitem a utilização desses métodos por usuários das mais diversas áreas do conhecimento e também que possibilitem a disseminação da cultura do uso de intervalos na matemática aplicada, engenharias, física, química e demais ciências que dessas ferramentas necessitem. Juntamente com a descrição dos métodos intervalares foram elaborados seus respectivos algoritmos.

Com esses estudos tornou-se possível se dar início a segunda etapa do trabalho, etapa que consistiu da elaboração de uma biblioteca aplicativa intervalar para a resolução de

sistemas de equações lineares, biblioteca esta denominada de *selint.p*. Esta biblioteca foi desenvolvida em um PC-486 e implementada utilizando o compilador Pascal-XSC, que entre suas características tem disponível a aritmética intervalar, aritmética de alta exatidão, produto escalar ótimo e a verificação automática do resultado.

Através do estudo dos métodos intervalares e suas respectivas metodologias de desenvolvimento a biblioteca *selint.p* foi estruturada modularmente, na qual cada módulo refere-se a uma metodologia ou grupo de métodos.

- a) Módulo *dirint*, incluindo métodos baseados em operações algébricas intervalares e propriedades intervalares. Também são conhecidos como métodos diretos intervalares;
- b) Módulo *refint*, incluindo métodos baseados em inclusões ou refinamentos intervalares da solução e do erro. Também podem ser chamados de métodos híbridos, uma vez que se pode utilizar a solução inicial calculada por métodos pontuais ou a matriz inversa pontual;
- c) Módulo *itrint*, incluindo métodos iterativos intervalares, conhecidos também como métodos de relaxação. Eles também se baseiam em inclusões monotônicas.

Além destes três módulos foi definido um quarto módulo, *equalg*, que deve conter rotinas do caso particular de sistema linear de ordem um, ou seja, resolução algébrica de equações por métodos intervalares, como as versões intervalares do método de Newton. Deve-se destacar, ainda, que para a implementação da biblioteca *selint.p* foi necessário a utilização dos módulos aritméticos avançados do Pascal-XSC (ver [HÖL95]), módulos que contém a aritmética intervalar e suas respectivas operações com matrizes e vetores de intervalos.

A documentação da biblioteca aplicativa intervalar é constituída do manual de utilização da *selint.p*. Ele é constituído por duas partes, a primeira contém informações sobre o uso e vantagens de utilização da biblioteca e a outra parte contém a documentação das rotinas que compõem essa biblioteca. A primeira parte visa atender um usuário que deseja maiores informações da biblioteca, enquanto que a segunda parte visa atender ao uso esporádico.

Na documentação das rotinas da biblioteca *selint.p* é usado o mesmo formato usado na documentação dos módulos aritméticos do Pascal-XSC. A documentação da biblioteca *selint.p* segue o padrão encontrado nos módulos aritméticos I_ARI, C_ARI, CI_ARI, MV_ARI, MVI_ARI, MVC_ARI, e MVCI_ARI. Este padrão estabelece que a biblioteca seja dividida em módulos, que são compostos por rotinas de um mesmo grupo.

Após esta implementação foram selecionados exemplos de sistemas de equações lineares a serem resolvidos pela biblioteca *selint.p* a fim de poder ter condições de analisar e

comparar os resultados obtidos pelos métodos intervalares implementados. Foram comparados os resultados intervalares com os resultados pontuais e os resultados da *selint.p* com os resultados da biblioteca *libselint.a*. Através destas comparações pode-se realizar uma análise de desempenho em termos quantitativos da biblioteca *selint.p*.

5.2 Considerações Finais

Neste item são apresentados as conclusões obtidas com a realização deste trabalho. Primeiramente procurou-se mostrar que o uso de métodos intervalares pode verificar a existência ou não existência da solução de sistemas de equações lineares, o que pode não acontecer com os métodos pontuais.

Outro aspecto a destacar é referente aos métodos baseados em refinamento, os quais utilizam a aritmética pontual para calcular uma aproximação da solução do sistema e depois realiza o refinamento intervalar desta aproximação. Este fato é importante pois em muitos casos os cálculos com matrizes e vetores de intervalos podem ser muito dispendiosos.

Quanto a implementação encontrou-se algumas dificuldades nos métodos iterativos, dificuldades quanto a convergência do método e a escolha da aproximação inicial da solução que será utilizada pelos métodos iterativos intervalares. Observou-se em alguns dos exemplos testados que o resultado do método iterativo é divergente. Devido a isto, estas dificuldades deverão ser estudadas e analisadas mais aprofundadamente na continuação deste trabalho.

Pode-se observar no Método de Eliminação de Gauss com e sem pivotamento o cuidado que deve-se ter com a estrutura da matriz A do sistema $AX = B$, pois se houver um zero pertencente ao intervalo localizado na diagonal principal dessa matriz, o método de Gauss falha. Isso aconteceu em alguns exemplos e às vezes nem com a troca de linhas e colunas o problema pôde ser solucionado.

No que se refere aos resultados obtidos pode-se observar que de uma maneira geral os métodos baseados em refinamento produzem melhores resultados comparados aos obtidos pelos métodos diretos e iterativos, embora alguns dos métodos diretos produzam resultados muito próximos aos resultados obtidos pelos métodos baseados em refinamento.

