

UM ESTUDO SOBRE UM SISTEMA COMPUTACIONAL DE  
GRANDE PORTE EM PROGRAMAÇÃO LINEAR

Dalva Maria de Oliveira Villarreal

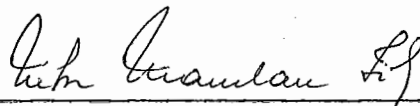
TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE  
PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE  
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO  
DO GRAU DE MESTRE EM CIÊNCIAS (M.Sc.) EM ENGENHARIA DE SISTEMAS  
E COMPUTAÇÃO.

Aprovada por:



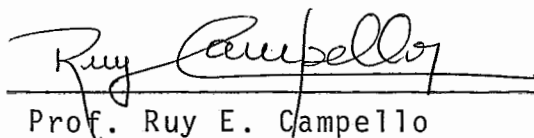
---

Prof. Claudio Thomás Bornstein  
(Presidente)



---

Prof. Nelson Maculan Filho



---

Prof. Ruy E. Campello

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 1984

VILLARREAL, DALVA MARIA DE OLIVEIRA

Um Estudo sobre um Sistema Computacional de Grande Porte em Programação Linear (Rio de Janeiro) 1984.

VII, 131 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas, 1984)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Programação Linear de Grande Porte

I. COPPE/UFRJ II. Um Estudo sobre um Sistema Computacional de Grande Porte em Programação Linear

Ao meu filho

Roque Oliveira Villarreal

A G R A D E C I M E N T O S

Agradeço aos prof. Claudio Thomás Bornstein pela orientação dedicada.

Agradeço a UNESP pela oportunidade que me deu de executar este trabalho.

Agradeço a CAPES pela ajuda financeira.

Agradeço a meus pais José e Dalva e ao meu esposo Francisco pelo apoio e incentivo dados durante a execução deste trabalho.

Agradeço também a todos que direta ou indiretamente colaboraram para a conclusão deste trabalho.

Resumo da Tese Apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UM ESTUDO SOBRE UM SISTEMA COMPUTACIONAL DE  
GRANDE PORTE EM PROGRAMAÇÃO LINEAR

Dalva Maria de Oliveira Villarreal

Março de 1984

Orientador: Claudio Thomás Bornstein

Programa: Engenharia de Sistemas e Computação

Neste trabalho foi desenvolvido um estudo detalhado de um sistema computacional de grande porte em Programação Linear, sendo feita a descrição de cada técnica utilizada, bem como a descrição e implementação da opção LU do método simplex, baseada no método de Forrest-Tomlin, no sistema computacional objeto de nosso estudo.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

A STUDY ABOUT A LARGE SCALE COMPUTATIONAL  
LINEAR PROGRAMMING SYSTEM

Dalva Maria de Oliveira Villarreal

March, 1984

Chairman: Claudio Thomás Borstein

Department: Engenharia de Sistemas e Computação

We report on a detailed study of a large size computational Linear Programming system. A description of each employed technique as well as the implementation of the simplex LU; i.e. the decomposition of the inverse of the Basis into elementary LU matrices, has been made. This implementation is based on the method developed by Forrest and Tomlin being the essential enhancement introduced in the computational system under examine.

I N D I C E

	<u>Páginas</u>
CAPÍTULO I - INTRODUÇÃO	1
CAPÍTULO II - O MÉTODO SIMPLEX REVISADO COM TRATAMENTO DE VARIÁVEIS LIMITADAS	4
2.1. Introdução . . . . .	4
2.2. O Método Simplex Revisado . . . . .	5
2.3. O Simplex Revisado com Tratamento de Variáveis limitadas . . . . .	7
CAPÍTULO III - A PROBLEMÁTICA DA MATRIZ INVERSA NO MÉTODO SIMPLEX	
3.1. Introdução . . . . .	14
3.2. O Problema da Atualização da Inversa . . . . .	14
3.3. A Forma do Produto da Inversa do Simplex Revisado . . . . .	19
3.4. A Reinversão da Matriz Básica . . . . .	24
3.4.1. O Método de Reinversão de Larsen . . . . .	47
CAPÍTULO IV - O SIMPLEX LU	54
4.1. Introdução . . . . .	54
4.1.1. Algumas Propriedades das Matrizes Triangulares . . . . .	57
4.2. A Forma de Eliminação da Inversa . . . . .	59
4.3. O Esquema de Atualização da Inversa de Forrest-Tomlin - Descrição Matemática . . . . .	65
4.4. O simplex LU . . . . .	87
4.5. Aspectos Computacionais . . . . .	97

	<u>Páginas</u>
CAPÍTULO V - ESTRUTURA DO ARMAZENAMENTO DA MATRIZ DOS COEFICIENTES E DAS MATRIZES ELEMENTARES	99
5.1. Introdução . . . . .	99
5.2. Estrutura de Dados do Sistema de FURNAS . . .	100
5.2.1. Matriz dos Coeficientes Tecnológicos (A) . . . . .	100
5.2.2. Matrizes Elementares . . . . .	105
CAPÍTULO VI - UM ESTUDO SOBRE PRICING MÚLTIPLO	109
6.1. Introdução . . . . .	109
6.2. O "Pricing" Múltiplo . . . . .	110
CAPÍTULO VII - UM ESTUDO SOBRE "SCALING" EM PROGRAMA- ÇÃO LINEAR . . . . .	115
7.1. Introdução . . . . .	115
7.2. O que é "scaling" em Programação Linear? . .	115
7.3. Razões para se fazer "scaling" . . . . .	117
7.3.1. Observação . . . . .	121
7.4. Alguns métodos de "scaling" em Programação Linear . . . . .	122
7.5. O "scaling" no software de FURNAS . . . . .	124
CAPÍTULO VIII - CONCLUSÃO	125
BIBLIOGRAFIA	128



## CAPÍTULO I

### INTRODUÇÃO

O objetivo principal deste trabalho foi estudar um sistema de grande porte em Programação Linear. Tal estudo se desenvolveu em duas fases:

1.<sup>a</sup> fase - Estudo do software propriamente dito, onde foi feita a descrição de cada técnica utilizada, ressaltando os motivos do uso da técnica, e a existência de outras técnicas semelhantes.

2.<sup>a</sup> fase - Implementação de uma técnica não existente no software.

O sistema objeto de nosso estudo foi desenvolvido por Ruy E. Campello e Maurício G.C. Resende [6], ambos engenheiros de FURNAS - Centrais Elétricas S.A.

O software de FURNAS possuía as seguintes características quando iniciamos o nosso estudo:

- Simplex Revisado na Forma do Produto da Inversa.
- Tratamento implícito de variáveis Limitadas.
- Esquema de Pricing múltiplo.
- Armazenamento de matrizes esparsas em estrutura de listas encadeadas.
- Recuperação da inversa da matriz básica para redução das filas e controle de erro.
- Scaling opcional do programa linear.

Atualmente além das características citadas acima, o software possui um esquema de atualização da matriz básica na forma LU, esquema este implementado por nós. Dessa forma, todos os tópicos desenvolvidos nesse texto constituem componentes do sistema de FURNAS.

No Capítulo II foram abordados os seguintes tópicos de Programação Linear: o simplex revisado e o simplex revisado com tratamento de variáveis limitadas. O objetivo nesse capítulo foi dar de forma resumida o funcionamento destes algoritmos, não nos preocupando a descrição matemática dos mesmos.

No Capítulo III tratamos inicialmente do problema da atualização da inversa, descrevendo duas formas de encarar o problema. Descrevemos também neste capítulo, a Forma do Produto da Inversa do simplex Revisado; tratamos do problema de reinversão da matriz básica, para redução das filas e controle de erro, e finalmente descrevemos o método de reinversão de Larsen <sup>[18]</sup>, método este utilizado no software de FURNAS.

No Capítulo IV descrevemos a forma de atualização da inversa de Forrest-Tomlin <sup>[10]</sup>, também conhecida como o método pseudo-LU de atualização do simplex. Tal esquema de atualização deu lugar a uma nova versão do algoritmo simplex, denominada por nós de simplex LU. Esta versão do simplex não existia anteriormente, no sistema de FURNAS, passando agora, a ser um componente do referido software. Ainda neste capítulo, descrevemos o método de inversão, conhecido como a forma de Eliminação da inversa <sup>[22]</sup>. Este método fornece a matriz inversa na forma de um produto de fatores triangulares e elementares,

ma esta necessária para o desenvolvimento do simplex LU. Trata<sub>u</sub>mos também neste capítulo, de alguns aspectos computacionais re<sub>u</sub>ferentes à implementação do simplex LU.

No Capítulo V descrevemos a estrutura de dados do sistema de FURNAS; tentamos a<sub>i</sub>, tornar mais simples os concei<sub>u</sub>tos computacionais de forma a deixar o texto mais acessível. Des<sub>u</sub>crevemos o armazenamento da matriz dos coeficientes tecnológicos e o armazenamento das matrizes elementares.

No Capítulo VI descrevemos o "pricing" múltiplo, técnica esta, que está implementada no software de FURNAS.

No capítulo VII fizemos um estudo sobre a técnica de "scaling", descrevendo o "scaling" utilizado no sistema de FURNAS.

Esperamos que este texto possa ser útil para aqueles que queiram estudar Programação Linear de Grande Porte, com vistas a uma implementação de um sistema deste tipo.

CAPÍTULO II  
O MÉTODO SIMPLEX REVISADO COM TRATAMENTO  
DE VARIÁVEIS LIMITADAS

2.1. Introdução

Seja o problema de Programação Linear:

$$\text{minimizar } z = c^t \cdot x \quad (2.1)$$

$$\text{sujeito a } Ax \begin{matrix} > \\ = \\ < \end{matrix} b \quad (2.2) \quad (I)$$

$$l \leq x \leq u \quad (2.3)$$

onde  $A$  é uma matriz  $(m \times n)$ ;  $x$ ,  $u$ ,  $c$  são vetores  $n$  dimensionais e  $b \in R^m$ .

Quando adotamos o método simplex tradicional na resolução de problemas do tipo acima surgem as seguintes desvantagens:

- (a) Ocorre um aumento de restrições provocado pelas desigualdades em (2.3);
- (b) Conseqüentemente teremos uma grande ocupação de memória e um gasto de tempo computacional enorme em cada iteração.

Mesmo a aplicação do simplex revisado ainda não solucionaria o problema do aumento de restrições. Tal problema é contornado se é dado um tratamento especial às variáveis limitadas no simplex revisado.

Neste tópico vamos descrever o método simplex revisado com tratamento de variáveis limitadas.

Estamos supondo o conhecimento do método simplex revisado sobre o qual faremos apenas um breve resumo.

Nas referências  $[01]$ ,  $[20]$  e  $[06]$  podemos encontrar detalhadamente o método simplex revisado, bem como o tratamento de variáveis limitadas no método simplex. Adotamos aqui a notação usada em  $[04]$ .

## 2.2. O Método Simplex Revisado

O método simplex revisado pode ser resumido nos seguintes passos, assumindo antes que se tenha:

Uma solução básica viável -  $x_B$

Uma representação da matriz inversa da base -  $B^{-1}$   
e o termo independente  $\bar{b} = B^{-1}b$

### Passo 1: Cálculo do vetor dual $\pi$

$$\pi = c_B^t \cdot B^{-1}$$

Observação: Esta operação, na literatura, é dita BTRAN (Backwards TRANSformation) por ser uma pós-multiplicação pela matriz  $B^{-1}$ .

### Passo 2: Cálculo dos custos reduzidos

Primeiramente calcula-se:

$$z_j = \pi^t \cdot a_j \quad , \quad (\text{para todo } j \text{ tal que } x_j \text{ seja variável não básica})$$

Depois calcula-se:

$$(c_j - z_j) \quad , \quad (\text{para todo } j \text{ tal que } x_j \text{ seja variável não básica}).$$

Passo 3: Escolha da variável não básica para entrar na base

A variável não básica com custo reduzido mais negativo será a escolhida.

Se não existe variável não básica com custo reduzido negativo então a solução é ótima; Pare. Caso contrário, seja  $j = s$  o índice da variável escolhida para entrar na base.

Passo 4: Atualização da coluna que entra na matriz básica

$$y_s = B^{-1} \cdot a_s$$

Observação: esta operação é dita FTRAN (Forward TRANSformation); é caracterizada pela pré-multiplicação por  $B^{-1}$ .

Passo 5: Escolha da variável que deixa a base

Seja:

$$\frac{\bar{b}_r}{y_{rs}} = \min_{i/y_{is} > 0} \left\{ \frac{\bar{b}_i}{y_{is}} \right\}$$

Se não existe  $y_{is} > 0$  então pare pois o problema é ilimitado; caso contrário  $x_r$  será a variável escolhida para deixar a base.

Passo 6: Pivoteamento

Cálculo da nova  $B^{-1}$  e atualização de  $\bar{b} = B^{-1} \cdot b$ .

A atualização da base  $B$  é feita com a saída do vetor  $a_r$  e a entrada do vetor  $a_s$ .

Voltar ao Passo 1.

### 2.3. O Simplex Revisado com Tratamento de Variáveis Limitadas

A característica do simplex revisado com tratamento de variáveis limitadas é que as variáveis não básicas ao invés de assumirem necessariamente o valor nulo, assume um dos seus limites, ou o inferior ou o superior. Isso ocorre tanto na inicialização (obtenção da solução básica inicial), quanto no decorrer do algoritmo. Portanto, é necessário ser registrado quais as variáveis não básicas que assumem seu limite inferior e quais assumem seu limite superior.

Com o propósito de tornar mais claro o algoritmo damos algumas definições iniciais.

Seja  $N$  o conjunto dos índices das variáveis não básicas:

$$N = \{j/x_j \text{ seja variável não básica}\}$$

$L$  é o conjunto dos índices das variáveis não básicas que assumem seu limite inferior:

$$L = \{j/x_j = \ell_j \text{ e } x_j \text{ é variável não básica}\}$$

Analogamente definimos  $K$ :

$$K = \{j/x_j = u_j \text{ e } x_j \text{ é variável não básica}\}$$

O sistema (2.2) colocado na forma padrão pode ser escrito como:

$$Bx_B + Rx_R + Qx_Q = b \quad (2.4)$$

$$x_B = B^{-1}b - B^{-1}Rx_R - B^{-1}Qx_Q \quad (2.5)$$

Onde  $B$  - é a matriz básica,

$R$  - é a matriz dos coeficientes das variáveis não básicas que assumem seu limite inferior.

$Q$  - é a matriz associada às variáveis não básicas que assumem seu limite superior

$x_B, x_R, x_Q$  são respectivamente as variáveis básicas, as variáveis não básicas que assumem seu limite inferior e as variáveis não básicas que assumem seu limite superior.

Substituindo os valores dos limites de  $x_R$  e  $x_Q$  ( $x_R = \ell_R$  e  $x_Q = u_Q$ ) em (2.5) temos:

$$x_B = B^{-1}b - B^{-1}R.\ell_R - B^{-1}Qu_Q \quad (2.6)$$

Por outro lado (2.1) pode ser escrita como:

$$z = c_B x_B + c_R x_R + c_Q x_Q \quad (2.7)$$

de onde teremos, substituindo (2.6) em (2.7):

$$z = c_B B^{-1}b + (c_R - c_B B^{-1}R)\ell_R + (c_Q - c_B B^{-1}Q)u_Q \quad (2.8)$$

O conceito de viabilidade de uma solução básica não deve deixar de levar em conta o tratamento de variáveis limitadas.

Definição: Uma solução básica  $x_B$  é dita viável se satisfaz (2.4) e cada componente de  $x_B$  está dentro do seu intervalo de variação, isto é:

$$\ell_B \leq x_B \leq u_B \quad (2.9)$$

Feitas tais considerações podemos passar ao algoritmo propriamente dito.



Passo 0: Inicialização

(a) Cada variável original assume seu limite inferior;

(b) Ajusta-se o termo independente convenientemente

$$\bar{b} = b - R \cdot \ell_R$$

(c) Introdução das variáveis de folga e artificiais para obtenção da forma padrão.

Observação: Tais operações são para obtenção de uma solução básica inicial viável; o algoritmo do m grande é aplicado até que não existam variáveis artificiais na base.

Tendo  $B^{-1}$ ,  $x_B$  e  $\bar{b}$

Passo 1: Cálculo do vetor dual  $\pi$ 

$$\pi = c_B^t \cdot B^{-1}$$

Passo 2: Cálculo dos custos reduzidos

Calcula-se

$$z_j = \pi^t \cdot a_j, \quad j \in N$$

e

$$(c_j - z_j), \quad j \in N$$

Passo 3: Escolha da variável não básica para entrar na base

Seja:

$$d_j = \begin{cases} (c_j - z_j) & j \in L \\ (z_j - c_j) & j \in K \end{cases}$$

e

$$d_s = \min_{j \in N} \{d_j\}$$

Se  $d_s \geq 0$  a solução é ótima, pare; caso contrário  $x_s$

entrará na base.

Observação: (2.8) pode ser escrito como:

$$Z = c_B B^{-1} b + \sum_{j \in L} (c_j - z_j) x_j + \sum_{j \in K} (c_j - z_j) x_j$$

Notemos que para as variáveis não básicas no seu limite inferior se  $(c_j - z_j) < 0$ ,  $z$  diminuirá se aumentarmos  $x_j$ . Por outro lado para as variáveis no seu limite superior se  $(c_j - z_j) > 0$  (isto é  $(z_j - c_j) < 0$ )  $z$  diminuirá se diminuirmos  $x_j$ . Escolho para entrar na base  $x_s$  que provoque a maior variação (diminuição) em  $z$ .

Passo 4: Atualização da coluna que entra na matriz básica

Calcular:

$$y_s = B^{-1} \cdot a_s$$

Passo 5: Escolha da variável que sai da base

Existem dois casos a considerar:

caso (1):  $x_s = l_s$  e sofre um acréscimo  $\delta$  e passa a ser

$$x_s = l_s + \delta \quad (\delta > 0)$$

caso (2):  $x_s = u_s$  e sofre um decréscimo  $\delta$  e passa a ser

$$x_s = u_s - \delta \quad (\delta > 0)$$

Nota: Desenvolveremos o caso (1) com detalhe; o caso (2) deve ser desenvolvido de forma semelhante e não o faremos nesse texto.

A variação  $\delta$  de  $x_s$  deve ser tal que a viabilidade da nova solução não seja afetada. Vamos analisar o que poderá acontecer no caso (1), isto é, se aumentarmos o valor de  $x_s$  do seu limite inferior  $l_s$ .

Como aumento de  $x_s$  podem ocorrer um dos três casos:

- (a) Uma variável básica atinge seu limite inferior antes que qualquer outra variável básica atinja um dos seus extremos e antes que  $x_s$  atinja seu extremo superior.
- (b) Uma variável básica atinge seu limite superior antes que qualquer outra variável básica atinja um dos seus extremos, e antes de  $x_s$  atingir seu limite superior.
- (c) A variável  $x_s$  atinge seu limite superior antes que qualquer variável básica atinja um dos seus extremos. Seja o novo valor de  $x_s = \ell_s + \delta$ .

A modificação provocada em  $x_B$  será:

$$x_B = \bar{b} - \delta \cdot y_s \quad (2.10)$$

Pela viabilidade (2.9) que tem de ser mantida temos:

$$\ell_B \leq \bar{b} - \delta \cdot y_s \leq u_B \quad (2.11)$$

Tomando (2.11) considerando o caso (a) e chamando de  $\gamma_1$  o valor de  $\delta$ , para este caso temos:

$$\delta \cdot y_s \leq \bar{b} - \ell_B$$

(a1) para  $y_s \leq 0$ ,  $\delta$  pode crescer infinitamente que a viabilidade não será transgredida, isto é, nenhuma variável básica atinge seu limite inferior. Fazemos  $\gamma_1 = \infty$

(a2) para  $y_s \neq 0$  temos:

$$\gamma_1 = \min_{i/y_{is} > 0} \left\{ \frac{\bar{b}_i - \ell_{B_i}}{y_{is}} \right\} = \frac{\bar{b}_r - \ell_{B_r}}{y_{rs}}$$

Tomando (2.11) considerando o caso (b) e chamando de  $\gamma_2$  o

valor de  $\delta$ , para este caso temos:

$$-\delta \cdot y_s \leq u_B - \bar{b}$$

(b1) para  $y_s \geq 0$ ,  $\delta$  poderá crescer tanto quanto se queira sem afetar viabilidade, isto é, nenhuma variável básica atinge seu limite superior. Faremos  $\gamma_2 = \infty$

(b2) para  $y_s \neq 0$  seja:

$$\gamma_2 = \min_{i/y_{is} < 0} \left\{ \frac{u_{B_i} - \bar{b}_i}{-y_{is}} \right\} = \frac{u_{B_r} - \bar{b}_r}{-y_{rs}}$$

Tomando agora o caso (c) e chamando de  $\gamma_3$  o valor de  $\delta$  para este caso temos que:

$$\gamma_3 = u_s - \ell_s$$

Fazendo:

$$\delta = \min\{\gamma_1, \gamma_2, \gamma_3\}$$

Se  $\delta = \infty$  temos que  $x_s$  é ilimitada; o problema de programação linear não tem solução. Pare.

Se  $\delta < \infty$  uma nova solução viável básica é dada por (2.10).

Nesse ponto temos dois casos a considerar:

(I) se  $\delta = \gamma_3$  não ocorrerá troca de variável de base, pois nenhuma variável básica atingirá um extremo seu para se tornar não básica - lembremos que as variáveis não básicas assumem um dos seus extremos em

qualquer solução básica viável - nesse caso ocorrerá apenas uma modificação no termo independente dada por  $\bar{b} \leftarrow b - \delta \cdot y_s$  onde  $\bar{b}$  é o novo valor de  $b$ .

(II) Se  $\delta = \gamma_1$  ou  $\delta = \gamma_2$  (e  $\delta < \infty$ ) ocorrerá uma troca de variáveis;  $x_r$  sairá da base assumindo seu limite inferior ou superior dependendo do valor de  $\delta$  ser  $\gamma_1$  ou  $\gamma_2$  respectivamente;  $x_s$  entrará na base.

Observação: Analogamente se considera o caso (2) quando  $x_s = u_s$ , e sofre um decréscimo  $\delta$ , e passa a

$$x_s = u_s - \delta \quad (\delta > 0)$$

#### Passo 6: Pivoteamento

Atualização de  $B^{-1}$  e de  $b$ , conforme os casos considerados em (I) e (II) do passo 5; voltar ao passo 1.

Observação: (a) As duas possíveis atualizações de  $b$  são:

$$\bar{b} = \begin{cases} b - \delta \cdot y_s & \text{se ocorrer o caso (1) do passo 5} \\ b + \delta \cdot y_s & \text{se ocorrer o caso (2) do passo 5} \end{cases}$$

(b) Os conjuntos  $L$  e  $K$  devem ser atualizados na passagem de uma iteração e outra.

CAPÍTULO III  
A PROBLEMÁTICA DA MATRIZ INVERSA  
NO MÉTODO SIMPLEX

### 3.1. Introdução

A maior parte dos cálculos no método simplex envolvem a matriz inversa da matriz básica, dessa forma é compreensível a preocupação com a obtenção e a manutenção da matriz inversa, tanto mais livre de erros, quanto possível, bem como o seu armazenamento de forma mais compacta.

São três os aspectos abordados nesse item e que envolvem a matriz inversa da base no simplex:

- (1) O problema da atualização da inversa;
- (2) A forma do produto da inversa do simplex revisado;
- (3) A reinversão da matriz básica.

O primeiro aspecto constitui uma peça da segunda e terceira abordagem, mas aqui resolvemos descrevê-lo separadamente para maior clareza do texto.

### 3.2. O problema de atualização da inversa

A atualização da inversa da matriz básica é um componente importante tanto no simplex revisado na sua forma do produto da inversa, como no algoritmo de reinversão da matriz básica, usada para contornar o problema dos erros acumulados em cada iteração do simplex.

O problema da atualização da inversa pode ser formulado da seguinte maneira:

Dada uma matriz  $B$  com inversa conhecida  $B^{-1}$ , deseja-se calcular a inversa da matriz  $\bar{B}$ , que é diferente de  $B$  em apenas uma coluna.

Expressando as matrizes por seus vetores colunas temos:

$$B = (a_1, a_2, \dots, a_p, \dots, a_m)$$

$$\bar{B} = (a_1, a_2, \dots, a_q, \dots, a_m)$$

O cálculo de  $\bar{B}^{-1}$  deve ser feito através de  $B^{-1}$ , que é dada.

