

ESTUDO ESTATÍSTICO DA PRECISÃO E DO TEMPO DE PROCESSAMENTO DE VÁRIOS MÉTODOS DE RESOLUÇÃO DE SISTEMAS LINEARES POR COMPUTADOR

AGOSTINHO ROBERTO DE ABREU

Engenheiro, Agrônomo

Orientador: Prof. Dr. Izaias Rangel Nogueira

Dissertação apresentada à Escola Superior de
Agricultura "Luiz de Queiroz", da Universidade
de São Paulo, para obtenção do Título de
Mestre em Estatística e Experimentação
Agrônômica.

PIRACICABA

Estado de São Paulo - Brasil

Agosto - 1979

À memória de meu pai,
a minha mãe,
à Ana Eliza e
à Luciana

AGRADECIMENTOS

À Escola Superior de Agricultura de Lavras, à Escola Superior de Agricultura "Luiz de Queiroz" e à Coordenação de Aperfeiçoamento de Pessoal Superior (CAPES), que tornaram possível a participação no Curso de Pós-Graduação.

Ao Departamento de Matemática da Faculdade de Medicina de Ribeirão Preto - USP e ao Instituto de Matemática da Escola de Engenharia de São Carlos - USP, onde iniciei o Curso de Pós Graduação.

Ao Dr. Izaias Rangel Nogueira, Professor Titular do Departamento de Matemática e Estatística da Escola Superior de Agricultura "Luiz de Queiroz", pela orientação e amizade.

Ao Professor Catedrático Dr. Frederico Pimentel Gomes, Coordenador do curso de Pós-Graduação em Estatística e Experimentação Agronômica, pela acolhida e boa vontade em resolver as dificuldades surgidas.

Ao Professor José Vitor Silveira, Vice-Diretor do Centro de Processamento de Dados da Escola Superior de Agricultura de Lavras, pela elaboração de toda programação usada neste trabalho e ainda pelas sugestões, incentivo e amizade.

Ao Dr. Humberto de Campos, Professor do Departamento de Matemática e Estatística da "ESALQ", pelas valiosas sugestões.

Aos Diretores e Funcionários do Centro de Pro
cessamento de Dados da "ESAL" , pela boa vontade demonstrada.

Aos colegas do curso de Pós-Graduação, pelo es
pírito de solidariedade e companheirismo.

Ao Professor Nelson Willibaldo Werlang, pela co
laboração.

Aos Acadêmicos, Elaine Botelho Carvalho e José
Rafael Filho, pela ajuda no decorrer do trabalho.

À Miriam Gomide Rezende e Maria Dulce Alvarenu
ga pelos serviços datilográficos.

Aos Funcionários da Gráfica da "ESAL" pelo cuidau
do com que procederam os serviços de impressão.

ÍNDICE

	Página
1. RESUMO	1
2. INTRODUÇÃO	4
3. REVISÃO DE LITERATURA	7
4. METODOLOGIA	12
4.1. De programação	12
4.2. De análise de resultados	14
4.2.1. Teste de Friedman	14
4.2.1.1. Aplicação	14
4.2.2. Comparações múltiplas	15
4.2.2.1. Aplicação	16
5. RESULTADOS E DISCUSSÃO	17
5.1. Matriz simétrica estritamente diagonalmente dominante	17
5.1.1. Ordens 3 a 30	18
5.1.1.1. Precisão	18
5.1.1.2. Tempo de processamento	20
5.1.2. Ordens 31 a 50	22
5.1.2.1. Precisão	22
5.1.2.2. Tempo de processamento	23
5.2. Matriz não simétrica estritamente diagonal - mente dominante	24

	Página
5.2.1. Ordens 3 a 30	25
5.2.1.1. Precisão	25
5.2.1.2. Tempo de processamento	26
5.2.2. Ordens 31 a 50	28
5.2.2.1. Precisão	28
5.2.2.2. Tempo de processamento	29
5.3. Matriz simétrica sem dominância diagonal ...	30
5.3.1. Ordens 3 a 30	31
5.3.1.1. Precisão	31
5.3.1.2. Tempo de processamento	32
5.3.2. Ordens 31 a 50	33
5.3.2.1. Precisão	33
5.3.2.2. Tempo de processamento	34
5.4. Matriz não simétrica sem dominância diagonal	35
5.4.1. Ordens 3 a 30	35
5.4.1.1. Precisão	35
5.4.1.2. Tempo de processamento	36
5.4.2. Ordens 31 a 50	37
5.4.2.1. Precisão	37
5.4.2.2. Tempo de processamento	38
6. CONCLUSÕES	39
7. SUMMARY	42
8. BIBLIOGRAFIA	45

	Página
9. APÊNDICE	47
9.1. Gráficos	48
9.2. Descrição dos métodos	65
9.3. Programas computacionais	89

1. RESUMO

Neste trabalho procurou-se determinar os melhores métodos de resolução de sistemas lineares, em relação a tempo gasto no processamento, e a precisão, ou seja, a diferença entre a solução exata e a solução encontrada pelo método usado na resolução, a qual foi medida através do erro percentual médio.

Trabalhou-se com alguns métodos mais comuns que são:

1. Método de Gauss-Jordan com pivotação máxima ou total
2. Método de Gauss-Jordan
3. Método de Cholesky
4. Método de Gauss-Seidel

5. Método de Jacobi
6. Método dos gradientes conjugados.

Como para resolver sistemas, determinados métodos exigem certos tipos de matrizes, procurou-se trabalhar com quatro tipos que são:

1. Matriz simétrica estritamente diagonalmente dominante
2. Matriz simétrica sem dominância diagonal
3. Matriz não simétrica estritamente diagonalmente dominante
4. Matriz não simétrica sem dominância diagonal

Dentro de cada tipo de matriz acima e para cada método que resolve este tipo, trabalhou-se com quatro repetições de cada ordem.

As ordens foram divididas em dois grupos, a saber: ordens de 3 a 30 e ordens de 31 a 50, isto devido ao fato do método da pivotação máxima ser muito lento para processar ordens altas, então no processamento por este método, foram submetidos sistemas até a ordem 30. Nos demais métodos notou-se as mesmas tendências, tanto no primeiro grupo de ordens como no segundo, isto é, os métodos que se destacaram no primeiro grupo, também destacaram no segundo grupo, como por exemplo, o método de CHOLESKY que se mostrou um dos mais eficientes em todos os casos.

Com respeito a precisão observou-se que, para os métodos que resolveram os sistemas apresentados, esta mostrou-se bastante satisfatória para todos eles, embora os testes estatísticos mostrassem diferenças entre métodos.

Nos tempos de processamentos de cada método, verificou-se diferenças acentuadas entre eles, nos diversos tipos de matrizes.

Neste trabalho são apresentados gráficos, tanto para as precisões, como para os tempos de processamentos, onde pode-se perceber as tendências de cada método, dentro de cada tipo de matriz.

2. INTRODUÇÃO

Vários são os métodos existentes para resolução de sistemas de equações lineares do tipo

$$\begin{aligned} a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n &= b_1 \\ a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n &= b_2 \\ \vdots & \\ a_{n1} x_1 + a_{n2} x_2 + \cdots + a_{nn} x_n &= b_n \end{aligned}$$

Considera-se aqui os $n \times n$ escalares a_{ij} e os n escalares b_i como fixos. Por solução do sistema acima, quer-se dizer uma n -upla de escalares x_j , para $j = 1, 2, \dots, n$, para os quais cada uma das n equações fique satisfeita.

O sistema pode ser escrito em forma compacta, usando a notação matricial $AX + B = 0$, onde

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_1 \\ a_{21} & a_{22} & \dots & a_2 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{n1} & a_{n2} & & a_{nn} \end{bmatrix} \quad X = \begin{bmatrix} X_1 \\ X_2 \\ \cdot \\ \cdot \\ X_n \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_n \end{bmatrix}$$

Um dos grandes problemas em computação é o de encontrar um método efetivo, para resolver um sistema de n equações simultâneas em n incógnitas, particularmente se n é muito grande.

Naturalmente não há um método ideal para resolver todos os problemas, pois a eficiência de um método depende principalmente, do particular sistema a ser resolvido, principalmente em relação ao tipo de matriz envolvida no referido sistema.

Quando depara-se com tais problemas, escolhe-se métodos baseados apenas em características matemáticas, sem preocupar-se, por exemplo, se o tempo gasto no processamento é satisfatório ou se a precisão dos resultados é aceitável e comprováveis na prática.

Quanto a precisão, ou seja, a maior ou menor diferença entre os valores calculados e os valores reais da solução do sistema, não cabe maiores comentários, pois a própria palavra já indica sua importância.

Se o resultado exato de um sistema de equações lineares é o vetor X e, o resultado encontrado por um determi

nado método e X_0 , deseja-se sempre que a diferença $|X - X_0|$ seja a menor possível. Tal precisão depende de vários fatores, entre os principais pode-se citar três: o método utilizado, o tipo de matriz do sistema e a ordem do sistema.

No presente trabalho procurou-se estabelecer as melhores combinações entre tais influências, para que se possa ter resultados confiáveis a tempo satisfatório.

O tempo é da maior importância, pois os custos de hora de computador são elevados, principalmente em centros de grande utilização.

13. REVISÃO DE LITERATURA

RALSTON (1965), afirma que o método dos gradientes conjugados na computação, não leva a uma solução exata, por acumulação de erros de arredondamento. E prova que, uma feitura útil deste método é que se a iteração $Q(X + \Delta X) - Q(X) = 1/2 (\Delta X' A \Delta X)$ e desenvolvendo para valores de $i > n$, então $\|\epsilon_{i+1}\| < \|\epsilon_i\|$, para todo i , onde $\epsilon_i = X_t - X_i$, X_t é solução exata e X_i é a aproximação na i -ésima iteração. Até que o arredondamento em cada iteração impede aproximações melhores posteriores.

O autor afirma ainda que, apesar da elegância dos métodos iterativos, como o gradiente conjugado, e de suas vantagens, como por exemplo, a convergência pode ocorrer em menos de n iterações, estes métodos são raramente usados em

em computadores digitais, para solução de sistemas de equações lineares simultâneas. A principal razão para isto, é que métodos diretos requerem menos computação para matrizes grandes e esparsas.

CARNAHAN, et alii (1969) diz que no método de Jacobi, o grau de aproximação pode ser melhorado pelo gasto de maior espaço computacional, isto é, perfazendo um maior número de iterações.

O autor prova ainda que, a convergência depende de ser verdadeira a expressão $\lim_{k \rightarrow \infty} A^k = 0$, onde k é o número de iterações, esta expressão é verdadeira, se e somente se, todos os auto valores (λ_k) de A em módulo forem menores que a unidade, e para que isto se dê, as condições suficientes são:

$$|\lambda_k| < \max \left(\sum_{j=1}^n |a_{ij}| \right) \text{ e } |\lambda_k| < \max \left(\sum_{i=1}^n |a_{ij}| \right)$$

e daí o desenvolvimento subsequente, temos as condições suficientes

$$\sum_{i=1}^n |a_{ij}| \leq \mu < 1, \quad 1 \leq j \leq n,$$

ou

$$\sum_{j=1}^n |a_{ij}| \leq \mu < 1, \quad 1 \leq i \leq n,$$

ou

$$\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2 \leq \mu < 1$$

O autor diz, ainda, que a solução de sistemas de equações por processos iterativos, tal como Jacobi e Gauss-Seidel, são muitas vezes denominados relaxação (os erros na estimativa inicial do vetor solução são diminuídos ou relaxados enquanto os cálculos continuam). Os métodos de Gauss-Seidel e similares são usados extensivamente na solução de grandes sistemas de equações lineares, gerados como resultados de aproximação de diferenças finitas de equações diferenciais parciais.

HEHL (1972), relaciona propriedades de um método automático de resolução de sistemas lineares, que são:

1. O método deve ser simples, requerendo um mínimo de memória.

2. O método deve assegurar uma rápida convergência se o número de passos necessários à solução é infinito. Se não ocorrerem erros de arredondamento, é preferível um método que produza a solução num número finito de passos.

3. O procedimento deve ser estável com respeito a erros de arredondamento. É possível diminuir erros de arredondamento pela repetição da mesma rotina, iniciando com os resultados obtidos como nova estimativa da solução.

4. Cada passo deve dar informações sobre a solução e deverá produzir uma nova e melhor estimativa do que o valor anterior.

5. O método deve usar tanto quanto possível os dados iniciais, isentos de erros de arredondamento, a fim

de preservar possíveis propriedades especiais do problema.

O autor citando Hestenes e Stiefel (1952), afirma que, de acordo com estes autores, existem dois métodos que mais se aproximam destes critérios: O método da eliminação de Gauss e o método de Gradientes Conjugados.

ALBRECHT (1973), diz que quando a ordem n do sistema é grande preferem-se os métodos iterativos pois, em geral, os métodos diretos acarretam erros de arredondamentos. Os métodos diretos tem a vantagem de se poder determinar, com antecedência, o número de operações necessárias para se chegar à solução.

O autor afirma ainda, que o método de Cholesky é uma modificação do método de eliminação de Gauss, e que o método tem estabilidade maior que o original.

O mesmo autor comparando três métodos iterativos, entre eles Jacobi e Gauss-Seidel, conclui que, "não se pode concluir, que o método de Jacobi é sempre inferior ao de Gauss-Seidel". Prova ainda que, para matrizes de diagonal estritamente dominante, os métodos de Jacobi e Gauss-Seidel, convergem.

BAJPAI, et alii (1977) afirma que o método de Cholesky é popular em programas de computação, devido seu espaço de armazenamento ser muito econômico. Pois não há necessidade de armazenar os zeros das matrizes de decomposição L e U e nem os l 's da matriz L . Aqueles elementos de L e U que têm de ser calculados, podem ser adaptados numa disposição (array)

retangular, além disso um elemento da matriz A do sistema , que já foi usado não é requisitado de novo, de modo que a sua posição no "array" original $n \times n$, pode ser usada para armazenagem do correspondente elemento quer seja de L ou de U.

4. METODOLOGIA

4.1. De programação

Para cada um dos quatro tipos de matrizes em estudo, foi escrita uma rotina para resolução de sistemas lineares, para cada método de resolução.

Procurou-se fazer a implementação destas rotinas, de modo que, o código (programa) obtido fosse o mais eficiente possível, eliminando desta forma o problema da influência (benefício ou prejuízo) de implementação em algum método, evitando-se com isto que este ou aquele método fosse beneficiado ou prejudicado, não por eficiência ou deficiência própria, mas por falta de uniformização de programação.

Estas rotinas foram reunidas em um programa ge

ral que:

1. Gera uma matriz de ordem n ($3 \leq n \leq 30$) do tipo em estudo e constrói a partir do vetor solução fornecido, um sistema linear.

2. O sistema linear gerado é então processado, simultaneamente, por cada um dos métodos.

3. Quando se consegue a solução do sistema dentro de um número máximo estipulado de iterações ou passos, respectivamente, para os métodos iterativos e não iterativos, calcula-se o tempo gasto no processamento e o erro percentual médio. Quando não consegue resolver, emite uma mensagem de causa da não resolução.

Esta mesma metodologia foi aplicada para sistemas de ordem n ($31 \leq n \leq 50$), esta divisão em dois grupos de ordens foi devida ao fato de que o método de Gauss-Jordam com pivotação máxima, que daqui por diante será chamado apenas de pivotação máxima, apesar de resolver todos os tipos de matrizes, gasta um tempo muito elevado, a partir de determinadas ordens, observou-se que sistemas de ordem 40 gastava cerca de 2.250 segundos no processamento.

Dentro de cada tipo de matriz, todos os métodos resolveram os mesmos sistemas, sendo que foram usados quatro sistemas de cada ordem, formando portanto um conjunto de 4 repetições para cada ordem.

De cada resolução anotou-se, para análises posteriores, o tempo de processamento e o erro percentual médio.

Quando no vetor solução fornecido tiver componente nulo, este não entra no cálculo do erro percentual médio, pois teria uma divisão por zero, então nestes casos o computador não considera a diferença deste componente.

4.2. De análise de resultados

A análise dos resultados, foram feitas através dos testes de FRIEDMAN (X^2 de Friedman) e TESTES DE COMPARAÇÕES MÚLTIPLAS, ambos descritos por CAMPOS (1979), como segue;

4.2.1. Teste de FRIEDMAN

As hipóteses a serem testadas são:

$$H_0: t_1 = t_2 = \dots = t_k$$

H_1 : pelo menos dois tratamentos diferem entre si.

4.2.1.1. Aplicação

Dentro de cada bloco (ordem), procede-se a classificação conjunta das K observações (métodos), dando nota 1 ao menor valor e K ao maior deles. É definido então:

$$X_r^2 = \frac{12}{nK(K+1)} \sum_{i=1}^K R_i^2 - 3n(K+1), \text{ onde } R_i \text{ é a soma das notas atribuídas aos dados do tratamento } i, \text{ nos } n \text{ blocos.}$$

No caso de empates na classificação dos tratamentos em um mesmo bloco utiliza-se a média das notas que seriam atribuídas, caso não houvesse empates. Além disso tem-se que aplicar ao valor X_r^2 a seguinte correção:

$$C = 1 - \frac{\sum_j T_j}{nK(K^2 - 1)}$$
, onde $T_j = t_{ij}^3 - K$, t_{ij} é o número de notas empatadas entre os K tratamentos nos n blocos, $i = 1, 2, \dots, K$ e $j = 1, 2, \dots, n$.

A nova expressão de X_r^2 fica:

$$X_r^2 = \frac{X_r^2}{C} = \frac{\frac{12}{nK(K+1)} \sum_{i=1}^K R_i^2 - 3n(K+1)}{1 - \frac{\sum_j T_j}{nK(K^2 - 1)}}$$

Os valores X_r^2 ou $X_r^{2'}$ são comparados com um valor X_0^2 tabelado para $K \leq 5$ e n (nº de blocos), a um nível α de significância.

Quando, porém, $K > 5$ ou o número de blocos não for previsto em tabela, deve-se utilizar a aproximação para grandes amostras.

Comprova-se que, sob a hipótese H_0 , para grandes amostras, X_r^2 tem uma distribuição de X^2 (qui-quadrado), com $K - 1$ graus de liberdade.

4.2.2. Comparações múltiplas

Este teste tem a finalidade de completar o teste anterior, quando nele se rejeita H_0 , procurando com isto,

localizar as possíveis diferenças entre pares de tratamentos.

Obviamente, os recursos destas comparações são limitados, pois ela permite apenas, confrontar os tratamentos dois a dois, não sendo possível, portanto, a comparação entre grupos de tratamentos.

4.2.2.1. Aplicação

Considera-se os C_k^2 pares de tratamentos e determina-se, para cada par, a diferença $|R_i - R_{i'}|$, com $i = 1, 2, \dots, K - 1$ e $i' = i + 1, i + 2, \dots, K$. Onde R_i e $R_{i'}$ representam as somas das notas atribuídas aos tratamentos i e i' , respectivamente, nos n blocos correspondentes.