Alguns dos métodos, em especial o método direto de *hansen4*, embora produza resultados intervalares que contenham a solução do sistema o diâmetro do resultado que por eles foi gerado é considerado muito grande o que o torna impróprio para poder se avaliar a solução do sistema.

Na comparação da biblioteca *selint.p* com a biblioteca *libselint.a* observou-se que os resultados obtidos, embora aproximados, possuem uma maior exatidão na biblioteca *selint.p*, devido as características da ferramenta utilizada na implementação da *selint.p*.

Ainda em comparação com a biblioteca *libselint.a*, um ponto negativo da biblioteca *selint.p* é o fato que a resolução de sistemas de grande porte pode tornar-se inviável devido a capacidade de processamento do equipamento no qual foi implementada a *selint.p*, fato este que não ocorre com a *libselint.a*, pois a mesma foi desenvolvida no supercomputador Cray Y-MP do CESUP/UFRGS.

Por fim concluiu-se que os resultados obtidos pela *selint.p* são melhores quando comparados com os da *libselint.a* e que na sua grande maioria os métodos intervalares implementados continham a solução exata do sistema. Este fato não impede que sejam realizadas melhorias na *selint.p*, bem como a inclusão de novos métodos intervalares, possível graças a característica modular da biblioteca.

5.3 Proposta para Novos Trabalhos

As propostas para novos trabalhos poderão ser apresentados em quatro diferentes itens:

- a) melhorias na biblioteca *selint.p*;
- b) melhorias na biblioteca *libselint.a*;
- c) estudos de métodos paralelos para a resolução de sistemas de equações lineares e a elaboração da versão intervalar dos mesmos;
- d) desenvolvimento de novos métodos intervalares.

Pôde-se notar com o desenvolvimento da biblioteca *selint.p* que alguns aspectos necessitam de um maior aprofundamento em seu estudo, como é o caso dos métodos intervalares iterativos. Outro detalhe é que a biblioteca *selint.p* utiliza apenas números reais, não aceitando com isso sistemas que possuam números complexos e este é um ponto a ser estudado e implementado posteriormente.

Com relação a biblioteca *libselint.a* poderá ser realizada as mesmas alterações citadas para a *selint.p* como também um estudo de um melhor aproveitamento das características e potencialidades existentes no supercomputador Cray Y-MP.

Aproveitando estes estudos deverá realizar-se pesquisas que abordem os métodos paralelos para a resolução de sistemas de equações lineares. Neste estudo procurar-se-á identificar as características dos mesmos, seus algoritmos de desenvolvimento e sua paralelização. Após realizado essa etapa do estudo poderão ser feitas pesquisas referentes a versão intervalar dos métodos paralelos, visando com isso identificar problemas na versão

intervalar desses métodos, como por exemplo a sua paralelização e o custo computacional da utilização de matrizes e vetores de intervalos em um ambiente de desenvolvimento paralelo.

Deverão, também, ser desenvolvidos novos métodos intervalares e com isso seus resultados comparados com os resultados que tem-se das bibliotecas *selint.p* e *libselint.a*. E com a disponibilidade de uma das bibliotecas em um supercomputador torna-se viável a resolução de aplicações reais que utilizam sistemas de equações lineares de grande porte e que por isso exigem um equipamento que possua uma grande velocidade de processamento.

Anexo 1: Algoritmos dos Métodos

1.1: Algoritmos dos Métodos Intervalares do Módulo *equalg*

a) Método para a resolução da equação $A + X = B$

Entrada de dados: A, B (intervalos) Calcular $X = B - A$

b) Método para a resolução da equação $AX + B = C$

Entrada de dados: A, B, C (intervalos) Calcular $X = (C - B) / A$, $0 \notin A$

c) Método para a resolução da equação $AX + BX = C$

Entrada de dados: A, B, C (intervalos) Calcular $X = C / (A + B)$
--

d) Método para a resolução da equação $AX + B = CX + D$

Entrada de dados: A, B, C, D (intervalos) Calcular $X = (D - B) / (A - C)$

e) Método de Newton Intervalar

Entrada de dados: X e X^0 (intervalos) $N_i(X) = X - f(X)/M$, onde $0 \notin M$ $X^{k+1} := N_i(X^k) \cap X^k$ para $k=1,2,\dots,n$
--

1.2: Algoritmos dos Métodos Intervalares do Módulo *dirint*

a) Método de Hansen2

Entrada de dados: A^I, B^I
 Calcular matriz centro de A^I , denominada A_c
 Calcular matriz centro de B^I , denominada B_c
 Calcular $Y = (A_c)^{-1} \cdot B_c$
 Calcular matriz inversa de A^I
 Calcular $Z^I = (B^I - A^I \cdot Y) \cdot (A^I)^{-1}$
 Calcular $X^I = Z^I + Y$ (solução do sistema)

b) Método de Hansen3

Entrada de dados: A^I, B^I
 Calcular matriz centro de A^I , denominada A_c
 Calcular matriz inversa pontual de A_c
 Calcular matriz inversa de A
 Calcular $X^I = (A^I)^{-1} \cdot B^I$ (solução do sistema)