A resolução deste problema pode ser feita da seguinte forma:

Como  $B$  é inversível os seus vetores colunas  $a_j (j=1, \dots, m)$  são linearmente independentes, portanto, formam uma base do espaço-coluna da matriz  $B$ . Seja  $y_q$  o vetor que expressa  $a_q$  em função dos vetores  $a_j$ ,  $j = 1, \dots, m$ ; isto é:

$$a_q = B \cdot y_q = \sum_{j=1}^m y_{jq} a_j \quad (3.1)$$

A condição necessária e suficiente para que  $\bar{B}^{-1}$  exista é que  $y_{pq} \neq 0$  |<sup>4</sup>|, pgs.36-37. Suponhamos  $y_{pq} \neq 0$ ; por (3.1) podemos escrever:

$$a_p = - \frac{1}{y_{pq}} \left( \sum_{\substack{j=1 \\ j \neq p}}^m y_{jp} a_j \right) + \frac{1}{y_{pq}} \cdot a_q$$

ou

$$a_p = \bar{B} \cdot \bar{y}_q$$

onde 
$$\bar{y}_{iq} = - \frac{y_{iq}}{y_{pq}} \quad , \quad i \neq p \quad , \quad i = 1, \dots, m$$

$$\bar{y}_{pq} = \frac{1}{y_{pq}}$$

Da definição de  $\bar{B}$  - ( $\bar{B}$  é diferente de  $B$  em apenas uma coluna, a coluna  $p$ ) - e de (3.2) temos:

$$B = \bar{B} \cdot (e_1, e_2, \dots, e_{p-1}, \bar{y}_q, e_{p+1}, \dots, e_m)$$

onde  $e_i$  é o  $i$ -ésimo vetor coluna da matriz identidade. Seja  $E_p = (e_1, e_2, \dots, e_{p-1}, \bar{y}_q, e_{p+1}, \dots, e_m)$ .  $E_p$  é dita matriz elemental coluna ou simplesmente matriz elemental e é caracterizada por ser igual à matriz identidade exceto quanto à  $p$ -ésima coluna. Dessa forma:

$$B = \bar{B} \cdot E_p \tag{3.3}$$

Multiplicando por  $\bar{B}^{-1}$  à esquerda e por  $B^{-1}$  à direita temos:

$$\bar{B}^{-1} = E_p \cdot B^{-1}$$

que é exatamente o que queríamos:  $\bar{B}^{-1}$  em função de  $B^{-1}$ . Outra maneira de encarar o mesmo problema é a que daremos a seguir, mas antes daremos algumas definições de operações com matrizes. Consideremos  $B$  uma matriz  $m \times m$ ;

$a_j$  - a  $j$ -ésima coluna de  $B$

$e_j$  - a  $j$ -ésima coluna da matriz identidade  $I$

(1) A  $j$ -ésima coluna de  $B$  pode ser escrita como:

$$a_j = B \cdot e_j$$



(2) O elemento  $a_{ij}$  de  $B$  pode ser escrito como:

$$a_{ij} = e_i^t \cdot B \cdot e_j$$

(3)  $B \cdot e_j \cdot e_j^t$  é uma matriz  $m \times m$  com todos os elementos nulos exceto os elementos da  $j$ -ésima coluna que são exatamente os elementos da  $j$ -ésima coluna de  $B$

$$B \cdot e_j \cdot e_j^t = \begin{bmatrix} \vdots & & \\ 0 & a_{1j} & \\ & \vdots & \\ & a_{mj} & 0 \end{bmatrix}$$

Observação: uma coluna  $a_k$  multiplicada pelo vetor  $e_j^t$  produz uma matriz com todos os elementos nulos exceto os elementos da  $j$ -ésima coluna que são iguais aos elementos da coluna  $a_k$ .

$$a_k \cdot e_j^t = \begin{bmatrix} \vdots & & \\ 0 & a_{1k} & \\ & \vdots & \\ & a_{mk} & 0 \end{bmatrix}$$

↑  
j-ésima coluna

$$(4) \quad I + B e_j \cdot e_j^t = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & \ddots & & & & \\ & & & 1+a_{jj} & & & \\ & & & \vdots & & & \\ & & & a_{mj} & & & \\ & & & & & & 1 \end{bmatrix}$$

(5) Seja o vetor  $a_q^t = (a_{1q}, a_{2q}, \dots, a_{mq})$

Eliminar de  $B$  o vetor  $a_p$  e inserir o vetor  $a_q$ , resultando uma matriz  $\bar{B}$ , é dado pela seguinte expressão:

$$\bar{B} = B - B \cdot e_p \cdot e_p^t + a_q \cdot e_p^t \quad (3.4)$$

Voltemos agora ao nosso problema que pode ser resu  
mido como o problema de obter  $\bar{B}^{-1}$  a partir de  $B^{-1}$ . No  
propósito de resolver o problema vamos aplicar os seguintes pas-  
sos:

(a) modificar  $B$  de forma a obter  $\bar{B}$ , trocando a  $p$ -ésima colu-  
na pela coluna  $a_q$

(b) conseguir  $\bar{B}^{-1}$  a partir de  $B^{-1}$ .

O passo (a) pode ser escrito segundo (3.4) pela seguinte expres-  
são:

$$\bar{B} = B \cdot \left[ I - e_p \cdot e_p^t + B^{-1} \cdot a_q \cdot e_p^t \right]$$

chamando de  $y_q = B^{-1} \cdot a_q$  temos:

$$\bar{B} = B \cdot \left[ e_1, \dots, e_{p-1}, y_q, e_{p+1}, \dots, e_m \right]$$

chamando

$$F = \left[ e_1, \dots, e_{p-1}, y_q, e_{p+1}, \dots, e_m \right]$$

temos:

$$\bar{B} = B \cdot F$$

Tomando a inversa de  $\bar{B}$  executamos o passo (b)

$$\bar{B}^{-1} = F^{-1} \cdot B^{-1}$$

$F^{-1}$  passaremos a chamar de  $E_p$  por ser uma matriz elementar  
coluna;  $F^{-1}$  existirá se e somente se  $y_{pq} \neq 0$  e por cálcu-  
los triviais verificamos que:

$$E_p = [e_1, \dots, e_{p-1}, \bar{y}_q, e_{p+1}, \dots, e_m]$$

onde:

$$\bar{y}_{iq} = - \frac{y_{iq}}{y_{pq}} \quad \text{para } i = 1, \dots, n ; i \neq p$$

e

$$\bar{y}_{pq} = \frac{1}{y_{pq}}$$

Portanto teremos:

$$\bar{B}^{-1} = E_p \cdot B^{-1}$$

### 3.3. A forma do produto da inversa do simplex revisado

A Forma do Produto da Inversa é uma variação do simplex revisado que tem como característica fundamental, a representação da matriz inversa da matriz básica por um produto de matrizes elementares. As vantagens, que o tornam um método computacionalmente atrativo, para ser usado nos sistemas de grande porte de Programação Linear são:

- (a) A matriz inversa da base não tem de estar na forma explícita; sua representação equivalente pode ser convenientemente armazenada sequencialmente em memória auxiliar na forma de filas etc.
- (b) Para problemas não densos, isto é, esparsos, a quantidade de cálculo necessário para que seja executada uma iteração do simplex usando a Forma do Produto da Inversa, pode ser menor que a quantidade necessária de cálculo usando a for-

ma do tableau explícito ou o método revisado com inversa explícita.

Inicialmente vamos considerar a maneira como se obtém uma representação da matriz inversa da base como um produto de matrizes elementares. Seja  $B_1 = I$  a base inicial; cada iteração é caracterizada por substituir na matriz básica atual um determinado vetor, por outro determinado vetor de fora da base, e obter a inversa da nova matriz resultante. Como  $B_1^{-1} = B_1 = I$ , e pelo item 3.2 temos que isto é conseguido da seguinte maneira se chamarmos de  $B_2$  a nova matriz obtida

$$B_2^{-1} = E_1 \cdot B_1^{-1}$$

O processo iterativo pode ser estabelecido por:

$$B_{k+1}^{-1} = E_k \cdot B_k^{-1}$$

Portanto a inversa da matriz básica quando forem feitas  $k+1$  trocas de vetores será dada por:

$$B_{k+1}^{-1} = E_k \cdot E_{k-1} \cdots E_1 \cdot I$$

Daremos agora um breve resumo do método simplex revisado na sua forma do produto da inversa. Iniciamos o algoritmo, com a matriz identidade como matriz básica inicial. Após a  $k$ -ésima iteração a inversa da base será dada por:

$$B^{-1} = E_k \cdot E_{k-1} \cdots E_2 \cdot E_1$$

Vamos supor que  $E_1, E_2, \dots, E_k$  estejam armazenadas

Passo 1: Cálculo da solução básica atual

$$x_B = \bar{b} = (E_k(E_{k-1}(\dots(E_1 b)\dots)))$$

Passo 2: Cálculo dos custos relativos:

Primeiramente calcular

$$\pi = (\dots((c_B^t \cdot E_k) \cdot E_{k-1}) \dots E_1)$$

depois calcular

$$(c_j - z_j) = (c_j - \pi \cdot a_j) \quad , \quad j \text{ tal que } x_j \text{ seja variável não básica.}$$

Passo 3: Escolha da variável que entra na base.

Seja  $x_s$  tal que:

$$c_s - z_s = \min_j \{(c_j - z_j)\} \quad , \quad j \text{ tal que } x_j \text{ seja não básica.}$$

Se  $c_s - z_s \geq 0$  a solução é ótima pare;

caso contrário  $x_s$  entra na base.

Passo 4: Atualização do vetor que entra na base

$$y_s = B^{-1} \cdot a_s$$

isto é

$$y_s = (E_k(E_{k-1} \dots (E_1 \cdot a_s) \dots))$$

Passo 5: Escolha da variável que deixa a base pelo mesmo método do simplex revisado (2.2). Definição de um novo vetor  $E_{k+1}$  e volta ao passo-1. A definição de  $E_{k+1}$  é feita da maneira indicada em (3.2): Seja  $x_p$  a variável que sai da base

$$E_{k+1} = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & \eta_{1p} & & \\ & & \vdots & & \\ & & \eta_{mp} & & 1 \\ & & \uparrow & & \\ & & \text{coluna } p & & \\ & & & & 1 \end{bmatrix}$$

onde

$$\eta_{ip} = \begin{cases} -\frac{y_{is}}{y_{ps}} & \text{para } i \neq p \\ \frac{1}{y_{ps}} & \text{para } i = p \end{cases}$$

### Observações:

(1) A operação BTRAN será:

$$(\dots(c_B^t \cdot E_k)E_{k+1}) \dots E_1$$

O cálculo executado recursivamente será:

$$\begin{aligned} c_B^t \cdot E_p &= (c_1, c_2, \dots, c_m) \begin{bmatrix} 1 & & & & \\ & \eta_{1p} & & & \\ & \eta_{2p} & & 1 & \\ & \vdots & & & \\ & \eta_{mp} & & & \\ & & & & 1 \end{bmatrix} \\ &= (c_1, \dots, c_{p-1}, c_B^t \cdot \eta_p, c_{p+1}, \dots, c_m) \end{aligned}$$

Isto é, a única componente que é modificada é a p-ésima.

A operação FTRAN será:

$$(E_k \dots (E_2(E_1 \cdot a_s)) \dots)$$

e o cálculo executado recursivamente será:

$$E_j \cdot a_s = \hat{a}_s + a_{ps} \cdot \eta_p$$

onde  $\hat{a}_s$  é o vetor  $a_s$  com zero na posição  $(p,s)$ ; e  $\eta_p$  é a coluna distinta de uma coluna da matriz identidade na

matriz elementar  $E_j$ .

- (2) As matrizes elementares são armazenadas nas denominadas filas eta, nas quais apenas a coluna diferente das colunas da matriz identidade e a sua respectiva posição dentro da matriz, são armazenadas.

Exemplo:

Sejam as matrizes elementares dadas por suas colunas

$$E_k = [e_1, \dots, e_{p-1}, \eta_p, e_{p+1}, \dots, e_m] \quad E_r = [e_1, \dots, e_{r-1}, \eta_r, e_{r+1}, \dots, e_m]$$

as filas eta correspondentes serão:

$$\eta_p = \begin{bmatrix} \eta_{1p} \\ \vdots \\ \eta_{mp} \end{bmatrix}, \quad \eta_r = \begin{bmatrix} \eta_{1r} \\ \vdots \\ \eta_{mr} \end{bmatrix}$$

Daqui para frente confundiremos matriz elementar e a sua correspondente fila eta.

- (3) O processo iterativo da Forma do Produto da Inversa deve ser interrompido, quando o número de iterações cresce muito. A quantificação desse "muito", geralmente é o valor  $(m+50)$ , onde  $m$  é o número de restrições do problema. Tal interrupção é necessária pois, uma seqüência muito grande de operações de produto interno, pode gerar erros de arredondamento relevantes. Outro motivo para a interrupção do método, é o crescimento intenso das transferências memória central  $\leftrightarrow$  memória auxiliar, já que as filas eta na maioria das vezes, são armazenadas em memória auxiliar. O crescimento dessas transferências, aumenta o tempo computacional do algoritmo, na resolução de problemas. No caso de armaze-

namento da inversa em memória central, quando cresce o número de iterações, a representação da inversa através de filas  $eta$  fica inadequada, do ponto de vista de ocupação de memória, em relação à representação explícita da inversa. Com o objetivo de contornar estas objeções, provocadas pelo crescimento do número de iterações, o processo é interrompido para a execução de uma rotina de reinversão, que fornecerá uma inversa representada por  $m$  filas  $eta$ , e que certamente será mais livre de erros que a anterior. Na seção 3.4 será tratada a questão da reinversão da matriz básica.

#### 3.4. A Reinversão da matriz básica

Quando a forma do produto da inversa é utilizada, e à medida que o número de iterações cresce, surgem os seguintes problemas:

- (a) O número de filas  $eta$  pode crescer muito fazendo com que as operações BTRAN e FTRAN aumentem e conseqüentemente o tempo de CPU gasto pelo algoritmo.
- (b) A instabilidade numérica provocada pelos erros de arredondamento acumulados em cada iteração pode se tornar muito perigosa para a confiabilidade do software.
- (c) Se a inversa está armazenada "in-core", o espaço ocupado pela sua representação através de filas  $etas$  pode ser muito grande em relação ao espaço gasto na representação explícita da matriz inversa, tornando o método da forma do produto da inversa inadequado.



O recurso utilizado para contornar tais problemas nos sistemas comerciais é recalculando a matriz inversa da matriz básica e continuar o método com esta nova matriz inversa que seguramente terá menos erros acumulados e será mais compacta pois para  $k$  iterações,  $k > m$  e  $B(m \times m)$  precisaremos de  $m$  operações de pivoteamento para obter a inversa, ou seja,  $m$  matrizes elementares, enquanto que a mesma inversa estava sendo representada por  $k$  matrizes elementares.

A maneira de calcular a inversa da matriz básica é, partindo da matriz identidade e utilizando o método da atualização da inversa descrito no item 3.2, construir a nova inversa da matriz básica.

O método que iremos descrever é baseado no método de Gauss-Jordan, e a inversa é determinada a partir de uma série de pivoteamentos feitos nas colunas da matriz  $B$  a ser invertida.

A ordenação dos vetores da base, tem um papel fundamental no método pois é esta ordenação que vai influir na esparsidade da representação da inversa. Dados os vetores  $b_1, b_2, \dots, b_m$  linearmente independentes existem muitas maneiras de ordená-los para formarem a base  $B$ . Por exemplo, podemos ter  $[b_m, b_{m-1}, \dots, b_1]$  ou  $[b_1, b_m, b_2, b_{m-1}, \dots]$ . Evidentemente trata-se de matrizes diferentes que obviamente possuirão inversas diferentes.

O objetivo principal do método abordado neste capítulo é ordenar os vetores  $b_1, \dots, b_m$  de tal maneira que a matriz  $B$  daí resultante tenha uma representação da inversa o mais espar-

sa possível. A ordenação de  $b_1, \dots, b_m$  e na verdade feita indiretamente através do pivoteamento. Isto é, dada uma matriz

$$B = [b_1, b_2, \dots, b_m]$$

a escolha dos elementos pivot correspondentes a cada coluna, necessários para o processo de determinação da inversa, implica na verdade em uma permutação, em uma reordenação dos vetores  $b_1, \dots, b_m$ . A inversa obtida será a inversa da matriz assim permutada.

Nos parágrafos que se seguem mostraremos primeiramente para um caso particular de uma matriz (3x3),  $B = [b_1, b_2, b_3]$ , como a escolha dos pivôs implica na verdade na obtenção da inversa de uma matriz obtida a partir da permutação das colunas de  $B$ . Mostraremos como a partir da escolha de pivôs é possível saber qual a permutação que foi feita.

A seguir, para um caso mais geral de uma matriz (mxm) mostraremos como obter a inversa a partir de pivoteamentos das colunas. Para este caso mais geral suporemos, sem perda de generalidade, que os pivôs já estão na diagonal principal. Isto evidentemente é sempre possível através de uma permutação adequada das colunas.

Primeiramente faremos uma análise, do que seja realmente a multiplicação de uma matriz  $B$ , por uma matriz elementar  $E$ , com o objetivo de transformar um vetor coluna de  $B$  num vetor da matriz identidade. É claro que esta transformação está associada a operações elementares nas linhas de  $B$ . Vejamos de que maneira:

Seja

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = [b_1, b_2, b_3]$$

$b_i$  - vetor coluna de  $B$

Seja  $E$  a matriz elementar que transforma o vetor coluna  $b_1$  de  $B$  no vetor  $e_2$ ; isto é

$$B^{(2)} = E \cdot B = \begin{bmatrix} 0 & b_{12}^{(2)} & b_{13}^{(2)} \\ 1 & b_{22}^{(2)} & b_{23}^{(2)} \\ 0 & b_{32}^{(2)} & b_{33}^{(2)} \end{bmatrix}; \text{ onde } E = \begin{bmatrix} 1 & -b_{11}/b_{21} & 0 \\ 0 & 1/b_{21} & 0 \\ 0 & -b_{31}/b_{21} & 1 \end{bmatrix}$$

A mesma matriz  $B^{(2)}$  pode ser conseguida com as seguintes operações elementares nas linhas de  $B$ :

(a) divisão da linha 2 por  $b_{21}$  (o pivot):

$$\begin{bmatrix} 1, b_{22}^{(2)}, b_{23}^{(2)} \end{bmatrix} = \frac{1}{b_{21}} \begin{bmatrix} b_{21}, b_{22}, b_{23} \end{bmatrix}$$

(b) substituição da linha 1 e da linha 3 pelas respectivas combinações lineares de linhas:

$$\begin{bmatrix} 0, b_{12}^{(2)}, b_{13}^{(2)} \end{bmatrix} = \frac{-b_{11}}{b_{21}} \begin{bmatrix} b_{21}, b_{22}, b_{23} \end{bmatrix} + \begin{bmatrix} b_{11}, b_{12}, b_{13} \end{bmatrix}$$

$$\begin{bmatrix} 0, b_{32}^{(2)}, b_{33}^{(2)} \end{bmatrix} = \frac{-b_{31}}{b_{21}} \begin{bmatrix} b_{21}, b_{22}, b_{23} \end{bmatrix} + \begin{bmatrix} b_{31}, b_{32}, b_{33} \end{bmatrix}$$

$$B^{(2)} = \begin{bmatrix} 0 & b_{12}^{(2)} & b_{13}^{(2)} \\ 1 & b_{22}^{(2)} & b_{23}^{(2)} \\ 0 & b_{32}^{(2)} & b_{33}^{(2)} \end{bmatrix}$$

Assim a multiplicação de uma matriz elementar  $E(m \times m)$  pela matriz  $B(m \times m)$  com o objetivo de transformar uma coluna de  $B$  em uma coluna de  $I(m \times m)$  nada mais é do que operações elementares efetuadas nas linhas de  $B$ .

Para uma matriz  $B(3 \times 3)$  veremos agora como a escolha dos pivôs implica na alteração da inversa da matriz  $B$  permutada.

Seja

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = [b_1, b_2, b_3]$$

### 1.<sup>a</sup> Iteração:

Consideremos o vetor coluna  $b_1$  de  $B$ ;

Suponhamos  $b_{21} \neq 0$  e o tomamos como o primeiro elemento pivot, teremos:

$$E_1 = \begin{bmatrix} 1 & \eta_{12} & 0 \\ 0 & \eta_{22} & 0 \\ 0 & \eta_{32} & 0 \end{bmatrix} \quad \text{onde} \quad \eta_{i2} = \frac{-b_{i1}}{b_{21}}, \quad i \neq 2$$

$$\eta_{22} = \frac{1}{b_{21}}$$

Teremos então:

$$B^{(2)} = E_1 \cdot B = \begin{bmatrix} 0 & b_{12}^{(2)} & b_{13}^{(2)} \\ 1 & b_{22}^{(2)} & b_{23}^{(2)} \\ 0 & b_{32}^{(2)} & b_{33}^{(2)} \end{bmatrix}$$

### 2.<sup>a</sup> Iteração:

Consideremos o segundo vetor de  $B^{(2)}$ ;

Suponhamos agora que  $b_{32}^{(2)} \neq 0$  seja adotado como pivot

$$E_2 = \begin{bmatrix} 1 & 0 & \eta_{13} \\ 0 & 1 & \eta_{23} \\ 0 & 0 & \eta_{33} \end{bmatrix}$$

onde

$$\eta_{i3} = \frac{-b_{i2}^{(2)}}{b_{32}^{(2)}}, \quad i \neq 3$$

$$\eta_{33} = \frac{1}{b_{32}^{(2)}}$$

teremos então

$$B^{(3)} = E_2 \cdot B^{(2)} = \begin{bmatrix} 0 & 0 & b_{13}^{(3)} \\ 1 & 0 & b_{23}^{(3)} \\ 0 & 1 & b_{33}^{(3)} \end{bmatrix}$$

### 3.<sup>a</sup> Iteração

Consideremos o terceiro vetor de  $B^{(3)}$ ;

Tomemos agora  $b_{13}^{(3)} \neq 0$  como elemento pivot:

$$E_3 = \begin{bmatrix} \eta_{11} & 0 & 0 \\ \eta_{21} & 1 & 0 \\ \eta_{31} & 0 & 1 \end{bmatrix}$$

onde

$$\eta_{11} = \frac{1}{b_{13}^{(3)}}$$

$$\eta_{i1} = \frac{-b_{i3}^{(3)}}{b_{13}^{(3)}}, \quad i = 2, 3$$

teremos então:

$$B^{(4)} = E_3 \cdot B^{(3)} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Isto é:

$$B^{(4)} = E_3 \cdot E_2 \cdot E_1 \cdot B \quad (3.7)$$

Notemos que  $B^{(4)}$  é a matriz identidade com as colunas permutadas. Seja  $P$  a matriz de permutação que faz com que  $I$  se transforme em  $B^{(4)}$ , isto é,

$$B^{(4)} = I \cdot P$$

Dessa forma (3.7) fica:

$$I \cdot P = E_3 \cdot E_2 \cdot E_1 \cdot B \quad (3.8)$$

Multiplicando à direita (3.8) pela matriz de permutação inversa  $P^{-1}$ , teremos:

$$I = E_3 \cdot E_2 \cdot E_1 \cdot B \cdot P^{-1} \quad (3.9)$$

(3.9) nos diz que

$$(B \cdot P^{-1})^{-1} = E_3 \cdot E_2 \cdot E_1 \quad (3.9.A)$$

Obtivemos assim a inversa de uma matriz  $BP^{-1}$  na forma de um produto de matrizes elementares, o que era o nosso objetivo inicial. As matrizes elementares são armazenadas na forma de filas etc. Cabe também lembrar que como  $P^{-1}$  é também uma matriz de permutação,  $BP^{-1}$  nada mais é do que a matriz  $B$  com as colunas permutadas.

Na prática não precisamos determinar  $P^{-1}$  veremos a seguir como determinar a permutação que foi feita em  $B$ .

Seja  $B$  dada por seus vetores colunas:

$$B = [\bar{b}_1, b_2, b_3]$$

Por (3.8) temos:

$$[e_2, e_3, e_1] = E_3 \cdot E_2 \cdot E_1 \cdot B = [E_3 \cdot E_2 \cdot E_1 \cdot b_1, E_3 \cdot E_2 \cdot E_1 \cdot b_2, E_3 \cdot E_2 \cdot E_1 \cdot b_3]$$

então:

$$E_3 \cdot E_2 \cdot E_1 \cdot b_1 = e_2 ; E_3 \cdot E_2 \cdot E_1 \cdot b_2 = e_3 ; E_3 \cdot E_2 \cdot E_1 \cdot b_3 = e_1$$

Podemos escrever:

$$I = [E_3 \cdot E_2 \cdot E_1 \cdot b_3, E_3 \cdot E_2 \cdot E_1 \cdot b_1, E_3 \cdot E_2 \cdot E_1 \cdot b_2]$$

e portanto

$$I = E_3 \cdot E_2 \cdot E_1 \cdot [\bar{b}_3, b_1, b_2] \quad (3.10)$$

Denominemos  $\bar{B} = [\bar{b}_3, b_1, b_2]$ . Pela unicidade da inversa vemos que:

$$\bar{B}^{-1} = E_3 \cdot E_2 \cdot E_1 \quad (3.10.A)$$

Comparando (3.9.A) com (3.10.A) temos:

$$\bar{B} = (BP^{-1}) = [\bar{b}_3, \bar{b}_1, \bar{b}_2] ,$$

o que nos permite facilmente identificar a permutação efetuada.