A uma taxa de erro experimental α , admite-se $t_i \neq t_{i'}$, se $|R_i - R_{i'}| \geq \Delta_1$, ou seja, a diferença mínima significativa (d.m.s.), a uma taxa α de probabilidade é: d.m.s. = Δ_1 , ou $P_0(|R_i - R_{i'}| \geq \Delta_1) = \alpha$, os valores de Δ_1 são obtidos em tabela.

Quando o número de ordens ou de tratamentos, ou ambos ultrapassam os valores previstos na tabela de Δ_1 , obtém-se a d.m.s. como segue:

$$\text{d.m.s.} = Q \sqrt{\frac{nK(K + 1)}{12}}$$

obtem-se o valor de Q em tabela própria, onde entra com o valor de K a um nível α de significância desejado.

5. RESULTADOS E DISCUSSÃO

Os resultados serão apresentados e discutidos, segundo cada tipo de matriz pesquisada.

Dentro de cada tipo de matriz, serão considerados dos dois intervalos de ordem, ou seja, 3 a 30 e 31 a 50.

5.1. Matriz simétrica estritamente diagonalmente dominante

Todos os métodos estudados mostraram-se eficientes para este tipo de matriz, porém a partir da ordem 30 ou pouco mais, o método dos gradientes conjugados não foi satisfatório, às vezes convergindo, outras não.

5.1.1. Ordens 3 a 30

Tem-se aqui 6 métodos em estudo, que serão representados pelos números:

1. Método da Pivotação Máxima
2. Método de Gauss-Jordan
3. Método de Cholesky
4. Método de Gauss-Seidel
5. Método de Jacobi
6. Método de Gradientes Conjugados

5.1.1.1. Precisão

O teste de FRIEDMAN forneceu o valor calculado (X^2_r) 98,6 e para 5 graus de liberdade tem-se o valor de X^2 (qui-quadrado) tabelado 16,75 a um nível de 0,5% de probabilidade, de onde pode-se afirmar que pelo menos dois métodos diferem entre si.

Nas comparações múltiplas obteve-se as d.m.s., 47,1 e 39,9, aos níveis de 1% e 5% de probabilidade, respectivamente, daí pode-se observar na tabela abaixo, quais métodos diferem entre si.

Tabela 1 - Diferenças de precisão - matriz simétrica estritamente diagonalmente dominante (ordens 3 a 30).

	$R_2 = 118,0$	$R_3 = 56,5$	$R_4 = 100,0$	$R_5 = 36,5$	$R_6 = 168,0$
$R_1 = 103,0$	N.S.	*	N.S.	**	**
$R_2 = 118,0$		**	N.S.	**	**
$R_3 = 56,5$			*	N.S.	**
$R_4 = 100,0$				**	**
$R_5 = 36,5$					**

Para todas as comparações múltiplas, R_i e $R_{i'}$, representam as somas das notas atribuídas aos tratamentos i e i' respectivamente, nas n ordens correspondentes.

Em todas as tabelas apresentadas, foram usadas as seguintes convenções:

*** - significativo a 0,5%

** - significativo a 1,0%

* - significativo a 5,0%

N.S.- não significativo

Dos resultados acima pode-se dizer que:

1. O método da pivotação máxima não difere em precisão dos métodos de Gauss-Seidel e Gauss-Jordan; é menos preciso que os de Cholesky e Jacobi; e ainda é mais preciso que o de gradientes conjugados.

2. O método de Gauss-Jordan não difere do método de Gauss-Seidel; é menos preciso que os de Cholesky e Jacobi; e mais preciso que o de gradientes conjugados.

3. O método de Cholesky não difere do de Jacobi, e é mais preciso que os de Gauss-Seidel e gradientes conjugados.

4. O Gauss-Seidel é menos preciso que o de Jacobi e mais preciso que o de gradientes conjugados.

5. O método de Jacobi é mais preciso que o de gradientes conjugados.

5.1.1.2. Tempo de processamento

O valor de FRIEDMAN calculado foi 126,0, enquanto que o de X^2 para 5 graus de liberdade a 0,5% de probabilidade é 16,75, de onde conclui-se que pelo menos dois métodos diferem entre si, em tempo de processamento.

Nas comparações múltiplas, as d.m.s. a 1% e 5% foram, respectivamente, 47,1 e 39,9, daí na tabela abaixo pode-se observar as diferenças a tais níveis.

Tabela 2 - Diferenças de tempo de processamento. Matriz simétrica estritamente diagonalmente dominante (ordem 3 a 30).

	$R_2 = 109,0$	$R_3 = 71,5$	$R_4 = 62,0$	$R_5 = 38,5$	$R_6 = 143,5$
$R_1 = 164,5$	**	**	**	**	N.S.
$R_2 = 109,0$		N.S.	*	**	N.S.
$R_3 = 71,5$			N.S.	N.S.	**
$R_4 = 62,0$				N.S.	**
$R_5 = 38,5$					**

Observando os resultados da Tabela 2, tem-se que:

1. O método da pivotação máxima não difere do método de gradientes conjugados, e é mais lento que os outros quatro métodos em estudo neste item.

2. O de Gauss-Jordan não difere do de Cholesky e do de gradientes conjugados, e é mais lento que os de Gauss-Seidel e Jacobi.

3. O método de Cholesky não difere dos de Gauss-Seidel e Jacobi, e é mais rápido que o de gradientes conjugados.

4. O método Gauss-Seidel não difere do Jacobi, sendo no entanto mais rápido que o de gradientes conjugados.

5. Jacobi é mais rápido que gradientes conjugados.

5.1.2. Ordens 31 a 50

Tem-se agora quatro métodos:

1. Método de Gauss-Jordan
2. Método de Cholesky
3. Método de Gauss-Seidel
4. Método de Jacobi

5.1.2.1. Precisão

O valor de FRIEDMAN foi 43,35, o de X^2 com 3 graus de liberdade a 0,5%, 12,84. Então pode-se afirmar neste nível de probabilidade que pelo menos dois métodos diferem entre si, em precisão.

Nos testes das comparações múltiplas, as d.m.s. foram 25,42 a 1% e 20,98 a 5%. Com isto pode-se ver na Tabela 3, quais métodos diferem entre si.

Tabela 3 - Diferenças de precisão - matriz simétrica estritamente diagonalmente dominante (ordens 31 a 50).

	$R_2 = 29,0$	$R_3 = 58,5$	$R_4 = 36,0$
$R_1 = 76,5$	**	N.S.	**
$R_2 = 29,0$		**	N.S.
$R_3 = 58,5$			*

Vê-se aí que:

1. O método de Gauss-Jordan não difere do de Gauss-Seidel, porém é menos preciso que os de Cholesky e Jacobi.

2. Cholesky não difere de Jacobi, mas é mais preciso que Gauss-Seidel.

3. Gauss-Seidel é menos preciso que Jacobi.

5.1.2.2. Tempo de processamento

O FRIEDMAN calculado foi 60,0 e o X^2 com 3 graus de liberdade a 0,5% de probabilidade, 12,84. Então pelo menos dois métodos são diferentes em tempo de processamento.

As d.m.s. das comparações múltiplas, foram 25,4 a 1% e 21,0 a 5% de probabilidade, daí pode-se mostrar os métodos diferentes na tabela seguinte.

Tabela 4 - Diferenças de tempo de processamento-matriz simétrica estritamente diagonalmente dominante (ordens 31 a 50).

	$R_2 = 60,0$	$R_3 = 40,0$	$R_4 = 20,0$
$R_1 = 80,0$	N.S.	**	**
$R_2 = 60,0$		N.S.	**
$R_3 = 40,0$			N.S.

Com isto vê-se que:

1. O método de Gauss-Jordan não difere do de Cholesky sendo, no entanto, mais lento que os de Gauss-Seidel e Jacobi.

2. Cholesky não difere de Gauss-Seidel e é mais lento que Jacobi.

3. Gauss-Seidel e Jacobi, não diferem entre si, em tempo de processamento.

5.2. Matriz não simétrica estritamente diagonalmente dominante

O método de gradientes conjugados não se mostrou eficiente para este tipo de matriz.

5.2.1. Ordens 3 a 30

Para estas ordens tem-se 5 métodos, que são:

1. Método da Pivotação Máxima
2. Método de Gauss-Jordan
3. Método de Cholesky
4. Método de Gauss-Seidel
5. Método de Gradientes Conjugados

5.2.1.1. Precisão

O valor de FRIEDMAN calculado foi 65,2, e com 4 graus de liberdade a 0,5% de probabilidade, o X^2 tabelado apresenta o valor 14,86, com isto pode-se afirmar que pelo menos dois métodos diferem entre si, em precisão.

Os valores das d.m.s. nas comparações múltiplas foram 38,5 e 32,3, aos níveis de 1% e 5% de probabilidade, respectivamente. Nestes níveis pode-se notar diferenças entre métodos na Tabela 5.

Tabela 5 - Diferenças de precisão - matriz não simétrica estritamente diagonalmente dominante (ordens 3 a 30)

	R ₂ = 117,5	R ₃ = 56,5	R ₄ = 92,0	R ₅ = 41,5
R ₁ = 112,5	N.S.	**	N.S.	**
R ₂ = 117,5		**	N.S.	**
R ₃ = 56,5			*	N.S.
R ₄ = 92,0				**

Com isto vê-se que:

1. A pivotação máxima não difere de Gauss-Seidel e nem de Gauss-Jordan, contando, é menos precisa que Cholesky e Jacobi.

2. Gauss-Jordan e Gauss-Seidel não diferem entre si, no entanto, Gauss-Jordan é menos preciso que Cholesky e Jacobi.

3. Cholesky não difere de Jacobi, mas é mais preciso que Gauss-Seidel.

4. Gauss-Seidel é menos preciso que Jacobi.

5.2.1.2. Tempo de processamento

O FRIEDMAN calculado foi 88,2 e o X² tabelado ao nível de 0,5% de probabilidade, com 3 graus de liberdade é

14,86, portanto, afirma-se que pelo menos dois métodos diferem entre si, em tempo de processamento.

Nas comparações múltiplas a 1% de probabilidade, obteve-se a d.m.s. 38,5 e a 5%, 32,3, portanto, com estes níveis pode-se ver qualquer diferença entre métodos na Tabela 6.

Tabela 6 - Diferenças de tempo de processamento - matriz não simétrica estritamente diagonalmente dominante (ordens 3 a 30).

	R ₂ = 105,0	R ₃ = 69,0	R ₄ = 68,0	R ₅ = 38,0
R ₁ = 140,0	*	**	**	**
R ₂ = 105,0		*	*	**
R ₃ = 69,0			N.S.	N.S.
R ₄ = 68,0				N.S.

Observando as diferenças pode-se escrever que:

1. O método da pivotação máxima é mais lento que os outros quatro estudados neste ítem.

2. O método de Gauss-Jordan é mais lento que os de Cholesky, Gauss-Seidel e Jacobi.

3. Cholesky, Gauss-Seidel e Jacobi, não diferem em tempo de processamento.

5.2.2. Ordens 31 a 50

Tem-se quatro métodos para estas ordens, que são:

1. Método de Gauss- Jordan
2. Método de Cholesky
3. Método de Gauss-Seidel
4. Método de Jacobi

5.2.2.1. Precisão

No cálculo do teste de FRIEDMAN obteve-se o valor 25,5; o valor de X^2 tabelado com 3 graus de liberdade a 0,5% de probabilidade é 12,84, daí pode-se dizer que tem pelo menos dois métodos com precisões diferentes.

No teste das comparações múltiplas, as d.m.s. foram 25,45 e 20,98, a 1% e 5% de probabilidade, respectivamente. Pode-se, portanto, montar a tabela de contrastes seguinte.

Tabela 7 - Diferenças de precisão - matriz não simétrica estritamente diagonalmente dominante (ordens 31 a 50).

	R ₂ = 35,0	R ₃ = 57,0	R ₄ = 37,5
R ₁ = 70,5	**	N.S.	**
R ₂ = 35,0		*	N.S.
R ₃ = 57,0			N.S.

Vê-se aí, que:

1. O método de Gauss-Jordan não difere do de Gauss-Seidel e é menos preciso que os de Cholesky e Jacobi.
2. Cholesky não difere de Jacobi e é mais preciso que Gauss-Seidel.
3. Gauss-Seidel e Jacobi, não diferem entre si, em precisão.

5.2.2.2. Tempo de processamento

FRIEDMAN forneceu o valor calculado 57,84 e para 3 graus de liberdade, ao nível de 0,5% de probabilidade, tem-se o valor tabelado de X^2 12,84, indicando que o tempo de processamento de pelo menos dois métodos, é diferente.

Nas comparações múltiplas as d.m.s. a 1% e 5%

de probabilidade, foram respectivamente, 25,42 e 20,98, daí o
bserva-se na Tabela 8 os métodos diferentes entre si.

Tabela 8 - Diferenças de tempo - matriz simétrica estritamente diagonalmente dominante (ordens 31 a 50):

	R ₂ = 60,0	R ₃ = 38,0	R ₄ = 22,0
R ₁ = 80,0	N.S.	**	**
R ₂ = 60,0		*	**
R ₃ = 38,0			N.S.

Dos resultados acima, pode-se escrever que:

1. Gauss-Jordan, não difere de Cholesky, mas é mais lento que Gauss-Seidel e Jacobi.

2. Cholesky, não difere de Gauss-Seidel, mas é mais lento que Jacobi.

3. Os métodos de Gauss-Seidel e Jacobi, não diferem.

5.3. Matriz simétrica sem dominância diagonal

Neste tipo de matriz, o método de gradientes conjugados só resolveu sistemas de ordens pequenas (até 16) e alguns de ordens maiores, porém sem regularidade (ora resol -

vendo, ora não).

Os métodos de Gauss-Seidel e Jacobi, não resolvem sistemas com este tipo de matriz.

5.3.1. Ordens 3 a 30

Temos aqui três métodos que são:

1. Método da Pivotação Máxima
2. Método de Gauss-Jordan
3. Método de Cholesky

5.3.1.1. Precisão

O valor de FRIEDMAN calculado foi 28,5 e com 2 graus de liberdade a 0,5% de probabilidade, χ^2 apresenta um valor tabelado de 10,60. Com isto pode-se concluir que, pelo menos dois métodos são diferentes em precisão.

As d.m.s. calculadas, a 1% e 5% de probabilidade foram, respectivamente, 21,8 e 17,5. Com isto pode-se montar a Tabela 9.

Tabela 9 - Diferenças de precisão - matriz não simétrica sem dominância diagonal (ordens 3 a 30).

	R ₂ = 66,0	R ₃ = 69,0
R ₁ = 33,0	**	**
R ₂ = 66,0		N.S.

Nota-se com isto, que:

1. O método da Pivotação máxima é mais preciso que os outros dois.

2. Entre Gauss-Jordan e Cholesky, não se nota diferença significativa, em precisão.

5.3.1.2. Tempo de processamento

O FRIEDMAN calculado foi 54,1, e o X^2 tabelado ao nível de 0,5% de probabilidade com 2 graus de liberdade 10,60, portanto, pelo menos dois métodos diferem entre si, em tempo de processamento.

Nas comparações múltiplas a 1% de probabilidade obteve-se a d.m.s. 21,8. Portanto, pode-se notar as tres diferenças na tabela abaixo.

Tabela 10 - Diferenças de tempo de processamento - matriz não simétrica sem dominância diagonal (ordens 3 a 30)

	$R_2 = 55,0$	$R_3 = 29,0$
$R_1 = 84,0$	**	**
$R_2 = 55,0$		**

Observando estas diferenças, nota-se que:

1. O método da pivotação máxima é mais lento que os de Gauss-Jordan e Cholesky.

2. Gauss-Jordan é mais lento que Cholesky.

5.3.2. Ordens 31 a 50

Nestas ordens só temos dois métodos:

1. Método de Gauss-Jordan

2. Método de Cholesky.

5.3.2.1. Precisão

No cálculo do teste de FRIEDMAN, obteve-se o valor 0,20, indicando que os dois métodos não diferem em precisão.

Tabela 11 - Diferença de precisão - matriz não simétrica sem dominância diagonal (ordens 31 a 50).

	R ₂ = 31,0
R ₁ = 29,0	N.S.

5.3.2.2. Tempo de processamento

FRIEDMAN forneceu o valor calculado 20,0. Para 1 grau de liberdade ao nível de 0,5% de probabilidade, tem-se o X^2 tabelado 11,52, mostrando que os dois métodos são diferentes em tempo de processamento.

Nota-se, portanto, que o método de Cholesky é mais rápido que o Gauss-Jordan, conforme Tabela 12.

Tabela 12 - Diferença de tempo de processamento - matriz não simétrica sem dominância diagonal (ordens 31 a 50)

	R ₂ = 20,0
R ₁ = 40,0	***

5.4. Matriz não simétrica sem dominância diagonal

Nos sistemas com tal tipo de matriz, só obtive soluções com os métodos da pivotação máxima, de Gauss-Jordan e de Cholesky.

O método da pivotação, não apresentou problemas nas resoluções, a não ser o seu demorado tempo de processamento. Gauss-Jordan e Cholesky, resolveram cerca de 90% dos sistemas propostos, como sempre com rapidez e alta precisão.

5.4.1. Ordens 3 a 30

Tem-se os três métodos que são:

1. Método da Pivotação Máxima
2. Método de Gauss-Jordan
3. Método de Cholesky

5.4.1.1. Precisão

O FRIENDMAN calculado foi 39,02 e o X^2 com 2 graus de liberdade a 0,5% de probabilidade 10,60. Daí vê-se que pelo menos dois métodos diferem em precisão.

As d.m.s. calculadas, a 1% e 5% de probabilidade foram, respectivamente, 21,01 e 16,90. Daí pode-se observar duas diferenças na Tabela 13.

Tabela 13 - Diferenças de precisão - matriz não simétrica sem dominância diagonal (ordens 3 a 30)

	R ₂ = 64,5	R ₃ = 65,5
R ₁ = 26,0	**	**
R ₂ = 64,5	.	N.S.

Nota-se com isto que:

1. O método da pivotação é mais preciso que os de Gauss-Jordan e Cholesky.

2. Gauss-Jordan e Cholesky não diferem em precisão

5.4.1.2. Tempo de processamento

O FRIEDMAN calculado foi $\underline{45,26,0X^2}$ com 2 graus de liberdade a 0,5% de probabilidade 10,60, então pelo menos dois métodos são diferentes em tempo de processamento.

Nas comparações múltiplas, a d.m.s. a 1% de probabilidade, foi 20,60, daí tem-se na Tabela 14 os três métodos diferentes entre si.