c) Método de Hansen4

Entrada de dados: A^I, B^I
 Calcular matriz centro de A^I , denominada A_c
 Calcular matriz inversa pontual de A_c
 Calcular $Y^I = (A_c)^{-1} \cdot A^I$
 Calcular $Z^I = (A_c)^{-1} \cdot B^I$
 Calcular matriz inversa intervalar de Y^I
 Calcular $X^I = (Y^I)^{-1} \cdot Z^I$ (solução do sistema)

d) Método de Hansen5

Entrada de dados: A^I, B^I
 Calcular matriz centro de A^I , denominada A_c
 Calcular matriz centro de B^I , denominada B_c
 Calcular $Y = (A_c)^{-1} \cdot B_c$
 Calcular $J^I = (A_c)^{-1} \cdot A^I$
 Calcular $K^I = (A_c)^{-1} \cdot (B^I - A^I \cdot Y)$
 Calcular matriz inversa intervalar de J^I
 Calcular $Z^I = K^I \cdot (J^I)^{-1}$
 Calcular $X^I = Y + Z^I$ (solução do sistema)

e) Método de Eliminação de Gauss

```

%% Etapa 1 - Triangularização
Entrada de Dados: AI, BI, n
Para k = 1(1)n - 1
  Para i = k + 1(1)n
    MI[i,k] ← -AI[i,k] / AI[k,k]
    Para j = k + 1(1)n
      AI[i,j] ← AI[i,j] + MI[i,k] * AI[k,j]
      YI[i] ← YI[i] + MI[i,k] * YI[k]
%% Etapa 2 - Retrossubstituição
XI[n] ← YI[n] / AI[n,n]
Para k = n - 1(-1)1
  XI[k] ← YI[k]
  Para i = k + 1(1)n
    XI[k] ← XI[k] - AI[k,i] * XI[i]
  XI[k] ← XI[k] / AI[k,k]
Saída [ XI[i] | i = 1(1)n | ]

```

f) Método de Eliminação de Gauss com Pivotamento

```

Entrada de Dados: AI, BI, n
Escolha do pivô
Se necessário troca linhas e/ou colunas
Utilizar algoritmo de Gauss sem pivotamento
Saída [ XI[i] | i = 1(1)n | ]

```

h) Cálculo da Matriz Inversa por Hansen

Entrada de Dados: A^{-1} , n
 Calcular a matriz pontual centro de A
 Calcular a aproximação da inversa pontual centro de A
 Calculo da matriz inversa intervalar:

{Passo 1}

$$E \leftarrow I - A \cdot A^{-1}$$

{Passo 2}

Para $k = 1(1)n$

Para $j = 1(1)n$

se $\text{abs}(\text{inf}(E[k,j])) \geq \text{abs}(\text{sup}(E[k,j]))$ então $Z[k,j] \leftarrow \text{abs}(\text{inf}(E[k,j]))$

senão $Z[k,j] \leftarrow \text{abs}(\text{sup}(E[k,j]))$

$\text{sumc} \leftarrow 0$; $r \leftarrow 0$

Para $k = 1(1)n$

Para $j = 1(1)n$

$\text{sumc} \leftarrow \text{sumc} + Z[j,k]$;

se $r < \text{sumc}$ então $r \leftarrow \text{sumc}$

$bl \leftarrow ((r * r) * r) / (1 - r)$

Para $k = 1(1)n$

Para $j = 1(1)n$

$P[k,j] \leftarrow \text{intval}(-bl, bl)$

{Passo 3}

$$S \leftarrow I + E * (I + E * (I + E))$$

{Passo 4}

$$C \leftarrow A^{-1} * (S + P)$$

1.3: Algoritmos dos Métodos Intervalares do Módulo *refint*

a) Método para Sistemas em Geral

```

1. {Cálculo de uma solução aproximada}
   Matinv(A,R,Err);
   if (Err ≠ "Sem erro") then
     return Err := "Matriz é provavelmente singular!"

2. {Iteração de resíduo real para uma solução aproximada}
   Kmax := 10; K := 0;  $\tilde{x}^U := R \cdot B$ ;
   repeat
     D :=  $\square(B - A \cdot \tilde{x}^K)$ ;
      $\tilde{x}^{K+1} := \square(\tilde{x}^K + R \cdot D)$ ;
     CheckForZeros( $\tilde{x}^K, \tilde{x}^{K+1}$ );
     Success := Accurate( $\tilde{x}^K, \tilde{x}^{K+1}$ );
     k := k + 1;
   until Success or (k ≥ Kmax);
    $\tilde{x} = \tilde{x}^K$ ;

3. {Cálculo de inclusões C e Z para C = I - RA e Z = R(B - A $\tilde{x}$ )}
   C :=  $\diamond(I - R \cdot A)$ ;
   D :=  $\square(B - A \cdot \tilde{x})$ ;
   D :=  $\diamond(B - A \cdot \tilde{x} - D)$ ;
   Z :=  $\diamond(R \cdot D + R \cdot D)$ ;

4. {Passo de verificação}
   if (Z = 0) then
     X :=  $\tilde{x}$ ; {Solução exata}
   else
     VerificationStep(X, Z, C, IsVerified);
     if (not IsVerified) then
       Err := "Verificação falhou, o sistema está provavelmente
mal-condicionado!";
       return Err;
     else
       X :=  $\tilde{x} + X$ ; {Aproximação mais correção intervalar}

5. {Return: Inclusão e código de erro}
   return X, Err;