Uma maneira prática de obtermos a permutação de B, provocada pela escolha do pivot, é a seguinte:

Na iteração k temos o vetor-coluna  $b_k^{(k)}$  considerado em  $B^{(k)}$ ; seja i a linha do pivot adotado. O vetor  $b_k$  de B deverá estar na i-ésima posição da matriz B permutada final. Representaremos isto através de  $\sigma(i) = k$ , isto é, o vetor  $b_k$  estará na posição i, ou, o que é o mesmo, o vetor que estará na i-ésima posição da matriz permutada será o vetor  $b_k$ . Para exemplificarmos esta regra prática, voltemos ao desenvolvimento do método, com escolha de pivot fora da diagonal. Na iteração 1, o primeiro vetor coluna de B é considerado; o elemento pivot foi escolhido na linha 2; desta forma definimos  $\sigma(\underline{2}) = \underline{1}$ . Na iteração 2 o segundo vetor coluna de  $B^{(2)}$  é considerado e o elemento pivot escolhido está na linha 3; definimos então

$$\sigma(\underline{3}) = \underline{2}$$

Analogamente na iteração 3 teremos

$$\sigma(\underline{1}) = \underline{3}$$

Então  $(BP^{-1})$  será dada por:

$$(BP^{-1}) = [b_{\sigma(1)}, b_{\sigma(2)}, b_{\sigma(3)}] = [b_3, b_1, b_2]$$



A escolha do elemento pivot em cada iteração, é um fator de grande importância dentro da reinversão. É a escolha adequada do elemento pivot em cada iteração, que irá levar-nos à obtenção de uma representação da inversa mais esparsa. Isto significará em termos computacionais, menos cálculos e menor ocupação de espaço no computador.

A seguir procuraremos generalizar o processo de determinação da inversa de uma matriz  $(m \times m)$  qualquer.

Seja  $B$  uma matriz a ser invertida:

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ \vdots & \vdots & & \vdots \\ b_{m1} & b_{m2} & & b_{mm} \end{bmatrix} \quad (3.5)$$

Sem perda de generalidade suporemos que as colunas de  $B$  estejam arrumadas de tal forma que o elemento pivot em cada iteração esteja na diagonal principal, para que tenhamos no final do método uma matriz identidade não permutada. Cabe lembrar que tal arrumação das colunas é sempre possível. O método de reinversão que iremos descrever é baseado no método de eliminação de Gauss-Jordan e é conhecido como a Forma do Produto de Gauss-Jordan. Este método pode ser descrito da seguinte forma: Seja a matriz  $B$  dada por (3.5):

Passo 1: Considerar o primeiro vetor coluna de  $B$  e calcular

$$E_1 = \begin{bmatrix} \eta_{11} & 0 & \dots & 0 \\ \eta_{21} & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ \eta_{m1} & 0 & & 1 \end{bmatrix}$$

onde

$$\eta_{i1} = \frac{-b_{i1}^{(1)}}{b_{11}^{(1)}} \text{ para } i \neq 1 \text{ e } b_{i1}^{(1)} = b_{i1}$$

$$\eta_{11} = \frac{1}{b_{11}^{(1)}}$$

Considerando  $B^{(1)} = B$  temos:

$$B^{(2)} = E_1 \cdot B^{(1)} = E_1 \cdot B = \begin{bmatrix} 1 & b_{12}^{(2)} & \dots & b_{1m}^{(2)} \\ 0 & b_{22}^{(2)} & & b_{2m}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & b_{m2}^{(2)} & & b_{mm}^{(2)} \end{bmatrix}$$

Passo k : Considerar o k-ésimo vetor coluna de  $B^{(k)}$  e calcular

$$E_k = \begin{bmatrix} 1 & \dots & \eta_{1k} & \dots & 0 \\ 0 & \dots & \eta_{2k} & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & & \eta_{mk} & & 1 \end{bmatrix} \quad \text{onde } \eta_{ik} = \frac{-b_{ik}^{(k)}}{b_{kk}^{(k)}}, \quad i \neq k$$

$$\eta_{kk} = \frac{1}{b_{kk}^{(k)}}$$

$$B^{(k+1)} = E_k \cdot B^{(k)} = E_k \dots E_2 \cdot E_1 \cdot B^{(1)} = \begin{bmatrix} 1 & b_{1,k+1}^{(k+1)} & \dots & b_{1,m}^{(k+1)} \\ \vdots & \vdots & & \vdots \\ 1 & b_{k,k+1}^{(k+1)} & & b_{k,m}^{(k+1)} \\ \vdots & \vdots & & \vdots \\ b_{m,k+1}^{(k+1)} & & & b_{mm}^{(k+1)} \end{bmatrix}$$

No passo  $m$  teremos: (lembrar que  $B = B^{(1)}$ )

$$I = B^{(m+1)} = E_m \dots E_2 \cdot E_1 \cdot B^{(1)} \quad (3.6)$$

portanto,  $B^{-1} = E_m \dots E_1$

Para caracterizar-mos a importância da escolha do elemento pivot, daremos a seguir dois exemplos de reinversão utilizando o método da Forma do Produto de Gauss-Jordan.

### Exemplo 1

Seja  $B$  a matriz a ser invertida:

$$B = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Tomando como pivot os elementos da diagonal principal temos:

$$B^{(1)} = B \quad ; \quad \text{pivot } b_{11} = 1$$

$$E_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad ; \quad \text{fila eta correspondente: } \eta_1 = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$$

$$B^{(2)} = E_1 \cdot B^{(1)} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{bmatrix}$$

Tomando como elemento pivot  $b_{22}^{(2)}$  teremos:

$$E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} ; \text{ fila eta correspondente } \eta_2 = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$$

$$B^{(3)} = E_2 \cdot B^{(2)} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 2 \end{bmatrix} ; \text{ tomando como pivot } b_{33}^{(3)}$$

temos:

$$E_3 = \begin{bmatrix} 1 & 0 & -1/2 \\ 0 & 1 & 1/2 \\ 0 & 0 & 1/2 \end{bmatrix} ; \text{ fila eta correspondente } \eta_3 = \begin{bmatrix} -1/2 \\ 1/2 \\ 1/2 \end{bmatrix}$$

$$B^{-1} = E_3 \cdot E_2 \cdot E_1 = \begin{bmatrix} 1/2 & 1/2 & -1/2 \\ -1/2 & 1/2 & 1/2 \\ 1/2 & -1/2 & 1/2 \end{bmatrix}$$

Este exemplo demonstra que apesar da matriz inversa ser totalmente densa, a sua representação por meio de filas eta não é totalmente densa.

Consideremos agora a mesma matriz  $B$ , mas com uma seqüência de elementos pivot distinta:

Seja  $b_{21} \neq 0$  o primeiro elemento pivot. Então teremos:  $\sigma(2) = 1$  ;

$$E_1 = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} ; \text{ fila eta correspondente: } \eta_2 = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$$

$$B^{(2)} = E_1 \cdot B = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

Seja  $b_{32}^{(2)}$  o segundo elemento pivot tomado. Teremos então:  $\sigma(3) = 2$  ;

$$E_2 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} ; \text{ fila eta correspondente: } \eta_3 = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

$$B^{(3)} = E_2 \cdot B^{(2)} = \begin{bmatrix} 0 & 0 & 2 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix}$$

Considerando  $b_{13}^{(3)}$  como o último pivot teremos:  $\sigma(1) = 3$

$$E_3 = \begin{bmatrix} 1/2 & 0 & 0 \\ 1/2 & 1 & 0 \\ -1/2 & 0 & 1 \end{bmatrix} ; \text{ fila eta correspondente: } \eta_1 = \begin{bmatrix} 1/2 \\ 1/2 \\ -1/2 \end{bmatrix}$$

$$B^{(4)} = E_3 \cdot B^{(3)} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$E_1 \cdot E_2 \cdot E_3 = \begin{bmatrix} 1/2 & -1/2 & 1/2 \\ 1/2 & 1/2 & -1/2 \\ -1/2 & 1/2 & 1/2 \end{bmatrix}$$

será a inversa da matriz  $\tilde{B} = [b_{\sigma(1)}, b_{\sigma(2)}, b_{\sigma(3)}] = [b_3, b_1, b_2]$

$$\tilde{B} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} ;$$

Esta escolha de pivot que nos levou a uma permutação das colunas da matriz  $B$ , não foi adequada, pois em relação a escolha de pivot anterior, forneceu uma representação da inversa pior, (mais densa que a anterior). Basta ver que as filas têm agora um menor número de zeros.

### Exemplo 2

Seja a matriz  $B$  dada por:

$$B = \begin{bmatrix} 2 & 0 & 1 \\ 6 & 0 & 0 \\ 2 & 4 & 2 \end{bmatrix}$$

Tomando  $b_{31}$  como o primeiro elemento pivot teremos:  $\sigma(3) = 1$ ;

$$E_1 = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -3 \\ 0 & 0 & 1/2 \end{bmatrix} ; \text{ fila eta correspondente: } \eta_3 = \begin{bmatrix} -1 \\ -3 \\ 1/2 \end{bmatrix}$$

$$B^{(2)} = E_1 \cdot B = \begin{bmatrix} 0 & -4 & -1 \\ 0 & -12 & -6 \\ 1 & 2 & 1 \end{bmatrix}$$

Tomando  $b_{12}^{(2)}$  como elemento pivot teremos:  $\sigma(1) = 2$

$$E_2 = \begin{bmatrix} -1/4 & 0 & 0 \\ -3 & 1 & 0 \\ 1/2 & 0 & 1 \end{bmatrix} ; \text{ fila eta correspondente: } \eta_1 = \begin{bmatrix} -1/4 \\ -3 \\ 1/2 \end{bmatrix}$$

$$B^{(3)} = E_2 \cdot B^{(2)} = \begin{bmatrix} 0 & 1 & 1/4 \\ 0 & 0 & -3 \\ 1 & 0 & 1/2 \end{bmatrix}$$

Tomando  $b_{23}^{(3)}$  como pivot teremos:  $\sigma(2) = 3$

$$E_3 = \begin{bmatrix} 1 & 1/12 & 0 \\ 0 & -1/13 & 0 \\ 0 & 1/6 & 1 \end{bmatrix} ; \text{ fila eta correspondente: } \eta_2 = \begin{bmatrix} 1/12 \\ -1/13 \\ 1/6 \end{bmatrix}$$

$$B^{(4)} = E_3 \cdot B^{(3)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Esta escolha de pivot nos leva à seguinte permutação das colunas de  $B$  : seja  $b_1, b_2, b_3$  as colunas de  $B$

$$\tilde{B} = [b_2, b_3, b_1] = \begin{bmatrix} 0 & 1 & 2 \\ 0 & 0 & 6 \\ 4 & 2 & 2 \end{bmatrix} = [b_{\sigma(1)}, b_{\sigma(2)}, b_{\sigma(3)}]$$

$$\tilde{B}^{-1} = E_3 \cdot E_2 \cdot E_1 = \begin{bmatrix} -1/2 & 1/2 & 1/4 \\ 1 & -1/3 & 0 \\ 0 & 1/6 & 0 \end{bmatrix}$$

Notemos que a representação da inversa através das filas e a com esta escolha de pivot é totalmente densa. Consideremos agora a mesma matriz mas com uma escolha do elemento pivot diferente.

$$B = \begin{bmatrix} 2 & 0 & 1 \\ 6 & 0 & 0 \\ 2 & 4 & 2 \end{bmatrix}$$

Seja  $b_{21}$  o primeiro elemento pivot escolhido:  $\sigma(2) = 1$

$$E_1 = \begin{bmatrix} 1 & -1/3 & 0 \\ 0 & 1/6 & 0 \\ 0 & -1/3 & 1 \end{bmatrix} ; \text{ fila eta correspondente: } \eta_2 = \begin{bmatrix} -1/3 \\ 1/6 \\ -1/3 \end{bmatrix}$$

$$B^{(2)} = E_1 \cdot B = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 4 & 2 \end{bmatrix}$$



Tomando  $b_{32}^{(2)}$  como o segundo elemento pivot teremos:  $\sigma(3) = 2$

$$E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/4 \end{bmatrix} ; \text{ fila eta correspondente: } \eta_3 = \begin{bmatrix} 0 \\ 0 \\ 1/4 \end{bmatrix}$$

$$B^{(3)} = E_2 \cdot B^{(2)} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1/2 \end{bmatrix}$$

Tomando como terceiro elemento pivot  $b_{13}^{(3)}$  teremos:  $\sigma(1) = 3$

$$E_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/2 & 0 & 1 \end{bmatrix} ; \text{ fila eta correspondente: } \eta_1 = \begin{bmatrix} 1 \\ 0 \\ -1/2 \end{bmatrix}$$

$$B^{(4)} = B^{(3)} \cdot E_3 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Esta escolha de pivot nos leva à seguinte permutação das colunas de  $B$  :

Seja  $b_1$ ,  $b_2$  e  $b_3$  as colunas de  $B$

$$\tilde{B} = [b_3, b_1, b_2] = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 6 & 0 \\ 2 & 2 & 4 \end{bmatrix} = [b_{\sigma(1)}, b_{\sigma(2)}, b_{\sigma(3)}]$$

$$\tilde{B}^{-1} = E_3 \cdot E_2 \cdot E_1 = \begin{bmatrix} 1 & -1/3 & 0 \\ 0 & 1/6 & 0 \\ -1/2 & 1/12 & 1/4 \end{bmatrix}$$

Notemos agora que esta escolha de pivot  $\tilde{e}$  é bem melhor que a anterior, pois a representação da inversa  $\tilde{B}^{-1}$  é muito menos densa. Tudo isto nos leva a concluir que, a propagação de elementos não nulos durante o método de reinversão, pode ser minimizada com uma escolha de elementos pivot adequada.

Uma pergunta que surge nesse ponto  $\tilde{e}$ : como se deve escolher o elemento pivot, numa dada iteração, para evitar o aparecimento de novos elementos não nulos?

Na tentativa de responder a esta questão, devemos voltar às raízes do método de reinversão que adotamos, que nada mais é que o método de Eliminação de Gauss-Jordan. Novamente vamos encarar o método da Forma do Produto de Gauss-Jordan como sucessivas pré-multiplicações da matriz dada por matrizes elementares, que transformam as colunas, da matriz dada, sucessivamente em colunas da matriz identidade

$$\begin{aligned} B &= B^{(1)} ; B^{(2)} = E_1 \cdot B^{(1)} ; B^{(3)} = E_2 \cdot B^{(2)} ; \dots B^{(m+1)} = \\ &= E_m \cdot B^{(m)} = I \end{aligned}$$

É claro que se durante o processo, não houver propagação de elementos não nulos, isto é, um elemento nulo em  $B^{(i)}$  não se torne não nulo em  $B^{(i+1)}$ , este fato refletirá nas definições das matrizes elementares subsequentes e portanto na esparsidade das filas etas. Isto porque, as matrizes elementares subsequentes e filas etas correspondentes serão definidas a partir dos

elementos da matriz  $B^{(i+1)}$ .

Seja  $B(4 \times 4)$  a matriz na qual estamos aplicando, o método de reinversão. Seja  $x$  a representação de um elemento não nulo, na matriz  $B$ .

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} = \begin{bmatrix} x & x & x & x \\ x & & & x \\ x & & & \\ x & x & & x \end{bmatrix}$$

Seja  $E_1$  a matriz elementar que vai transformar a primeira coluna de  $B$  numa coluna da matriz identidade. A multiplicação  $E_1 \cdot B$ , do ponto de vista de operações elementares, efetua o seguinte:

- (a) se o elemento pivot está na linha  $i$ , esta linha é dividida pelo elemento pivot.
- (b) cada linha  $k$  diferente da linha pivot, é substituída por uma combinação linear da linha pivot e da linha  $k$ .

(Todas estas operações já foram detalhadas no início da seção 3.4).

Suponhamos inicialmente que o pivot escolhido seja  $b_{11}$ ; e seja  $B^{(2)} = E_1 \cdot B$ ; teremos então:

$$B^{(2)} = \begin{bmatrix} 1 & b_{12}^{(2)} & b_{13}^{(2)} & b_{14}^{(2)} \\ 0 & b_{22}^{(2)} & b_{23}^{(2)} & b_{24}^{(2)} \\ 0 & b_{32}^{(2)} & b_{33}^{(2)} & b_{34}^{(2)} \\ 0 & b_{42}^{(2)} & b_{43}^{(2)} & b_{44}^{(2)} \end{bmatrix}$$

As equações abaixo definem os elementos de  $B^{(2)}$

$$b_{1j}^{(2)} = \frac{b_{1j}}{b_{11}}, \quad j = 1, 2, 3, 4 \quad (3.11)$$

$$b_{kj}^{(2)} = b_{kj} - \frac{b_{1j} \cdot b_{k1}}{b_{11}}; \quad \begin{matrix} k = 2, 3, 4 \\ j = 1, 2, 3, 4 \end{matrix} \quad (3.12)$$

Observando as equações (3.11) e (3.12) verificamos que:

(a) na linha pivot não existe a possibilidade de um elemento nulo em  $B$  se tornar não nulo em  $B^{(2)}$ ;

(b) nas linhas distintas da linha pivot teremos:

se  $b_{kj} = 0$ ,  $b_{kj}^{(2)}$  poderá ser nulo se  $b_{1j}$  ou  $b_{k1}$  forem nulos.

Verifiquemos que  $b_{1j}$  está na linha do elemento pivot e  $b_{k1}$  está na coluna do elemento pivot.

Adotando esta escolha de pivot  $B^{(2)}$  poderá ter a seguinte configuração:

$$B^{(2)} = \begin{bmatrix} 1 & X & X & X \\ 0 & X & X & X \\ 0 & X & X & X \\ 0 & X & X & X \end{bmatrix}$$

Isto não é interessante pois as filas e as seguintes tenderão a serem totalmente densas.

Se ao invés de adotarmos  $b_{11}$  como elemento pivot, adotarmos o elemento  $b_{31}$ , teríamos as seguintes equações definindo os elementos de  $B^{(2)}$ :

$$b_{3j}^{(2)} = \frac{b_{3j}}{b_{31}} ; j = 1, 2, 3, 4 \quad (3.13)$$

$$b_{kj}^{(2)} = b_{kj} - \frac{b_{3j} \cdot b_{k1}}{b_{31}} ; \begin{matrix} k = 1, 2, 4 \\ j = 1, 2, 3, 4 \end{matrix} \quad (3.14)$$

Observando as equações (3.13) e (3.14) verificamos que:

$$b_{3j} = 0 \text{ para } j = 2, 3, 4 \text{ isto implica que}$$

$$b_{3j}^{(2)} = 0 \text{ para } j = 2, 3, 4 ; \text{ além disso se } b_{kj} = 0 ,$$

então  $b_{kj}^{(2)} = 0$ . Notemos que  $b_{3j}$ ;  $j = 1, 2, 3, 4$  é exatamente a linha do elemento pivot; e se ela é esparsa nenhum elemento nulo em  $B$  se tornará não nulo em  $B^{(2)}$ . Nesse caso  $B^{(2)}$  poderá ter a seguinte configuração:

$$B^{(2)} = \begin{bmatrix} 0 & X & X & X \\ 0 & & & X \\ 1 & & & \\ 0 & X & & X \end{bmatrix}$$

É fácil ver que se fixarmos a linha e tomarmos o pivot na coluna mais esparsa, o raciocínio análogo é válido.

Podemos agora definir três estratégias, de escolha do elemento pivot em cada iteração, do método da Forma do Produto de Gauss-Jordan, com o intuito de evitarmos a propagação de elementos não nulos.

Estratégia 1: escolher como elemento pivot, na iteração  $k$ , o elemento que se encontra, na linha mais esparsa da matriz  $B^{(k)}$ , no  $k$ -ésimo vetor coluna fixado. Levando em conta que uma linha só pode conter um elemento pivot durante todo o processo, e que o pivot tem de ser não nulo.

Estratégia 2: escolher como elemento pivot, na iteração  $k$ , o elemento que se encontra na coluna mais esparsa de  $B^{(k)}$ , na  $k$ -ésima linha fixada; levando em conta as considerações finais da estratégia 1.

Estratégia 3: escolher como elemento pivot, na iteração  $k$ , o elemento que se encontra na linha ou coluna mais esparsa possível da matriz  $B^{(k)}$ ; levando em conta as considerações finais da estratégia 1.

Um fato que pode ser facilmente verificado é que, se  $B$  tem forma triangular inferior, e se pivotarmos na diagonal principal, não haverá propagação de elementos não nulos, com a aplicação da Forma do Produto de Gauss-Jordan. Na realidade toda manipulação de escolha de elemento pivot, visa conseguir o mesmo efeito do pivoteamento na diagonal em uma matriz triangular inferior. Os métodos de escolha do elemento pivot, na reinversão, que podemos encontrar com maior frequência são: o método de Merit <sup>[13]</sup>, o método de Hellerman-Rarick <sup>[13]</sup> e o método de Larsen <sup>[14]</sup>. Todos objetivam a não propagação de elementos não nulos na representação da inversa.

Um método de escolha bastante comum é o que adota como pivot, o elemento de maior valor absoluto nas colunas e linhas ainda não consideradas; este método objetiva a diminuição dos erros de arredondamento.

O método de reinversão da matriz básica implementado no sistema de FURNAS é o de Larsen <sup>[14]</sup>, que descreveremos a seguir.

#### 3.4.1. O método de Reinversão de Larsen

O método de Larsen, ao invés de adotar como critério para a escolha da linha pivot, aquela que tem o maior elemento em valor absoluto, adota o critério que leva em conta a densidade da matrix cuja inversa se quer obter, - escolhe como linha pivot aquela que tem a menor quantidade de elementos não nulos. Na prática foi verificado <sup>[18]</sup>, que o uso deste método diminui a densidade das filas eta, fato este bastante importante do ponto de vista computacional. A densidade das filas eta

é que vai influenciar no espaço necessário para armazenar a representação da inversa, bem como afetar a quantidade de cálculo em cada iteração do algoritmo simplex. Desta forma reduzir a densidade das filas é desejável a fim de reduzir o tempo computacional.

O método de reinversão de Larsen é basicamente o método da Forma do Produto de Gauss-Jordan com uma escolha do elemento pivot diferente. Este método pode ser descrito da seguinte forma:

#### Passo 1: Inicialização

Seja  $B(m \times m)$  a matriz a ser invertida e dada por seus vetores colunas

$$B = [b_1, b_2, \dots, b_m]$$

Seja  $\bar{B}^{(1)} = B$  e  $\sigma(1 \times m)$  um vetor de  $m$  posições para guardar as permutações das colunas de  $B$ , provocadas pela escolha do pivot.

Passo 2: Determinar o número de elementos não nulos em cada linha da matriz  $\bar{B}^{(k)}$ . Seja  $N_i$  o número de elementos não nulos na linha  $i$  da matriz  $\bar{B}^{(k)}$ .

Passo 3: Atualização do vetor  $b_k$  de  $B$ .

Multiplicar as matrizes elementares já geradas pelo vetor  $b_k$ ,

$$y_k = E_{k-1} \cdots E_{k-2} \cdots E_1 \cdot b_k.$$

Na primeira iteração quando ainda não foi gerada nenhuma matriz elementar,



$$y_k = b_k$$

Passo 4: Escolha da linha pivot.