Tabela 14 - Diferenças de tempo de processamento - matriz não simétrica sem dominância diagonal (ordens 3 a 30)

	$R_2 = 51,5$	$R_3 = 25,5$
$R_1 = 73,0$	**	**
$R_2 = 51,5$		**

Pode-se escrever que:

1. O método da pivotação máxima é mais lento que os de Gauss-Jordan e Cholesky.

2. Gauss-Jordan é mais lento que Cholesky.

5.4.2. Ordens 31 a 50

Nestas ordens só foram usados dois métodos:

1. Método de Gauss-Jordan

2. Método de Cholesky

5.4.2.1. Precisão

O Teste de FRIEDMAN forneceu um valor calculado 1,66, indicando que os dois métodos não diferem em precisão.

Tabela 15 - Diferença de precisão - matriz não simétrica sem dominância diagonal (ordens 31 a 50).

	R ₂ = 25,0
R ₁ = 20,0	N.S.

5.4.2.2. Tempo de processamento

O valor do teste de FRIEDMAN calculado foi 17,0. O valor de X^2 tabelado com 1 grau de liberdade a 0,5% de probabilidade, é 7,88. Daí vê-se que os dois métodos são diferentes em tempo de processamento.

Observa-se portanto, que o método de Cholesky é mais rápido para processar que o método de Gauss-Jordan, conforme Tabela 16.

Tabela 16 - Diferença de tempo de processamento - matriz não simétrica sem dominância diagonal (ordens 31 a 50)

	R ₂ = 17,0
R ₁ = 34,0	***

6. CONCLUSÕES

Com base nos estudos realizados em resultados e discussão, pode-se concluir que:

6.1. Matriz simétrica estritamente diagonalmente dominante.

6.1.1. Precisão

Para os dois grupos de ordens, os métodos mais precisos foram os de Cholesky e Jacobi.

6.1.2. Tempo de processamento

Nos dois grupos de ordens o método de Jacobi foi

o mais rápido, sendo que o de Gauss-Seidel mostrou-se também rápido para as ordens 3 a 30.

6.2. Matriz não simétrica estritamente diagonalmente dominante

6.2.1. Precisão

Tem-se o Cholesky como o mais preciso nos dois grupos de ordens e o Jacobi também aparecendo nas ordens 3 a 30.

6.2.2. Tempo de processamento

Tanto para o primeiro como para o segundo grupo, o mais rápido foi Jacobi, embora Cholesky e Gauss-Seidel mostrassem também rápidos, para as ordens 3 a 30.

6.3. Matriz simétrica sem dominância diagonal

6.3.1. Precisão

Para o primeiro grupo de ordens o mais preciso foi o da Pivotação Máxima, e no segundo foram os de Gauss-Jordan e Cholesky.

6.3.2. Tempo de processamento

Para todas as ordens o mais rápido foi Cholesky.

6.4. Matriz não simétrica sem dominância diagonal

6.4.1. Precisão

Nas ordens 3 a 30, o mais preciso foi o da Pivotação Máxima, e de 31 a 50, os de Gauss-Jordan e Cholesky.

6.4.2. Tempo de processamento

Em todas as ordens o mais preciso foi o de Cholesky.

Quando não se conhece o tipo de matriz envolvida no sistema, deve-se usar o método de Cholesky, que foi o mais regular para todos os tipos estudados, tanto no que se refere a precisão, como em tempo de processamento.

7. SUMMARY

This study was carried out with the objective of determining the methods of calculating the solution of linear systems with regard to both time taken for processing and the precision, that is the difference between the exact result and that found by using the method this difference was calculated as the average percentage error.

Several of the most common methods used were

1. Gauss-Jordan's method with maximum pivot strategy
2. Gauss-Jordan's method
3. Cholesky's method
4. Gauss-Seidel's method
5. Jacobi's method
6. Method of Conjugate Gradient

Since some methods require certain types of matrices to calculate the systems, 4 of those were used:

1. Symmetric matrix with diagonal dominance

2. Symetric Matrix without diagonal dominance
3. Non-symetric matrix with diagonal dominance
4. Non-symetric matrix without diagonal dominance

Within each type of those matrices, as well as for each method which solves each one of them four repetitions of each order were used.

The orders were divided into 2 groups: a 3 to 30 order group and a 31 to 50 one, because the maximum pivot strategy method is too slow to process high orders when this method was used for processing, systems were submitted until the 30 th order. The same tendencies were found with order methods, either in the first or in the second group, which means that those most evident method in the first group were also in the second group, for instance, "Choleshy Method", which has been show one of the most effective in all cathegories.

Preciseness was found to show itself satisfactory enough for all of those methods which resolved the presented systems, although statistic tests had showed differences among the methods.

During processing times of each method striking difference among them were found in the several types of matrices.

In this study graphs either for precision or for pro -

cessing time were used the tendencies of each method may be found for each type of matrix.

8. BIBLIOGRAFIA

AYRES JUNIOR, F., 1971. Matrizes. Rio de Janeiro, McGraw-Hill do Brasil. 286 p.

ALBRECHT, P., 1973. Análise Numérica-um Curso Moderno. Rio de Janeiro, Livros Técnicos e Científicos/PUC, 240 p.

BAJPAI, A.C. et alii, 1977. Numerical Methods for Engineers and Scientists. London, John Wiley & Sons. 380 p.

CAMPOS, H. de, 1979. Estatística Experimental Não-Paramétrica. 3. ed. Piracicaba, ESALQ/USP. 343 p.

CARNAHAN, B. et alii, 1969. Applied Numerical Methods. New York, John Wiley & Sons. 604 p.

FINKBEINER, D.T., 1970. Introdução às Matrizes e Transformações Lineares. Rio de Janeiro, Ao Livro Técnico/USP. 321 p.

FORSYTHE, G.E. e C.B. Moler, 1967. Computer Solution of Linear Algebraic Systems. Englewood CLIFFS, Prentice - Hall, 148 p.

HEHL, M.E., 1972. Métodos de Gradientes Conjugados Aplicação para Sistemas Lineares. São Paulo, Instituto de Física/USP. 20 p. (Série: Aplicações de Computadores - Publicação SEMA, 1).

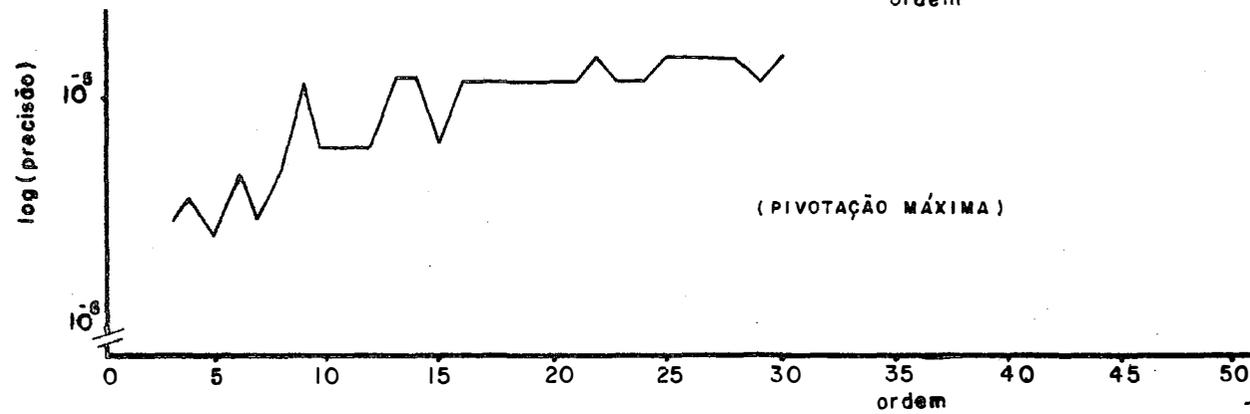
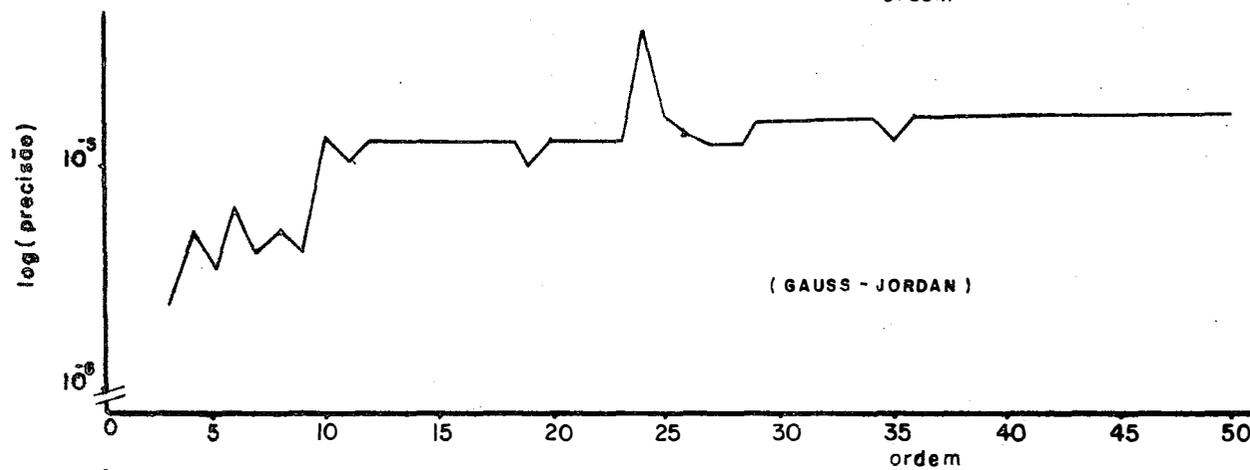
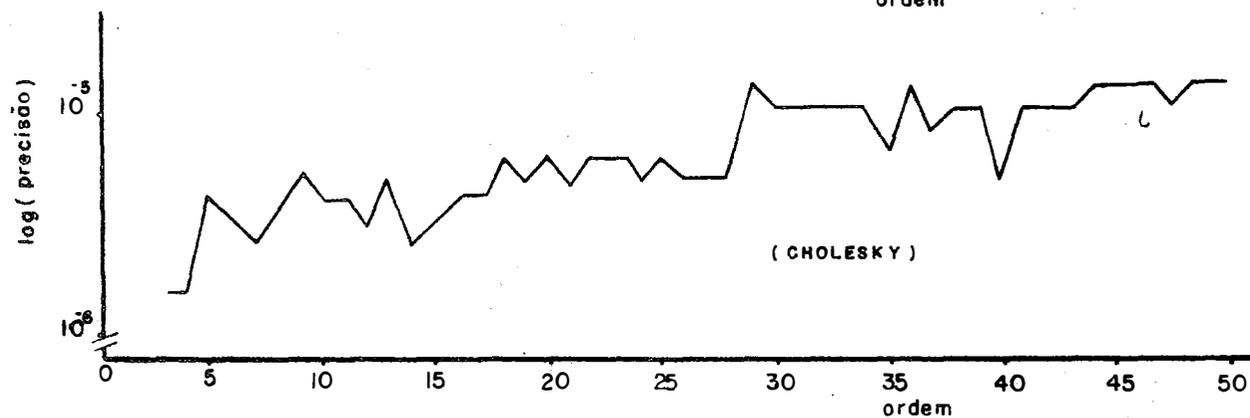
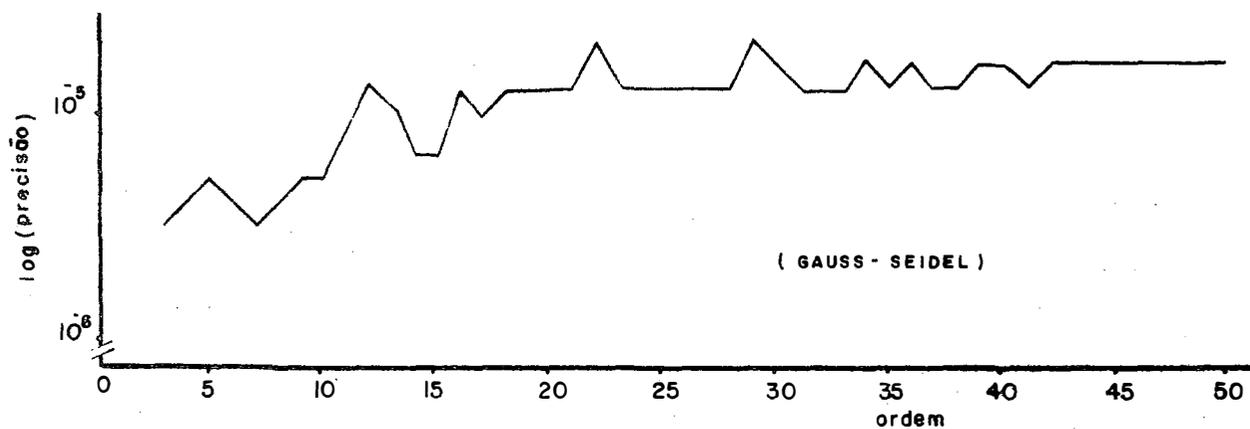
RALSTON, A., 1965. A First Course in Numerical Analysis. New York, McGraw-Hill Kogakusha. 578 p.

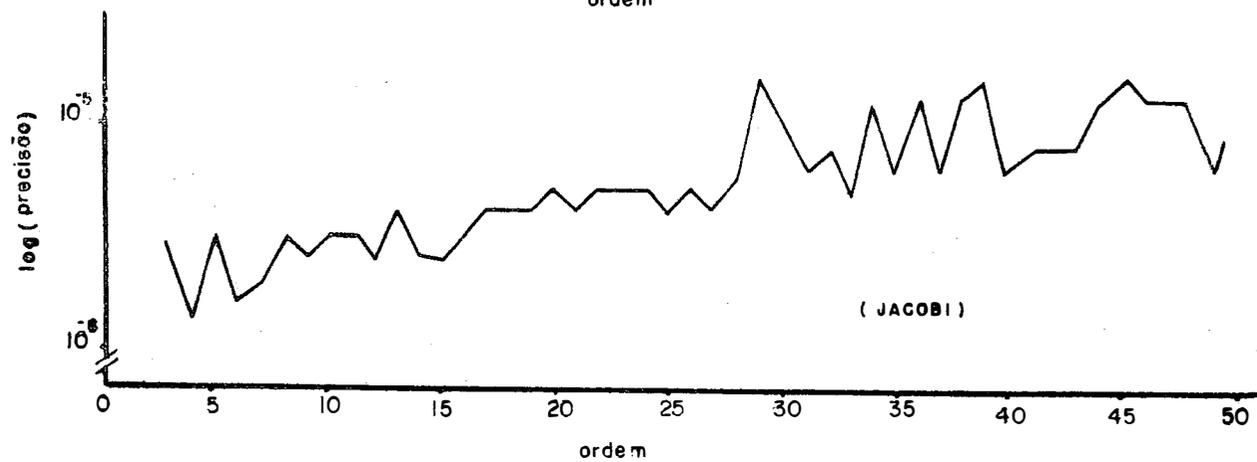
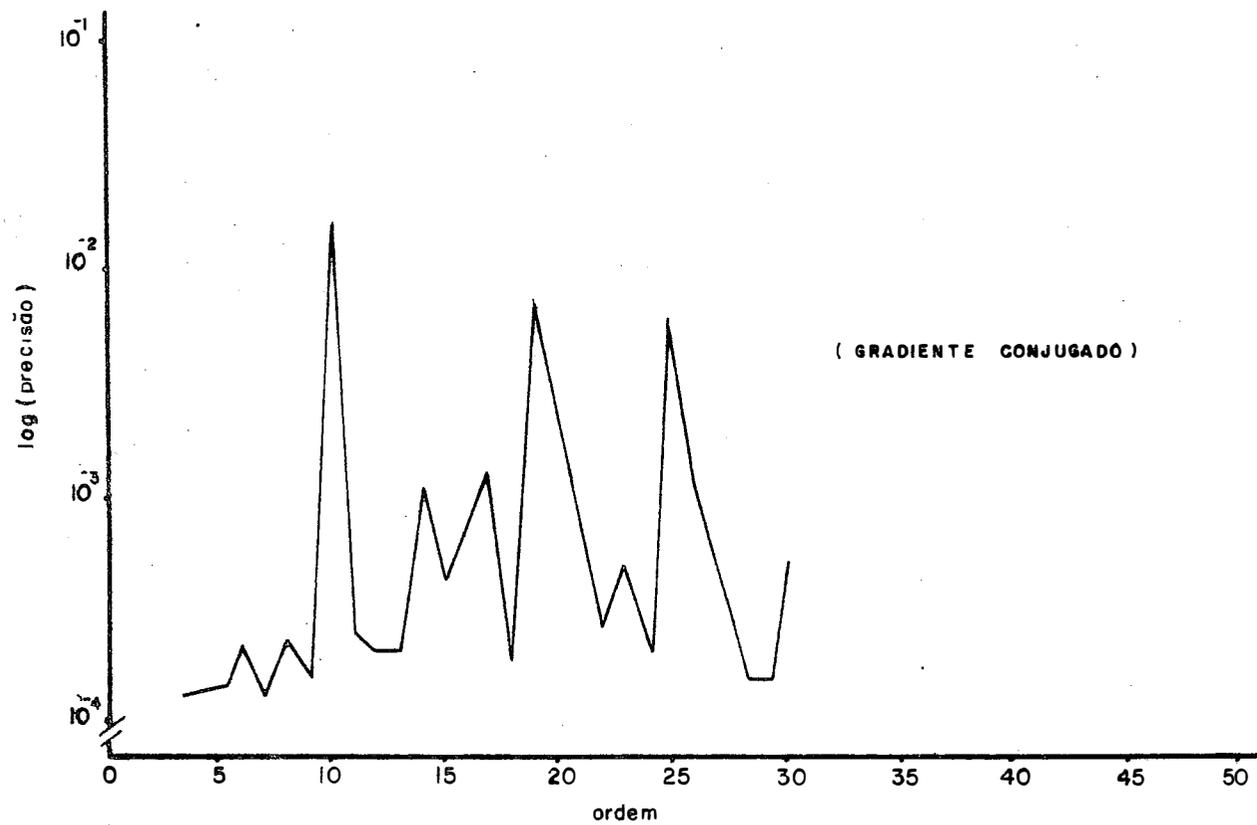
SCHWARZ, H.R., 1973. Numerical Analysis of Symmetric Matrices. Englewood CLIFFS, Prentice-Hall. 275 p.