```

b) Método para Sistemas com Matrizes Densas

{Calcula a inclusão para a solução de um sistema linear quadrado $AX = B$ }

Parte 1: cálculo da aproximação da matriz inversa pontual de A

- 1.1 calcule uma aproximação R da inversa de A
- 1.2 calcule uma aproximação $\tilde{X} = R \cdot B$ de X
melhora \tilde{X} por uma correção de defeito iterativa:
repeat
 $\tilde{X} = \tilde{X} + R(B - A\tilde{X})$
until $\tilde{X} <$ exatidão ou contador de iterações for excedido
- 1.3 calcula a inclusão para o resíduo:
 $Z = R \diamond (B - A\tilde{X})$
e para a iteração da matriz:
 $C = \diamond(I - R \cdot A)$
- 1.4 Iteração intervalar
 $Y = Z$
repeat
 $Y_A = \text{blow}(Y, \epsilon)$ { ϵ - inflação}
 $Y = Z + C \cdot Y_A$
until $Y \subset \text{int}(Y_A)$ ou contador de iteração for excedido
- 1.5 If $Y \subset \text{int}(Y_A)$ then
existe uma solução única X e $X \in \tilde{X} + Y$
else
if na Parte 1 then
Parte 1 falhou, vá para Parte 2 com $R_1 = R$
else
algoritmo falhou, matriz A mal-condicionada ou singular

Parte 2: aproximação da inversa de tamanho duplo

- 2.1 calcule a aproximação da inversa $R = R_1 + R_2$ de A:
 $S = R_1 \cdot A$
calcule uma aproximação da inversa S1 para S
 $S = S_1 \cdot R_1$
 $R_2 = S_1 \cdot R_1 - S$
 $R_1 = S$
- 2.2 vá para 1.2 da Parte 1

c) Método para Sistemas com Matrizes Bandas

entrada de dados: A,B,tamanho das bandas
calcula a matriz inversa pela técnica de decomposição LU e o erro $LU - A$
calcula a aproximação da solução X_{app}
calcule o erro: $B - AX_{app}$ da solução aproximada X_{app}
calcule $Z = \text{erro}$
calcule $X = X_{app} + Z$
saída: X (solução obtida pelo método para sistemas com matrizes bandas)

d) Cálculo da Matriz Inversa

{Calcula inclusão para a inversa da matriz quadrada A}

1. (resolva $AX = I$, coluna por coluna)
para $i := 1$ até n faça
 inicio
 $B_i = E_i$ (=i-th vetor unitário)
 resolva $Ax_i = B_i$ pelo uso do algoritmo do
 método para sistemas com matrizes densas
 fim
2. $X = (x_1, \dots, x_n)$ é a inclusão procurada

1.4: Algoritmos dos Métodos Intervalares do Módulo *itrint*

a) Método de Jacobi

Entrada $\{A, B \mid i = 1(1)n, j = 1(1)n\}$
 Entrada $\{\text{limite}, X \mid i = 1(1)n\}$

$k \leftarrow 1$
 Enquanto $k < \text{limite}$ faça

$k \leftarrow k + 1$
 $X^{k+1} = AX^k + B$

Saída $\{X\}$

$i = 1(1)n$

$X[i] \leftarrow B[i]$

b) Método de Gauss-Seidel

Entrada $\{A, B \mid i = 1(1)n, j = 1(1)n\}$
 Entrada $\{\text{limite}, X \mid i = 1(1)n\}$

$k \leftarrow 1$
 Enquanto $k < \text{limite}$ faça

$k \leftarrow k + 1$

Saída $\{k\}$

$i = 1(1)n$

$B[i] \leftarrow V[i]$

$j = 1(1)n$

Se $j \neq i$ então

$B[i] \leftarrow B[i] - A[i, j] * X[j]$

$B[i] \leftarrow B[i] / A[i, i]$

Saída $\{B[i]\}$

$i = 1(1)n$

$X[i] \leftarrow B[i]$

Exemplo 3: (sistema apresentado em [HÖL95])

$$\alpha = 0.7071, \beta = 0.8660 \text{ e } \delta = 0.5$$

$$A = \begin{pmatrix} \alpha & 0 & 0 & -1 & -\beta & 0 & 0 & 0 & 0 \\ \alpha & 0 & 1 & 0 & -\delta & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -\alpha \\ 0 & 0 & 0 & 0 & \beta & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -\delta & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ -1000 \\ 0 \\ 0 \\ 500 \\ 0 \\ 0 \\ -500 \\ 0 \end{pmatrix}$$

Solução do exemplo 3

X1 =	-1.035280949936208E+002
X2 =	7.320471596998927E+001
X3 =	0.000000000000000E+000
X4 =	-2.679528403001069E+001
X5 =	-5.359056806002145E+001
X6 =	7.320471596998927E+001
X7 =	5.267952840300107E+002
X8 =	2.679528403001069E+001
X9 =	-3.789461749400466E+001