(a) Determinar o conjunto  $P$ , dos índices das linhas que ainda não foram tomados elementos pivot, e que possuam elementos do vetor  $b_k$  cujos valores absolutos sejam maiores que uma certa tolerância  $T$

$$P = \{i / [b_{ik}] > T, \text{ e } i \text{ ainda não tenha sido adotada como linha de elemento pivot}\};$$

$$i = 1, \dots, m$$

(b) Escolher como elemento pivot o elemento  $b_{qk}$  tal que  $q \in P$  e que  $N_q$  seja o menor possível;

$$N_q = \min_{i \in P} \{N_i\}$$

(c) Fazer  $\sigma(q) = k$

Passo 5: Geração de uma matriz elementar a partir do vetor  $y_k$ , tomando como pivot o elemento  $y_{qk}$ .

$$E_k = \begin{bmatrix} 1 & & \eta_{1q} & & \\ & 1 & \eta_{2q} & & \\ & & \vdots & & \\ & & \eta_{mq} & & \\ & & & & 1 \end{bmatrix}$$

$$\text{onde: } \eta_{iq} = \begin{cases} \frac{-y_{ik}}{y_{qk}} & \text{para } i \neq q, i = 1, \dots, m \\ \frac{1}{y_{qk}} & \text{para } i = q \end{cases}$$

Passo 6: Fazer  $k = k+1$  ;

Se  $k > m$  pare, pois temos que

$$\tilde{B}^{-1} = E_{k-1} \cdot E_{k-2} \cdots E_2 \cdot E_1$$

é a inversa procurada da matriz  $B$  permutada ( $\tilde{B}$ ),  
dada por:

$$\tilde{B} = [b_{\sigma(1)}, b_{\sigma(2)}, \dots, b_{\sigma(m)}]$$

Se  $k \leq m$  considerar

$$\bar{B}^{(k)} = [b_k, b_{k+1}, \dots, b_m]$$

e voltar ao passo 2.

Deve ter sido observado que o método de Larsen, procura adotar como elemento pivot na iteração  $k$ , aquele que está na linha mais esparsa de  $\bar{B}^{(k)}$  e na  $k$ -ésima coluna de  $B^{(k)}$ ; seja

$$B = [b_1, \dots, b_m]$$

então

$$\bar{B}^{(k)} = [b_k, \dots, b_m]$$

e

$$B^{(k)} = E_k \cdot E_{k-1} \cdots E_1 \cdot B$$

Isto é, o método de Larsen não adota a matriz  $B^{(k)}$  para determinar a linha mais esparsa, mas sim  $\bar{B}^{(k)}$ .

Primeiramente observemos que se o método calcula-

se em cada passo  $k$  a matriz  $B^{(k)}$  completa e a mantivesse na memória para determinar a linha mais esparsa, perderia muito em eficiência do ponto de vista computacional.

Observemos também que  $\bar{B}^{(k)}$  é na verdade formada a partir das colunas de  $B^{(1)} = B$ , eliminando-se as colunas  $b_1, \dots, b_{k-1}$ . Se não há propagação de elementos não nulos, o cálculo do número de elementos não nulos, nas linhas ainda não pivoteados de  $\bar{B}^{(k)}$  e  $B^{(k)}$  é o mesmo. Como o método de Larsen procura evitar a propagação de elementos não nulos, é de se supor que os elementos nulos em  $B^{(1)}$ , continuem nulos em  $B^{(2)}$ ,  $B^{(3)}$ , ...,  $B^{(k)}$ , na sua grande maioria. Este fato nos possibilita utilizar  $\bar{B}^{(k)}$  ao invés de  $B^{(k)}$  para o cálculo de  $N_i$ , com a vantagem de não ser necessário calcularmos  $B^{(k)}$ .

A seguir daremos um exemplo do método de Larsen.

### Exemplo:

Seja  $T = 0$  (tolerância)

Seja a matriz:

$$B = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 0 & 6 \\ 2 & 4 & 2 \end{bmatrix} ; b_1 = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} ; b_2 = \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix} ; b_3 = \begin{bmatrix} 2 \\ 6 \\ 2 \end{bmatrix}$$

1.<sup>a</sup> iteração: Seja  $\bar{B}^1 = B$  ;  $k = 1$

$b_1$  vetor coluna considerado

$$N_1 = 2$$

$$N_2 = 1$$

$$N_3 = 3$$

$y_1 = b_1$  pois não existe matrizes elementares geradas  
 $P = \{1, 3\}$

$N_1 = \min\{N_1, N_3\}$ , portanto  $q = 1$

$b_{11}$  é escolhido como pivot;  $\sigma(1) = 1$

Matriz elementar gerada:

$$E_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 0 \end{bmatrix}$$

2.<sup>a</sup> iteração:  $k = 2$ ;

$$\bar{B}^2 = [\bar{b}_2, \bar{b}_3]$$

$b_2$  vetor considerado

$$N_1 = 1$$

$$N_2 = 1$$

$$N_3 = 2$$

Atualização do vetor considerado

$$y_2 = E_1 \cdot b_2 = \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}$$

$P = \{3\}$ ,  $N_3 = \min\{N_3\}$ ; portanto  $q = 3$  e  $\sigma(3) = 2$ .

O pivot escolhido será  $b_{32}$

A matriz elementar gerada será:

$$E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/4 \end{bmatrix}$$

3.<sup>a</sup> iteração:  $k = 3$

$$\bar{B}^3 = \begin{bmatrix} \bar{b}_3 \end{bmatrix}$$

$b_3$  vetor considerado

$$N_1 = 1$$

$$N_2 = 1$$

$$N_3 = 1$$

Atualização do vetor considerado

$$y_1 = E_2 \cdot E_1 \cdot b_3 = \begin{bmatrix} 2 \\ 6 \\ -1/2 \end{bmatrix}$$

$$P = \{2\} ; N_2 = \min\{N_2\} ; q = 2 ; \sigma(2) = 3$$

o pivot será  $b_{23}$ .

A matriz elementar gerada será:

$$E_3 = \begin{bmatrix} 1 & -1/3 & 0 \\ 0 & 1/6 & 0 \\ 0 & 1/12 & 1 \end{bmatrix}$$

4.<sup>a</sup> iteração:  $k = 4$  ; então

$$\tilde{B}^{-1} = E_3 \cdot E_2 \cdot E_1 = \begin{bmatrix} 1 & -1/3 & 0 \\ 0 & 1/6 & 0 \\ -1/2 & 1/12 & 1/4 \end{bmatrix} ;$$

e

$$\tilde{B} = \begin{bmatrix} \bar{b}_{\sigma(1)} & \bar{b}_{\sigma(2)} & \bar{b}_{\sigma(3)} \end{bmatrix} ; \tilde{B} = \begin{bmatrix} \bar{b}_1 & \bar{b}_3 & \bar{b}_2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 6 & 0 \\ 2 & 2 & 4 \end{bmatrix}$$

CAPÍTULO IV  
O SIMPLEX LU

4.1. Introdução

Existem dois métodos fundamentais de inversão de matriz, que são com mais frequência encontrados nas rotinas de reinversão, geralmente denominadas INVERT, dos sistemas comerciais. São eles: O método da Forma do Produto de Gauss-Jordan, e o método da Forma da Eliminação da Inversa. O método da Forma do Produto de Gauss-Jordan, está descrito no Capítulo III; o método da Forma da Eliminação da Inversa <sup>[22]</sup> descreveremos neste capítulo.

Dantzig et alii <sup>[07]</sup> constataram uma grande vantagem do método da Forma de Eliminação da Inversa sobre o método da Forma do Produto de Gauss-Jordan, no que diz respeito à rapidez, precisão e esparsidade.

Ambos os métodos calculam a inversa da matriz básica apresentando-a como um produto de matrizes elementares. A diferença entre estes dois métodos reside no fato de que, o método da Forma da Eliminação da Inversa apresenta a inversa como um produto de matrizes elementares que também são triangulares. A apresentação de  $B^{-1}$  é dada por:

$$B^{-1} = (U_1^{-1} \cdot U_2^{-1} \cdot \dots \cdot U_m^{-1}) \times (L_m^{-1} \cdot L_{m-1}^{-1} \cdot \dots \cdot L_2^{-1} \cdot L_1^{-1}) \quad (4.1)$$

onde  $U_i^{-1}$  é uma matriz elementar e triangular superior;

$L_i^{-1}$  é uma matriz elementar e triangular inferior;

$$i = 1, \dots, n \text{ e } B(m \times m)$$

Verificada a eficiência do método da Forma da Eliminação da Inversa, surgiu o interesse em procurar uma maneira de continuar o algoritmo simplex, (após uma reinversão, usando a Forma da Eliminação da Inversa), com a matriz inversa da matriz básica na forma fatorizada (4.1). Procurava-se uma maneira eficiente e geral de atualizar a matriz inversa da base dada por (4.1). Os métodos surgidos então, foram: o método de Bartel e Golub <sup>[2]</sup>; e o método de Brayton et alii <sup>[5]</sup> que foi implementado por Tomlin <sup>[27]</sup>.

O uso dessas novas maneiras de atualização, da inversa da base no método simplex, dão lugar a novas versões do algoritmo simplex, que neste texto denominaremos de Simplex LU. Devemos observar, que estaremos sempre nos referindo, ao simplex na sua forma do Produto da Inversa, com a inversa mantida como um produto de fatores triangulares e elementares quando nos referimos ao simplex LU.

Em problemas de grande porte densos, nos quais a resolução requer um número muito grande de iterações, e conseqüentemente, um número muito grande de filas e, é conveniente após uma reinversão adequada, adotar o simplex LU. O motivo disto é que este método tem como característica, a manutenção da esparsidade na representação de  $B^{-1}$ .

Os sistemas comerciais são munidos de um esquema de escolha automática, para decidir qual versão do simplex é a mais conveniente, para prosseguir na resolução de um problema. Este esquema adota o simplex LU, para problemas densos, e o Sim

plex na Forma do Produto da Inversa para problemas esparsos.

O sistema de FURNAS original não era munido da opção LU do simplex. Esta opção foi implementada por nós no referido sistema, com o intuito de deixá-lo mais eficiente. Cabe também salientar que a implementação do simplex LU requereu uma rotina de reinversão usando o método da Forma da Eliminação da Inversa, que também não existia no sistema de FURNAS.

O esquema de escolha, que adotamos no sistema de FURNAS, para adoção do simplex LU ou a adoção do simplex na Forma do Produto da Inversa, é o sugerido em [3]:

- calcula-se a razão entre o número de elementos não nulos na representação da inversa ( $\# B^{-1}$ ) e o número de elementos não nulos da matriz básica ( $\# B$ ); se a razão calculada for maior que 1.1, o simplex LU é adotado; caso contrário o simplex na Forma do Produto da inversa o será..

$$\frac{(\# B^{-1})}{(\# B)} \left\{ \begin{array}{l} > 1.1 - \text{simplex LU} \\ \leq 1.1 - \text{simplex na F.P.I} \end{array} \right.$$

Este esquema foi implementado no MPSX/370 da IBM [3].

Neste capítulo iremos descrever o esquema de atualização da inversa de Forrest-Tomlin [10], a Forma da Eliminação da Inversa e o simplex LU; antes disto porém, daremos algumas propriedades das matrizes triangulares, que usaremos com freqüência.



#### 4.1.1. Algumas Propriedades das Matrizes Triangulares

##### (a) Decomposição de matrizes triangulares

Dada uma matriz triangular inferior  $L(m \times m)$ , esta pode ser decomposta, num produto de matrizes triangulares inferiores e elementares, da seguinte forma:

Seja  $m = 3$

$$L = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & l_{22} & 0 \\ 0 & l_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & l_{33} \end{bmatrix}$$

$$L = L_1 \cdot L_2 \cdot L_3$$

Analogamente temos o seguinte para matrizes triangulares superiores  $U(m \times m)$

Seja  $m = 3$

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

$$U = U_3 \cdot U_2 \cdot U_1$$

No caso geral teremos, para  $L$  e  $U$  respectivamente triangular inferior e triangular superior ( $m \times m$ ):

$$L = L_1 \cdot L_2 \cdots L_m ;$$

onde cada  $L_i$  é triangular inferior e elementar, definida pela  $i$ -ésima coluna de  $L$ ;  $i = 1, \dots, m$ ;

$$U = U_m \cdot U_{m-1} \cdots U_2 \cdot U_1 ;$$

onde cada  $U_i$  é triangular superior e elementar, definida pela  $i$ -ésima coluna de  $U$ ;  $i = 1, \dots, m$ .

(b) Inversão de uma matriz elementar

Dada uma matriz elementar ( $m \times m$ ):

$$E_p = \begin{bmatrix} 1 & 0 & \dots & \eta_{1p} & \dots & 0 \\ 0 & 1 & & \eta_{2p} & \dots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ \vdots & \vdots & & \eta_{pp} & & \vdots \\ \vdots & \vdots & & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & & \eta_{mp} & & 1 \end{bmatrix}$$

A inversa desta matriz elementar é dada por:

$$E_p^{-1} = \begin{bmatrix} 1 & 0 & \dots & \bar{\eta}_{1p} & \dots & 0 \\ 0 & 1 & & \bar{\eta}_{2p} & \dots & 0 \\ \vdots & \vdots & & \vdots & & \vdots \\ \vdots & \vdots & & \bar{\eta}_{pp} & & \vdots \\ \vdots & \vdots & & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & & \bar{\eta}_{mp} & & 1 \end{bmatrix}$$

onde  $\bar{\eta}_{ip} = -\frac{\eta_{ip}}{\eta_{pp}}$ ,  $i = 1, \dots, m$ ;  $i \neq p$

$$\bar{\eta}_{pp} = \frac{1}{\eta_{pp}}, \quad \eta_{pp} \neq 0$$

(c) Inversão de um produto de matrizes

Dado  $L = L_1 \cdot L_2 \cdot \dots \cdot L_m$

temos que:

$$L^{-1} = L_m^{-1} \dots L_2^{-1} \cdot L_1^{-1} ;$$

para

$$U = U_m \cdot U_{m-1} \dots U_2 \cdot U_1$$

temos que:

$$U^{-1} = U_1^{-1} \cdot U_2^{-1} \dots U_{m-1}^{-1} \cdot U_m^{-1}$$

#### 4.2. A Forma de Eliminação da Inversa

O método de inversão denominado Forma de Eliminação da Inversa, e também conhecido como o método da Decomposição Triangular, está baseado no método de Eliminação de Gauss. Este método vem sendo muito utilizado nas rotinas de reinversão dos sistemas de grande porte, por sua grande eficiência em relação aos outros métodos conhecidos. Sua principal característica é fornecer uma representação da inversa na forma fatorizada (4.1).

A Forma da Eliminação da Inversa pode ser descrita da seguinte maneira:

Dada uma matriz  $B(m \times m)$ , para ser invertida, denominamos  $B = B^{(1)}$ , isto é:

$$B^{(1)} = B = \begin{bmatrix} b_{11}^{(1)} & b_{12}^{(1)} \dots b_{1m}^{(1)} \\ \vdots & \vdots & \vdots \\ b_{m1}^{(1)} & b_{m2}^{(1)} \dots b_{mm}^{(1)} \end{bmatrix}$$

Passo 1: Considerar o primeiro vetor de  $B^{(1)}$ ; definir

$$L_1 = \begin{bmatrix} b_{11}^{(1)} & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ b_{m1}^{(1)} & & & 1 \end{bmatrix}, \text{ e calcular } L_1^{-1} \text{ que é dada por:}$$

$$L_1^{-1} = \begin{bmatrix} 1/b_{11}^{(1)} & 0 & \dots & 0 \\ -b_{21}^{(1)}/b_{11}^{(1)} & 1 & & \vdots \\ \vdots & & & \vdots \\ -b_{m1}^{(1)}/b_{11}^{(1)} & 0 & \dots & 1 \end{bmatrix}$$

$$B^{(2)} = L_1^{-1} \cdot B^{(1)} = \begin{bmatrix} 1 & b_{12}^{(2)} & \dots & b_{1m}^{(2)} \\ 0 & b_{22}^{(2)} & & b_{2m}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & b_{m2}^{(2)} & & b_{mm}^{(2)} \end{bmatrix}$$

Passo 2: Considerar o segundo vetor de  $B^{(2)}$ ; definir

$$L_2 = \begin{bmatrix} 1 & 0 & & \\ 0 & b_{22}^{(2)} & & \\ \vdots & \vdots & 1 & \\ 0 & b_{m2}^{(2)} & & 1 \end{bmatrix}, \text{ e calcular } L_2^{-1} \text{ dada por}$$

$$L_2^{-1} = \begin{bmatrix} 1 & & & \\ & 0 & & \\ 0 & 1/b_{22}^{(2)} & & \\ & \vdots & 1 & \\ 0 & -b_{m2}^{(2)}/b_{22}^{(2)} & & 1 \end{bmatrix}$$

$$B^{(3)} = L_2^{-1} \cdot B^{(2)} = \begin{bmatrix} 1 & b_{12}^{(3)} & b_{13}^{(3)} & \dots & b_{1m}^{(3)} \\ 0 & 1 & b_{23}^{(3)} & & b_{2m}^{(3)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & b_{m3}^{(3)} & & b_{mm}^{(3)} \end{bmatrix}$$

Passo k : Considerar o k-ésimo vetor de  $B^{(k)}$ , onde:

$$\begin{bmatrix} 1 & b_{12}^{(k)} & b_{13}^{(k)} & \dots & b_{1k}^{(k)} & \dots & b_{1m}^{(k)} \\ & 1 & b_{23}^{(k)} & & b_{2k}^{(k)} & & b_{2m}^{(k)} \\ & & \vdots & & \vdots & & \vdots \\ & & & & b_{kk}^{(k)} & & b_{km}^{(k)} \\ & & & & \vdots & & \vdots \\ & & & & b_{mk}^{(k)} & & b_{mm}^{(k)} \end{bmatrix}$$

Definir:

$$L_k = \begin{bmatrix} 1 & \dots & 0 & & \\ & & b_{kk}^{(k)} & & \\ & & \vdots & & \\ & & b_{mk}^{(k)} & & \\ 0 & & & & 1 \end{bmatrix}; \text{ e calcular } L_k^{-1} \text{ dada por:}$$

$$L_k^{-1} = \begin{bmatrix} 1 & & & & \\ \vdots & & & & \\ \vdots & & & & \\ \vdots & & 1/b_{kk}^{(k)} & & \\ \vdots & & \vdots & & \\ \vdots & & & & 1 \\ 0 & & -b_m^{(k)}/b_{kk}^{(k)} & & \\ & & & & 1 \end{bmatrix}$$

$$B^{(k+1)} = L_k^{-1} \cdot B^{(k)}$$

Após  $m$  passos teremos  $B^{(m+1)}$  uma matriz triangular superior; isto é, chamando  $B^{(m+1)}$  de  $U$  teremos:

$$U = B^{(m+1)} = L_m^{-1} \dots L_2^{-1} \cdot L_1^{-1} \cdot B.$$

Podemos então escrever, usando as propriedades (a) e (c) da seção 4.1.1, ( $U = U_m \dots U_1$  ;  $U^{-1} = U_1^{-1} \dots U_m^{-1}$ ) :

$$B^{-1} = U_1^{-1} \dots U_m^{-1} \cdot L_m^{-1} \dots L_1^{-1}$$

O que se pode notar no desenvolvimento do método, é que, numa dada iteração  $k$ , a multiplicação da matriz elementar  $L_k^{-1}$ , pela matriz  $B^{(k)}$ , nada mais é que as seguintes operações elementares efetuadas em  $B^{(k)}$  :

(a) Dividir a linha  $k$  pelo elemento  $b_{kk}^{(k)}$  ; (elemento pivot)

$$\left[ 0, \dots, 1, b_{k,k+1}^{(k+1)}, \dots, b_{km}^{(k+1)} \right] = \frac{1}{b_{kk}^{(k)}} \cdot \left[ 0, \dots, b_{kk}^{(k)}, b_{k,k+1}^{(k)}, \dots, b_{km}^{(k)} \right]$$

(b) Para cada linha  $i$ ,  $i > k$  executar a operação elementar de substituição da linha  $i$  pela combinação linear das linhas  $i$  e  $k$ , dada por:

$$\left[ 0, \dots, 0, b_{i,k+1}^{(k+1)}, \dots, b_{im}^{(k+1)} \right] = \frac{-b_{ik}^{(k)}}{b_{kk}^{(k)}} \cdot \left[ 0, \dots, b_{kk}^{(k)}, \dots, b_{km}^{(k)} \right] + \left[ 0, \dots, b_{ik}^{(k)}, \dots, b_{im}^{(k)} \right]$$

Os elementos da matriz  $B^{(k+1)} = L_k^{-1} \cdot B^{(k)}$ , podem ser dados pelas equações:

$$b_{ij}^{(k+1)} = b_{ij}^{(k)}, \text{ para } i < k ; j = 1, 2, \dots, m \quad (4.2)$$

$$b_{kj}^{(k+1)} = \frac{b_{kj}^{(k)}}{b_{kk}^{(k)}}, \text{ } j = 1, \dots, m \quad (4.3)$$

$$b_{ij}^{(k+1)} = b_{ij}^{(k)} - \frac{b_{ik}^{(k)} \cdot b_{kj}^{(k)}}{b_{kk}^{(k)}}, \text{ para } i > k, j=1, \dots, m \quad (4.4)$$

Podemos verificar pela equação (4.2), que os elementos nas linhas já pivoteadas, não se modificam. Desta maneira os elementos nulos nestas linhas não se tornarão não nulos no decorrer do processo. Cabe lembrar que num método de reinversão estamos sempre preocupados com a não propagação de elementos não nulos, assunto este, já abordado no capítulo III.

A seguir daremos um exemplo de aplicação do método.

Exemplo 1:

Seja:

$$B = \begin{bmatrix} 2 & 2 & 1 \\ 0 & 2 & 0 \\ 1 & 4 & 2 \end{bmatrix} \quad ; \quad B = B^{(1)}$$

Passo 1: Considerando o primeiro vetor coluna de  $B^{(1)}$  ;

$$L_1 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad ; \quad L_1^{-1} = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1 & 0 \\ -1/2 & 0 & 1 \end{bmatrix}$$

$$B^{(2)} = L_1^{-1} \cdot B^{(1)} = \begin{bmatrix} 1 & 1 & 1/2 \\ 0 & 2 & 0 \\ 0 & 3 & 3/2 \end{bmatrix}$$

Passo 2: Considerando o segundo vetor coluna de  $B^{(2)}$

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 3 & 1 \end{bmatrix} \quad ; \quad L_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & -3/2 & 1 \end{bmatrix}$$

$$B^{(3)} = L_2^{-1} \cdot B^{(2)} = \begin{bmatrix} 1 & 1 & 1/2 \\ 0 & 1 & 0 \\ 0 & 0 & 3/2 \end{bmatrix}$$

Passo 3 : Consideremos o terceiro vetor coluna de  $B^{(3)}$  ;

$$L_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3/2 \end{bmatrix} \quad ; \quad L_3^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2/3 \end{bmatrix}$$

$$B^{(4)} = L_3^{-1} \cdot B^{(3)} = \begin{bmatrix} 1 & 1 & 1/2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} ;$$

$$U = B^{(4)} \quad ; \quad U = U_3 \cdot U_2 \cdot U_1$$

$$\text{onde } U_3 = \begin{bmatrix} 1 & 0 & 1/2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} ; U_2 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} ; U_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$U^{-1} = U_1^{-1} \cdot U_2^{-1} \cdot U_3^{-1}$$

$$U_1^{-1} = I \quad ; \quad U_2^{-1} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad ; \quad U_3^{-1} = \begin{bmatrix} 1 & 0 & -1/2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$U^{-1} = \begin{bmatrix} 1 & -1 & -1/2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Por outro lado temos:

$$L^{-1} = L_3^{-1} \cdot L_2^{-1} \cdot L_1^{-1}$$

$$\therefore L^{-1} = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ -1/3 & -1 & 2/3 \end{bmatrix}$$

Finalmente temos:

$$B^{-1} = U^{-1} \cdot L^{-1}$$

$$\therefore B^{-1} = \begin{bmatrix} 2/3 & 0 & 1/3 \\ 0 & 1/2 & 0 \\ -1/3 & -1 & 2/3 \end{bmatrix}$$

#### 4.3. O Esquema de Atualização da Inversa de Forrest-Tomlin - Descrição Matemática

Suponhamos que a rotina de reinversão tenha sido executada, que tenhamos  $B^{-1}(mxm)$  na forma de um produto de fatores triangulares e elementares:

$$B^{-1} = U_1^{-1} \cdot U_2^{-1} \dots U_m^{-1} \cdot L_m^{-1} \cdot L_{m-1}^{-1} \dots L_1^{-1} \quad (4.5)$$

O problema de atualização da inversa pode ser colocado da seguinte maneira: Seja a matriz básica  $B(mxm)$  dada por seus vetores colunas,

$$B = (a_1, a_2, \dots, a_{p-1}, a_p, a_{p+1}, \dots, a_m) ;$$

Seja  $a_p$  o vetor coluna de  $B$ , que será substituído pelo vetor  $a_q$ , não pertencente à base, formando assim, a matriz  $\bar{B}$  ( $m \times m$ ) dada por:

$$\bar{B} = (a_1, a_2, \dots, a_{p-1}, a_q, a_{p+1}, \dots, a_m)$$

O método de atualização que denominamos aqui de Forrest-Tomlin, é devido a Brayton et alii<sup>[5]</sup>, e é também conhecido como método pseudo-LU. Nele partimos com a matriz  $B^{-1}$  na forma dada por (4.5), mas no entanto a forma triangular não se mantém durante o processo; mantendo somente a forma do produto de fatores elementares.