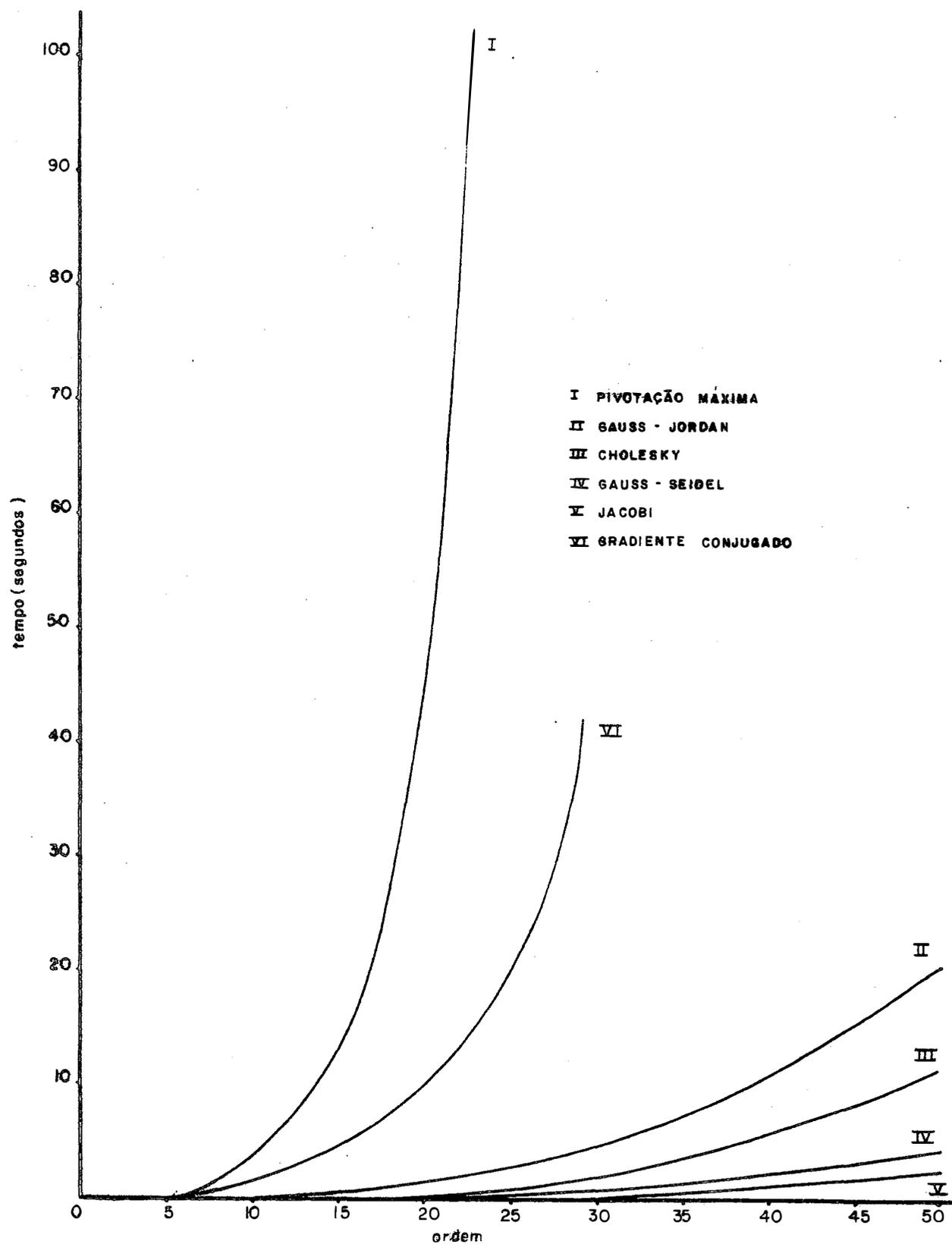
9. APÊNDICE

9.1. Gráficos

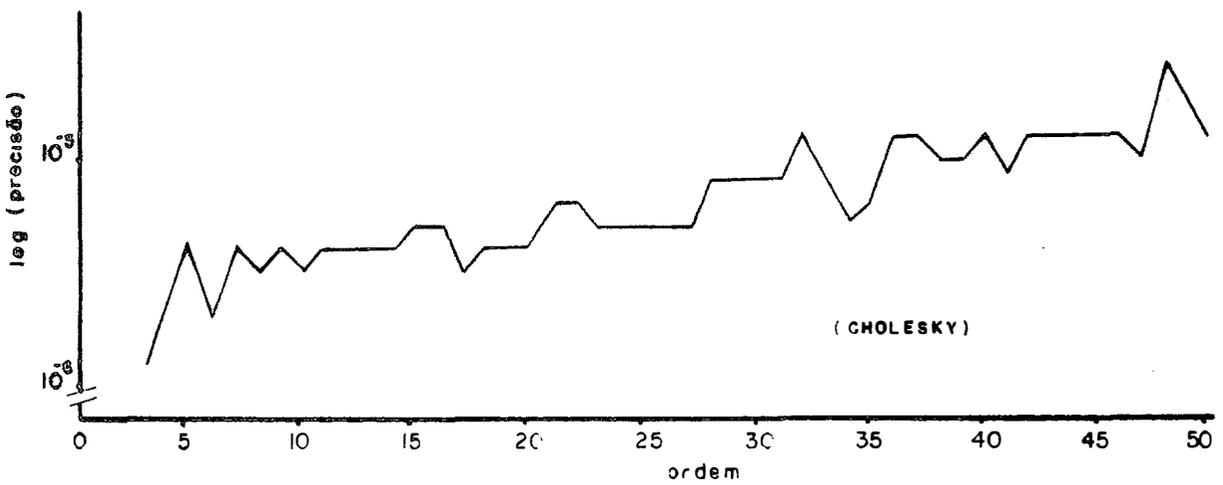
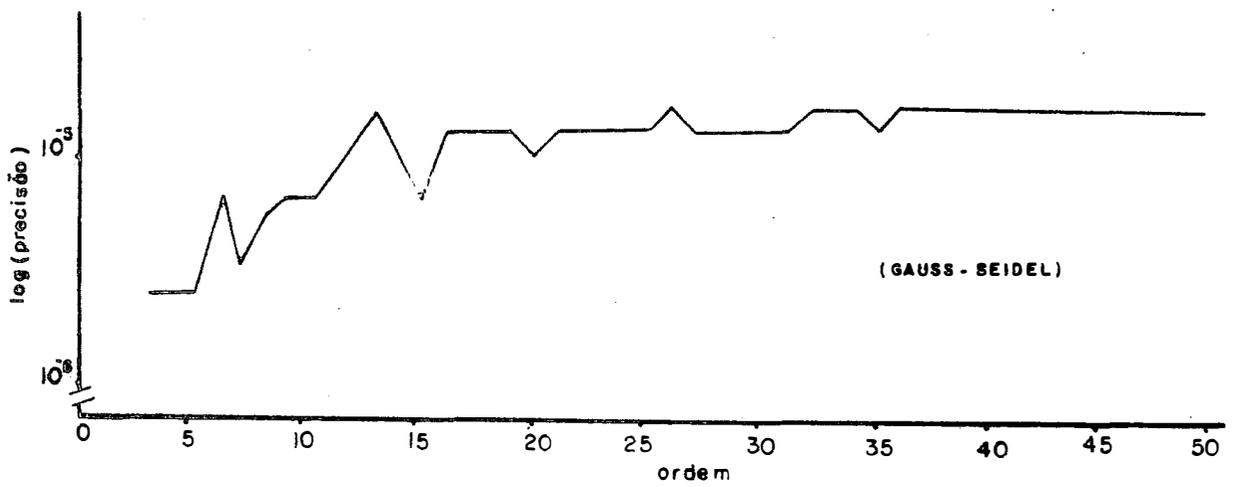
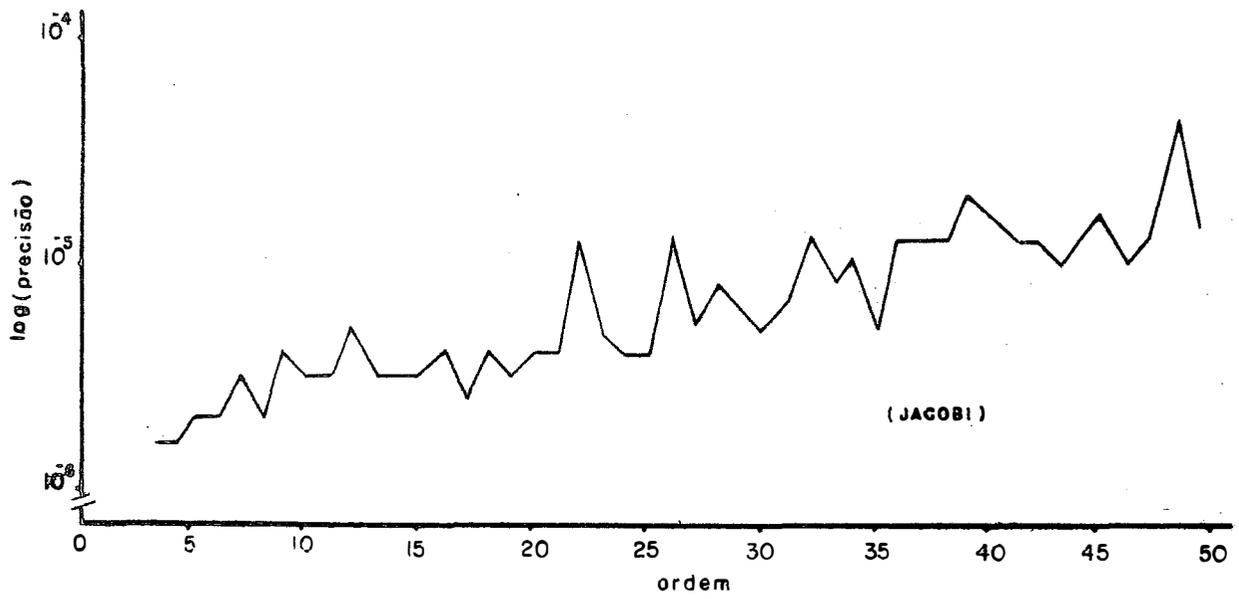
9.1.1. Os gráficos das páginas 49 e 50, correspondem à precisão das matrizes simétricas estritamente diagonalmente dominante e o da página 51 ao tempo de processamento das mesmas matrizes.

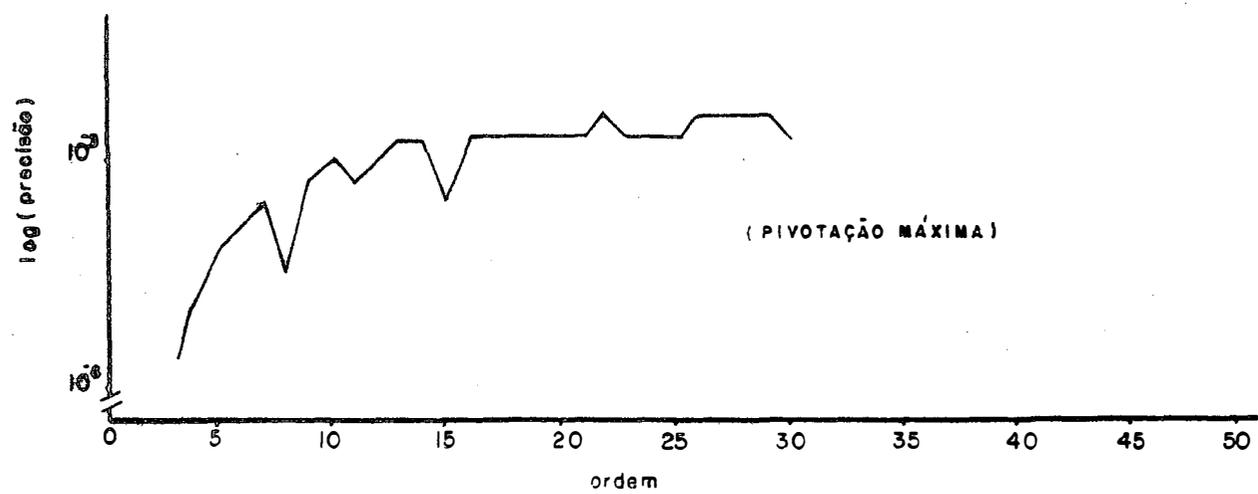
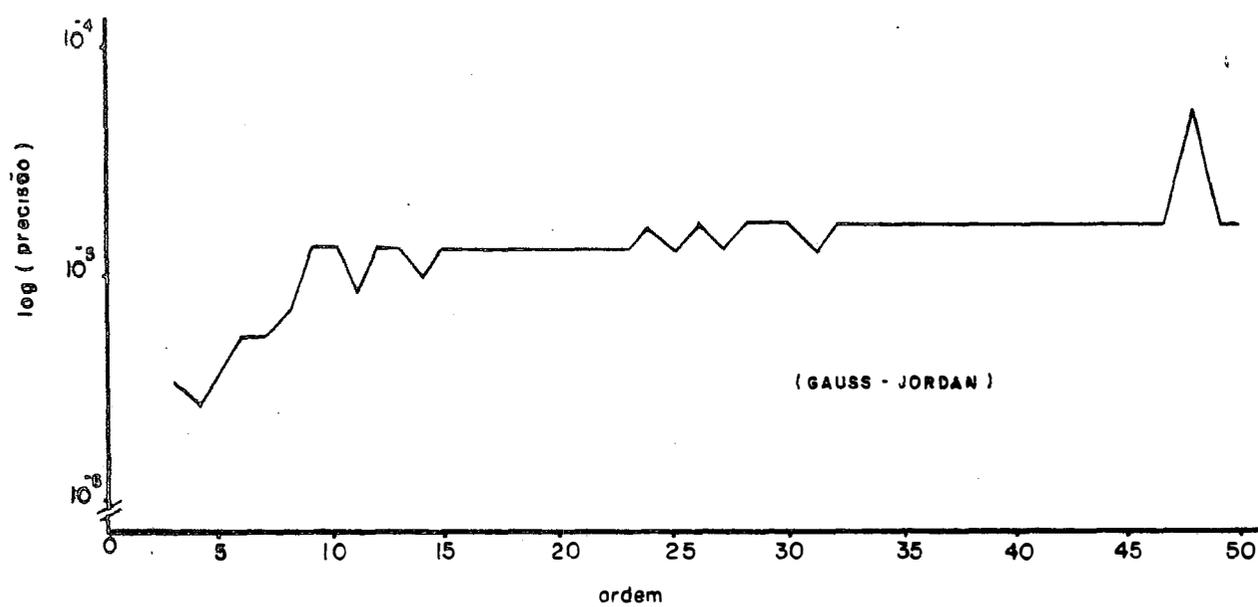