Exemplo 5: (sistema apresentado em [HÖL95])

$$A = \begin{bmatrix} \alpha & 0 & 0 & -1 & -\alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha & 0 & 1 & 0 & +\alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & -\alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +\alpha & 1 & 0 & 0 & -\alpha & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\alpha & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha & 0 & 0 & -1 & -\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\alpha & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -10 \\ 0 \\ 0 \\ -15 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -10 \\ 0 \\ 0 \end{bmatrix}$$

Solução do exemplo 5

X1 =	-1.414227124876255E+001
X2 =	2.500000000000000E+001
X3 =	1.000000000000000E+001
X4 =	-1.000000000000000E+001
X5 =	0.000000000000000E+000
X6 =	2.500000000000000E+001
X7 =	1.500000000000000E+001
X8 =	-2.500000000000000E+001
X9 =	2.121340687314383E+001
X10 =	1.000000000000000E+001
X11 =	0.000000000000000E+000
X12 =	-1.000000000000000E+001
X13 =	0.000000000000000E+000
X14 =	1.000000000000000E+001
X15 =	1.000000000000000E+001
X16 =	-1.414227124876255E+001
X17 =	1.000000000000000E+001

Exemplo 6:

$$A = \begin{pmatrix} 2.4759 & 1.6235 & 4.6231 \\ 1.4725 & 0.9589 & -1.3253 \\ 2.6951 & 2.8965 & -1.4794 \end{pmatrix}, B = \begin{pmatrix} 0.00647 \\ 1.0473 \\ -0.6789 \end{pmatrix}, X = \begin{pmatrix} 1.8282 \\ -2.0646 \\ -0.2527 \end{pmatrix}$$

Exemplo 7:

$$A = \begin{pmatrix} -20 & -20 & -34 \\ -30 & 23 & 1 \\ -23 & 2 & -2 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, X = \begin{pmatrix} -0.143 \\ -0.101 \\ -0.0382 \end{pmatrix}$$

Exemplo 8: (sistema apresentado em [HAN67])

$$A = \begin{pmatrix} [2,3] & [0,1] \\ [1,2] & [2,3] \end{pmatrix}, B = \begin{pmatrix} [0,120] \\ [60,240] \end{pmatrix}, X = \begin{pmatrix} [-120,90] \\ [-60,240] \end{pmatrix}$$

Exemplo 9: (sistema apresentado em [RUM94])

$$A = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 2\varepsilon & 2\varepsilon \\ 1 & 2\varepsilon & -\varepsilon \end{pmatrix}, B = \begin{pmatrix} 3 + 3\varepsilon \\ 6\varepsilon \\ 2\varepsilon \end{pmatrix}, X = \begin{pmatrix} \varepsilon \\ 1 \\ 1 \end{pmatrix}$$

Exemplo 10: (sistema apresentado em [HAN69])

$$A = \begin{pmatrix} [4.325,4.335] & [-1.125,-1.115] & [-1.085,-1.075] & [1.135,1.145] \\ [-1.125,-1.115] & [4.325,4.335] & [0.235,0.245] & [-1.225,-1.215] \\ [-1.085,-1.075] & [0.235,0.245] & [7.205,7.215] & [-3.225,-3.215] \\ [1.135,1.145] & [-1.225,-1.215] & [-3.225,-3.215] & [5.425,5.435] \end{pmatrix}, B = \begin{pmatrix} [3.515,3.525] \\ [1.565,1.575] \\ [0.535,0.545] \\ [-1.095,-1.085] \end{pmatrix}$$

$$X' = \begin{pmatrix} [1.0407,1.0518] \\ [0.55667,0.56892] \\ [0.10563,0.11639] \\ [-0.23523,-0.22104] \end{pmatrix}$$

Exemplo 11: (sistema apresentado em [HAM93])

$$A = \begin{pmatrix} 5 & 10 & 10 & 5 & 1 \\ 15 & 40 & 45 & 24 & 5 \\ 35 & 105 & 126 & 70 & 15 \\ 70 & 224 & 280 & 160 & 35 \\ 126 & 420 & 540 & 315 & 70 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}, X = \begin{pmatrix} 0 \\ 1 \\ -2 \\ 3 \\ -4 \end{pmatrix}$$

Exemplo 12: (sistema apresentado em [GEÖ93])

Matriz A:

10	45	120	210	252	210	120	45	10	1
55	330	990	1848	2310	1980	1155	440	99	10
220	1485	4752	9240	11880	10395	6160	2376	540	55
715	5148	17160	4320	45045	40040	24024	9360	2145	220
2002	15015	51480	105105	140140	126126	76440	30030	6930	715
5005	38610	135135	280280	378378	343980	210210	83160	19305	2002
11440	90090	320320	672672	917280	840840	517440	205920	48048	5005
24310	194480	700128	1485120	2042040	1884960	1166880	466752	109395	11440
48620	393822	1432080	3063060	4241160	3938220	2450448	984555	231660	24310
92378	755820	2771340	5969040	8314020	7759752	4849845	1956240	461890	48620

$B = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)^T$, solução pontual encontrada $X = (0, 1, -2, 3, -4, 5, -6, 7, -8, 9)^T$.