A expressão (4.5) pode ser escrita da seguinte maneira:

$$B^{-1} = U^{-1} \cdot L^{-1} ;$$

dessa forma temos que:

$$L^{-1} \cdot B = U ,$$

com  $U$  triangular superior. Por outro lado temos que  $(L^{-1}, \bar{B})$  não será necessariamente triangular superior, pois  $(L^{-1} \cdot a_q)$  pode ser um vetor qualquer. Seja:

$$\alpha^p = L^{-1} \cdot a_q ;$$

podemos escrever:

$$L^{-1} \cdot B = (L^{-1} \cdot a_1, L^{-1} \cdot a_2, \dots, L^{-1} \cdot a_p, L^{-1} \cdot a_{p+1}, \dots, L^{-1} \cdot a_n)$$

$$L^{-1} \cdot \bar{B} = (L^{-1} \cdot a_1, L^{-1} \cdot a_2, \dots, L^{-1} \cdot a_{p-1}, L^{-1} \cdot a_q, L^{-1} \cdot a_{p+1}, \dots, L^{-1} \cdot a_m)$$

$$L^{-1} \bar{B} = \begin{array}{|c|c|c|} \hline & 1 & 2 \\ \hline & & 3 \\ \hline & & & \\ \hline \end{array} \rightarrow p\text{-ésima linha} \quad (4.6)$$

O objetivo do método de Forrest-Tomlin é definir matrizes de transformações elementares, que aplicadas a  $L^{-1} \bar{B}$  a tornem triangular superior, sem contudo, modificar os elementos nas regiões 1, 2 e 3 da matriz dada por (4.6).

A primeira transformação executada é a multiplicação de  $L^{-1} \bar{B}$  à direita por uma matriz de permutação  $Q$ , que transfira a coluna  $\alpha^p$  para a posição  $m$ , deslocando as colunas  $p+1, \dots, m$  uma posição à esquerda; seja:

$$H = L^{-1} \bar{B} \cdot Q = \begin{array}{|c|c|c|} \hline & p & m \\ \hline & 1 & 2 \\ \hline & & 3 \\ \hline & & & \\ \hline \end{array} \quad (4.7)$$

$\downarrow$   
 $\alpha^p$

A matriz  $H$  é uma matriz de Hessenberg superior, e pode ser transformada em uma matriz triangular superior de diversas maneiras. O método de Forrest-Tomlin adota a estratégia de eliminar em  $H$ , os elementos da linha  $p$ , nas colunas  $p$  a  $m-1$ , para obter uma matriz triangular superior permutada.

Suponhamos que as modificações em  $H$  descritas no

parágrafo anterior sejam provocadas por uma matriz que denominaremos de  $R^{-1}$ ; - mais adiante especificaremos que  $\bar{e}$  é  $R^{-1}$ ; seja:

$$V = R^{-1} \cdot H = \begin{array}{|c|c|c|} \hline & 1 & 2 \\ \hline & & 3 \\ \hline & & & \\ \hline \end{array} \quad (4.8)$$

↓  
 $R^{-1} \cdot \alpha^p$

$V$  é uma matriz triangular superior permutada. Aplicando a permutação  $Q^{-1}$  nas linhas de  $V$ , teremos uma matriz completamente triangular superior  $\bar{U}$ . A permutação  $Q^{-1}$  deslocará a  $p$ -ésima linha para a posição  $m$ , deslocando as linhas  $p+1$  a  $m$  uma posição acima

$$\bar{U} = Q^{-1} V = \begin{array}{|c|c|c|} \hline & 1 & 2 \\ \hline & & 3 \\ \hline & & & \\ \hline \end{array} \quad (4.9)$$

De (4.7) podemos escrever:

$$\bar{B} = L \cdot H \cdot Q^{-1}$$

invertendo teremos:

$$\bar{B}^{-1} = Q \cdot H^{-1} \cdot L^{-1} \quad (4.10)$$

De (4.8) podemos escrever

$$H = R \cdot V ;$$

invertendo teremos:

$$H^{-1} = V^{-1} . R^{-1} \quad (4.11)$$

Substituindo (4.11) em (4.10) teremos  $B^{-1}$  dada por:

$$\bar{B}^{-1} = Q . V^{-1} . R^{-1} . L^{-1} \quad (4.12)$$

De (4.9) podemos escrever:

$$V = Q . \bar{U} ;$$

invertendo teremos:

$$V^{-1} = \bar{U}^{-1} . Q^{-1} \quad (4.13)$$

Substituindo (4.13) em (4.12) teremos também  $\bar{B}^{-1}$  dada por:

$$\bar{B}^{-1} = Q . \bar{U}^{-1} . Q^{-1} . R^{-1} . L^{-1} \quad (4.14)$$

Deveremos agora obter a forma do produto de  $\bar{B}^{-1}$ . Se obtivermos a forma do produto de  $\bar{B}$ , a forma do produto de  $\bar{B}^{-1}$  é imediata, uma vez que, inverter matrizes elementares é uma operação simples. A expressão (4.14) pode ser tomada como:

$$\bar{B} = L . R . Q . \bar{U} . Q^{-1} \quad (4.15)$$

Na tentativa de sermos mais explícitos faremos antes de prosseguirmos, algumas observações sobre as permutações  $Q$  e  $Q^{-1}$  utilizadas durante o processo.

Seja  $Q$  a permutação que leva a coluna  $p$  da matriz,  $\bar{a}$  qual está sendo aplicada, para a posição  $m$ , deslocando as colunas  $p+1$  a  $m$  uma posição  $\bar{a}$  esquerda. Neste caso  $Q$  é aplicada  $\bar{a}$  direita da matriz.

Seja  $A$  uma matriz ( $m \times m$ ) dada por seus vetores colunas, e apliquemos a permutação  $Q$  a  $A$  :

$$A.Q = [\bar{a}_1, \dots, \bar{a}_{p-1}, \bar{a}_p, \bar{a}_{p+1}, \dots, \bar{a}_m].Q = [\bar{a}_1, \dots, \bar{a}_{p-1}, \bar{a}_{p+1}, \dots, \bar{a}_m, \bar{a}_p]$$

A permutação  $Q^{-1}$  quando aplicada à direita de  $A$  efetuará a seguinte modificação em  $A$  :

$$A.Q^{-1} = [\bar{a}_1, \dots, \bar{a}_{p-1}, \bar{a}_p, \bar{a}_{p+1}, \dots, \bar{a}_m].Q^{-1} = [\bar{a}_1, \dots, \bar{a}_{p-1}, \bar{a}_m, \bar{a}_p, \bar{a}_{p+1}, \dots, \bar{a}_{m-1}]$$

Quando as mesmas permutações são aplicadas à esquerda da matriz  $A$  ocorrem as seguintes modificações nas linhas de  $A$  : (tomemos agora  $A$  dada por suas linhas)

$$Q.A = Q. \begin{bmatrix} \bar{a}_1 \\ \vdots \\ \bar{a}_{p-1} \\ \bar{a}_p \\ \bar{a}_{p+1} \\ \vdots \\ \bar{a}_{m-1} \\ \bar{a}_m \end{bmatrix} = \begin{bmatrix} \bar{a}_1 \\ \vdots \\ \bar{a}_{p-1} \\ \bar{a}_m \\ \bar{a}_p \\ \bar{a}_{p+1} \\ \vdots \\ \bar{a}_{m-1} \end{bmatrix} ; \quad Q^{-1}.A = Q^{-1} \begin{bmatrix} \bar{a}_1 \\ \vdots \\ \bar{a}_{p-1} \\ \bar{a}_p \\ \bar{a}_{p+1} \\ \vdots \\ \bar{a}_{m-1} \\ \bar{a}_m \end{bmatrix} = \begin{bmatrix} \bar{a}_1 \\ \vdots \\ \bar{a}_{p-1} \\ \bar{a}_{p+1} \\ \vdots \\ \bar{a}_m \\ \bar{a}_p \end{bmatrix}$$

Voltemos agora à expressão (4.15) para a partir daí conseguirmos a forma do produto de  $\bar{B}$ . Podemos verificar no entanto, que a forma do produto de  $L$ , já a temos, através da forma do produto de  $B$ ;  $R$  é uma matriz elementar por construção, como veremos; portanto só nos falta a decomposição de  $Q.U.Q^{-1}$  num produto de matrizes elementares para termos a forma do produto de  $\bar{B}$ .

Tomemos então  $\bar{U}$ . Segundo o desenvolvimento do método temos que  $\bar{U}$  é dada por:

$$\bar{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1p-1} & u_{1p+1} & \cdots & \bar{\alpha}_1^p \\ 0 & u_{22} & & u_{2p-1} & u_{2p+1} & \cdots & \bar{\alpha}_2^p \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & & u_{p-1,p-1} & u_{p-1,p+1} & \cdots & \bar{\alpha}_{p-1}^p \\ 0 & 0 & & 0 & u_{p+1,p+1} & \cdots & \bar{\alpha}_{p+1}^p \\ & & & & & & \vdots \\ & & & & & & \bar{\alpha}_m^p \\ 0 & 0 & & 0 & 0 & & \bar{\alpha}_p^p \end{bmatrix}$$

Como  $\bar{U}$  é triangular superior pode ser decomposta da seguinte forma:

$$\bar{U} = \bar{U}_m \cdot \bar{U}_{m-1} \cdots \bar{U}_{p+1} \cdot \bar{U}_p \cdot \bar{U}_{p-1} \cdots \bar{U}_1 \quad (4.15.A)$$

onde  $\bar{U}_m$  é dada por:

$$\bar{U}_m = \begin{bmatrix} 1 & & & & & & \bar{\alpha}_1^p \\ & 1 & & & & & \vdots \\ & & & & & & \vdots \\ & & & & & & \bar{\alpha}_{p-1}^p \\ & & & & & & \bar{\alpha}_{p+1}^p \\ & & & & & & \vdots \\ & & & & 1 & & \bar{\alpha}_m^p \\ & & & & & & \bar{\alpha}_p^p \end{bmatrix}$$

$\bar{U}_j$  para  $j = p, p+1 \dots m-1$  é dada por:

$$\bar{U}_j = \begin{bmatrix} 1 & 0 & \dots & u_{1,j+1} & 0 \\ 0 & 1 & & u_{2,j+1} & 0 \\ & & & \vdots & \vdots \\ & & & u_{j-1,j+1} & 0 \\ & & & u_{j+1,j+1} & 0 \\ & & & \vdots & \vdots \\ & & & 0 & 1 \end{bmatrix} ;$$

↓  
j-ésima coluna

e  $\bar{U}_j$  para  $j = 1, \dots, p-1$  é dada por:

$$\bar{U}_j = \begin{bmatrix} 1 & u_{1j} & 0 & 0 \\ 0 & u_{2j} & 0 & 0 \\ & \vdots & \vdots & \vdots \\ 0 & u_{jj} & 0 & 0 \\ & \vdots & 1 & 0 \\ & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

↓  
j-ésima coluna

Cabe observar que estamos procurando obter a forma do produto de  $\bar{B}$  a partir da forma do produto de  $B$ . Isto é, nos interessa sobretudo, que a forma do produto de  $\bar{B}$  seja derivada da forma do produto de  $B$  de maneira simples e direta. Com este objetivo vamos tentar associar a forma do produto de  $\bar{U}$  com a forma do produto de  $U$  dada por (4.5.A).

$$\text{Seja } U = U_m \dots U_1 ,$$



onde  $U_{j+1}$  é dada por:

$$U_{j+1} = \begin{bmatrix} 1 & 0 & u_{1,j+1} & 0 \\ & & u_{2,j+1} & 0 \\ & & \vdots & \vdots \\ 0 & 1 & u_{p-1,j+1} & 0 \\ & & u_{p,j+1} & 0 \\ & & u_{p+1,j+1} & 0 \\ & & \vdots & \vdots \\ & & u_{j+1,j+1} & 0 \\ & & \vdots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

↓  
j+1-ésima coluna

Seja  $\tilde{U}_{j+1}$  a matriz elementar idêntica a  $U_{j+1}$ , somente com o elemento  $u_{pj+1}$  anulado.

$$\tilde{U}_{j+1} = \begin{bmatrix} 1 & 0 & u_{1,j+1} & 0 \\ & & u_{2,j+1} & 0 \\ & & \vdots & \vdots \\ 0 & 0 & u_{p-1,j+1} & 0 \\ & & 0 & 0 \\ & & u_{p+1,j+1} & 0 \\ & & \vdots & \vdots \\ & & u_{j+1,j+1} & 0 \\ & & \vdots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

↓                                  ↓  
p-ésima                                  j+1-ésima  
coluna                                  coluna

Multiplicando  $\tilde{U}_{j+1}$  por  $Q$  à direita temos:

$$\tilde{U}_{j+1} \cdot Q = \begin{bmatrix} 1 & u_{1,j+1} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ u_{p-1,j+1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ u_{p+1,j+1} & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ u_{j,j+1} & 0 & 0 & 0 \\ u_{j+1,j+1} & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

↑  
j-ésima coluna

Multiplicando  $\tilde{U}_{j+1} \cdot Q$  à esquerda por  $Q^{-1}$  temos:

$$Q^{-1} \tilde{U}_{j+1} Q = \begin{bmatrix} 1 & u_{1,j+1} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & u_{p-1,j+1} & 0 & 0 \\ 0 & u_{p+1,j+1} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & u_{j+1,j+1} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & 1 & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix} = \tilde{U}_j$$

↓  
j-ésima  
coluna

Portanto temos que:

$$\bar{U}_j = Q^{-1} \cdot \tilde{U}_{j+1} Q \quad ; \quad j = p, \dots, m-1 \quad (4.16)$$

Isto é conseguimos definir  $\bar{U}_j$  através de  $\tilde{U}_{j+1}$ , que por sua vez está associada a  $U_{j+1}$ .

Tomemos agora  $\bar{U}_j$ ,  $j = 1, \dots, p-1$ . Verificamos que  $\bar{U}_j = U_j$ ,  $j = 1, \dots, p-1$ . Por outro lado também podemos verificar que  $\bar{U}_j = Q^{-1} \bar{U}_j Q$ ,  $j = 1, \dots, p-1$ . Seja:

$$\bar{U}_j = \begin{bmatrix} 1 & & & & 0 \\ & u_{1j} & & & 0 \\ & \vdots & & & \vdots \\ & u_{jj} & & & \vdots \\ 0 & & & 1 & 0 \\ & & & \vdots & \vdots \\ 0 & & & 0 & 1 \end{bmatrix} ;$$

↑  
p-ésima coluna

aplicando  $Q$  e  $Q^{-1}$  a  $\bar{U}_j$  temos:

$$Q^{-1} \cdot \bar{U}_j = \begin{bmatrix} 1 & u_{1j} & 0 & \dots & 0 \\ & u_{2j} & & & \\ & \vdots & & & \\ & u_{jj} & & & \\ 0 & & 0 & & 1 \\ 0 & & 1 & & 0 \end{bmatrix} ; \quad Q^{-1} \cdot \bar{U}_j \cdot Q = \begin{bmatrix} 1 & & & & 0 \\ & u_{1j} & & & 0 \\ & u_{2j} & & & 0 \\ & \vdots & & & \vdots \\ & u_{jj} & & & 0 \\ & & & & 1 \end{bmatrix} = \bar{U}_j$$

↑  
p-ésima coluna

Isto é:

$$Q^{-1} \cdot \bar{U}_j \cdot Q = \bar{U}_j = U_j, \quad \text{para } j = 1, \dots, p-1 \quad (4.17)$$

Substituindo (4.16) e (4.17) em (4.15.A) temos:

$$\begin{aligned}\bar{U} &= \bar{U}_m \cdot Q^{-1} \cdot \tilde{U}_m \cdot Q \dots Q^{-1} \cdot \tilde{U}_{p+1} \cdot Q \cdot Q^{-1} \cdot \bar{U}_{p-1} \cdot Q \dots Q^{-1} \cdot \bar{U}_1 \cdot Q \\ \bar{U} &= \bar{U}_m \cdot Q^{-1} \cdot \tilde{U}_m \dots \tilde{U}_{p+1} \cdot U_{p-1} \dots U_1 \cdot Q\end{aligned}\quad (4.18)$$

Substituindo (4.18) em (4.15) temos:

$$\bar{B} = L \cdot R \cdot Q \cdot \bar{U}_m \cdot Q^{-1} \cdot \tilde{U}_m \dots \tilde{U}_{p+1} \cdot U_{p-1} \dots U_1 \cdot Q \cdot Q^{-1}$$

Portanto

$$\bar{B} = L \cdot R \cdot Q \cdot \bar{U}_m \cdot Q^{-1} \cdot \tilde{U}_m \cdot Q^{-1} \cdot \tilde{U}_m \dots \tilde{U}_{p+1} \cdot U_{p-1} \dots U_1$$

Chamando  $Q \cdot \bar{U}_m \cdot Q^{-1}$  de  $W$  temos que:

$$W = Q \cdot \bar{U}_m \cdot Q^{-1} = \begin{bmatrix} 1 & & & \bar{\alpha}_1^p & & & & \\ & 1 & & \vdots & & & & \\ & & & \bar{\alpha}_{p-1}^p & & & & \\ & & & \bar{\alpha}_p^p & & & & \\ & & & \bar{\alpha}_{p+1}^p & & & & \\ & & & \vdots & & & & \\ & & & \bar{\alpha}_m^p & & & & 1 \end{bmatrix}$$

↑  
p-ésima coluna

e

$$\bar{B} = L \cdot R \cdot W \cdot \tilde{U}_m \dots \tilde{U}_{p+1} \cdot U_{p-1} \dots U_1$$

Obtivemos portanto a forma do produto para  $\bar{B}$ , a partir de (4.14):

$$\bar{B} = L \cdot R \cdot Q \cdot U \cdot Q^{-1}$$

L tem a seguinte forma do produto:

$$L = L_1 \cdot L_2 \cdot \dots \cdot L_m$$

$R^{-1}$  por sua construção como veremos a seguir, é uma matriz elementar, portanto R também o será.

A forma do produto de  $Q \bar{U} Q^{-1}$  pode ser obtida da forma do produto de U, da seguinte forma:

$$\text{Seja } U = U_m \cdot \dots \cdot U_{p+1} \cdot U_p \cdot U_{p-1} \cdot \dots \cdot U_1$$

- (a) eliminar a p-ésima matriz elementar ( $U_p$ ), da decomposição de U ;
- (b) anular nas matrizes elementares  $U_{p+1}, \dots, U_m$  os respectivos elementos:  $u_{pp+1}, \dots, u_{pm}$ , produzindo as matrizes elementares  $\tilde{U}_{p+1}, \dots, \tilde{U}_m$  ;
- (c) acrescentar ao produto a matriz elementar W, definida pelo vetor  $\bar{\alpha}^p = R^{-1} \cdot \alpha^p$  na p-ésima coluna.

A expressão (4.14) pode ser escrita como:

$$\bar{B} = L_1 \cdot \dots \cdot L_m \cdot R \cdot W \cdot \tilde{U}_m \cdot \dots \cdot \tilde{U}_{p+1} \cdot U_{p+1} \cdot \dots \cdot U_1$$

Consequentemente  $\bar{B}^{-1}$  é dada por:

$$\bar{B}^{-1} = U_1^{-1} \cdot \dots \cdot U_{p-1}^{-1} \cdot \tilde{U}_{p+1}^{-1} \cdot \dots \cdot \tilde{U}_m^{-1} \cdot W^{-1} \cdot R^{-1} \cdot L_m^{-1} \cdot \dots \cdot L_1^{-1}$$

Vamos agora esclarecer como se obtêm  $R^{-1}$ . Convém lembrar que  $R^{-1}$  foi a matriz utilizada para zerar em H os elementos da linha p, nas colunas p a m-1. Lembremos também que:

$$H = L^{-1} \cdot \bar{B} \cdot Q = (L^{-1} \cdot a_1, \dots, L^{-1} \cdot a_{p-1}, L^{-1} \cdot a_{p+1}, L^{-1} \cdot a_{p+2}, \dots, L^{-1} \cdot a_{pm}, L^{-1} \cdot a_q) =$$

$$= (h_1, \dots, h_m, \alpha^p)$$

e

$$U = L^{-1} \cdot B = (L^{-1} \cdot a_1, \dots, L^{-1} \cdot a_{p-1}, L^{-1} \cdot a_p, L^{-1} \cdot a_{p+1}, \dots, L^{-1} \cdot a_{pm}) =$$

$$= (u_1, u_2, \dots, u_m)$$

Se  $R^{-1}$  fosse aplicado a  $U$  zeraria os elementos  $u_{pp+1}, u_{pp+2}, \dots, u_{pm}$  que são respectivamente iguais a  $h_{pp}, h_{pp+1}, \dots, h_{pm-1}$ . Usaremos este fato, para deduzir  $R^{-1}$ .

A expressão de  $U$  com os elementos  $u_{pi} = 0$ ,  $i = p+1, \dots, m$  denominaremos  $R^{-1} \cdot U$  e é dada por:

$$R^{-1} \cdot U = U - e_p \cdot \bar{u}^t \quad (4.19)$$

onde  $\bar{u}^t = (0, \dots, 0, u_{pp+1}, \dots, u_{pm})$

e  $e_p$  é o  $p$ -ésimo vetor coluna da matriz  $I(m \times m)$ . Podemos verificar isto com um exemplo:

Seja

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{24} \\ & u_{22} & u_{23} & u_{24} \\ & & u_{33} & u_{34} \\ & & & u_{44} \end{bmatrix}$$

suponhamos  $p = 2$

$$\bar{u}^t = (0, 0, u_{23}, u_{24}) \quad ; \quad e_p = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$e_p \cdot \bar{u}^t = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & u_{23} & u_{24} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$U - e_p \cdot \bar{u}^t = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ & u_{22} & 0 & 0 \\ & & u_{33} & u_{34} \\ & & & u_{44} \end{bmatrix}$$

Como  $U$  é inversível (4.19) pode ser escrita como:

$$R^{-1} \cdot U = U - e_p \cdot \bar{u}^t \cdot U^{-1} \cdot U$$

$$R^{-1} \cdot U = (I - e_p \cdot \bar{u}^t \cdot U^{-1}) \cdot U$$

$$R^{-1} = I - e_p \cdot \bar{u}^t \cdot U^{-1}$$

Chamando  $r^t = -\bar{u}^t \cdot U^{-1} = (0, \dots, 0, r_{p+1}, \dots, r_m)$ , a matriz  $R^{-1}$  será uma matriz linha elementar, e dada por:

$$R^{-1} = \begin{bmatrix} 1 & & & & & & \\ \vdots & & & & & & \\ 0 & \dots & 0 & 1 & r_{p+1} & \dots & r_m \\ & & & & 1 & & \\ & & & & & & 1 \end{bmatrix}$$

Na tentativa de sermos mais explícitos aplicaremos as transformações do método de Forrest-Tomlin, numa matriz genérica (5x5).

Exemplo:

Seja  $B$  uma matriz (5x5) dada por seus vetores colunas

$$B = [a_1, a_2, a_3, a_4, a_5]$$

Seja  $\bar{B}$  a matriz que se obtém substituindo o vetor  $a_3$  de  $B$  pelo vetor  $a_q$

$$\bar{B} = [a_1, a_2, a_3, a_4, a_5]$$

Faremos, para este caso particular, todo o desenvolvimento do método de Forrest-Tomlin, com o intuito de esclarecermos a obtenção da forma do produto de  $\bar{B}^{-1}$ .

Seja  $L^{-1}$  tal que:

$$U = L^{-1}B = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} & u_{15} \\ & u_{22} & u_{23} & u_{24} & u_{25} \\ & & u_{33} & u_{34} & u_{35} \\ & & & u_{44} & u_{45} \\ & & & & u_{55} \end{bmatrix} \quad (4.20)$$

Fazendo  $\alpha^p = L^{-1} \cdot a_q$  teremos



$$L^{-1} \cdot \bar{B} = \begin{bmatrix} u_{11} & u_{12} & \alpha_1^p & u_{14} & u_{15} \\ & u_{22} & \alpha_2^p & u_{24} & u_{25} \\ & & \alpha_3^p & u_{34} & u_{35} \\ & & \alpha_4^p & u_{44} & u_{45} \\ & & \alpha_5^p & & u_{55} \end{bmatrix} \quad (4.21)$$

Nesse caso ( $p=3$ ).