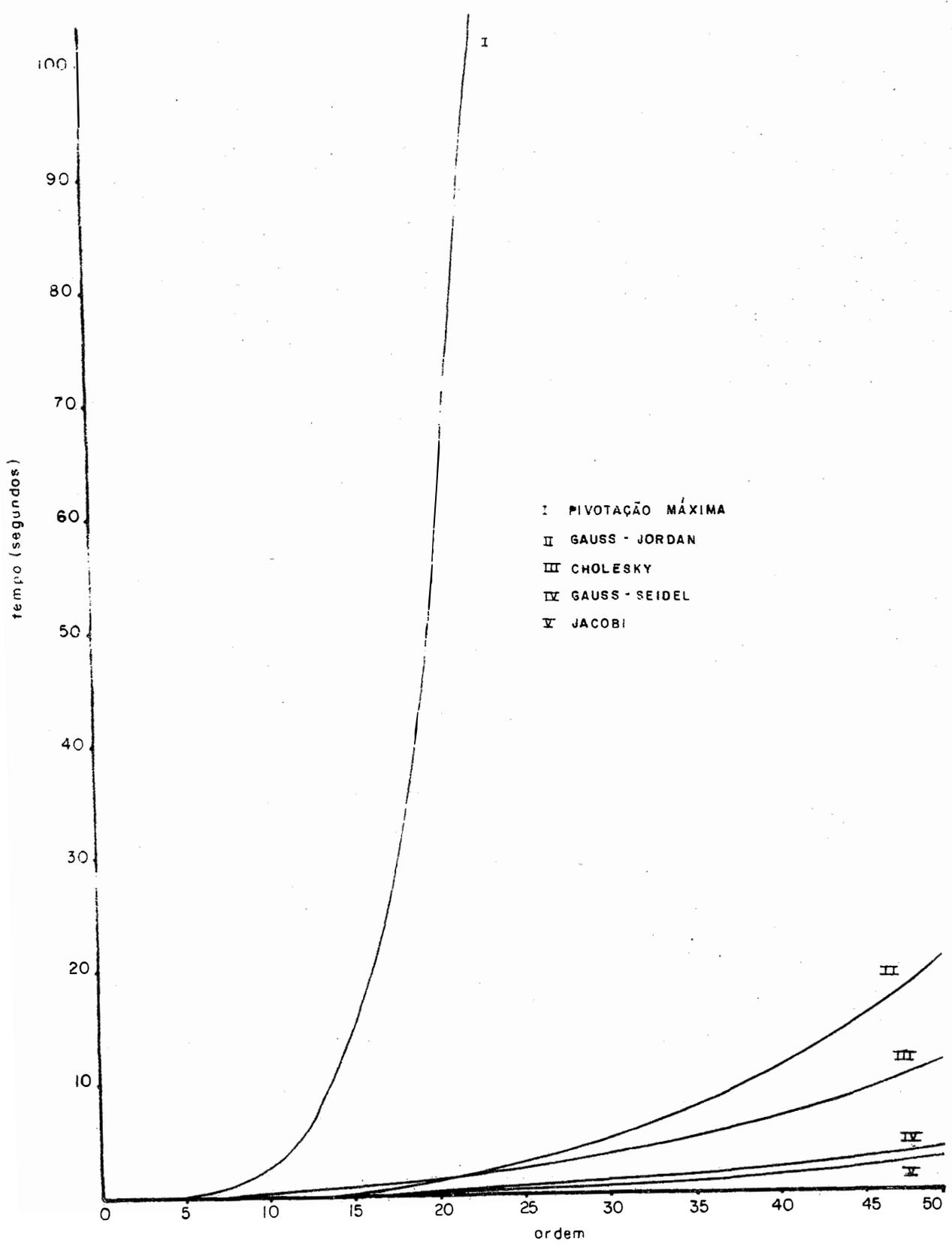




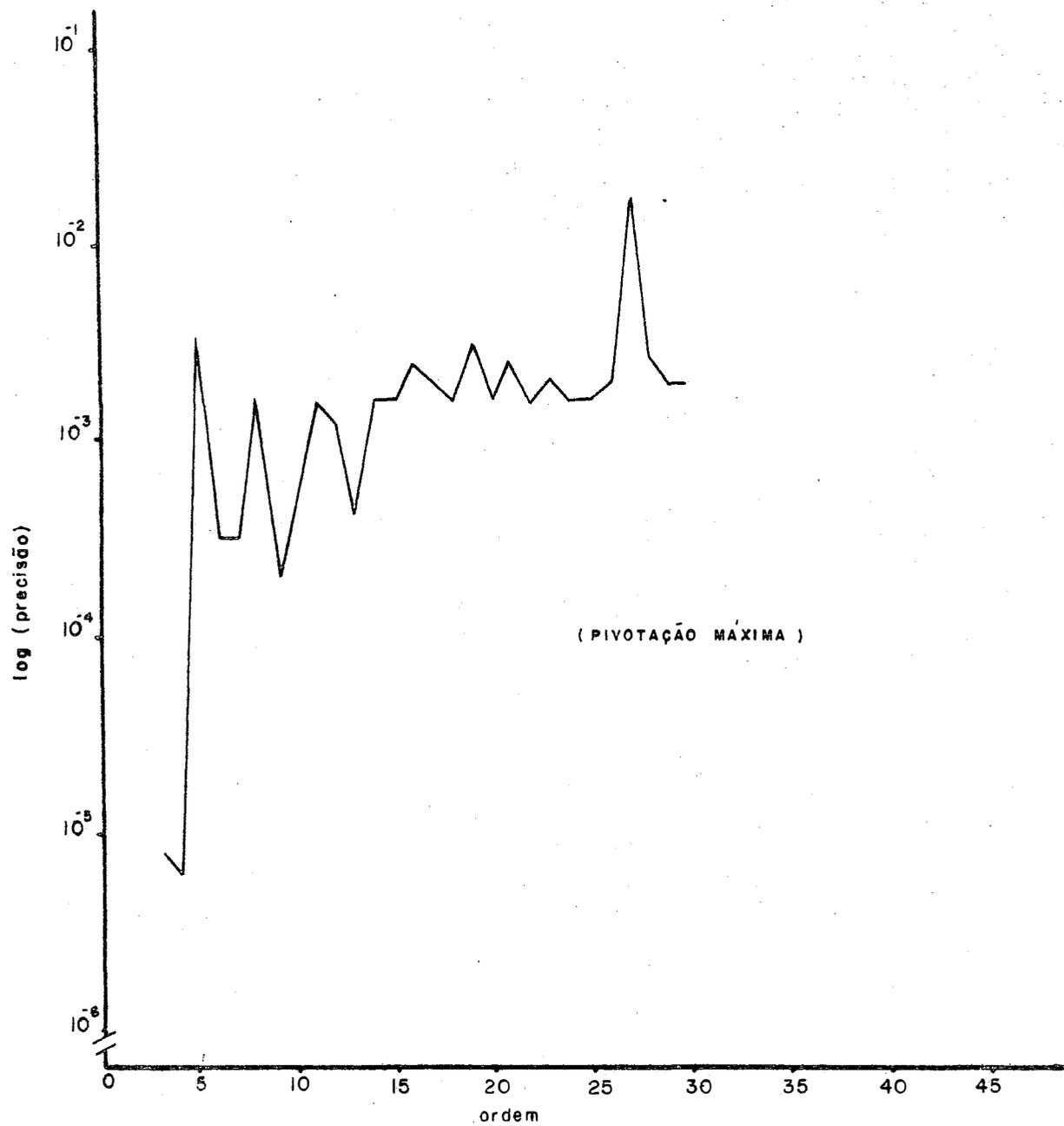
9.1.2. Os gráficos das páginas 53 e 54 correspondem à precisão das matrizes não simétricas estrita - mente diagonalmente dominante e o da página 55 ao tempo de processamento das mesmas matrizes.

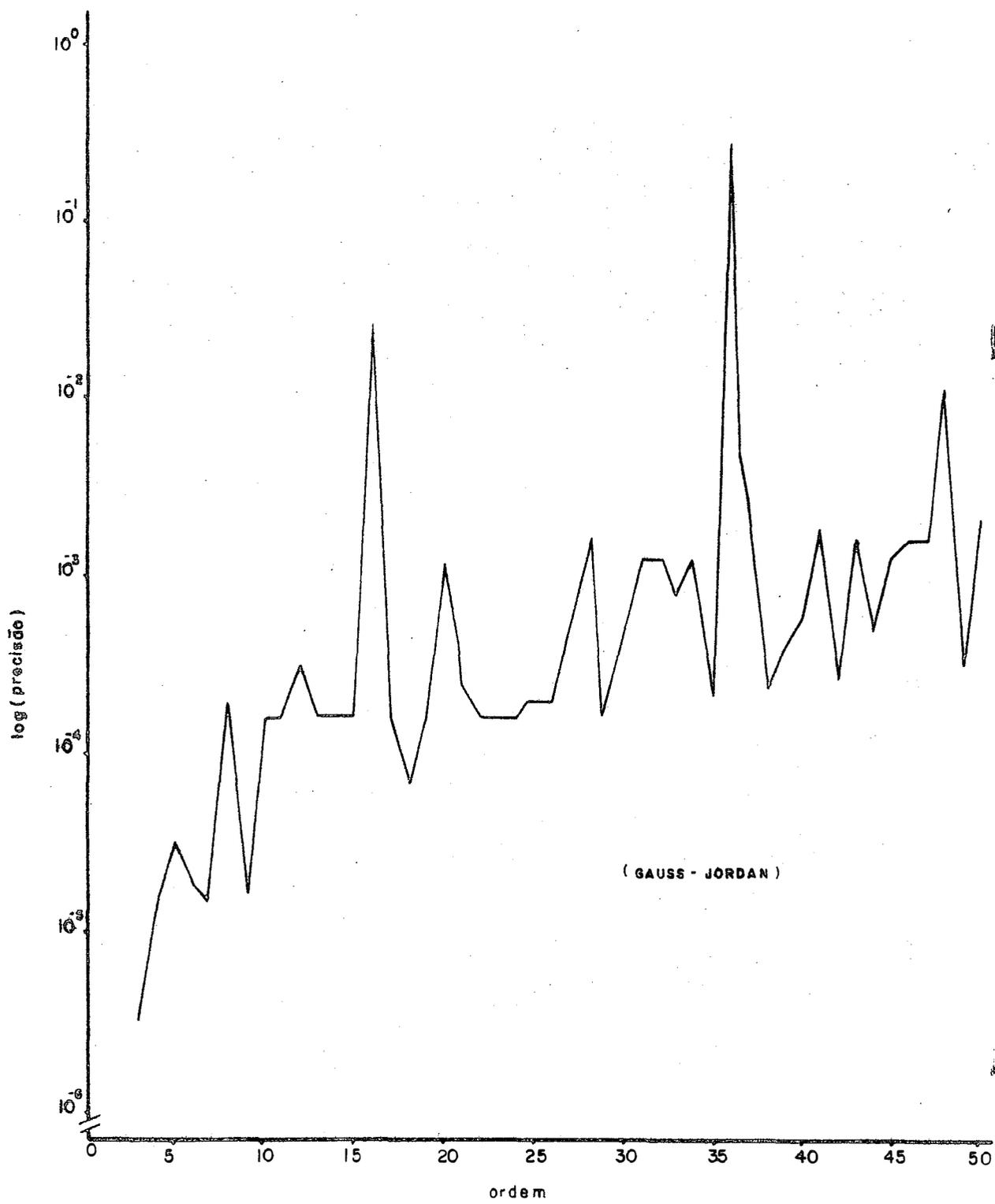


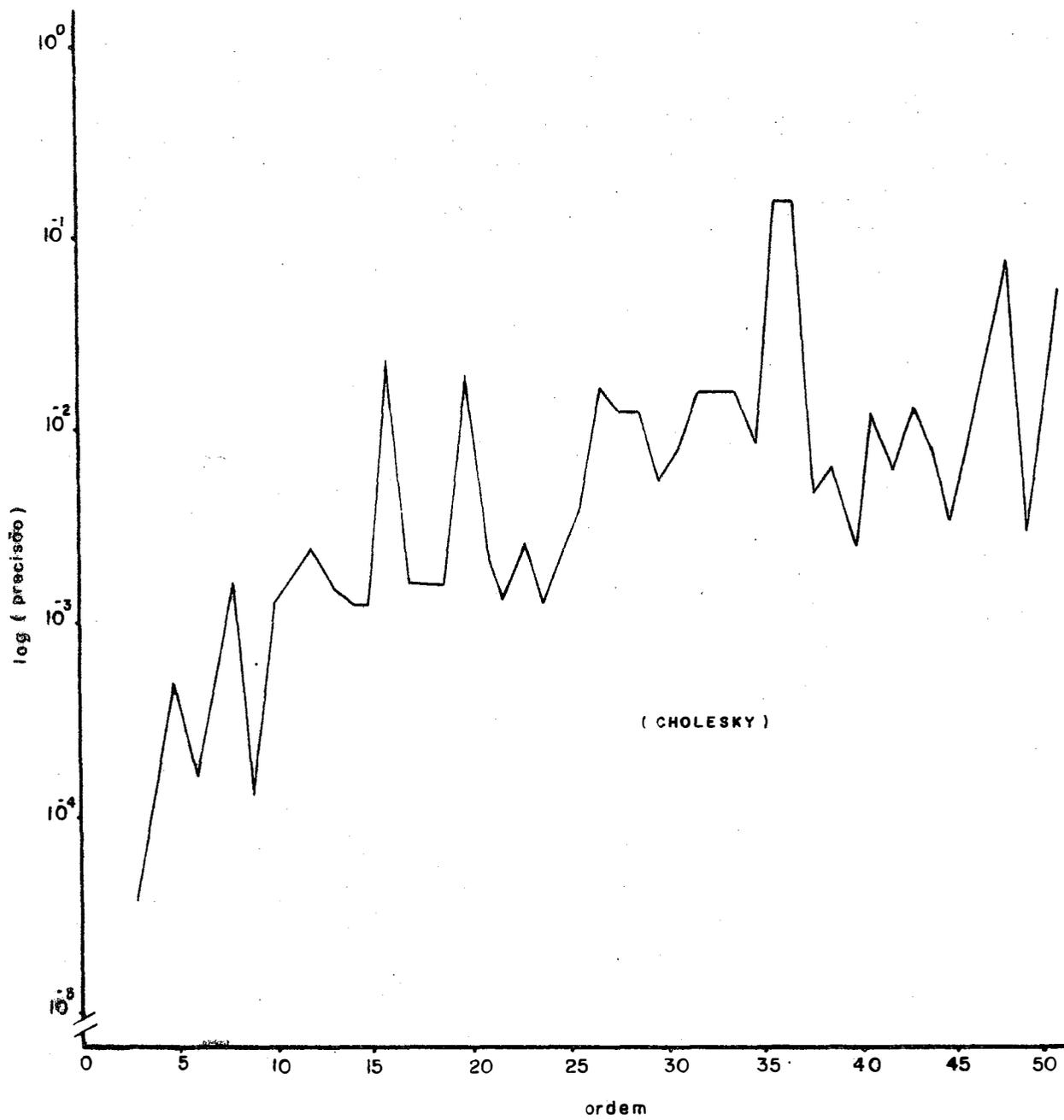


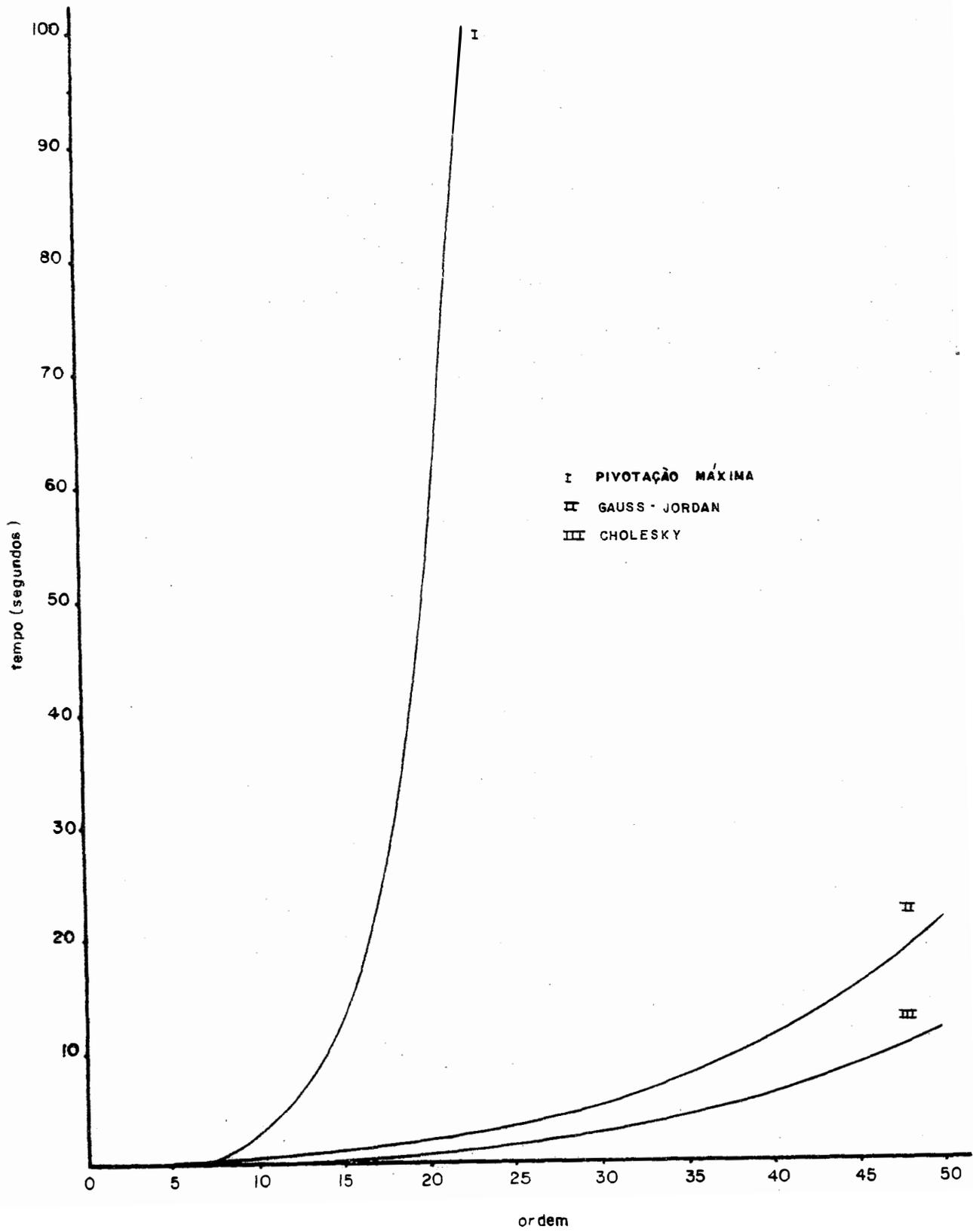


9.1.3. Os gráficos das páginas 57, 58 e 59 correspondem à precisão das matrizes simétricas sem dominância diagonal e o da página 60 ao tempo de processamento das mesmas matrizes.

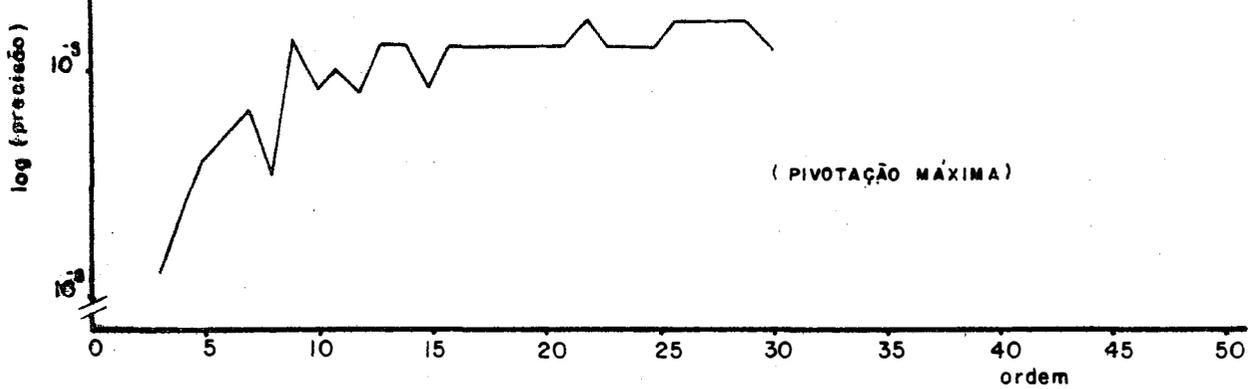
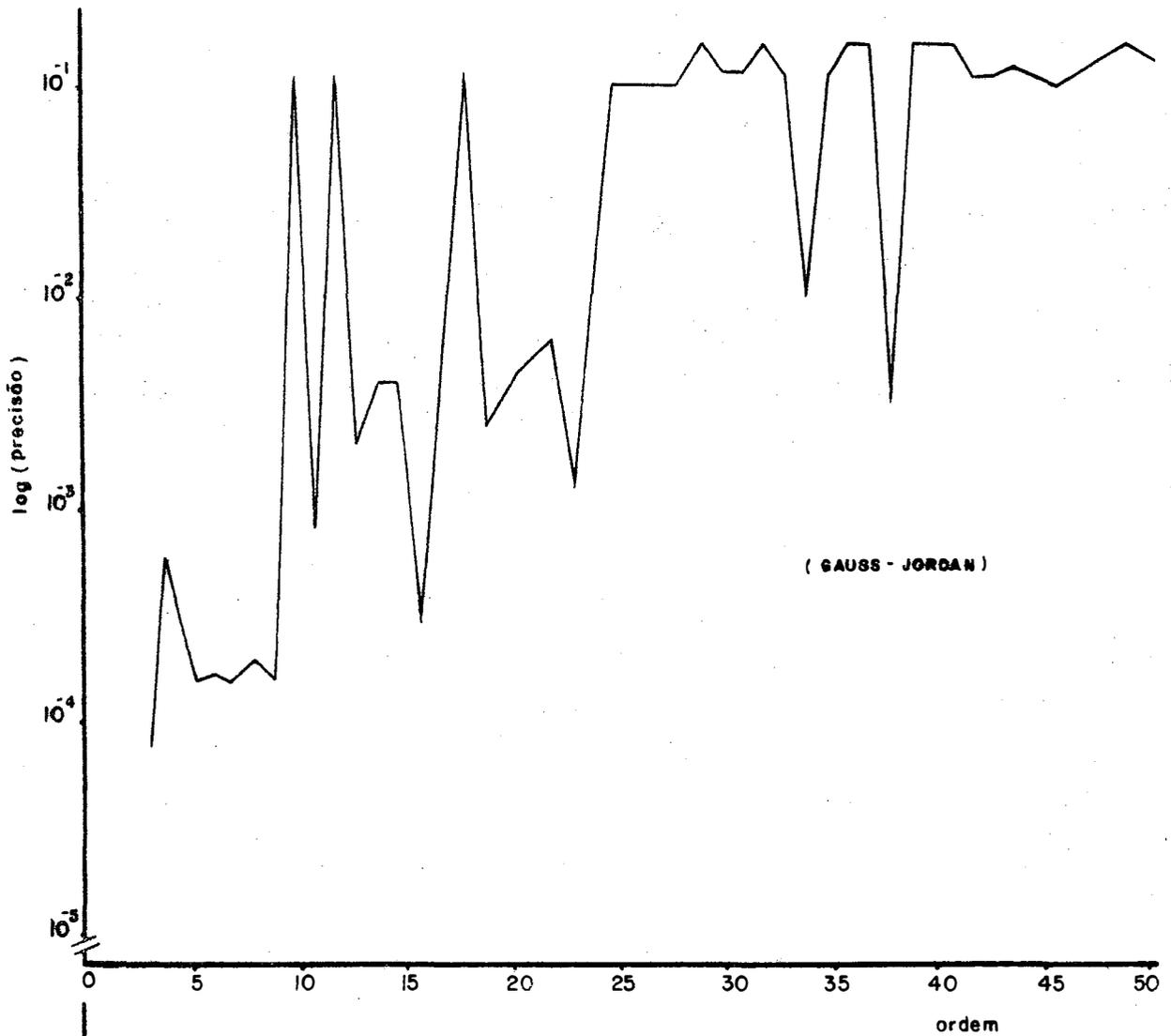