Exemplo 13: (exemplo apresentado em [NEU90])

$$A = \begin{pmatrix} 2 & [-1, 0] \\ [-1, 0] & 2 \end{pmatrix}, B = \begin{pmatrix} 1.2 \\ -1.2 \end{pmatrix}, X = \begin{pmatrix} [0.3, 0.6] \\ [-0.6, -0.3] \end{pmatrix}$$

Exemplo 14: (exemplo apresentado em [NEU90])

$$A = \begin{pmatrix} [0, 1] & 0.1 \\ 0.1 & [1, 2] \end{pmatrix}, B = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, X = \begin{pmatrix} [-1, 1] \\ [-1, 1] \end{pmatrix}$$

Exemplo 15: (exemplo apresentado em [NEU90])

$$A = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}, B = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, X = \begin{pmatrix} [-1, 1] \\ [-1, 1] \end{pmatrix}$$

Exemplo 16: (exemplo apresentado em [NEU90])

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}, B = \begin{pmatrix} [-1,1] \\ [-1,1] \\ 0 \end{pmatrix}, X = \begin{pmatrix} [-1,1] \\ [-2,2] \\ [2,3] \end{pmatrix}$$

Exemplo 17: (exemplo apresentado em [NEU90])

$$A = \begin{pmatrix} 2 & -1 \\ [-6,-2] & 3.5 \end{pmatrix}, B = \begin{pmatrix} -2 \\ 5 \end{pmatrix}, X = \begin{pmatrix} [-2,2] \\ [-2,6] \end{pmatrix}$$

Exemplo 18: (exemplo apresentado em [CLA89])

$$A = \begin{pmatrix} 3 & 2 & 0 & 1 \\ 9 & 8 & -3 & 4 \\ -6 & 4 & -8 & 0 \\ 3 & -8 & 3 & -4 \end{pmatrix}, B = \begin{pmatrix} 3 \\ 6 \\ -16 \\ 18 \end{pmatrix}, X = \begin{pmatrix} 2 \\ -1 \\ 0 \\ -1 \end{pmatrix}$$

Exemplo 19: (exemplo apresentado em [CLA89])

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 2 & -1 & 1 \\ 3 & 2 & -1 \end{pmatrix}, B = \begin{pmatrix} 6 \\ 3 \\ 4 \end{pmatrix}, X = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Exemplo 20: (exemplo apresentado em [CLA89])

$$A = \begin{pmatrix} 5 & 0 & 0 & -3 & -1 \\ -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 2 & -1 & 0 \\ -1 & 0 & 0 & 4 & -2 \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix}, B = \begin{pmatrix} 2 \\ 3 \\ -1 \\ 0 \\ -1 \end{pmatrix}, X = \begin{pmatrix} 0.09 \\ 0.61 \\ -0.65 \\ -0.30 \\ -0.65 \end{pmatrix}$$

Exemplo 21: (exemplo apresentado em [CLA89])

$$A = \begin{pmatrix} 4 & -2 & -1 & 0 \\ -2 & 9 & 0 & -5 \\ -1 & 0 & 6 & -3 \\ 0 & -5 & -3 & 10 \end{pmatrix}, B = \begin{pmatrix} 10 \\ -5 \\ 0 \\ 0 \end{pmatrix}, X = \begin{pmatrix} 2.76 \\ 0.22 \\ 0.61 \\ 0.29 \end{pmatrix}$$

Exemplo 22: (exemplo apresentado em [CLA89])

$$A = \begin{pmatrix} 10 & 5 & 2 & 1 \\ 5 & 10 & -3 & 2 \\ 2 & -3 & 12 & -5 \\ 1 & 2 & -5 & 12 \end{pmatrix}, B = \begin{pmatrix} 18 \\ 14 \\ 6 \\ 10 \end{pmatrix}, X = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Exemplo 23:

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{pmatrix}, B = \begin{pmatrix} 7 \\ 8 \\ 5 \end{pmatrix}, X = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}$$

Bibliografia

- [ADA93] ADAMS,E.; KULISCH,U. (Eds.). **Scientific Computing with automatic result verification**. San Diego: Academic Press, 1993. 612p.
- [ALE83] ALEFELD,G.; HERZBERGER,J. **Introduction to Interval Computations**. New York: Academic Press, 1983. 333p.
- [AYR71] AYRES Jr, F. **Matrizes**. São Paulo: McGraw-Hill, 1971. 286p.
- [BAR87] BARROSO, L.C. et al. **Cálculo numérico (com aplicações)**. São Paulo: Habra, 1987. 367p.
- [CLA89] CLÁUDIO, D.M.; MARINS, J.M. **Cálculo Numérico Computacional: teoria e prática**. São Paulo: Atlas, 1989. 464p.
- [DEI86] DEIF,A. **Sensitivity Analysis in Linear Systems**. Berlim: Springer-Verlag, 1986.
- [DEM85] DEMMEL, J.W.; KRUCKEBERG, F. An Interval Algorithm for Solving Systems of Linear Equations to Prespecified Accuracy. **Computing**, New York, v.34, p.117-129, 1985.
- [DIV86] DIVERIO,T.A. **Software Numérico Aplicativo e Instrucional**. Porto Alegre: CPGCC da UFRGS, 1986. Dissertação de mestrado.
- [DIV90] DIVERIO, T.A. **LEPMAC - Material Didático de Apoio - módulo de SELAS**. Porto Alegre: GRUPO DE MATEMATICA COMPUTACIONAL, CPGCC da UFRGS, 1990. 61p.
- [DIV91] DIVERIO, T. A. et al. **Versões intervalares do método de Newton**. Porto Alegre: CPGCC da UFRGS, 1991. 86p. (RP - 161).