Aplicando a permutação  $Q$  a (4.21) temos:

$$H = L^{-1} \cdot \bar{B} \cdot Q = \begin{bmatrix} u_{11} & u_{12} & u_{14} & u_{15} & \alpha_1^p \\ & u_{22} & u_{24} & u_{25} & \alpha_2^p \\ & & u_{34} & u_{35} & \alpha_3^p \\ & & u_{44} & u_{45} & \alpha_4^p \\ & & & u_{55} & \alpha_5^p \end{bmatrix}$$

Seja  $R^{-1}$  a matriz de transformação que zerará os elementos  $u_{34}$  e  $u_{35}$ , isto é,  $u_{pp+1}, \dots, u_{pm}$ .  $R^{-1}$  é tal que além de zerar os elementos especificados, somente modifica os elementos da última coluna de  $H$ .

$$\text{Seja } R^{-1} \cdot \begin{bmatrix} \alpha_1^p \\ \alpha_2^p \\ \alpha_3^p \\ \alpha_4^p \\ \alpha_5^p \end{bmatrix} = \begin{bmatrix} \bar{\alpha}_1^p \\ \bar{\alpha}_2^p \\ \bar{\alpha}_3^p \\ \bar{\alpha}_4^p \\ \bar{\alpha}_5^p \end{bmatrix} \quad \text{onde } \bar{\alpha}_i^p = \alpha_i^p, \quad i=1, \dots, 5; \\ i \neq 3 \\ \bar{\alpha}_3^p \neq \alpha_3^p$$

Teremos então:

$$V = R^{-1} \cdot H \quad \begin{bmatrix} u_{11} & u_{12} & u_{14} & u_{15} & \bar{\alpha}_1^p \\ & u_{22} & u_{24} & u_{25} & \bar{\alpha}_2^p \\ & & 0 & 0 & \bar{\alpha}_3^p \\ & & & u_{44} & u_{45} & \bar{\alpha}_4^p \\ & & & & u_{55} & \bar{\alpha}_5^p \end{bmatrix} \quad (4.22)$$

Aplicando  $Q^{-1}$  a  $V$  obtemos:

$$\bar{U} = Q^{-1} \cdot V = \begin{bmatrix} u_{11} & u_{12} & u_{14} & u_{15} & \bar{\alpha}_1^p \\ & u_{22} & u_{24} & u_{25} & \bar{\alpha}_2^p \\ & & u_{44} & u_{45} & \bar{\alpha}_4^p \\ & & & u_{55} & \bar{\alpha}_5^p \\ & & & & \bar{\alpha}_3^p \end{bmatrix}$$

$$\bar{U} = \bar{U}_5 \cdot \bar{U}_4 \cdot \bar{U}_3 \cdot \bar{U}_2 \cdot \bar{U}_1$$

onde

$$\bar{U}_5 = \begin{bmatrix} 1 & & & & \alpha_1^p \\ & 1 & & & \alpha_2^p \\ & & 1 & & \alpha_4^p \\ & & & 1 & \alpha_5^p \\ & & & & \alpha_3^p \end{bmatrix}; \bar{U}_4 = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{matrix} u_{15} \\ u_{25} \\ u_{45} \\ u_{55} \\ 1 \end{matrix}; \bar{U}_3 = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{matrix} u_{14} \\ u_{24} \\ u_{44} \\ 1 \\ 1 \end{matrix}$$

$$\bar{U}_2 = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \begin{matrix} u_{12} \\ u_{22} \\ 1 \\ 1 \\ 1 \end{matrix}; \bar{U}_1 = \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix}$$

Por (4.15) temos  $\bar{B}$  dada por:

$$\bar{B} = L \cdot R \cdot Q \cdot \bar{U} \cdot Q^{-1} \quad (4.24)$$

onde  $Q \cdot \bar{U} \cdot Q^{-1}$  pode ser escrita como:

$$Q \cdot \bar{U} \cdot Q^{-1} = Q \cdot \bar{U}_5 \cdot \bar{U}_4 \cdot \bar{U}_3 \cdot \bar{U}_2 \cdot \bar{U}_1 \cdot Q^{-1};$$

que pode ser escrita como:

$$Q \cdot \bar{U} \cdot Q^{-1} = Q \cdot \bar{U}_5 \cdot Q^{-1} \cdot Q \cdot \bar{U}_4 \cdot Q^{-1} \cdot Q \cdot \bar{U}_3 \cdot Q^{-1} \cdot Q \cdot \bar{U}_2 \cdot Q^{-1} \cdot Q \cdot \bar{U}_1 \cdot Q^{-1} \quad (4.25)$$

onde:

$$Q \cdot \bar{U}_5 \cdot Q^{-1} = \begin{bmatrix} 1 & & & & \\ & \alpha_1^p & & & \\ & & 1 & & \\ & & & \alpha_2^p & \\ & & & & \alpha_3^p \\ & & & & & \alpha_4^p & 1 \\ & & & & & & & \alpha_5^p & \\ & & & & & & & & & 1 \end{bmatrix} ; \quad Q \cdot \bar{U}_4 \cdot Q^{-1} = \begin{bmatrix} 1 & & & & & & & & & & u_{15} \\ & 1 & & & & & & & & & u_{25} \\ & & 1 & & & & & & & & 0 \\ & & & 1 & & & & & & & u_{45} \\ & & & & 1 & & & & & & u_{55} \end{bmatrix}$$

$$Q \cdot \bar{U}_3 \cdot Q^{-1} = \begin{bmatrix} 1 & & & & & & & & & & \\ & 1 & & & & & & & & & \\ & & 1 & & & & & & & & \\ & & & 1 & & & & & & & \\ & & & & 1 & & & & & & \\ & & & & & 1 & & & & & \\ & & & & & & 1 & & & & \\ & & & & & & & 1 & & & \\ & & & & & & & & 1 & & \\ & & & & & & & & & 1 & \\ & & & & & & & & & & 1 \end{bmatrix} ; \quad Q \cdot \bar{U}_2 \cdot Q^{-1} = \begin{bmatrix} 1 & & & & & & & & & & \\ & 1 & & & & & & & & & \\ & & 1 & & & & & & & & \\ & & & 1 & & & & & & & \\ & & & & 1 & & & & & & \\ & & & & & 1 & & & & & \\ & & & & & & 1 & & & & \\ & & & & & & & 1 & & & \\ & & & & & & & & 1 & & \\ & & & & & & & & & 1 & \\ & & & & & & & & & & 1 \end{bmatrix} ;$$

$$Q \cdot \bar{U}_1 \cdot Q^{-1} = \begin{bmatrix} u_{11} & & & & & & & & & & \\ & 1 & & & & & & & & & \\ & & 1 & & & & & & & & \\ & & & 1 & & & & & & & \\ & & & & 1 & & & & & & \\ & & & & & 1 & & & & & \\ & & & & & & 1 & & & & \\ & & & & & & & 1 & & & \\ & & & & & & & & 1 & & \\ & & & & & & & & & 1 & \\ & & & & & & & & & & 1 \end{bmatrix}$$

comparando a decomposição de  $Q \cdot \bar{U} \cdot Q^{-1}$  com a decomposição de  $U$  vemos que:

$$Q \cdot \bar{U}_4 \cdot Q^{-1} = \tilde{U}_5 \quad \text{onde} \quad \tilde{U}_5 \quad \text{é} \quad U_5 \quad \text{com o elemento} \quad u_{35} = 0 ;$$

$$Q \cdot \bar{U}_3 \cdot Q^{-1} = \tilde{U}_4 \quad \text{onde} \quad \tilde{U}_4 \quad \text{é} \quad U_4 \quad \text{com o elemento} \quad u_{34} = 0 ;$$

$$Q \cdot \bar{U}_2 \cdot Q^{-1} = U_2 \quad \text{e} \quad Q \cdot \bar{U}_1 \cdot Q^{-1} = U_1$$

Fazendo

$$W = Q \cdot \bar{U}_5 \cdot Q^{-1}$$

podemos escrever (4.25) como:

$$Q \cdot \bar{U} \cdot Q^{-1} = W \cdot \tilde{U}_5 \cdot \tilde{U}_4 \cdot U_2 \cdot U_1 \quad (4.26)$$

substituindo (4.26) em (4.24) temos:

$$\bar{B} = L \cdot R \cdot W \cdot \tilde{U}_5 \cdot \tilde{U}_4 \cdot U_2 \cdot U_1 ;$$

e teremos portanto

$$\bar{B}^{-1} = U_1^{-1} \cdot U_2^{-1} \cdot \tilde{U}_4^{-1} \cdot \tilde{U}_5^{-1} \cdot W^{-1} \cdot R^{-1} \cdot L^{-1}$$

O processo pode ser continuado a partir de (4.12) usando o fator  $V \cdot Q^{-1}$  em lugar de  $U$ . Para darmos a continuação do processo consideremos a seguinte mudança de notação:

$$\bar{B} = L \cdot R \cdot V \cdot Q^{-1} \quad \bar{B} \text{ é modificada para}$$

$$\bar{B}_1 = L \cdot R_1 \cdot V \cdot Q_1^{-1}, \text{ fazendo } R_1 = R, Q_1^{-1} = Q^{-1}.$$

Tomando  $V \cdot Q_1^{-1} = V^{(1)}$  teremos:

$$\bar{B}_1 = L \cdot R_1 \cdot V^{(1)}$$

Na iteração  $k$  teremos portanto:

$$\bar{B}_k = L \cdot R_1 \dots R_k \cdot V^{(k)}$$

ou alternativamente

$$\bar{B}_k = L \cdot R_1 \dots R_k \cdot V^{(k-1)} \cdot Q_k^{-1}$$

$$(\bar{B}_k)^{-1} = Q_k \cdot (V^{(k-1)})^{-1} \cdot R_k^{-1} \dots R_1^{-1} \cdot L^{-1}$$

Ao iniciarmos o método de Forrest-Tomlin temos as filas e armazenadas em duas estruturas distintas: uma estrutura na qual armazenamos as filas e correspondentes às matrizes triangulares superiores  $U_i$ ; e outra estrutura na qual armazenamos as filas e correspondentes às matrizes triangulares inferiores  $L_i$ . Na continuação do método faz-se o armazenamento de cada  $W^{-1}$  gerada no grupo das filas e  $U_i$ ; e cada  $R^{-1}$  gerada é armazenada no grupo das filas e  $L_i$ .

A seguir daremos a descrição do algoritmo simplex com a atualização de Forrest-Tomlin, que denominaremos de simples LU. Cabe observar que o método de Forrest-Tomlin não mantém a forma LU da matriz inversa da base, mas conserva no entanto a forma do produto de matrizes elementares.

#### 4.4. O simplex LU

A forma do produto da inversa do simplex revisado com a atualização da inversa de Forres-Tomlin <sup>[10]</sup> pode ser descrita da seguinte forma:

Suponhamos  $B^{-1}$  dada como um produto de fatores triangulares e elementares:

$$B^{-1} = U_1^{-1} \dots U_m^{-1} \cdot L_m^{-1} \dots L_1^{-1}$$

$U_i^{-1}$  é triangular superior e elementar;

$L_i^{-1}$  é triangular inferior e elementar;

$$i = 1, \dots, m$$

Inicializar  $k = 0$

Passo 1: Cálculo da solução básica atual

$$x_B = \bar{b} = B^{-1}b = U_1^{-1} \dots U_m^{-1} \cdot L_m^{-1} \dots L_1^{-1} \cdot b$$

Passo 2: Cálculo dos custos relativos

Calcular primeiramente

$$\pi = c_B^t \cdot B^{-1} = c_B^t \cdot U_1^{-1} \dots U_m^{-1} \cdot L_m^{-1} \dots L_1^{-1}$$

depois

$$(c_j - z_j) = (c_j - \pi \cdot a_j) \quad j \text{ tal que } x_j \text{ seja V.N.B.}$$

Passo 3: Escolha da variável que entra na base:

Seja  $x_q$  tal que

$$c_q - z_q = \min\{c_j - z_j\}, j \text{ tal que } x_j \text{ seja variável não básica}$$

Se  $c_q - z_q \geq 0$  a solução atual é ótima; pare caso contrário  $x_q$  entra na base.

Passo 4: Atualização do vetor que entra na base:

Isto é feito em duas etapas:

Calcular primeiramente

$$\alpha^p = L^{-1} \cdot a_q = L_m^{-1} \dots L_1^{-1} \cdot a_q$$

O vetor  $\alpha^p$  será utilizado na construção de  $R^{-1}$  mais adiante;

continuar a atualização do vetor que entra na base:

$$y_q = U_1^{-1} \dots U_m^{-1} \cdot \alpha^p$$

Isto é, foi calculado na verdade  $y_q = B^{-1} \cdot a_q = U_1^{-1} \dots U_m^{-1} \cdot L_m^{-1} \dots L_1^{-1} \cdot a_q$

Passo 5: Escolha da variável que deixa a base

Seja

$$\frac{\bar{b}_p}{y_{pq}} = \min \left\{ \frac{\bar{b}_i}{y_{iq}} \right\}, i \text{ tal que } y_{iq} > 0$$

Se não existe  $y_{iq} > 0$ , pare; o problema é ilimitado. Caso contrário  $x_p$  será a variável que deve deixar a base.



Passo 6: Atualização da inversa:

(1) Definição das matrizes elementares  $R^{-1}$  e  $W^{-1}$

$$\text{de } U = U_m \cdot U_{m-1} \cdots U_1$$

$$\text{tiramos } \bar{u}^t = (0, \dots, 0, u_{p \ p+1}, u_{p \ p+2}, \dots, u_{p \ m})$$

$$\underline{R^{-1} = I - e_p \cdot \bar{u}^t \cdot U^{-1}}$$

onde  $I$  é a matriz identidade

$e_p$  é o  $p$ -ésimo vetor da matriz identidade

$$U^{-1} = U_1^{-1} \cdot U_2^{-1} \cdots U_m^{-1}$$

Seja:

$$\bar{\alpha}^p = R^{-1} \cdot \alpha^p = \begin{bmatrix} \bar{\alpha}_1^p \\ \vdots \\ \bar{\alpha}_p^p \\ \vdots \\ \bar{\alpha}_m^p \end{bmatrix}$$

construir a matriz elementar  $W^{-1}$  com pivot  $\bar{\alpha}_p^p$

$$W^{-1} = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & \ddots & & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

↓  
coluna p

$$\text{onde } \begin{cases} w_i = -\frac{\bar{\alpha}_i^p}{\bar{\alpha}_p^p} & i = 1, \dots, m ; i \neq p \\ w_p = \frac{1}{\bar{\alpha}_p^p} \end{cases}$$

a expressão de  $W^{-1}$  pode ser dada por:

$$W^{-1} = I - \frac{1}{\alpha_p} (\bar{\alpha}^p - e_p) e_p^t$$

(2) Modificação da matriz  $U^{-1}$

$$U^{-1} = U_1^{-1} \dots U_{p-1}^{-1} \cdot U_p^{-1} \cdot U_{p+1}^{-1} \dots U_m^{-1}$$

é modificada para:

$$\bar{U}^{-1} = U_1^{-1} \dots U_{p-1}^{-1} \cdot \bar{U}_{p+1}^{-1} \dots \bar{U}_m^{-1} \cdot W^{-1}$$

$U_p^{-1}$  é eliminada do produto

$\bar{U}_j$ ,  $j = p+1, \dots, m$  é exatamente igual à matriz elementar  $U_j$  mas com o  $p$ -ésimo elemento na coluna que a define zerado.  $W^{-1}$  é incluída no produto de  $\bar{U}^{-1}$ .

A nova matriz básica obtida substituindo o vetor coluna  $a_p$  pelo vetor coluna  $a_q$ , tem a sua representação dada por:

$$\bar{B}^{-1} = (U_1^{-1} \dots U_{p-1}^{-1} \cdot \bar{U}_{p+1}^{-1} \dots \bar{U}_m^{-1} \cdot U^{-1}) \cdot (R^{-1} \cdot L_m^{-1} \dots L_1^{-1})$$

Agregar  $W^{-1}$  ao conjunto de filas eta  $U_i$ , e  $R^{-1}$  ao conjunto de filas eta  $L_i$ . Voltar ao passo 1.

A seguir daremos um exemplo de aplicação do método de Forrest-Tomlin.

Exemplo:

Problema:

$$3x_1 + 5x_2 \leq 15$$

$$5x_1 + 2x_2 \leq 10$$

$$x_1 \geq 0, \quad x_2 \geq 0$$

$$5x_1 + 3x_2 = Q'(x) \rightarrow \max!$$

Forma Padrão:

$$3x_1 + 5x_2 + x_3 = 15$$

$$5x_1 + 2x_2 + x_4 = 10$$

$$-5x_1 - 3x_2 = Q(x) \rightarrow \min!$$

Resolução pelo simplex normal

	$x_1$	$x_2$	$x_3$	$x_4$	
$x_3$	3	5	1	0	15
$x_4$	5	2	0	1	10
	-5	-3	0	0	$Q(x)$

1a. solução

$$x_1 = x_2 = 0 ; x_3 = 15 ; x_4 = 10$$

$$Q(x) = 0$$

	$x_1$	$x_2$	$x_3$	$x_4$	
$x_3$	0	19/5	1	-3/5	9
$x_1$	1	2/5	0	1/5	2
	0	-1	0	1	$Q(x)+10$

2a. solução

$$x_1 = 2 ; x_3 = 9 ; x_2 = x_4 = 0$$

$$Q(x) = -10$$

	$x_1$	$x_2$	$x_3$	$x_4$	
$x_2$	0	1	5/19	-3/19	45/19
$x_1$	1	0	-2/19	5/19	20/19
	0	0	5/19	16/19	$Q(x) + \frac{235}{19}$

3a. solução

$$x_1 = \frac{20}{19} ; x_2 = \frac{45}{19} ;$$

$$x_3 = x_4 = 0$$

$$Q(x) = -\frac{235}{19}$$

a solução é ótima.

Agora daremos a resolução do mesmo problema pelo simplex LU.

Resolução do mesmo problema pelo simplex LU

1.<sup>a</sup> iteração

Variáveis básicas:  $x_{B1} = x_3$  ,  $x_{B2} = x_4$  ;

Variáveis não básicas  $x_1$  ,  $x_2$

$$B = B^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} ; b = \begin{bmatrix} 15 \\ 10 \end{bmatrix} ; c_B^t = [0, 0] ,$$

que são respectivamente, a matriz básica; o vetor dos termos independentes, e os custos das variáveis básicas.

$$U = L = I ;$$

Passo 1: Cálculo da solução:

$$x_B = U^{-1} \cdot L^{-1} \cdot b = \begin{bmatrix} 15 \\ 10 \end{bmatrix} ;$$

Passo 2: Cálculo dos custos relativos

$$\pi = c_B^t \cdot U^{-1} \cdot L^{-1} = [0, 0]$$

$$c_1 - z_1 = -5 , c_2 - z_2 = -3$$

Passo 3: Escolha da variável que entra na base:

$x_1$  entra na base

Passo 4: Atualização do vetor que entra na base:

$$\alpha^p = L^{-1} \cdot a_1 = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

$$y_1 = U^{-1} \cdot \alpha^p = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

Passo 5: Escolha da variável que sai da base:

$x_4$  sai da base; isto é  $x_1$  entra no lugar de  $x_{B2}$ .

Passo 6: Atualização da base: ( $p=2$ )

$$\bar{u}^t = [0, 0]$$

$$R^{-1} = I, \quad \bar{\alpha}^p = R^{-1} \cdot \alpha^p = \begin{bmatrix} 3 \\ 5 \end{bmatrix}; \quad \bar{\alpha}^p \text{ é o elemento pivot.}$$

$W^{-1}$  é construída a partir de  $\bar{\alpha}^p$  e com  $\bar{\alpha}_2^p$  como pivot.

$$W^{-1} = \begin{bmatrix} 1 & -3/5 \\ 0 & 1/5 \end{bmatrix}$$

Obtenção da nova  $U^{-1}$  com a eliminação de  $U_2^{-1}$  e o adicionamento de  $W^{-1}$  no cálculo da nova  $U^{-1}$ , ( $U_N^{-1}$ ):

$$U^{-1} = U_1^{-1} \cdot U_2^{-1} = I \cdot I$$

$$U_N^{-1} = U_1^{-1} \cdot W^{-1} = \begin{bmatrix} 1 & -3/5 \\ 0 & 1/5 \end{bmatrix}$$

Nova atribuição:  $U^{-1} \leftarrow U_N^{-1}$ ;  $U_1^{-1} \leftarrow U_1^{-1}$ ;  $U_2^{-1} \leftarrow W^{-1}$

Seja:

$$L_N^{-1} = R^{-1} \cdot L^{-1} = I \cdot I = I$$

Nova atribuição:  $L^{-1} \leftarrow L_N^{-1}$

Teremos então:

$$B^{-1} = U^{-1} \cdot L^{-1} = \begin{bmatrix} 1 & -3/5 \\ 0 & 1/5 \end{bmatrix}; \quad B = \begin{bmatrix} 1 & 3 \\ 0 & 5 \end{bmatrix}$$

2ª Iteração:

Variáveis básicas:  $x_{B1} = x_3$  ;  $x_{B2} = x_1$

Variáveis não básicas:  $x_2$  e  $x_4$

Passo 1: Cálculo da solução atual:

$$x_B = B^{-1} \cdot b = \begin{bmatrix} 1 & -3/5 \\ 0 & 1/5 \end{bmatrix} \cdot \begin{bmatrix} 15 \\ 10 \end{bmatrix} = \begin{bmatrix} 9 \\ 2 \end{bmatrix}$$

Passo 2: Cálculo dos custos reduzidos

$$\pi = c_B^t \cdot B^{-1} = [0, -5] \cdot \begin{bmatrix} 1 & -3/5 \\ 0 & 1/5 \end{bmatrix} = [0, -1]$$

$$(c_4 - z_4) = 1 ; (c_2 - z_2) = -1 .$$

Passo 3 : escolha da variável que entra na base

$x_2$  entra na base.

Passo 4 : atualização do vetor que entra na base:

$$\alpha^p = L^{-1} \cdot a_2 = I \cdot a_2 = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

$$y_2 = U^{-1} \alpha^p = \begin{bmatrix} 1 & -3/5 \\ 0 & 1/5 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ 2 \end{bmatrix} = \begin{bmatrix} 19/5 \\ 2/5 \end{bmatrix}$$

Passo 5: escolha da variável que sai da base:

$x_3$  sai da base;  $x_2$  entra no lugar de  $x_{B1}$

Passo 6: Atualização da inversa: (p=1)

Temos que:

$$U^{-1} = \begin{bmatrix} 1 & -3/5 \\ 0 & 1/5 \end{bmatrix} ; \text{ então } U = \begin{bmatrix} 1 & 3 \\ 0 & 5 \end{bmatrix}$$

$$\bar{u}^t = [0, 3]$$

$$R^{-1} = I - e_1 \cdot \bar{u}^t \cdot u^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 3/5 \\ 0 & 0 \end{bmatrix}$$

$$\bar{\alpha}^p = R^{-1} \cdot \alpha^p = \begin{bmatrix} 1 & -3/5 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ 2 \end{bmatrix} = \begin{bmatrix} 19/5 \\ 2 \end{bmatrix}$$

Obtenção de  $W^{-1}$  a partir de  $\bar{\alpha}^p$  com elemento pivot  $\bar{\alpha}_1^p$

$$W^{-1} = \begin{bmatrix} 5/19 & 0 \\ -10/19 & 1 \end{bmatrix}$$

Obtenção da nova  $U^{-1}$  com a eliminação de  $U_1^{-1}$  e o anulamento do elemento  $u_{12}$  em  $U_2^{-1}$  e o adicionamento de  $W^{-1}$  no cálculo da nova  $U^{-1}$  ( $U_N^{-1}$ )

$$U^{-1} = U_1^{-1} \cdot U_2^{-1} = I \cdot \begin{bmatrix} 1 & -3/5 \\ 0 & 1/5 \end{bmatrix}$$

$$U_N^{-1} = \bar{U}_2^{-1} \cdot W^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1/5 \end{bmatrix} \cdot \begin{bmatrix} 5/19 & 0 \\ -10/19 & 1 \end{bmatrix} = \begin{bmatrix} 5/19 & 0 \\ -2/19 & 1/5 \end{bmatrix}$$

$\bar{U}_2^{-1}$  foi conseguida de  $U_2^{-1}$ , zerando o elemento  $u_{12}$ . Nova atribuição:

$$U^{-1} \leftarrow U_N^{-1} ; U_1^{-1} \leftarrow \bar{U}_2^{-1} ; U_2^{-1} \leftarrow W^{-1}$$

Acrescentando  $R^{-1}$  às matrizes elementares  $L_i$  temos:

$$L_N^{-1} = R^{-1} \cdot L^{-1} = R^{-1} \cdot I = \begin{bmatrix} 1 & -3/5 \\ 0 & 1 \end{bmatrix}$$

Nova atribuição:  $L^{-1} \leftarrow L_N^{-1}$

A inversa da nova matriz básica será dada por:

$$B^{-1} = U^{-1} \cdot L^{-1}$$

$$B^{-1} = \begin{bmatrix} 5/19 & 0 \\ -2/19 & 1/5 \end{bmatrix} \cdot \begin{bmatrix} 1 & -3/5 \\ 0 & 1 \end{bmatrix}$$

$$B^{-1} = \begin{bmatrix} 5/19 & -3/19 \\ -2/19 & 5/19 \end{bmatrix} ; \quad B = \begin{bmatrix} 5 & 3 \\ 2 & 5 \end{bmatrix}$$

3ª iteração:  $x_{B1} = x_2$  ;  $x_{B2} = x_1$

Passo 1: Cálculo da solução atual:

$$x_B = B^{-1} \cdot b = \begin{bmatrix} 5/19 & -3/19 \\ -2/19 & 5/19 \end{bmatrix} \cdot \begin{bmatrix} 15 \\ 10 \end{bmatrix} = \begin{bmatrix} 45/19 \\ 20/19 \end{bmatrix}$$

Passo 2: Cálculo dos custos relativos

$$\pi = c_B^t \cdot B^{-1} = [-3, -5] \cdot \begin{bmatrix} 5/19 & -3/19 \\ -2/19 & 5/19 \end{bmatrix} = [-5/19, -16/19]$$

$$(c_3 - z_3) = 0 - \left(-\frac{5}{19}\right) = \frac{5}{19}$$



$$(c_4 - z_4) = 0 - \left(-\frac{16}{19}\right) = \frac{16}{19}$$

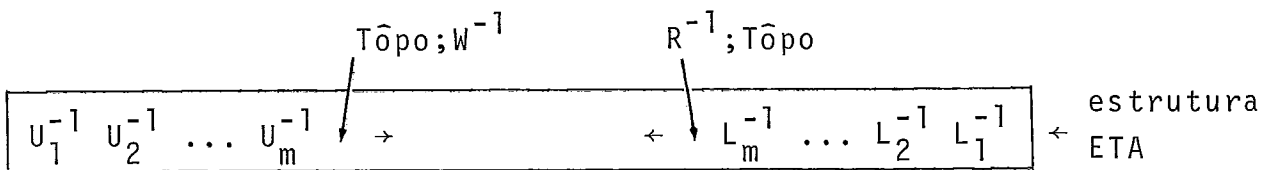
Todos positivos, portanto a solução é ótima.