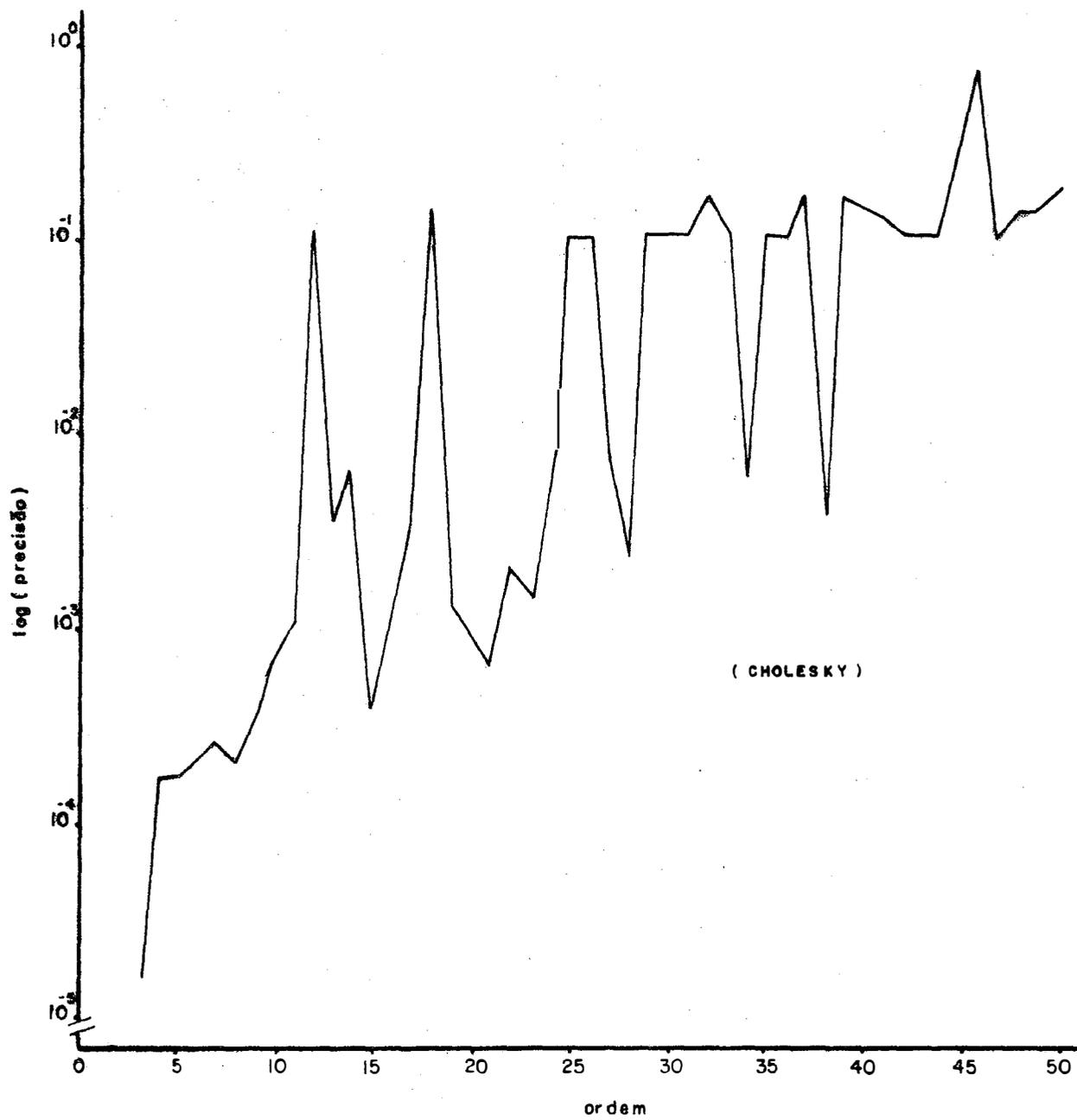


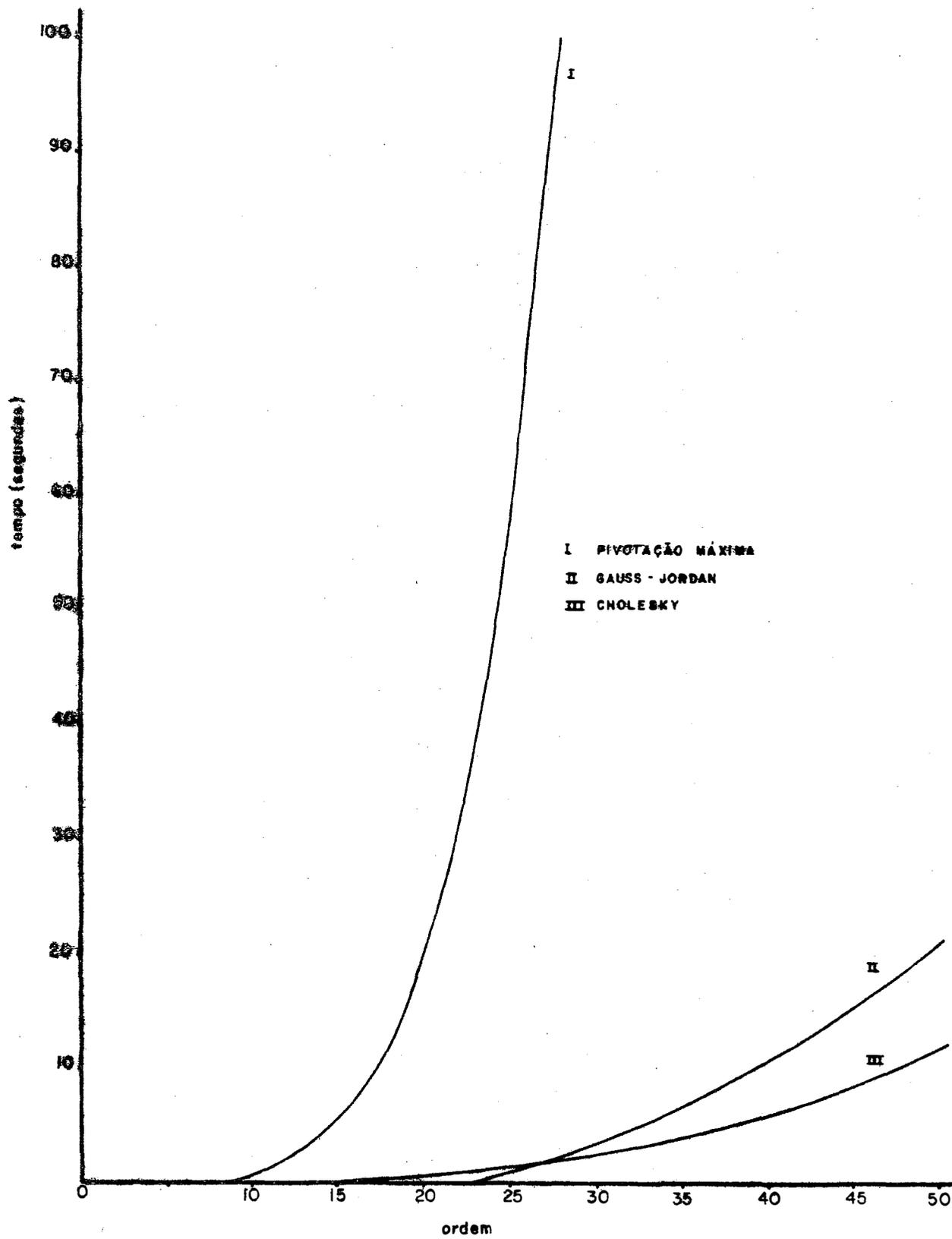




9.1.4. Os gráficos das páginas 62 e 63 correspondem à precisão das matrizes não simétricas sem dominância diagonal e o da página 64 ao tempo de processamento das mesmas matrizes.







9.2. Descrição dos Métodos

9.2.1. Método de Gauss-Jordan segundo CARNAHAN (1969)

Uma variação do método de eliminação que faz a retro-substituição simultânea, com a redução dos elementos sub diagonais, será ilustrado com um sistema de ordem 3 :

$$2x_1 - 7x_2 + 4x_3 = 9$$

$$x_1 + 9x_2 - 6x_3 = 1$$

$$-3x_1 + 8x_2 - 5x_3 = 6$$

Suponha que deseja-se B^{-1} , e forme a matriz aumentada $|B:U:I|$

$$\left[\begin{array}{ccc|ccc} 2 & -7 & 4 & 9 & 1 & 0 & 0 \\ 1 & 9 & -6 & 1 & 0 & 1 & 0 \\ -3 & 8 & 5 & 6 & 0 & 0 & 1 \end{array} \right]$$

Normaliza a primeira linha, dividindo-a pelo pivot, elemento 2, a seguir reduza os elementos restantes da primeira coluna a zero, pela subtração da nova primeira linha da segunda linha, e assim também por subtração de 3 vezes a nova primeira linha da terceira linha, o resultado é :

$$\left[\begin{array}{ccc|ccc} 1 & -7/2 & 2 & 9/2 & 1/2 & 0 & 0 \\ 0 & 25/2 & -8 & -7/2 & -1/2 & 1 & 0 \\ 0 & -5/2 & 11 & 39/2 & 3/2 & 0 & 1 \end{array} \right]$$

De maneira análoga, procura-se colocar 1 na posição a_{22} e zero na posição a_{32} e assim sucessivamente até que obtenha a matriz resultante $|I:XB^{-1}|$, onde X é o vetor solução e B^{-1} é a inversa da matriz original dos coeficientes, assim obtém-se :

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 4 & 93/235 & 67/235 & 6/235 \\ 0 & 1 & 0 & 1 & 13/235 & 22/235 & 16/235 \\ 0 & 0 & 1 & 2 & 7/47 & 1/47 & 5/47 \end{array} \right]$$

Conclui-se esta descrição desenvolvendo o algoritmo para o procedimento acima, que é chamado de eliminação de Gauss-Jordan.

Seja a matriz aumentada inicial $A_{n \times (n+1)}$, formada pelos $n \times n$ coeficientes da matriz original e pela coluna dos termos independentes.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & a_{2,n+1} \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & a_{n,n+1} \end{bmatrix}$$

Seja $m = n + 1$ e façamos $k = 1, 2, \dots, n$, ser o índice do pivot, de tal forma que o elemento a_{kk} seja o elemento pivot, para o k -ésimo passo de redução. Subentende-se, que os valores dos elementos de A serão mudados durante o processamento. O algoritmo é :

normalizando

$$a_{kj} = \frac{a_{kj}}{a_{kk}}, \quad j = m, m-1, \dots, k$$

redução

$$a_{ij} = a_{ij} - a_{ik} a_{kj}, \quad \begin{array}{l} j = m, m-1, \dots, k \\ i = 1, 2, \dots, n, \\ i \neq k \end{array}$$

$$K = 1, 2, \dots, n.$$

Observações :

- a) Como não aparecem elementos não nulos à esquerda de a_{kk} no k -ésimo passo, é desnecessário normalizar a_{kj} para $j < k$.
- b) Com a finalidade de evitar modificações prematuras de elementos da coluna pivot, o índice

ce j da coluna é sempre decrementado do seu maior valor m , até que a coluna pivot seja atingida.

Se um pivot for encontrado e sua grandeza for menor que ε (um valor bem pequeno, p.ex. 10^{-15}), os cálculos são interrompidos, pois a matriz pode ser singular.

9.2.2. Redução de Gauss-Jordan com uso da pivotação máxima segundo CARNAHAN (1969)

Na redução de Gauss-Jordan com a estratégia do pivot diagonal, para pivot no k -ésimo ciclo era escolhido o elemento da posição a_{kk} , isto é, um elemento da diagonal. Esta estratégia pode levar a dificuldades computacionais, p.ex. divisão por zero, para alguns sistemas de equações. Se fosse $a_{ik} = 0$, então os passos de normalização do algoritmo, não deveriam ser executados. Quando a_{kk} é de uma grandeza relativamente pequena, comparadas com a de outros elementos nas primeiras colunas da matriz A , as operações de normalização e redução, podem muitas vezes ser efetuadas, mas os resultados computacionais, podem não conduzir à solução, por causa do erro de arredondamento.

Em tais casos, desde que a matriz dos coeficientes não seja quase-singular, é sempre possível, superar o problema, pelo rearranjo das linhas ou colunas da matriz dos coeficientes, isto é equivalente a reordenação de equações

ou renumeração das variáveis, respectivamente. Uma alternativa possível é, escolher algum elemento de grandeza significativa, que ainda esteja disponível para pivotamento (não em uma linha ou coluna já contendo pivot), como pivot, para o k-ésimo passo. Geralmente, o maior de tais elementos é o preferido, uma vez que seu uso conduz ao menor erro de arredondamento no passo de redução.

É necessário somente manter a lista dos índices de linhas e colunas dos sucessivos elementos pivot. Seja r_k e c_k , respectivamente, linha e coluna subscrita do k-ésimo elemento pivot, p_k .

O algoritmo é :

Realização

$$d = 1$$

escolha do pivot

$$p_k = a_{rkck}, |a_{rkck}| = \max_{i,j} |a_{ij}|,$$

$$i = 1, 2, \dots, n,$$

$$i \neq r_1, r_2, \dots, r_{k-1}$$

$$j = 1, 2, \dots, n,$$

$$j \neq c_1, c_2, \dots, c_{k-1} \quad k = 1, 2, \dots, n$$

atualizando o determinante

$$d = dp_k$$

normalizando

$$a_{rkj} = a_{rkj}/p_k \quad j = 1, 2, \dots, n+1$$

reduzindo

$$a_{ij} = a_{ij} - a_{ick} a_{rkj}$$

$$j = 1, 2, \dots, n + 1$$

$$i = 1, 2, \dots, n$$

$$i \neq r_k$$

Aos elementos r_k e c_k devem ser assinalados valores apropriados, uma vez que o k -ésimo pivot tenha sido selecionado. Assim d é o determinante (possivelmente com o sinal incorreto) da matriz dos coeficientes. Após a redução ter sido completada, as primeiras n linhas de A conterão possivelmente uma matriz identidade permutada. A $(n + 1)$ -ésima coluna conterá o vetor solução $x = [x_1, x_2, \dots, x_n]'$, embora os elementos, geralmente, estejam fora de ordem. As soluções podem ser postas em ordem, pelo uso das informações contidas nos vetores r e c como segue :

$$x_{ck} = a_{rk, n+1}, \quad k = 1, 2, \dots, n. \quad (1)$$

A matriz dos coeficientes, transformada resultante, tem um determinante igual ± 1 (é uma matriz elementar a segunda classe ou matriz identidade). Em consequência disto, o valor computado para o determinante d , pode estar com o sinal incorreto. É preciso determinar se o número de permutações entre linhas, que é exigido para reordenar a matriz permutação final, como matriz identidade, é par ou ímpar. O problema pode ser resolvido, formando um vetor auxiliar j , tal que

$$j_{rk} = c_k, \quad k = 1, 2, \dots, n \quad (2)$$

e em seguida contando o número de pares de trocas requeridas, para colocar o vetor j numa sequência ascendente. Se o número de permutações é par, o sinal de d está correto, se é ímpar está incorreto.

O método é provavelmente melhor ilustrado com um exemplo numérico.

Consideremos o conjunto de equações abaixo, cuja solução exata é o vetor $[4, 1, 2]^t$

$$-3x_1 + 8x_2 + 5x_3 = 6$$

$$2x_1 - 7x_2 + 4x_3 = 9$$

$$x_1 + 9x_2 - 6x_3 = 1$$

A matriz de coeficientes aumentada, para este sistema é :

$$\left[\begin{array}{ccc|c} -3 & 8 & 5 & 6 \\ 2 & -7 & 4 & 9 \\ 1 & 9 & -6 & 1 \end{array} \right]$$

Então, para $k = 1$, isto é, para o primeiro passo, o elemento a_{32} , tem a maior magnitude dos nove elementos possíveis e $p_1 = 9$, $r_1 = 3$, $c_1 = 2$. O valor do determinante d é atualizado, de seu valor inicial 1 até 9, em ordem crescente. Após os passos de normalização, para o primeiro ciclo, a matriz transformada é :

$$\begin{bmatrix} -3 & 8 & 5 & 6 \\ 2 & -7 & 4 & 9 \\ 1/9 & 1 & -2/3 & 1/9 \end{bmatrix}$$

os passos de redução levariam a

$$\begin{bmatrix} -35/9 & 0 & \underline{31/3} & 46/9 \\ 25/9 & 0 & -2/3 & 88/9 \\ 1/9 & 1 & -2/3 & 1/9 \end{bmatrix}$$

Para o segundo ciclo, somente a_{11} , a_{13} , a_{21} e a_{23} estão disponíveis para serem pivots. Uma vez que, a_{13} tem a maior magnitude dos quatro candidatos, $p_2 = 31/3$, $r_2 = 1$ e $c_2 = 3$. O valor do determinante é tomado de $9 \cdot 31/3 = 93$. Após os passos de redução e normalização, a matriz se torna :

$$\begin{bmatrix} -35/93 & 0 & 1 & 46/93 \\ \underline{235/93} & 0 & 0 & 940/93 \\ -13/93 & 1 & 0 & 41/93 \end{bmatrix}$$

Para o terceiro e último ciclo, somente a_{21} está disponível para pivotamento. Então $p_3 = 235/93$, $r_3 = 2$ e $c_3 = 1$. O valor determinante é tomado até $93 \cdot 235/93 = 235$, e após redução e normalização, a matriz é :

$$\begin{bmatrix} 0 & 0 & 1 & | & 2 \\ 1 & 0 & 0 & | & 4 \\ 0 & 1 & 0 & | & 1 \end{bmatrix} .$$

As três primeiras colunas contêm uma matriz identidade permutada e a quarta contêm as soluções, embora não em ordem natural. Os vetores r e c contêm os valores

k	r_k	c_k
1	3	2
2	1	3
3	2	1

De (1), as soluções são

$$x_1 = a_{24} = 4$$

$$x_2 = a_{34} = 1$$

$$x_3 = a_{14} = 2$$

O vetor auxiliar j , formado de acordo com (2),

é:

$$j_1 = 3$$

$$j_2 = 1$$

$$j_3 = 2$$

Uma vez que, são exigidos dois pares de trocas, para colocar j na sequência ascendente, o sinal de d está correto, e o determinante da matriz original dos coeficientes é 235.