- [DIV94] DIVERIO, T.A. **PASCAL XSC - Pascal para computação científica: descrição, instalação e aplicações.** Porto Alegre: CPGCC da UFRGS, 1994. 65p. (RP-245).
- [DIV95] DIVERIO, T. A. **Uso efetivo da matemática intervalar em supercomputadores vetoriais.** Porto Alegre: CPGCC da UFRGS, 1995. 290p. Tese de doutorado.
- [DIV95a] DIVERIO, T. A. **LIBAVIA Manual de utilização.** Porto Alegre: CPGCC da UFRGS, 1995. 430p.
- [DIV95b] DIVERIO, T. A. **LIBAVIA Verificação e Validação.** Porto Alegre: CPGCC da UFRGS, 1995. 217p.
- [DOR81] DORN, W.S.; McCracken, D.D. **Cálculo numérico com estudos de casos em Fortran IV.** Rio de Janeiro: Campus, 1981. 568p.
- [DUF86] DUFF, I.S.; ERISMAN, A.M.; REID, J.K. **Direct Methods for Sparse Matrices.** Oxford: Oxford University Press, 1986. 341p.
- [DUF88] DUFF, I.S. et al. Sparsity structure and Gaussian elimination. **ACM SIGNUM NEWLETTER**, New York, v.23, p.2-8, 1988.
- [FER94] FERNANDES, U.A.L.; DIVERIO, T.A. **Aplicações de Intervalos.** Porto Alegre: CPGCC da UFRGS, 1994. 64p. (RP- 235).
- [FOR67] FORSYTHE, G.; MOLLER, C.B. **Computer solution of linear algebraic systems.** Englewood Cliffs: Prentice Hall, 1967.
- [FOR77] FORSYTHE, G.E.; MALCOLM, M.A.; MOLLER, C.B. **Computer Methods for mathematical computations.** Englewood Cliffs: Prentice Hall, 1977. 259p.
- [GEÖ93] GEÖRG, S. et al. **PASCAL-XSC: A PASCAL Extension for Scientific Computation and Numerical Data Processing.** Karlsruhe: Institute of Applied Mathematics of the University of Karlsruhe (Germany), 1993. 38p.

- [HAH85] HAHN, W.; MOHR, K.; SCHAUER, U. Some Techniques for Solving Linear Equations Systems with Guarantee. **Computing**, New York, v.34, p.375-379, 1985.
- [HAM62] HAMMING, R.W. **Numerical methods for scientists and engineers**. New York: McGraw Hill, 1962.
- [HAM93] HAMMER, R. et al. **Numerical Toolbox for Verified Computing I: basic numerical problems**. Berlin: Springer-Verlag, 1993. 337p.
- [HAN65] HANSEN, E. Interval arithmetic in matrix computation Part 1. **SIAM J NUMER ANAL**, Philadelphia, v.2, p.308-320, 1965.
- [HAN67] HANSEN, E.; SMITH, R. Interval arithmetic in matrix computation Part 2. **SIAM J NUMER ANAL**, Philadelphia, v.4, p.1-9, 1967.
- [HAN69] HANSEN, E. On the Solution of Linear Algebraic Equations with Interval Coefficients. **Linear Algebra Appl.**, New York, v.2, p.153-165, 1969.
- [HAN81] HANSEN, E.; SENGUPTA, S. Bounding Solutions of Systems of Equations Using Interval Analysis. **Bit**, Copenhagen, v.21, p.203-221, 1981.
- [HÖL93] HÖLBIG, C.A.; DIVERIO, T.A. **Aritmética Intervalar - uma nova abordagem**. Porto Alegre: CPGCC da UFRGS, 1993.
- [HÖL94] HÖLBIG, C.A.; DIVERIO, T.A.; NORONHA, C.R. **Sistema de Ponto-Flutuante e o Padrão IEEE 754**. Porto Alegre: CPGCC da UFRGS, 1994. 58p. (RP-225).
- [HÖL94a] HÖLBIG, C.A.; FERNANDES, U.A.L.; DIVERIO, T.A. **Sistemas de Equações Lineares: Instabilidade, Análise Sensitiva e Métodos Intervalares**. Porto Alegre: CPGCC da UFRGS, 1994. 56p. (RP-228).
- [HÖL94b] HÖLBIG, C.A.; DIVERIO, T.A.; CLAUDIO, D.M. A Proximidade da Singularidade dos Sistemas de Equações Lineares. In: CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 17., 1994, Vitória, ES. **Anais ...** Vitória: SBMAC, 1994. v.1, p.304-308.