#### 4.5. Aspectos Computacionais

Na implementação do simples LU, utilizamos o armazenamento das filas eta em duas estruturas de filas da seguinte forma: a rotina de inversão fornece  $B^{-1}$  como o produto

$$B^{-1} = U_1^{-1} \dots U_m^{-1} \cdot L_m^{-1} \dots L_1^{-1} ;$$

as filas eta  $U_i^{-1}$ ,  $i = 1, \dots, m$  são armazenados no início da estrutura de fila denominada ETA; e as filas eta  $L_i^{-1}$ ,  $i = 1, \dots, m$  são armazenadas no fim da estrutura ETA:



A fila eta  $W^{-1}$  criada é acrescentada na parte detrás da estrutura, no tôpo; a fila eta  $R^{-1}$ , quando criada, é acrescentada na parte da frente da estrutura, no tôpo.

A superioridade do esquema LU em relação ao simples na forma do Produto da Inversa foi comprovada na prática por Forrest-Tomlin <sup>[27]</sup>; tal estudo comparativo, por si só justifica a implementação em sistemas de grande porte da opção LU de atualização da inversa da base. Daremos agora o estudo comparativo feito em <sup>[26]</sup>, no qual podemos avaliar de maneira direta

a propagação de elementos não nulos na representação da inversa, utilizando o simplex LU e a Forma do Produto da Inversa do simplex:

Problemas	A	B	C	D
Número de linhas do prob.	602	1154	712	2787
Número de coluna do prob.	1350	2452	1156	6596
Número de elementos não nulos no problema	10503	12395	9987	22471
Número de elementos não nulos na inversa após a reinversão	4281	6583	2566	8091
Número de elementos não nulos na inversa após 50 iterações com a FPI	23824	18633	6050	8402
Número de elementos não nulos na inversa após 50 iterações com o esquema LU	6436	9192	3142	8375

As duas últimas linhas da tabela nos dão uma idéia da vantagem de utilização do simplex LU, no tocante ao problema de propagação de elementos não nulos, na representação da inversa, quando este é comparado com a Forma do Produto da Inversa.

CAPÍTULO V  
ESTRUTURA DO ARMAZENAMENTO DA MATRIZ  
DOS COEFICIENTES E DAS MATRIZES ELEMENTARES

5.1. Introdução

Inicialmente vamos definir alguns conceitos que usaremos com certa frequência:

Taxa de esparsidade de uma matriz (mxn) é a razão entre o número de elementos nulos da matriz e o produto do número de linhas pelo número de colunas

$$\% \text{Esparsidade} = \frac{\text{nº de zeros}}{mxn} \quad (5.1)$$

Taxa de densidade de uma matriz é um menos a esparsidade. Denominaremos por  $\delta$  a taxa de densidade

$$\delta = 1 - \% \text{Esparsidade} \quad (5.2)$$

Um fato importante, observado em modelos reais, é que a matriz dos coeficientes é sempre esparsa; isto é, tem uma grande quantidade de elementos nulos (ou  $\delta$  baixa). Em <sup>13</sup> encontramos as seguintes tendências de taxas de densidade

tipo de modelo	número de linhas (m)	número de colunas (n)	taxa de densidade $\delta$	(5.3)
pequenos	300 a 800	500 a 1200	0,5 a 2%	
médios	800 a 1500	1200 a 7000	0,05 a 0,3%	
grandes	+ 2500	+ 30000	menos 0,01%	

As matrizes esparsas quando armazenadas de forma completa ocupam memória em excesso e sem necessidade pois as operações de soma e multiplicação que envolvem seus elementos nulos não preci

sam ser calculadas pois tem resultados conhecidos.

Existe uma grande preocupação nos sistemas computacionais de grande porte com a estrutura, bem como com o "acessamento" dos dados, pois existe uma grande possibilidade de economizar memória e tempo de CPU usando estruturas de armazenamento adequadas.

Em [17] e [08] podemos encontrar métodos de armazenamentos sofisticados aplicados à programação linear.

## 5.2. Estrutura de Dados do Sistema de FURNAS

### 5.2.1. Matriz dos coeficientes tecnológicos (A)

No sistema de FURNAS foi utilizado para armazenagem da matriz dos coeficientes tecnológicos, uma estrutura de listas interligadas que descreveremos a seguir:

Observações: (1) Somente os elementos não nulos de A são armazenados - isto dará uma economia considerável de espaço de memória.

(2) Consideraremos para ilustrar o comportamento de cada lista (ou vetor) a matriz A como sendo:

$$A = \begin{bmatrix} a_{11} & 0 & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{bmatrix} \quad (5.4)$$

Os elementos não nulos da matriz A estão armazenados em ordem sequencial por linhas, no vetor COFINF. Para a matriz dada temos:

$$\text{COFINF} = \{a_{11}, a_{13}, a_{22}, a_{23}, a_{33}\}$$

ou seja:  $\text{COFINF}(1) = a_{11}$  ,  $\text{COFINF}(2) = a_{13}$  ,  $\text{COFINF}(3) = a_{22}$

$$\text{COFINF}(4) = a_{23} \text{ , } \text{COFINF}(5) = a_{33}$$

Observação: Sempre que nos referenciarmos à lista de elementos não nulos estaremos nos referindo ao vetor COFINF.

No vetor COFIND estão armazenados os índices das linhas dos elementos não nulos da matriz A, na mesma seqüência dos elementos do vetor

COFINF

$$\text{COFIND} = [1, 1, 2, 2, 3]$$

Por exemplo:  $\text{COFIND}(4) = 2$ , isto é, o quarto elemento na lista dos elementos não nulos ( $\text{COFINF}(4)$ ) está na linha 2 da matriz A.

O vetor COFLNK contém índices da lista de elementos não nulos; estes índices são os índices do antecessor não nulo de cada elemento, em sua coluna na matriz A. O vetor COFLNK é tal que,  $\text{COFLNK}(j) = k$  significa que  $\text{COFINF}(k)$  é o primeiro elemento não nulo que precede  $\text{COFINF}(j)$ , na coluna de  $\text{COFINF}(j)$ . Por exemplo consideremos a terceira coluna da matriz A

$$a_3 = \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix} \begin{array}{l} \curvearrowright \\ \curvearrowright \end{array} \text{COFLNK}$$

$\text{COFLNK}(4) = 2$  , isto é, o quarto elemento na lista de elementos não nulos (COFINF) tem como antecessor na coluna (na matriz dada) o segundo elemento na lista de elementos não nulos (COFINF). Analogamente,  $\text{COFLNK}(5) = 4$ , pois  $\text{COFINF}(5) = a_{33}$  e  $\text{COFINF}(4) = a_{23}$ . Quando  $\text{COFLNK}(j) = 0$ , isto significa que  $\text{COFINF}(j)$  é

o primeiro elemento não nulo de sua coluna na matriz A.

Para o exemplo (4.4), temos:

$$\text{COFLNK} = [0, 0, 0, 2, 4]$$

INICOL é um vetor cujos elementos são índices de COFINF.

INICOL (I) é o índice de COFINF relativo ao último elemento não nulo da coluna I da matriz A.

Para o nosso exemplo de A, dado em (4.4) temos:

INICOL(1) = 1, pois  $\text{COFINF}(1) = a_{11}$  é o único elemento não nulo na coluna 1 da matriz A.

INICOL(3) = 5 pois  $\text{COFINF}(5) = a_{33}$  é o último elemento não nulo da coluna 3 da matriz A.

A estrutura de armazenamento da matriz dada em (4.4) pode ser resumida na seguinte tabela:

Índices	1	2	3	4	5
COFINF	$a_{11}$	$a_{13}$	$a_{22}$	$a_{23}$	$a_{33}$
COFIND	1	1	2	2	3
COFLNK	0	0	0	2	4
INICOL	1	3	5		

Os motivos que levaram a adotar esta estrutura foram:

- (a) Economia de memória - este é um motivo facilmente perceptível, uma vez que apenas os elementos não nulos são armazenados.

(b) Economia de tempo de CPU - este não é tão perceptível quanto o motivo (a), por isto vamos nos estender um pouco mais na tentativa de uma explicação.

Seja o produto escalar  $w \cdot a_j$  onde  $w$  é um vetor  $(1 \times m)$  e  $a_j$  é a  $j$ -ésima coluna da matriz  $A$  que é  $(m \times m)$ . O produto escalar calculado com os dados armazenados na estrutura de listas interligadas, descrita anteriormente, pode ser executado usando a rotina:

```

WA = 0
IPTR = INICOL(I)
40 IF(IPTR.EQ.0) GO TO 50
WA = WA + W(COFIND(IPTR))*COFINF(IPTR)
IPTR=COFLNK(IPTR)
GO TO 40
50 CONTINUE

```

No caso de usarmos, para o mesmo produto escalar, uma rotina tradicional teremos:

```

IFIM = M+1
PRODUT = 0
I = 1
50 IF(I.EQ.IFIM) GO TO 80
PRODUT = PRODUT + W(I)*A(I,J)
I = I+1
GO TO 50
80 CONTINUE

```

Comparando os dois métodos de cálculo do produto escalar de  $w \cdot a_j$ , levando em consideração a taxa de densidade da matriz A, (pois somente as operações com os elementos não nulos são efetuadas no primeiro método), e os tempos de CPU das operações do FORTRAN-H IBM 370/158 [06], temos:

operações	Tempo de CPU(m seg)	Nº de operações do método tradicional	Nº de operações do método usado
soma	3,53	$2M+1$	$\delta M$
multiplicação	6,93	M	$\delta M$
acesso à matriz unidimensional	4,18	$2M$	$4\delta M+1$
acesso à matriz bidimensional	12,88	M	0
IF() GO TO	5,38	$M+1$	$\delta M+1$
GO TO	2,2	M	$\delta M+1$
Inicializações reais	2,3	1	1
Inicializações inteiras	2,13	2	0
TOTAL		$A = (42,81 M + 15,47)$	$B = (34,76 \delta M + 14,06)$

onde A e B são respectivamente os tempos totais de CPU para execução do produto escalar do método tradicional e do método usado no software.

$$A > B \quad \text{se} \quad 42,81M \geq 34,74\delta M$$

$$\text{então} \quad \delta \leq \frac{42,81}{34,76} \quad \delta \leq 1,23$$

Pela tabela (5.3), com  $\delta \leq 1$  temos que a estrutura de dados utilizada diminui o tempo de CPU.



### 5.2.2. Matrizes elementares

O armazenamento das matrizes elementares no sistema de FURNAS é feito de maneira a tirar proveito da característica básica das matrizes elementares, isto é, a esparsidade. Lembremos que uma matriz elementar é exatamente igual à matriz identidade exceto quanto a uma coluna. Já observamos anteriormente, a necessidade de armazenagem apenas da coluna distinta e da sua posição dentro da matriz elementar.

A estrutura de armazenamento utilizada no software de FURNAS é de listas interligadas semelhante à do armazenamento da matriz dos coeficientes. Antes de passarmos à descrição da estrutura utilizada devemos observar que existem  $m$  operações de divisão no cálculo da matriz elementar:

$$E = \begin{bmatrix} 1 & & \eta_{1p} & & \\ & & \eta_{2p} & & \\ & & \vdots & & \\ & & \eta_{mp} & & 1 \end{bmatrix}_{m \times m}$$

↑  
coluna p

onde:

$$\eta_{ip} = \begin{cases} \frac{-y_{ip}}{y_{pp}} & \text{para } i \neq p, i = 1, \dots, n \\ \frac{1}{y_{pp}} & \text{para } i = p \end{cases} \quad \text{pivot: } \eta_{pp}$$

Como divisão é uma operação cara, pode-se modificar a definição do vetor  $\eta_p$  de modo a economizar tempo de CPU. A modi-

ficação é a seguinte: efetua-se o cálculo do pivot (suponhamos que o pivot seja  $1/y_{pp}$ ), armazenando o resultado numa variável chamada PIVOT que será uma componente da fila eta correspondente; a nova definição do vetor  $\bar{\eta}_p$  será:

$$\bar{\eta}_{1p} = \text{PIVOT}$$

$$\bar{\eta}_{ip} = -y_{i-1,p} \quad i = 2, \dots, m+1 ; i \neq p$$

$$\bar{\eta}_{pp} = 1$$

A economia de tempo feita com esta nova definição da fila eta, (já que com a definição anterior tínhamos  $m$  divisões, e adotando (9,43 m.sec) que é o tempo de CPU da operação divisão do FORTRAN-H IBM 370/158 [06]), será de  $(m-1) \times 9,43$  m.sec, pois nessa nova definição temos apenas uma divisão.

Vamos descrever agora, de forma resumida, a estrutura de listas ligadas adotada para armazenagem das matrizes elementares no sistema de FURNAS.

ETAINF é o vetor que contém as informações, (os elementos não nulos) da fila e também o valor do PIVOT. Consideremos as seguintes matrizes elementares para exemplificarmos o uso de cada vetor da estrutura:

$$E_1 = \begin{bmatrix} 1 & 0 & 0 & \eta_{14} & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \eta_{44} & 0 \\ 0 & 0 & 0 & \eta_{45} & 1 \end{bmatrix} ; E_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \eta_{22} & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & \eta_{42} & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Seja  $\text{PIVOT}_4$  e  $\text{PIVOT}_2$  os valores dos elementos pivot respectivamente para  $E_1$  e  $E_2$ .

$$\text{ETAINF} = \left[ \text{PIVTO}_{4, \eta_{14}, \eta_{44}, \eta_{54}}, \text{PIVOT}_{54, \eta_{22}, \eta_{42}} \right]$$

isto é,  $\text{ETAINF}(1) = \text{PIVOT}_4$ ,  $\text{ETAINF}(2) = \eta_{14}$  ...  $\text{ETAINF}(7) = \eta_{42}$

ETAIND é o vetor que contém as seguintes informações sobre a matriz elementar:

- (a) A coluna ou posição do vetor eta dentro da matriz elementar;
- (b) O índice da linha de cada elemento do vetor eta de uma matriz elementar.

As informações do vetor ETAIND, estão na mesma seqüência das informações do vetor ETAINF.

$\text{ETAIND}(1) = 4$ , indica que a primeira fila eta armazenada em ETAINF, é referente a uma matriz elementar cujo vetor eta está na quarta coluna.

$\text{ETAIND}(2) = 1$ , porque  $\text{ETAINF}(2) = \eta_{14}$  está na primeira linha de sua matriz elementar.

ETALNK é o vetor que contém o índice, relativo a ETAINF, da próxima informação de uma matriz elementar.

$\text{ETALNK}(J) = k$  então  $\text{ETAINF}(k)$  é a próxima informação da mesma matriz elementar.

$$\text{ETALNK} = [2, 3, 4, 0, 6, 7, 0]$$

PTRETA é um vetor de apontadores, isto é, contém os índices de ETAINF que determinam o início das informações de uma matriz elementar

$PTRETA(I) = k$  significa que a  $I$ -ésima matriz tem a sua informação do PIVOT, (que é a primeira), armazenada em  $ETAINF(k)$

Exemplo:

$$PTRETA(1) = 1 \quad , \quad PTRETA(2) = 5$$

Para o exemplo (4.5) podemos construir a tabela:

	1	2	3	4	5	6	7
ETAINF	PIVOT <sub>4</sub>	$\eta_{14}$	$\eta_{44}$	$\eta_{54}$	PIVOT <sub>2</sub>	$\eta_{22}$	$\eta_{42}$
ETAIND	4	1	4	5	2	2	4
ETALNK	2	3	4	0	6	7	0
PTRETA	1	5					

## CAPÍTULO VI

### UM ESTUDO SOBRE PRICING MÚLTIPLO

#### 6.1. Introdução

O maior gasto de tempo computacional do algoritmo simplex, quando utilizado em problemas de grande porte reside nas operações que exigem a consulta dos dados  $(A, b, c)$ , dados estes que geralmente estão armazenados em memória auxiliar. Vamos denominar a transferência de dados da memória auxiliar para a memória principal, e vice-versa de operações de "pass"  $|^{19}|$ .

Os passos do algoritmo simplex revisado que mais exigem a operação de "pass" são os passos 1, 2, 4 (cap. III sec. 3.3); estes passos envolvem cálculos com a matriz inversa que geralmente está também armazenada em memória auxiliar na forma de Filas etc. Por outro lado, mesmo quando o problema está totalmente residente na memória central os cálculos que mais exigem tempo computacional são os cálculos dos custos reduzidos  $(c_j - z_j)$  para as variáveis não básicas. Estes cálculos são denominados operações de "pricing" ou simplesmente "pricing" do algoritmo. Portanto, o grande esforço computacional de uma iteração simplex quando aplicado à qualquer tipo de problema está nas operações de "pricing"; por este motivo existe um cuidado especial quando se desenvolvem sistemas de grande porte, com a rotina de cálculo de "pricing".

No sistema de FURNAS foi implementada uma rotina de pricing múltiplo que iremos descrever, antes porém devemos

ressaltar a existência de outros métodos de "pricing" tais como: o método de pricing "Devex", o método de pricing "steepest-edge" e também o método de "pricing" parcial [23].

## 6.2. O "Pricing" Múltiplo

O "pricing" múltiplo é uma técnica usada em sistemas de grande porte com o objetivo de diminuir o tempo computacional gasto em cada iteração por meio da diminuição das transferências de informações da memória central para a memória auxiliar. Na prática a utilização do "pricing" múltiplo tem provocado a redução do número de iterações do Simplex [16].

A seguir vamos descrever o "pricing" múltiplo individualmente e depois daremos a Forma do Produto da Inversa com o "pricing" múltiplo.

O "pricing" múltiplo pode ser descrito da seguinte maneira:

### Passo 1: Cálculo dos custos reduzidos e teste de otimalidade

Calcular  $(c_j - z_j)$  para todas as variáveis não básicas.

Determinar:

$$(c_r - z_r) = \min_j (c_j - z_j)$$

Se  $(c_r - z_r) \geq 0$  a solução obtida é ótima pare; caso contrário vá para o passo 2.

### Passo 2: Escolha do conjunto "pricing" (P)

Selecionar  $\bar{k}$  colunas não básicas correspondentes aos menores custos reduzidos, para formar o conjunto P .

Passo 3: Resolução do Problema "Relaxado"

O problema relaxado é constituído das variáveis básicas atuais e das variáveis não básicas cujas colunas fazem parte do conjunto P.

Resolve-se o problema "relaxado" pelo simplex revisado usual sempre considerando como variáveis não básicas as variáveis cujas colunas estão no conjunto P; considerar também que quando uma coluna do conjunto P entra na base, ela é excluída do conjunto P, no que concerne às futuras iterações da resolução do problema "relaxado". Obtida uma solução do problema "relaxado" volta-se ao passo 1.

A forma do Produto da Inversa com "pricing" múltiplo pode ser escrita como:

$$\text{Seja } B^{-1} = E_k \cdot E_{k-1} \cdots E_2 \cdot E_1$$

Suponhamos que  $E_1, E_2, \dots, E_k$  estejam armazenadas,

Passo 1: Cálculo da solução básica atual.

$$x_B = \bar{b} = (E_k(E_{k-1}(\dots(E_1 \cdot b)\dots)))$$

Passo 2: Cálculo dos custos relativos e teste de otimalidade

calcular  $(c_j - z_j)$  para todo  $j$  tal que  $x_j$  seja variável não básica. (Lembrar que  $z_j = \pi \cdot a_j$ ,  $\pi = c_B^t \cdot B^{-1}$  onde  $B^{-1}$  é a inversa da base atual e  $a_j$  é a coluna correspondente à variável não básica  $x_j$ ).

Seja

$(c_s - z_s) = \min\{(c_j - z_j)\}$ , para  $j$  tal que  $x_j$  seja variável não básica.

Se  $c_s - z_s \geq 0$  a solução é ótima, pare; caso con-

trário  $\bar{v}$  para o passo 3.

Passo 3: Escolha do conjunto P

Escolher  $\bar{k}$  vetores  $a_j$ , correspondentes aos menores  $(c_j - z_j)$  calculados no passo 2 para formarem o conjunto P.

Passo 4: Determinação da coluna que entra na base dentre as colunas do conjunto P.

Seja

$$(c_s - z_s) = \min(c_j - z_j), \text{ para } j \text{ tal que } a_j \in P$$

Se  $(c_s - z_s) \geq 0$  ou  $P = \emptyset$   $\bar{v}$  para o passo 2 testar a otimalidade da solução obtida; caso contrário  $x_s$  entra na base.

Passo 5: Atualização da coluna que entra na base

Calcular:

$$y_s = E_k \cdot E_{k-1} \cdots E_2 \cdot E_1 \cdot a_s$$

Eliminar  $a_s$  do conjunto P

Passo 6: Determinação da variável que sai da base

Seja:

$$\frac{\bar{b}_r}{y_{rs}} = \min \left\{ \frac{\bar{b}_i}{y_{is}} \right\}, \text{ para todo } i \text{ tal que } y_{is} > 0$$

Se não existe  $y_{is} > 0$  o problema é ilimitado, pare; caso contrário  $x_r$  deixa a base.



Passo 7: Definição de uma matriz elementar  $E_{k+1}$  a partir do vetor  $y_s$  e tomando como pivot o elemento  $y_{rs}$ .

Calcular o novo termo independente

$$x_B = E_{k+1} \cdot \bar{b}$$

Calcular o novo vetor dual

$$\pi = c_B^t \cdot B^{-1} = (\dots((c_B^t \cdot E_{k+1}) \cdot E_k) \dots E_1);$$

Fazer  $k = k+1$  e voltar ao passo 4.

### Observações

(1) Uma iteração do simplex com "pricing" múltiplo é caracterizada cada vez que se calcula os custos reduzidos de todas as variáveis não básicas no passo 2.

(2) Pode-se manter também em memória central as atualizações das colunas do conjunto P, nesse caso pode ser adotado como critério de entrada na base o critério do maior decréscimo da função objetivo:

$$(c_s - z_s) \cdot \frac{\bar{b}_r}{y_{rs}} = \min (c_j - z_j) \cdot \frac{x_i}{y_{ij}}, \text{ para } j$$

tal que  $(c_j - z_j) < 0$  e  $a_j \in P$ , e  $i$  tal que  $y_{ij} > 0$ .

Com este novo critério de escolha da variável que entra na base a resolução do problema "relaxado" é denominada de subotimização.

(3) A grande vantagem do "pricing" múltiplo está em que a sua utilização não necessita de buscas de dados intensas em memória auxiliar; tais buscas são muito lentas e consomem um tempo computacional muito grande.