Na implementação o valor do determinante não é usado.

9.2.3. Método iterativo de Jacobi segundo CARNAHAN (1969).

Considere o sistema $BX = U$. (1)

Formulamos agora, o método iterativo de Jacobi, para resolver o sistema linear (1).

Primeiramente resolvemos para x_1 , resultando,

$$x_1 = (U_1 - b_{12}x_2 - b_{13}x_3 - \dots - b_{1n}x_n) / b_{11}$$

$$x_2 = (U_2 - b_{21}x_1 - b_{23}x_3 - \dots - b_{2n}x_n) / b_{22}$$

$$\vdots \quad (2)$$

$$x_n = (U_n - b_{n1}x_1 - b_{n2}x_2 - \dots - b_{n,n-1}x_{n-1}) / b_{nn}$$

O sistema (2), pode ser escrito de maneira mais consisa, como,

$$x_i = \frac{\left(U_i - \sum_{\substack{j=1 \\ j \neq i}}^n b_{ij} x_j \right)}{b_{ii}}, \quad i = 1, 2, \dots, n \quad (3)$$

Note que o rearranjo acima exige b_{ii} diferente de zero. Em geral tentamos reordenar as equações e as incógnitas, de modo a obter, a dominância diagonal, isto é, de forma que cada elemento b_{ii} é maior em valor absoluto que todos os outros coeficientes na linha e na coluna i .

Depois fazemos tentativas iniciais, para os x 's, e inserimos estes valores nos elementos do segundo membro de (2). As novas aproximações resultantes para os x 's, são resubstituídas no segundo membro de (2), e o processo é repetido.

Provavelmente, os x 's assim computados, mostra

rão pequenas variações, após várias iterações feitas.

Para se estabelecer um critério de convergência, considere as equações rearranjadas (2), com o sistema $X = AX + V$, no qual

$$A = \begin{bmatrix} 0 & b_{12}/b_{11} & b_{13}/b_{11} & \dots & b_{1n}/b_{11} \\ b_{21}/b_{22} & 0 & b_{23}/b_{22} & \dots & b_{2n}/b_{22} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1}/b_{nn} & b_{n2}/b_{nn} & b_{n3}/b_{nn} & \dots & 0 \end{bmatrix}, \quad V = \begin{bmatrix} V_1/b_{11} \\ V_2/b_{22} \\ \vdots \\ V_n/b_{nn} \end{bmatrix} \quad (4)$$

Se o vetor inicial X_0 está próximo do vetor de solução, a convergência será rápida. Em cada elemento definimos

$$x_{k+1} = Ax_k + v \quad (5)$$

no qual o subscrito k é o número de iterações. Isto significa que

$$x_k = A^k x_0 + [I + A + A^2 + \dots + A^{k-1}]v.$$

Daí vemos que a convergência exige, normalmente que, $\lim_{k \rightarrow \infty} A^k = 0$

(6) e a condição necessária e suficiente é:

$$\lim_{k \rightarrow \infty} [I + A + A^2 + \dots + A^k] = (I - A)^{-1}.$$

Assim quando (6) é satisfeita, $X = \lim_{k \rightarrow \infty} X_k$ existe e $X = 0 + (I - A)^{-1}v$,

isto é $(I - A)x = v$ ou $x = Ax + v$. Desta forma a convergência depende de ser verdadeira (6).

Pode-se provar que (6) é verdadeira, se e somente se, todos os auto-valores de A, em módulo, forem menores que a unidade. Para que isto se dê, temos a condição suficiente, tomando

$$|\lambda_k| \leq \max_{j=1}^n \sum_{i=1}^n |a_{ij}| \quad e$$

$|\lambda_k| \leq \max_{i=1}^n \sum_{j=1}^n |a_{ij}|$ e desenvolvendo subsequente;

$$\sum_{i=1}^n |a_{ij}| \leq \mu < 1, \quad 1 \leq j \leq n,$$

ou

$$\sum_{j=1}^n |a_{ij}| \leq \mu < 1, \quad 1 \leq i \leq n, \quad (7)$$

ou

$$\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2 \leq \mu < 1.$$

Pelo emprego de (4), as condições de suficiência podem também ser transformadas, num conjunto equivalente de condições, sobre os elementos da matriz original B dos coeficientes. Por exemplo a segunda condição de (7), torna-se

$$\sum_{j=1}^n |a_{ij}| < |b_{ii}|, \quad \text{para } 1 \leq i \leq n, \quad (8).$$

$i \neq j$

Se, como ocorre frequentemente, a matriz B for

irredutível (isto é, uma matriz da forma

$$\begin{bmatrix} B_1 & 0 \\ B_2 & B_3 \end{bmatrix},$$

onde B_1 é quadrada, 0 é a matriz nula; não pode ser encontrada permutando as linhas e colunas de B), a condição suficiente pode ser relaxada, para

$$\sum_{\substack{j=1 \\ i \neq j}}^n |b_{ij}| \leq |a_{ii}| \quad 1 \leq i \leq n$$

com a desigualdade valendo estritamente para pelo menos um valor de i .

9.2.4. Método de Gauss-Seidel segundo CARNAHAN (1969)

Consideremos o sistema linear (1), descrito em Jacobi, refrazeando na forma (2) ou na forma $X = AX + V$. Nas iterações, contudo, os elementos recomputados do vetor X , são sempre usados do lado direito do sinal de igual, tão logo sejam obtidos. Isto contrasta com o método de Jacobi, no qual os novos componentes não são usados até que todos os n componentes tenham sido encontrados.

Para ilustrar, aplicaremos o método a um exemplo numérico; a fórmula usada é:

$$x_1 = 11/4 - x_2/2 - x_3/4,$$

$$x_2 = 3/2 + x_1/2,$$

$$x_3 = 4 - x_1/2 - x_2/4,$$

com a compreensão de que os x 's mais recentemente calculados, são sempre usados no segundo membro da igualdade. Agora x_0 inicial é escolhido, podendo ser $[1, 1, 1,]'$. A primeira iteração fornece,

$$x_{11} = 11/4 - (1/2).1 - (1/4).1 = 2$$

$$x_{21} = 3/2 + (1/2).2 = 5/2$$

$$x_{31} = 4 - (1/2).2 - (1/4)(5/2) = 19/8$$

isto é:

$$x_1 = [2, 5/2, 19/8]'$$

Analogamente, as próximas iterações fornecem

$$x_2 = [29/32, 125/64, 783/256]'$$

$$x_3 = [1033/1024, 4095/2048, 24541/8192]'$$

Observe que neste exemplo a razão de convergência é mais rápida que Jacobi.

A fim de investigar a condição de convergência do método de Gauss-Seidel, nós primeiramente frazeamos a iteração em termos de componentes individuais. Seja x_{ik} , denotamos a k -ésima aproximação ao i -ésimo componente do vetor solução $x = [x_1, x_2, \dots, x_n]'$. Seja $[x_{10}, x_{20}, \dots, x_{n0}]'$ uma aproximação inicial arbitrária (tal como o método de Jacobi, se uma boa estimativa é conhecida, poderia ser resolvido com eficiência). Sejam A e V os mesmos como dados em (4) no Jacobi, e defina,

$$x_{ik} = \sum_{j=1}^{i-1} a_{ij}x_{jk} + \sum_{j=i+1}^n a_{ij}x_{j,k-1} + v_i \quad (9)$$

para $1 \leq i \leq n$ e $1 \leq k$. Quando $i=1$, $\sum_{j=1}^{i-1} a_{ij}x_{jk}$ é interpreta-

do como zero

Escrevemos $A = A_L + A_R$, onde

$$A_L = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ a_{21} & 0 & \dots & 0 & 0 \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ a_{n1} & a_{n2} & \dots & a_{n,n-1} & 0 \end{bmatrix},$$

$$A_R = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ 0 & 0 & \dots & a_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & \dots & 0 \end{bmatrix}$$

Desta forma A_L é uma matriz estritamente triangular inferior, cujas entradas subdiagonais, são os elementos de A nas suas posições naturais. Uma descrição similar se aplica a A_R . Vê-se que se $x_k = [x_{1k}, x_{2k}, \dots, x_{nk}]'$, $x_k = A_L x_k + A_R x_{k-1} + v$. Isto pode ser parafraseado como:

$$x_k = (I - A_L)^{-1} A_R x_{k-1} + (I - A_L)^{-1} v,$$

o qual é da forma de Jacobi. Isto significa que uma condição necessária e suficiente, para a convergência de (9), é que os auto-valores de $(I - A_L)^{-1} A_R$ sejam menores que a unidade em módulo. Os auto-valores de $(I - A_L)^{-1} A_R$ encontram-se pela solução do determinante

$$\left[(I - A_L)^{-1} A_R - \lambda I \right] = 0$$

ou determinante

$$(I - A_L)^{-1} \cdot [A_R - \lambda(I - A_L)] = 0,$$

ou determinante

$$(A_R - \lambda I + \lambda A_L) = 0.$$

Desta forma o método de Gauss-Seidel converge, se e somente se, as raízes do determinante de

$$\begin{bmatrix} -\lambda & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21}\lambda & -\lambda & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32}\lambda & -\lambda & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1}\lambda & a_{n2}\lambda & a_{n3}\lambda & \dots & -\lambda \end{bmatrix} \quad (10)$$

forem menores que 1 (um) em valores absolutos.

Uma vez que $a_{ii} = 0$, $1 \leq i \leq n$ e $a_{ij} = -b_{ij}/b_{ii}$, para $i \neq j$, o determinante de (10), tem as mesmas raízes que o determinante de

$$\begin{bmatrix} b_{11}\lambda & b_{12} & b_{13} & \dots & b_{1n} \\ b_{21}\lambda & b_{22}\lambda & b_{23} & \dots & b_{2n} \\ b_{31}\lambda & b_{32}\lambda & b_{33}\lambda & \dots & b_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n1}\lambda & b_{n2}\lambda & b_{n3}\lambda & \dots & b_{nn}\lambda \end{bmatrix} \quad (11)$$

Isto mostra que condições análogas às duas pri

meiras de (7), em Jacobi, são suficientes para garantir convergência a saber:

$$\sum_{\substack{j=1 \\ j \neq i}}^n |b_{ij}/b_{ii}| \leq \mu < 1 \quad \text{ou} \quad \sum_{\substack{j=1 \\ j \neq i}}^n |b_{ji}/b_{jj}| \leq \mu < 1, \quad 1 \leq i \leq n \quad (12)$$

A primeira destas, pode ser demonstrada como segue. É sabido que se

$$|b_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |b_{ij}|,$$

B é não singular. Assim o vetor solução x existe, tal que, $x = Ax + v$, donde

$$x_i = \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j + v_i, \quad \text{no qual } a_{ij} = -b_{ij}/b_{ii}.$$

Substituindo isto em (9), obtem-se

$$|x_{ik} - x_i| \leq \sum_{j=1}^{i-1} |a_{ij}| |x_{jk} - x_j| + \sum_{j=i+1}^n |a_{ij}| |x_{j,k-1} - x_j| \quad (13)$$

Façamos e_k denotar o máximo dos números $|x_{ik} - x_i|$ de acordo com a variação de i . Então

$$|x_{1k} - x_1| \leq \sum_{j=2}^n |a_{1j}| e_{k-1} \leq \mu e_{k-1} < e_{k-1}$$

Substituindo isto em (13) obtem-se

$$|x_{2k} - x_2| \leq |a_{21}| e_{k-1} + \sum_{j=3}^n |a_{2j}| e_{k-1} \leq \mu e_{k-1}$$

Continuando como indicado chega-se $|x_{ik} - x_i| \leq \mu e_{k-1}$, $1 \leq i \leq n$.

isto significa, naturalmente, que $|x_{ik} - x_i| \leq \mu^k e_0$, donde, uma

vez que $0 < \mu < 1$, $\lim_{k \rightarrow \infty} x_{ik} = x_i$

9.1.5. Método de Cholesky segundo BAJPAI et alii (1977)

Seja o sistema $AX = B$. Sabe-se que a matriz A pode ser decomposta num produto de duas matrizes triangulares, uma superior L e uma inferior U .

Além disso as matrizes L e U não são únicas, pois podemos escolher à vontade quais serão os elementos das diagonais principais de L e U , e isto deve ser feito de modo a obter a forma mais simples possível.

Temos

$$A = LU$$

podemos fazer todos os elementos da diagonal de L iguais a 1, então L terá a forma :

$$L = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ l_{12} & 1 & 0 & \dots & 0 \\ l_{31} & l_{32} & 1 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & \dots & 1 \\ n_1 & n_2 & n_3 & \dots & \cdot \end{bmatrix}$$

e U será :

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \dots & u_{nn} \end{bmatrix}$$

O sistema $AX = B$ (1), pode ser escrito da forma, $LUX = B$ (2)

fazendo

$$UX = Y \quad (3)$$

podemos reescrever (2)

$$LY = B,$$

cuja solução é facilmente encontrada por substituição direta.

Obtido o vetor Y , solução de (4), podemos resolver o sistema (3), por retrossubstituição, o que nos conduzirá ao vetor solução X .

O algoritmo então é :

Fatoração

$$A = LU$$

$$u_{1j} = a_{1j}, \quad \text{para } j = 1, 2, \dots, n.$$

$$j = 1$$

enquanto $j < n - 1$

fazemos

$$m = j + 1$$

$$\text{se } j = 1, \text{ então } l_{ij} = a_{ij}/u_{jj}$$

$$\text{se } j > 1, \text{ então } l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj})/u_{jj} \quad i=m, m+1, \dots, n$$

$$u_{mi} = a_{mi} - \sum_{k=1}^{m-1} l_{mk} u_{ki}$$

$$\text{fim } j = j + 1$$

$$\text{resolver } LY = B$$

$$\text{resolver } UX = Y$$

9.1.6. Método dos gradientes conjugados segundo RALSTON (1965).

Se A é uma matriz quadrada, simétrica e positiva definida o sistema $Ax = b$, pode ser escrita assim

$$Q = 1/2(x'Ax) - x'b$$

Podemos procurar uma sequência de vetores $\{x_i\}$ tal que $Q(x_{i+1}) < Q(x_i)$ se $Q(x_i) \neq 0$. Esta sequência conduz a solução de $Ax = b$. Na prática arredondamentos nos impedem chegar a esta convergência teórica.

Seja v_1, v_2, \dots, v_n uma base para o espaço Euclídiano de dimensão n , e seja x_t a solução verdadeira do sistema $Ax = b$. Então se x_1 é nossa aproximação inicial da solução, escrevemos

$$x_t - x_1 = \sum_{j=1}^n \alpha_j v_j \quad (1)$$

Se por exemplo, os v_j 's são ortogonais, então

$$\alpha_j = \frac{v_j^t (x_t - x_1)}{v_j^t v_j}, \quad j = 1, 2, \dots, n \quad (2)$$

mas isto é de pouca ajuda, uma vez que, estamos procurando pelo x_t . Por outro lado suponhamos que

$$v_j^t A v_i = 0 \quad i \neq j \quad (3)$$

Neste caso os v_j são chamados A-ortogonal ou A-conjugados. Então

$$\alpha_j = \frac{v_j^t A (x_t - x_1)}{v_j^t A v_j} = \frac{v_j^t r_1}{v_j^t A v_j}, \quad j = 1, 2, \dots, n \quad (4)$$

do qual α_j 's são facilmente calculados. (Note que $v_j^t A v_j \neq 0$, se A é positiva definida). A idéia básica do método dos gradientes conjugados é gerar uma sequência $\{v_j\}$ dos vetores A-ortogonais, calcular α_j 's como em (4), e desta forma atingir x_t .

Para gerar um conjunto de vetores A-ortogonal, supõe-se que existe um conjunto de vetores u_1, u_2, \dots, u_n que forma uma base. Então emprega-se o procedimento de ortogonalização de Gram-Schmidt, para calcular

$$v_1 = u_1$$

$$v_{i+1} = u_{i+1} + \sum_{k=1}^i \beta_{i+1,k} v_k, \quad i = 1, 2, \dots, n-1 \quad (5)$$

onde

$$\beta_{i+1,k} = - (v_k' A u_{i+1} / v_k' A v_k) \quad (6)$$

Uma vez que os u_i 's são linearmente independentes, pode-se mostrar que os v_i 's também são. Resta então escolher a sequência $\{u_j\}$. Uma possibilidade poderia ser, empregar os n vetores coordenados. Se for feito isto pode-se mostrar que, o procedi-
mento resultante é funcionalmente equivalente a eliminação de Gauss. Uma outra alternativa que leva a um algoritmo mais conveniente, contudo, será a seguinte, define-se:

$$x_i = x_1 + \sum_{j=1}^{i-1} \alpha_j v_j' \quad i = 1, 2, \dots, n \quad (7)$$

como os α_j 's dado por (4). Então $x_{i+1} = x_i + \alpha_i v_i$, $i=1, 2, \dots, n$
(8) de acordo com $(x_{i+1} = x_i + \alpha_i v_i)$. O i -ésimo resíduo é da-
do por $r_i = b - A x_i$, $i = 1, 2, \dots, n$ (9)

O princípio do método de gradientes conjugados, está em escolher $u_i = r_i$, $i = 1, 2, \dots, n$. Pode-se provar que, quando a sequência $\{r_i\}$ termina (isto é, tem algum $r_i = 0$) antes de $i = n+1$, neste caso temos uma solução, ou a sequên-
cia forma uma base ortogonal para um espaço n -dimensional. Tam-
bém pode-se mostrar que $Q(x_{i+1}) < Q(x_i)$. A priori, nós não conhecemos a sequência $\{r_i\}$, mas podemos determinar os v_i 's e r_i 's por um processo iterativo

, Da aproximação inicial temos

$$r_1 = b - A x_1$$

de (5)

$$v_1 = r_1$$

de (4) e (7)

$$\alpha_1 = v_1' r_1 / v_1' A v_1$$

e

$$x_2 = x_1 + \alpha_1 v_1$$

então

$$r_2 = b - A x_2 = r_1 - \alpha_1 A v_1$$

e

$$v_2 = r_2 + \beta_{21} v_1$$

onde

$$\beta_{21} = - v_1' A r_2 / v_1' A v_1$$

As equações do algoritmo geral são:

$$\alpha_i = v_i' r_i / v_i' A v_i$$

$$x_{i+1} = x_i + \alpha_i v_i$$

$$r_{i+1} = r_i - \alpha_i A v_i \quad i = 1, 2, \dots, n$$

$$\beta_i = - v_i' A r_{i+1} / v_i' A v_i \quad v_1 = r_1 = b - A x_1 \quad (10)$$

$$v_{i+1} = r_{i+1} + \beta_i v_i$$

Note que α_i dado acima, difere de α_i dado por (4). Contudo, usando a ortogonalidade dos v_i 's, pode-se mostrar que as duas formas tornam-se idênticas. Também v_{i+1} em (10) envolve somente um único β_i em contraste com (5). Mas novamente usando a ortogonalidade dos v_i 's, pode ser mostrado que todos os outros β_{ik} 's desaparecem. De (1) e (10) segue que $x_{n+1} = x_t$.