- [HÖL95] HÖLBIG, C.A. **Sistemas Lineares: Aplicabilidade, Resolução e Verificação Automática**. Porto Alegre: CPGCC da UFRGS, 1995. 98p (TI-472).
- [HÖL96] HÖLBIG, C.A.; DIVERIO, T.A.; CLAUDIO, D.M. Algumas Aplicações de Sistemas de Equações Lineares. Porto Alegre: CPGCC da UFRGS, 1996. (ainda não publicado).
- [HÖL96a] HÖLBIG, C. A. **SELINT.P Manual de utilização**. Porto Alegre: CPGCC da UFRGS, 1996. (ainda não publicado).
- [KEA90] KEARFOTT, R.B. Preconditioners for the Interval Gauss-Seidel Method. **Siam J. of Numerical Analysis**, Philadelphia, v.27, p.804-822, 1990.
- [KEH76] KEHAT,E.; SCHACHAM, M. A direct method for the solution of large sparse systems of linear equations. **Computer Journal**, London, v.19, n.4, p.353-359, Nov. 1976.
- [KEL67] KELLY,L.G. **Handbook of numerical methods and applications**. Massachusetts: Addison Wesley, 1967.
- [KLA91] KLATTE, R. et al. **PASCAL-XSC : Language Reference with Examples**. Berlin, Springer-Verlag, 1991.
- [KLA92] KLATTE,R. et al. **PASCAL-XSC language reference with examples**. Berlin: Springer Verlag, 1992. 344p.
- [KOR94] KORZENOWSKI, H. **Estudo sobre Resolução de Equações de Coeficientes Intervalares**. Porto Alegre: CPGCC da UFRGS, 1994. 132p. Dissertação de Mestrado.
- [KRÄ96] KRÄMER, W.; KULISCH, U.; LOHNER, R. **Numerical Toolbox for Verified Computing II: advanced numerical problems**. Berlin: Springer-Verlag, 1996. 385p.
- [KUO72] KUO,S.S. **Computer Applications of numerical methods**. New York: Addison Wesley, 1972.

- [MAY88] MAYER, G. Enclosing the Solutions of Systems of Linear Equations for Interval Iterative Process. **Computing Suppl.**, New York, v.6; p.47-58, 1988.
- [NEU90] NEUMAIER, A. **Interval methods for systems of equations**. Cambridge: Cambridge University Press, 1990. 260p.
- [OST84] OSTRBY, O.; ZLATEV, Z. **Direct Methods for Sparse Matrices**. Berlin: Springer Verlag, 1984.
- [RAT82] RATSCHEK, K.; SAVER, W. Linear Interval Equations. **Computing**, New York, v.28, p.105-115, 1982.
- [ROH88] ROHN, J. Solving Systems of Linear Equations. **Reliability in Computing**, p.171-182, 1988.
- [RUM80] RUMP, S.M.; KAUCHER, E. Small Bounds for the Solution of Systems of Equations. **Computing**, New York, Suppl.2, p.157-164, 1980.
- [RUM94] RUMP, S.M. **Verification Methods for Dense and Sparse Systems of Equations**. Hamburg: Technische Universität Hamburg-Harburg, 1994. 73p.
- [SCH87] SCHWANDT, B. Iterative Methods for System of Equations with Interval Coefficients and Linear Form. **Computing**, New York, v.38, p.143-161, 1987.
- [STA67] STATON, R.G. **Numerical methods for science and engineering**. New Delhi: Prentice Hall, 1967.
- [TEW73] TEWARSON, R.P. **Sparse Matrices**. New York: Academic Press, 1973.
- [TAR94] TARTARUGA, I. G. P.; TOSMANN, I. R.; DIVERIO, T. A. Métodos Intervalares iterativos para resolução de sistemas de equações lineares. In: CONGRESSO NACIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL, 17., 1994, Vitória, ES. **Anais ...** Vitória: SBMAC, 1994. v.1, p.325-329.

- [WES68] WESTLAKE, J.R. **A Handbook of Numerical Matrix Inversion and Solution of Linear Equations.** New York: Willy, 1968.
- [WIL71] WILKINSON, J.M.; REINSCH, C. **Handbook for Automatic Computation Linear Algebraic.** Berlin: Spring Verlag, 1971.



Métodos Intervalares para a Resolução de Sistemas de Equações Lineares

por

Carlos Amaral Hölbig

Dissertação apresentada aos Senhores:

Prof. Dr. Rudnei Dias da Cunha (Inst. Mat./UFRGS)

Prof. Dr. Philippe Olivier Alexandre Navaux

Profa. Dra. Maria Angélica de Oliveira Camargo Brunetto (DCC/UEL)

Vista e permitida a impressão.

Porto Alegre, 12 / 04 / 96.

Prof. Dr. Dalcídio Moraes Claudio,
Orientador.

Prof. Dr. Tiarajú Asmuz Diverio,
Co-orientador.

Prof. Flávio Rech Wagner
Coordenador do Curso de Pós-Graduação
em Ciência da Computação - CPGCC
Instituto de Informática - UFRGS