- (4) O valor de  $\bar{k}$  varia de sistema para sistema; no sistema de FURNAS  $\bar{k} = 10$ .
- (5) Segundo  $|^{24}|$  o trabalho extra que é executado com a utilização do "pricing" múltiplo não chega a ser uma desvantagem diante do enorme sucesso da técnica na diminuição do tempo computacional.

## CAPÍTULO VII

### UM ESTUDO SOBRE "SCALING" EM PROGRAMAÇÃO LINEAR

#### 7.1. Introdução

O "scaling" aparece nos software de Programação Linear de grande porte nas mais diferentes formas e com muita freqüência; no entanto a teoria que fundamenta "scaling" e sua eficiência não é clara e na verdade é consenso geral que muito pouco se sabe sobre "scaling" em Programação Linear.

Nosso objetivo nesse estudo é descrever o "scaling" usado no software de FURNAS, dando antes uma breve idéia do que seja "scaling" e apresentando também suas formas mais habituais.

#### 7.2. O que é "scaling" em Programação Linear?

As restrições de um problema de Programação Linear podem ser escritas da seguinte forma:

$$[I, \bar{A}] \cdot x = b \quad (7.1)$$

$$L \leq x \leq u \quad (7.2)$$

onde  $\bar{A}$  é uma  $(m \times n - m)$  matriz, também conhecida como matriz associada às variáveis originais, (ditas, também, variáveis estruturais), do problema.  $I$  é a matriz identidade  $(m \times m)$  associada às variáveis artificiais e de folga do problema, chamadas também de variáveis lógicas.  $x$  e  $b$  são vetores: respectivamente  $(n \times 1)$  e  $(m \times 1)$ . Para maior facilidade de notação (7.1) pode ser

escrita como:

$$Ax = b \quad (7.3)$$

onde  $A = \begin{bmatrix} I & \bar{A} \end{bmatrix}$

O "scaling" em Programação Linear na maioria das vezes é apenas uma transformação linear. Nesse capítulo vamos nos deter na forma de "scaling" que se reduz a uma transformação simples de variáveis. Esta forma de "scaling" é efetuada no sistema (7.3) através das seguintes operações:

- (a) multiplicação de uma equação por um número;
- (b) multiplicação de uma coluna de  $A$  por um número<sup>1</sup>

Estas operações efetuadas no sistema (7.3) são traduzidas do ponto de vista matricial respectivamente por:

- (a') Pré-multiplicação de (7.3) por uma matriz  $R(m \times m)$ , diagonal e não singular;
- (b') A pós-multiplicação de  $A$  por uma matriz  $D(n \times n)$  diagonal e não singular; (7.3) ficará então:

$$RADx = R.b \quad (7.4)$$

Do ponto de vista de mudança de variáveis conseguimos chegar à mesma expressão (7.4). Consideremos a mudança de variável introduzida:

$$x = Dy \quad (7.5)$$

---

<sup>1</sup> Esta é a única operação que precisa ser desfeita através de um "unscaling".

Substituindo (7.5) em (7.3) teremos:

$$A Dy = b \quad (7.6)$$

Multiplicando por  $R$  à esquerda temos:

$$R A Dy = Rb \quad (7.7)$$

Notemos que (7.4) e (7.7) constituem o mesmo sistema. A modificação provocada em (7.2) pela mudança de variável (7.5) será:

$$D^{-1} L \leq y \leq D^{-1} u \quad (7.8)$$

Tudo isso, nos esclarece o fato dessa forma de "scaling" ser uma simples mudança de variáveis. Convém observar que os coeficientes das variáveis lógicas (artificiais e/ou de folga) não devem ser modificados com as operações de "scaling", ou melhor que as operações do tipo (b) nas variáveis lógicas devem ser de terminadas pelas operações do tipo (a) nessas mesmas variáveis, pois em caso contrário teríamos mais trabalho na obtenção da so lução básica inicial.

### 7.3. Razões para se fazer "scaling"

Uma pergunta natural nesse ponto é: para que se faz "scaling"? Em [27] e [09] encontramos as razões mais usuais que levam a adoção de "scaling" em Programação Linear:

- (a) Tornar mais compacta a representação dos limites das variáveis; por exemplo fazer uma mudança de variáveis que coloque todas nas novas variáveis com limite inferior zero. Isto equivale a fazer em (7.7) e (7.8) a seguinte mudança de variáveis:

$$z = Dy - L$$

Fazendo a substituição (7.7) e (7.8) ficam:

$$RAz = Rb - RAL \quad (7.9)$$

$$0 \leq z \leq (u-L) \quad (7.10)$$

Ao tornar mais compacta a representação dos limites, economiza-se memória para armazenamento dos dados; em contrapartida podem ser introduzidas grandes distorções nos dados. Isto pode acontecer se para algum  $j$ ,  $(u_j - \ell_j)$  tiver ordem de grandeza muito elevada. Para exemplificarmos tal fato, seja:

$$\begin{aligned} u_j &= 5569798987 \quad ; \quad \ell_j = 5569743598 \\ u_j - \ell_j &= 55389 \end{aligned} \quad (7.11)$$

Suponhamos que o computador usado tenha base  $\beta = 10$  e admita a representação numérica  $\hat{x} = m.\beta^e$ , com 6 dígitos na mantissa. Seja

$\hat{\alpha}$  = representação do número  $\alpha$  em forma exponencial;

$m$  = mantissa da representação do número  $\alpha$  ;

$\beta$  = base do computador; no caso  $\beta = 10$  ;

$e$  = expoente .

As representações de  $u_j$  e  $\ell_j$  serão:

$$\begin{aligned} \hat{u}_j &= 556980 \times 10^4 \\ \hat{\ell}_j &= 556974 \times 10^4 \\ \hat{u}_j - \hat{\ell}_j &= 6. \times 10^4 \end{aligned} \quad (7.12)$$

Comparando (7.11) e (7.12) verificamos uma diferença muito grande nos resultados obtidos; (7.11) é a operação feita manualmente e (7.12) é a mesma operação feita pelo computador. A diferença entre as duas operações é de 4.611.

No caso de usarmos o "scaling" com o objetivo de tornar mais compacta a representação dos limites das variáveis como pode ser visto em (7.10), teremos de utilizar a operação diferença entre os limites  $u$  e  $L$ , isto provocará uma modificação na região viável do problema proposto inicialmente. Portanto usar "scaling" com a finalidade de tornar mais compacta a representação de variáveis limitadas não é recomendável.

(b) Reduzir o número de iterações na resolução de problemas

O critério de escolha da variável para entrar na base nos softwares de programação linear é feito a partir de algum subconjunto de variáveis que possuam custos reduzidos mais negativos

$$c_j - z_j = c_j - (c_B)^t \cdot B^{-1} \cdot a_j$$

Caso não tenha sido feito o "scaling", as colunas  $a_j$  das variáveis estruturais podem ter elementos de valores muito distintos, (caso por exemplo de variáveis medidas por unidades diferentes); esse fato poderia afetar o cálculo de  $z_j$  no sentido de fazer com que uma variável que faça parte da solução ótima seja introduzida muito mais tarde na base. Com o "scaling" os  $z_j$  darão uma indicação mais exata da contribuição potencial que cada variável fará na melhoria da função objetivo. Isto poderá ficar mais claro com o exemplo a seguir:

Exemplo:

$$\text{Seja } a_j = \begin{bmatrix} 1000 \\ 1 \end{bmatrix} \quad a_k = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

Suponhamos  $x_j$  e  $x_k$  variáveis não básicas

$$(c^B)^t \cdot B^{-1} = [1, 1] ; c_j = c_k = 1$$

Teremos portanto

$$z_j = 1001 ; z_k = 2$$

$$c_j - z_j = -1000 ; c_k - z_k = -1$$

A variável  $x_j$  será escolhida para entrar na base. Se  $x_k$  fizer parte da solução ótima e  $x_j$  não,  $x_k$  entrará na base depois de  $x_j$ . Se fizermos "scaling" coluna em  $a_j$  teremos:

$$a_j = \begin{bmatrix} 1 \\ 0,001 \end{bmatrix} \quad z_j = 1,001 \quad \text{e} \quad c_j - z_j = -0,001$$

Nesse caso  $x_k$  entrará na base ao invés de  $x_j$ .

(c) Melhorar o comportamento numérico do algoritmo

Esta razão para se usar o "scaling" é a mais importante e também a mais ambiciosa.

Os erros de arredondamento no uso do simplex podem produzir a singularidade da base ou a perda de viabilidade. O que se faz contra isso nos "softwares" de Programação Linear é introduzir certas tolerâncias, algumas relativas ou tras absolutas; por exemplo, qualquer valor menor que  $\epsilon$  será considerado zero. Para assumir os valores dessas tolerâncias devemos estabelecer certas propriedades para os elementos da matriz dos coeficientes, por exemplo, fazer



com que os elementos da matriz estejam distribuídos em torno da unidade (através do "scaling").

### 7.3.1. Observação

O conceito de matriz que teve um bom "scaling" - matriz adequada - está associado ao fato de que seus elementos em valor absoluto estejam próximos uns dos outros.

Analogamente, uma matriz é mal adequada - teve um "scaling" inadequado - quando seus elementos têm valores absolutos muito distintos. Segundo [27] essa noção de matriz adequada tem a vantagem de poder razoavelmente ser aplicado na Programação Linear a matriz dos coeficientes  $A$ , com a expectativa de que a propriedade se estenda para as diversas bases. Embora este conceito de bom "scaling" seja amplamente usado, não existe uma concordância completa do que isto significa. Outros critérios para verificar se uma matriz é bem adequada aparecem em [27]:

(a) Uma matriz é bem adequada se:

$$\max_{i,j} |a_{ij}| / \min_{i,j} |a_{ij}| \leq R$$

para um  $R$  pré-fixado (normalmente  $R = 10^6$ )

(b) Uma matriz é bem adequada se:

$$\sum_{a_{ij} \neq 0} (\log |a_{ij}|)^2 \leq V$$

para um  $V$  determinado.

(c) Um outro método de medir a efetividade de um "scaling" feito, é verificar o quanto ele reduz  $\sigma^2$  - a variância dos elementos da matriz:

$$\sigma^2 = \left( \sum_{i,j} a_{ij}^2 - \left( \sum_{i,j} |a_{ij}| \right)^2 / \text{NUMPOS} \right) / \text{NUMPOS}$$

onde NUMPOS é o número de elementos não nulos da matriz.

#### 7.4. Alguns métodos de "scaling" em Programação Linear

##### (a) Método do equilíbrio

Consiste em executar o método de equilíbrio em cada linha e a seguir executar o método de equilíbrio em cada coluna.

O método de equilíbrio numa linha  $i$  consiste em dividir todos os elementos não nulos da  $i$ -ésima equação (coeficientes e termo independente) do sistema (7.3) por:

$$\max_j |a_{ij}|, \quad a_{ij} \neq 0$$

Do ponto de vista matricial tal operação consiste na pré-multiplicação por  $R$  no sistema (7.3), e cada elemento diagonal de  $R$  é dado por:

$$r_{ii} = \left( \max_j |a_{ij}| \right)^{-1}, \quad a_{ij} \neq 0$$

O método de equilíbrio da coluna  $j$  é feito dividindo os elementos não nulos da coluna  $j$  por:

$$\max_i |a_{ij}|, \quad a_{ij} \neq 0$$

Analogamente do ponto de vista matricial da matriz  $D$  em (7.4) é dado por:

$$d_{jj} = (\max_i |a_{ij}|)^{-1}, \quad a_{ij} \neq 0$$

(b) Método Geométrico

Consiste em executar o método geométrico em cada linha e a seguir em cada coluna.

O método geométrico numa linha  $i$  consiste em dividir cada elemento não nulo da  $i$ -ésima equação do sistema em questão por:

$$\sqrt{\max_j |a_{ij}| \cdot \min_j |a_{ij}|}, \quad a_{ij} \neq 0$$

Do ponto de vista matricial temos que em (6.4) os elementos diagonais da matriz  $R$  são dados por:

$$r_{ii} = (\sqrt{\max_j |a_{ij}| \cdot \min_j |a_{ij}|})^{-1}, \quad a_{ij} \neq 0$$

O método geométrico na coluna  $j$  consiste em dividir cada elemento não nulo da coluna  $j$  por:

$$\sqrt{\max_i |a_{ij}| \cdot \min_i |a_{ij}|}$$

Analogamente, do ponto de vista matricial, em (7.4) os elementos diagonais de  $D$  são dados por:

$$d_{jj} = (\sqrt{\max_i |a_{ij}| \cdot \min_i |a_{ij}|})^{-1}, \quad a_{ij} \neq 0$$

(d) Método da média

O método da média consiste em aplicar o método da média em cada linha e a seguir em cada coluna da matriz. O método da média numa linha  $i$  consiste em dividir cada elemento não nulo da linha  $i$  e o termo independente ( $b_i$ ) pela média aritmética dos elementos não nulos da linha  $i$ .

Analogamente se faz o método da média numa coluna  $j$ .

Os métodos geométrico e da média podem ser seguidos pela aplicação do método do equilíbrio.

Na prática foi observado [27] que o método geométrico seguido do método de equilíbrio tem conseguido os menores valores de  $\sigma^2$  (variância dos elementos da matriz dos coeficientes).

#### 7.5. O "scaling" no software de FURNAS

No "software" de FURNAS a técnica de "scaling" é opcional, isto é, o usuário é quem decide se deseja ou não o "scaling" do problema através da entrada de dados. Dentre os dados existe uma variável SCALE:

se SCALE = 1 , o "scaling" será feito nos dados;

se SCALE = 0 , a rotina de "scaling" não é executada.

A rotina de "scaling" do "pacote" de FURNAS tem a seguinte sequência de operações:

- (a) O método geométrico é aplicado no máximo 4 vezes;
- (b) Após cada aplicação do método geométrico calcula-se a variância dos elementos da matriz dos coeficientes

$$\sigma^2 = \left( \sum_{i,j} a_{ij}^2 - \left( \sum_{i,j} |a_{ij}| \right)^2 / \text{NUMPOS} \right) / \text{NUMPOS}$$

onde  $\sigma^2 \equiv$  variância dos elementos da matriz

$a_{ij} \equiv$  elemento não nulo da matriz dos coeficientes

NUMPOS  $\equiv$  número de elementos não nulos na matriz dos coeficientes.

Se a variância for menor que 10 encerra-se o "scaling" geométrico e passa-se a (c).

Se a variância não for menor que 10, prossegue-se com o "scaling" geométrico até obter-se uma variância menor que 10 ou então até que o método geométrico seja executado 4 vezes.

(c) Aplica-se o método do equilíbrio e termina a rotina de "scaling".

Após a obtenção da solução ótima se foi acionada a rotina de "scaling" é necessário recuperar a solução real do problema, isto é, anular as modificações introduzidas pela rotina "scaling".

A rigor as únicas operações que afetam a solução do sistema (7.3) são as operações feitas nas colunas da matriz A através da matriz D. Portanto, uma rotina de "unscaling", para recuperação da solução real do problema, necessita apenas manter a matriz D guardada; obtida a solução  $y = D^{-1}.x$  a solução real será:

$$x = D.y$$

CAPÍTULO VIII  
CONCLUSÕES

A finalidade deste capítulo é fazer alguns comentários a respeito da implementação e implantação do software de FURNAS no B-6700 da UFRJ.

Inicialmente devemos ressaltar que a "abertura do pacote", foi um trabalho árduo, podendo-se dizer que 80% de todo o trabalho residiu na compreensão, descrição e implantação do software.

A seguir nos dedicamos à descrição e implementação da opção LU do simplex, baseada em <sup>[10]</sup>, implementação esta que está voltada para finalidades didáticas e não comerciais; mesmo porque segundo a nossa ótica, o software de FURNAS não é ainda um software comercial, faltando para isto algum trabalho adicional principalmente no tocante à dualidade e entrada de dados.

O software com a opção LU foi testado com problemas pequenos, inclusive o de Beale, retirados de <sup>[4]</sup>. Testamos também o software com uma família de problemas que denominamos família D que pode ser descrita da seguinte maneira:

Seja N ímpar; a matriz dos coeficientes tecnológicos é dada por:

$$\left. \begin{array}{l} a_{11} = 1 = a_{1N} \quad ; \quad a_{1j} = 0 \quad , \quad j = 2, \dots, N-1 \\ a_{ii-1} = a_{ji} = 1 \\ a_{ij} = 0 \quad , \quad \text{para todas as} \\ \quad \quad \quad \text{outras} \end{array} \right\} \quad i = 2, \dots, N \quad ;$$

o termo independente e o vetor de custos são dados por:

$$\begin{array}{l} b_i = 1 \quad , \quad i = 1, \dots, N \\ c_i = 1 \quad , \quad i = 1, \dots, N \quad ; \end{array}$$

a  $i$ -ésima restrição do problema  $\bar{e}$  do tipo ( $\leq$ ) se  $i$   $\bar{e}$  impar, e do tipo ( $\geq$ ) se  $i$   $\bar{e}$  par.

A solução de um problema da família D  $\bar{e}$  dada por:

variáveis básicas:  $x_1, \dots, x_{N-1}, x_n$ ; onde  $n = \frac{5 \cdot N - 1}{2}$

$$\left. \begin{array}{l} x_i = 0 \quad \text{se } i \text{ } \bar{e} \text{ par} \\ x_i = 1 \quad \text{se } i \text{ } \bar{e} \text{ impar} \end{array} \right\} i = 1, \dots, N-1$$

$$x_n = 1 \quad (x_n \text{ } \bar{e} \text{ uma variável de folga})$$

Todas as outras variáveis são nulas.

O valor ótimo  $\bar{e}$  dado por:

$$Z = \left(\frac{N-1}{2}\right) \quad \text{se o problema } \bar{e} \text{ de minimização}$$

$$Z = -\left(\frac{N-1}{2}\right) \quad \text{se o problema } \bar{e} \text{ de maximização.}$$

Os testes foram feitos para comparar o comportamento da opção LU do simplex em relação à Forma do Produto da Inversa, quanto a número de iterações, taxa de utilização das filas e tempo de solução. Para isto fizemos com que o software adotasse o simplex LU a partir da segunda iteração, na resolução dos problemas; depois rodamos os mesmos problemas normalmente. Os resultados obtidos com os testes são resumidos na tabela abaixo.

Problema	N	densidade	Número de iterações	Taxa de utilização das filas e $\tau$	tempo de solução/ minutos
A FPI	5	0.40	5	0.0030	0.01
A LU			5	0.0065	0.01
B FPI	55	0.364	55	0.0405	0.02
B LU			55	0.0940	0.03
C FPI	99	0.0202	99	0.0735	0.028
C LU			99	0.1745	0.037

Cabe ressaltar que os problemas da tabela acima não deixam transparecer o comportamento numérico do Simplex LU. Isto acontece porque o gerador de problemas testes que foi utilizado é muito simples. Portanto devemos deixar claro que o software necessita de uma fase de testes complementar, trabalho este que pode ser desenvolvido gradativamente dentro da filosofia do Programa de Engenharia de Sistemas e Computação da UFRJ.



BIBLIOGRAFIA

- [<sup>1</sup>] BAZARAA, M.Sc. e JARVIS, J.J., "Linear Programming and Network Flows, J. Wiley, New York, (1977).
- [<sup>2</sup>] BARTELS, R.H. e GOLUB, G.H., "The Simplex Method of Linear Programming using LU Decomposition", Communications ACM 13, pp. 266-286, (1969).
- [<sup>3</sup>] BENICHOU, M., GAUTHIER, J.M., HENTGES, G. e RIBIERE, G., "The Efficient Solution of Large-Scale Linear Programming Problems - Some Algorithmic Techniques and Computational Results", Mathematical Programming, 12, pp. 280-322, North-Holland Publishing Company, (1977).
- [<sup>4</sup>] BORNSTEIN, C.T., OLIVEIRA, A.A.F., BREGALDA, P.F., "Introdução à Programação Linear", Ed. Campus, RJ, (1981).
- [<sup>5</sup>] BRAYTON, R.K., GUSTAVSON, F.G., WILLOUGHBY, R.A., "Some Results on Sparse Matrices", RC-2332, IBM Research Center, Yorktown Heights, N.Y., February, 14 (1969).
- [<sup>6</sup>] CAMPELLO, R. e RESENDE, M., "Software para Programação Linear de Médio Porte: Desenvolvimento e Implementação", Coordenação de Planejamento - CP.T, Informe Técnico AMM.T-006.81, FURNAS - Centras Elétricas S.A. Dezembro (1981).

- [<sup>7</sup>] DANTZIG, G.B., HARVEY, R.P., MCKNIGHT, R.D., SMITH, S.S.,  
"Sparse matrix techniques in two mathematical  
Programming Codes", in: Sparse matrix Proceeding,  
Ed. R. Willoughby, (RA-1, IBM Research Center,  
Yorktown Heights, N.Y., pp.85-95 (1969)).
- [<sup>8</sup>] DeBUCHET, J., "How to take into account the low density  
of matrices to design a Mathematical Programming  
Package", in J.K.Reid (ed), Large Sparse Sets of  
Linear Equations, Academic Press London,  
pp.211-218
- [<sup>9</sup>] ESCUDERO, L.F., "An Algorithm for Large-Scale Quadratic  
Programming Problems and its extensions to the  
Linearly constrained Non-Linear Case", IBM -  
Madrid Scientific Center, (1982).
- [<sup>10</sup>] FORREST, J.J.H., TOMLIN, J.A., "Updated triangular  
factors of the basis to maintain sparsity in the  
Product form simplex method", Mathematical  
Programming 2, pp.263-278, North-Holland  
Publishing Company, (1972).
- [<sup>11</sup>] FORSYTHE, G.E., MOLLER, C.B., "Computer Solution of  
Linear Algebraic Systems", Englewood Cliff,  
Prentice-Hall, (1967).
- [<sup>12</sup>] FULKERSON, F. e WOLFE, P., "An Algorithm for Scaling  
Matrices", SIAM Review, 4, pp.142-146, (1962).
- [<sup>13</sup>] GREENBERG, H.J., "Tutorial on Matricial Packing", in  
Design and Implementation of Optimization Soft-  
ware, H.Greenberg (ed.), NATO, Advanced Studies  
Institute Series, E.Applied Science, (1978).

- |<sup>14</sup>| HELLERMAN, E. e RARICK D., "Reinversion with the Preassigned Pivot Procedure", *Mathematical Programming* 1, pp.195-216, (1971).
- |<sup>15</sup>| HAMMING, R.W., "Introduction to Applied Numerical Analysis", MacGraw-Hill, New York, (1971).
- |<sup>16</sup>| HENTGES, G., "MPS/360 - Matriz Scaling", IBM France Research Memorandum, (March 1969).
- |<sup>17</sup>| KALAN, J.E., "Aspects of Large-Scale in core Linear Programming", in *Proceeding of the 1971 Annual Conference of the ACM Chicago, ILL*, pp.304-313, (August 1971).
- |<sup>18</sup>| LARSEN, L.J., "A Modified Inversion Procedure for Product Form of the Inverse Linear Programming Codes", Comm.ACM, (1962).
- |<sup>19</sup>| LASDON, L. S., "Optimization Theory for Large Systems", Macmillan Publ.Co.Inc., New York. (1970)
- |<sup>20</sup>| LUENBERGER, D.G., "Introduction to Linear and Nonlinear Programming", Ed. Addison-Wesley P.C., (1973).
- |<sup>21</sup>| MACULAN, F.N., BRAGA, L.P.V., "Aspectos Computacionais em Programação Linear", 5.<sup>a</sup> Escuela Latino-Americana de Matemática, Mar del Plata, Argentina, (1980).
- |<sup>22</sup>| MARKOWITZ, H.M., "The Elimination Form of the Inverse and its Application to Linear Programming", Management Science, vol.3, nº 3, pp.255-269, April (1957).

- |<sup>23</sup>| MURTAGH, B. A., "Advanced Linear Programming", ed.  
McGraw-Hill, (1981).
- |<sup>24</sup>| ORCHARD-HAYS, W., "Advanced Linear Programming Computing  
Techniques", McGraw-Hill, New York, (1968).
- |<sup>25</sup>| ROSANDER, R.R., "Multiple Pricing and Suboptimizations  
in Dual Linear Programming Algorithms",  
Mathematical Programming Study 4, pp.108-117,  
(1975).
- |<sup>25</sup>| SCHNEIDER, H., BARKER, G.P., "Matrices and Linear  
Algebra", Holt Rinehard and Wiston, Inc.(1967).
- |<sup>26</sup>| TOMLIN, J.A., "On Scaling Linear Programming Problems",  
Mathematical Programming Study 4, pp.146-166,  
(1975).
- |<sup>27</sup>| TOMLIN, J.A., "On Pricing and Backward Transformation  
in Linear Programming", Mathematical Programming  
6, pp.42-47, (1974).