O método dos gradientes conjugados, por isto ,

é um método iterativo finito.

9.3. Programas computacionais dos métodos de resolução de sistemas lineares.

```

SUBROUTINE PIVOTA (N,A,X,INDICA)
INTEGER LINHA(50), COLUNA(50), LIN, COL, LINI, COLI, INDICA
REAL A(50,51), X(50), PIVO, AUX

```

```

C
C
C
C
C ==> ESTA SUBROTINA RESOLVE SISTEMAS LINEARES PELO METODO DA
C PIVOTACAO MAXIMA.
C SE DURANTE OS CALCULOS FOR ENCONTRADO UM PIVO MENOR QUE 10**-15
C A MATRIZ PASSA A SER CONSIDERADA SINGULAR.
C
C
C -----
C ---->>DESCRICAO DAS VARIAVEIS PRINCIPAIS
C
C
C ==> N      ; NUMERO DE INCOGNITAS (EQUACOES) DO SISTEMA
C ==> A      ; MATRIZ AMPLIADA DO SISTEMA
C ==> X      ; VETOR CONTENDO A SOLUCAODO SISTEMA QUANDO ESTA EXISTE
C ==> INDICA ; ANORMALIDADE NA SOLUCAO;
C              INDICA = 0  ----> SOLUCAO ENCONTRADA
C              INDICA = 1  ----> A MATRIZ FOI CONSIDERADA SINGULAR
C
C -----
C ==> LINHA  ; VETOR CONTENDO OS INDICES DAS LINHAS DOS PIVOS
C ==> COLUNA ; VETOR CONTENDO OS INDICES DAS COLUNAS DOS PIVOS
C ==> PIVO   ; VALOR CORRENTE DO PIVO
C
C
C
C
C
C
C
C
C
C

```

```

C
C
INDICA = 0
NM1 = N+1
C   ==> INICIA O PROCEDIMENTO DE ELIMINACAO
DO 30 K = 1,N
    KM1 = K - 1
C   ==> PROCURA O ELEMENTO A SER TOMADO COMO PIVO
    PIVO = 0.0
    DO 10 I = 1,N
    DO 10 J = 1,N
        IF( K .EQ. 1 ) GO TO 9
        DO 8 IPESQ = 1,KM1
        DO 8 JPESQ = 1,KM1
C   ==> PROCURA NOS VETORES "LINHA" E "COLUNA"
C   OS SUBSCRITOS DE PIVO INVÁLIDOS
        IF( I .EQ. LINHA(IPESQ) ) GO TO 10
        IF( J .EQ. COLUNA(JPESQ) ) GO TO 10
    8   CONTINUE
    9   IF( ABS( A(I,J) ) .LE. ABS( PIVO ) ) GO TO 10
        PIVO = A(I,J)
        LINHA(K) = I
        COLUNA(K) = J
    10  CONTINUE
C   ==> VERIFICA SE O ELEMENTO PIVO E' MAIOR QUE 10**-15
        IF( ABS( PIVO ) .LE. 1.0E-15 ) GO TO 90
        LIN = LINHA(K)
        COL = COLUNA(K)
C   ==> NORMALIZA OS ELEMENTOS NA LINHA DO PIVO
        DO 12 J = 1,NM1
            A(LIN,J) = A(LIN,J)/PIVO
    12  CONTINUE
C   ==> EXECUTA A FASE DE ELIMINACAO
        A(LIN,COL) = 1.0/PIVO
        DO 20 I = 1,N
            AUX = A(I,COL)
            IF( I .EQ. LIN ) GO TO 20
            A(I,COL) = - AUX/PIVO
            DO 15 J = 1,NM1
                IF( J .NE. COL ) A(I,J) = A(I,J) - AUX*A(LIN,J)
    15  CONTINUE
    20  CONTINUE
    30  CONTINUE
C   ==> ORDENA A SOLUCAO (SE EXISTIR)
        DO 35 I = 1,N
            LINI = LINHA(I)
            COLI = COLUNA(I)
            X(COLI) = A(LINI,NM1)
    35  CONTINUE
        RETURN
    90  CONTINUE
        INDICA = 1
        RETURN
        END

```



```

SUBROUTINE CHOLES(N,A,X)
REAL A(50,51),U(50,51),L(50,51),Y(50),X(50),SOMA
EQUIVALENCE (A,U,L), (Y,X)
C
C
C      ==> ESTA SUBROTINA RESOLVE SISTEMAS LINEARES PELO METODO DE
C      CHOLESKI
C
C----->>> DESCRICAO DAS VARIAVEIS PRINCIPAIS
C
C      ==>      N  : NUMERO DE INCOGNITAS (EQUACOES) DO SISTEMA
C      ==>      A  : MATRIZ AMPLIADA DO SISTEMA
C      ==>      X  : VETOR CONTENDO A SOLUCAO DO SISTEMA
C      ==>      Y  : VETOR AUXILIAR CONTENDO A SOLUCAO INTERMEDIARIA
C      ==>      U  : MATRIZ TRIANGULAR SUPERIOR
C      ==>      L  : MATRIZ TRIANGULAR INFERIOR
C
C
C              'A MATRIZ A FICA DECOMPOSTA NO PRODUTO A = L*U'
C              PARA ECONOMIA DE MEMORIA AS MATRIZES A,L E U
C              OCUPAM O MESMO ESPACO DE MEMORIA.
C
C
C      NM1 = N + 1
C      ==> DECOMPOE A EM L*U
C      J = 1
10  M = J + 1
    DO 15 I = M,N
      IF(J .GT. 1) GO TO 11
      L(I,J) = A(I,J)/U(J,J)
      GO TO 13
11  CONTINUE
    SOMA = 0.0
    K = 0
12  K = K + 1
    SOMA = SOMA + L(I,K)*U(K,J)
    IF(K .LT. (J-1) ) GO TO 12
    L(I,J) = ( A(I,J) - SOMA )/U(J,J)
13  CONTINUE
    SOMA = 0.0
    K = 0
14  K = K + 1
    SOMA = SOMA + L(M,K)*U(K,I)
    IF(K .LT. (M-1) ) GO TO 14
    U(M,I) = A(M,I) - SOMA
15  CONTINUE
    J = J + 1
    IF( J .LT. N) GO TO 10
C
C-----> RESOLVE O SISTEMA L*Y = B
C
C      Y(1) = A(1,NM1)
C      DO 17 I = 2,N
C        SOMA = 0.0
C        DO 16 J = 1,I-1
C          SOMA = SOMA + L(I,J)*Y(J)
16  CONTINUE
    Y(I) = A(I,NM1) - SOMA
17  CONTINUE
C
C

```

```
C
C
C-----> RESOLVE O SISTEMA  U*X= Y
C
      X(N) = Y(N)/U(N,N)
      DO 19 I = N-1,1,-1
        SOMA = 0.0
        DO 18 J = N,I+1,-1
          SOMA = SOMA + U(I,J)*X(J)
18      CONTINUE
        X(I) = (Y(I) - SOMA)/U(I,I)
19 CONTINUE
      RETURN
      END
```

```

SUBROUTINE SEIDEL (N,A,X,INDICA,ITER)
REAL A(50,51), X(50), AUX, EPS
INTEGER CONVER
DATA EPS/1.0 E-7/, NUMITE/100/

```

```

==> ESTA SUBROTINA RESOLVE SISTEMAS LINEARES PELO METODO DE
    GAUUS-SEIDEL.
    ° A MATRIZ DO SISTEMA DEVE SER REORDENADA DE FORMA A SE
    OBTER A DOMINANCIA DIAGONAL °

```

-----> DESCRICAO DAS VARIAVEIS

```

==>      N      : NUMERO DE INCOGNITAS (EQUACOES) DO SISTEMA
==>      A      : MATRIZ AMPLIADA DO SISTEMA
==>      X      : VETOR SOLUCAO DO SISTEMA SE HOVER
==>      INDICA  : ANORMALIDADE NA SOLUCAO
            INDICA = 0      ----> SOLUCAO ENCONTRADA
            INDICA = 1      ----> NAO HOVE CONVERGENCIA
            INDICA = 2      ----> ELEMENTO NA DIAGONAL IGUAL A ZERO

==>      NUMITE  : NUMERO MAXIMO DE ITERACOES
==>      EPS     : CRITERIO DE CONVERGENCIA
==>      ITER    : CONTADOR DE ITERACOES REALIZADAS

```

```

INDICA = 0
NM1 = N+1
==> DETERMINA UM VALOR INICIAL PARA X
    SE O ELEMENTO DA DIAGONAL FOR ZERO, RETORNA COM ESTA INDICACAO
DO 8 I = 1, N
    IF( A(I,I) .NE. 0.0 ) GO TO 7
    INDICA = 2
    RETURN
7  CONTINUE
    X(I) = A (I,NM1) /A(I,I)
8  CONTINUE

```

```

==> NORMALIZA OS ELEMENTOS DE CADA LINHA
DO 9 I=1,N
  AUX=A(I,I)
  DO 9 J=1,NM1
    A(I,J)=A(I,J)/AUX
2 CONTINUE
==> TENTAR ENCONTRAR A SOLUCAO DENTRO DO NUMERO DE ITERACOES
PERMITIDO
DO 20 ITER= 1,NUMITE
  CONVER=1
  DO 13 I=1,N
    XAUX = X(I)
    X(I) = A(I,NM1)
    DO 10 J= 1,N
      IF(I.EQ.J) GO TO 10
      X(I) = X(I) - A(I,J)*X(J)
0 CONTINUE
==> VERIFICA SE O CRITERIO DE CONVERGENCIA FOI ALCANCADO
    CASO POSITIVO TERMINA O PROCESSAMENTO
    IF(ABS(XAUX - X(I)) .LE. EPS) GO TO 13
    CONVER = 0
3 CONTINUE
==> SE HOUVE A CONVERGENCIA RETORNA COM A SOLUCAO
    SENAO TENTA NOVAMENTE
IF(CONVER .NE. 1) GO TO 20
  RETURN
0 CONTINUE
==> SE NAO CONSEGUIU A CONVERGENCIA DENTRO DO NUMERO DE ITERACOES
    PERMITIDO,RETORNA COM ESTA INDICACAO.

INDICA = 1
ITER = NUMITE
RETURN
END

```

```

SUBROUTINE JACOBI(N,A,X,INDICA,ITER)
REAL A(50,51),XAUX(50),X(50)
REAL PREC,SOMA,MAIOR1,MAIOR2,AUX1,AUX2
DATA NUMITE/100/,PREC/1.0E-7/
==> ESTA SUBROTINA RESOLVE SISTEMAS LINEARES PELO METODO DE JACOBI
    A MATRIZ DO SISTEMA DEVE SER REORDENADA DE FORMA A OBTER-SE A
    DOMINANCIA DIAGONAL.

```

-----> DESCRICAO DAS VARIAVEIS

```

==> N      : NUMERO DE INCOGNITAS (EQUACOES) DO SISTEMA
==> A      : MATRIZ AMPLIADA DO SISTEMA
==> X      : VETOR SOLUCAO DO SISTEMA (SE EXISTIR)
==> INDICA : ANORMALIDADE NA SOLUCAO
        INDICA = 0    ----> SOLUCAO ENCONTRADA
        INDICA = 1    ----> NAO HOUE CONVERGENCIA
        INDICA = 2    ----> ELEMENTO NA DIAGONAL IGUAL A ZERO

==> PREC   : CRITERIO DE CONVERGENCIA
            (NORMA DO VETOR DIFERENCA/NORMA DO VETOR SOLUCAO)

==> ITER   : CONTADOR DO NUMERO DE ITERACOES REALIZADAS
==> NUMITE : NUMERO DE ITERACOES PERMITIDAS

```

```
INDICA = 0
```

```
MM1 = N+1
```

```
==> CALCULA UM VALOR INICIAL PARA O VETOR X
    ZERA O VETOR XAUX QUE CONTERA' OS NOVOS VALORES DO VETOR X
```

```
DO 8 I = 1,N
```

```
IF( A(I,I) .NE. 0.0 ) GO TO 6
```

```
INDICA = 2
```

```
RETURN
```

```
6 CONTINUE
```

```
X(I) = A(I,MM1)/A(I,I)
```

```
XAUX(I) = 0.0
```

```
8 CONTINUE
```

```

C
C
C
C
C
C      ==> EXECUTA AS ITERACOES ATE' O MAXIMO IGUAL A NUMITE
DO 70 ITER = 1,NUMITE
C      ==> CALCULA OS VALORES PARA O VETOR XAUX
      DO 20 I = 1,N
      SOMA = 0.0
      DO 10 J = 1,N
      IF( I .NE. J ) SOMA = SOMA + A(I,J)*X(J)
10     CONTINUE
      XAUX(I) = ( A(I,NM1) - SOMA )/A(I,I)
20     CONTINUE
C      ==> VERIFICA SE O CRITERIO DE CONVERGENCIA FOI ALCANÇADO
      MAIOR1 = -1.5D38
      MAIOR2 = -1.5D38
      DO 30 I = 1,N
      AUX1 = DABS( XAUX(I) - X(I) )
      IF( AUX1 .GE. MAIOR1 ) MAIOR1 = AUX1
      AUX2 = DABS( X(I) )
      IF( AUX2 .GE. MAIOR2 ) MAIOR2 = AUX2
C      ==> ATUALIZA O VETOR SOLUCAO
      X(I) = XAUX(I)
30     CONTINUE
      IF( AUX1/AUX2 .GT. PREC ) GO TO 70
C      ==> SE CONSEGUIU CONVERGENCIA DENTRO DO NUMERO PERMITIDO DE
C      ITERACOES RETORNA COM A SOLUCAO
      RETURN
70 CONTINUE
C      ==> SE NAO CONSEGUIU CONVERGENCIA APOS O NUMERO MAXIMO DE
C      ITERACOES RETORNA COM ESTA INDICACAO
      INDICA = 1
      ITER = NUMITE
      RETURN
      END

```

```

C
C
C
C
C
SUBROUTINE GRADE(N,A,X,INDICA)
REAL  A(50,51),B(50),X(50),V(50),R(50),ALFA,BETA,MAIORX,MAIORD
REAL  VITA(50) , XAUX(50),VIR,VIAVI,AUX,AUX1,AUX3,PREC
DATA NUMITE/100/, PREC/1.0E-5/
C
C
C
C----->  DESCRICAO DAS VARIÁVEIS PRINCIPAIS
C
C
C
C ==>      N      ; NUMERO DE INCOGNITAS (EQUACOES) DO SISTEMA
C ==>      A      ; MATRIZ AMPLIADA DO SISTEMA
C ==>      X      ; VETOR SOLUCAO DO SISTEMA
C ==>      INDICA ; ANORMALIDADE NA SOLUCAO
C              INDICA = 0  ---> SOLUCAO ENCONTRADA
C              INDICA = 1  ---> NAO HOUE CONVERGENCIA
C
C ==>      NUMITE ; NUMERO MAXIMO DE ITERACOES PERMITIDO
C ==>      PREC   ; CRITERIO DE CONVERGENCIA
C              (NORMA DO VETOR DIFERENCA/NORMA DO VETOR SOLUCAO)
C
C ==>      V,R,B, ;
C              VITA ; VETORES AUXILIARES
C ==>      XAUX  ; VETOR AUXILIAR PARA O CALCULO DA SOLUCAO
C
C
C=====>>>>      PARA MAIORES ESCLARECIMENTOS VEJA
C                   A FIRST COURSE IN NUMERICAL ANALYSIS
C                   ANTHONY RALSTON
C                   1965 - IES - MACGRAW-HILL
C
C
C
INDICA = 0
NM1 = N+1
C ==> INICIALIZA OS VETORES B,V, R E VITA
DO 10 I = 1,N
  B(I) = A(I,NM1)
  V(I) = B(I)
  R(I) = B(I)
  VITA(I) = 0.0
) CONTINUE

```

```

C ==> TENTA ENCONTRAR A SOLUCAO DENTRO DO NUMERO MAXIMO DE ITERACOES
C PERMITIDO
DO 80 ITER = 1,NUMITE
AUX = 0.0
VIR = 0.0
VIAVI = 0.0
C ==> CALCULA O PRODUTO DO VETOR V TRANSPOSTO PELA MATRIZ A
C O RESULTADO E' O VETOR VITA
C A SEGUIR CALCULA OS PRODUTOS ESCALARES V POR R
C E VITA POR V
DO 30 I = 1,N
VIR = VIR + V(I)*R(I)
XAUX(I) = X(I)
DO 20 J = 1,N
VITA(I) = VITA(I) + V(J)*A(J,I)
20 CONTINUE
VIAVI = VIAVI + VITA(I)*V(I)
30 CONTINUE
C ==> CALCULA AS CORRECOES ALFA E BETA
ALFA = VIR/VIAVI
MAIORD = -1.7D38
MAIORX = -1.7D38
DO 50 I = 1,N
X(I) = X(I) + ALFA*V(I)
AUX = ABS( XAUX(I) - X(I) )
IF( AUX .GE. MAIORD ) MAIORD = AUX
AUX = ABS( X(I) )
IF( AUX .GE. MAIORX ) MAIORX = AUX
AUX1 = 0.0
DO 40 J = 1,N
AUX1 = AUX1 + A(I,J)*V(J)
40 CONTINUE
R(I) = R(I) - ALFA*AUX1
50 CONTINUE
AUX3 = 0.0
DO 60 I = 1,N
VITA(I) = 0.0
DO 55 J = 1,N
VITA(I) = VITA(I) + V(J)*A(J,I)
55 CONTINUE
AUX3 = AUX3 + VITA(I)*R(I)
60 CONTINUE
BETA = -AUX3/VIAVI
DO 70 I = 1,N
V(I) = R(I) + BETA*V(I)
70 CONTINUE
C VERIFICA SE O CRITERIO DE CONVERGENCIA FOI ALCANCADO
C IF( MAIORD/MAIORX .GE. PREC ) GO TO 80
C ==> SE HOUVE CONVERGENCIA RETORNA COM A SOLUCAO
RETURN
80 CONTINUE
C ==> SE A CONVERGENCIA NAO FOI ALCANCADA COM O NUMERO
C MAXIMO DE ITERACOES, RETORNA COM ESTA INDICACAO
ITER = NUMITE
INDICA = 1
RETURN
END

```