

## 4 Solucionador Iterativo Proposto

### 4.1 Introdução

Após terem sido apresentados nos capítulos anteriores os fundamentos necessários para a compreensão das características do problema, este capítulo descreve o solucionador iterativo proposto para resolver o subproblema linear esparso associado ao método de Newton-Raphson do problema de fluxo de carga. O solucionador é desenvolvido buscando agregar eficiência e robustez, evitando as limitações das técnicas numéricas analisadas nos capítulos anteriores.

A fim de atingir as metas acima, é proposto um pré-condicionador baseado no algoritmo de Doolittle (Bodewig, 1956; Wilkinson, 1965; Dahlquist & Björk, 1974; Hoffman, 2001), denominado deste ponto em diante de  $ILU(\xi)$ . Aliado ao pré-condicionador, também se propõe uma regra de preenchimento baseada no erro resultante após o descarte de elementos explorando certas características do método de Doolittle, visando facilitar o cálculo do referido erro. Conseqüentemente, a regra pode decidir eficientemente se um dado elemento em ambos os fatores triangulares L e U deve ser preenchido ou descartado.

Neste capítulo também é proposto um método para ajuste dos valores dos parâmetros dos solucionadores iterativos. O método permite identificar intervalos de valores permissíveis para cada um dos parâmetros, visando garantir a robustez e melhorar o desempenho do solucionador.

### 4.2 O Estado da Arte dos Solucionadores de Sistemas Lineares

Um solucionador é um termo genérico associado a um programa computacional que soluciona um problema matemático. Dentro deste contexto, um solucionador de sistemas de equações lineares é um programa computacional que inclui um

método numérico de solução de sistemas de equações lineares (direto ou iterativo) e estratégias numéricas desenvolvidas para melhorar seu desempenho computacional.

#### 4.2.1 Solucionadores Diretos

Embora o estudo e avaliação dos métodos diretos não seja um dos objetivos desta tese, existe a necessidade de adotar um solucionador direto para realizar comparações com o solucionador iterativo que está sendo proposto aqui. Na Tabela 4.1 apresenta-se uma síntese das principais características dos solucionadores diretos mais usados em diversas áreas da engenharia e da ciência. Como se pode observar, na primeira coluna são listados os nomes dos solucionadores, na segunda o tipo de método direto adotado pelo solucionador, na terceira os tipos de reordenamento da matriz de coeficientes adotados pelo solucionador e, finalmente, na última coluna são listadas as referências para cada solucionador, nas quais pode-se obter maiores detalhes.

Pode-se afirmar que existem basicamente 4 grandes categorias de métodos diretos usados nos solucionadores, conhecidos como atualização imediata (*right-looking*), atualização retrasada (*left-looking*), método supernodal, método multifrontal, ou combinações destes. Pode-se dizer também que a maioria utiliza o conjunto de sub-rotinas BLAS, disponibilizada por (Dongarra et al., 1990) para realizar as principais operações esparsas de álgebra linear (vetor, matriz-vetor e matriz-matriz).

Os solucionadores realizam fatorações do tipo LU, sendo que o SPARSPAK e os solucionadores do Matlab podem também realizar fatorações do tipo QR. Todos os solucionadores são capazes de operar com matrizes simétricas e não-simétricas. Os métodos de reordenamento disponíveis são do tipo grau mínimo (*minimum degree*) e variantes como o mínimo preenchimento (*minimum fill*), grau mínimo por colunas (*column minimum degree*), Markowitz e outros; também são considerados o reordenamento por divisão encaixada/único-sentido (*nested/one-way dissection*), permutações por blocos triangulares e de redução de largura de banda, como o conhecido Cuthill Mckee reverso RCM (*Reverse Cuthill Mckee*).

Tabela 4.1 – Características básicas dos solucionadores diretos mais usados

Solucionador	Tipo de Método Direto	Estratégias Implementadas	Referência
MA28	Atualização imediata Markowitz	Mínimo grau e Bloco triangular	(Duff & Reid, 1979)
SPARSPAK	Atualização retrasada	Mínimo grau, Divisão encaixada e Cuthill Mckee reverso	(George & Liu, 1981)
Solucionadores de MATLAB	Vários	Mínimo grau, Bloco triangular e Cuthill Mckee reverso	(Gilbert et al., 1992)
MA48	Atualização retrasada	Mínimo grau, Bloco triangular	(Duff & Reid, 1996)
SuperLU	Atualização retrasada Supernodal	Mínimo grau	(Demmel et al., 1999)
SuperLU _DIST	Atualização imediata Supernodal	Mínimo grau	(Li & Demmel, 2003)
PARDISO	Atualização imediata/retrasada Supernodal	Mínimo grau e Divisão encaixada	(Schenk & Gartner, 2001)
WSMP	Multifrontal	Mínimo grau, Divisão encaixada e Bloco triangular	(Gupta, 2002)
Solucionadores de Mathematica	Vários	Mínimo grau, Divisão encaixada, e Bloco triangular	(Wolfram, 2003)
UMFPACK	Multifrontal	Mínimo grau e Escalonamento	(Davis, 2004)

Nestes solucionadores, os aspectos a serem levados em consideração para desenvolver os algoritmos complexos, heurísticas e estratégias apresentadas na Tabela 4.1, basicamente são: a geração do menor número de elementos não-nulos possível durante o processo de fatoração e também a conservação da estabilidade numérica mediante as estratégias de reordenamento e monitoramento/eleição dos elementos pivôs, respectivamente.

Neste trabalho, o solucionador MA28 representará os solucionadores diretos quando realizadas as comparações de desempenho com o solucionador iterativo proposto nesta tese. Este solucionador foi escolhido principalmente porque permite calcular o número de operações de ponto flutuante, pela facilidade para ser acondicionado dentro do programa de fluxo de carga, pela disponibilidade

total do código do solucionador em linguagem FORTRAN e desempenho similar a outros solucionadores diretos. O MA28 permuta tanto linhas como colunas na matriz de coeficientes A, utilizando duas matrizes P e Q. Este solucionador usa principalmente as estratégias de Markowitz e mínimo grau, visando manter as propriedades esparsas e o controle da perda da exatidão devido aos erros de arredondamento (Duff & Reid, 1979). Ambas as estratégias são executadas de forma dinâmica durante o processo de fatoração ( $P \cdot A \cdot Q = L \cdot U$ ).

No próximo capítulo, são usados dois programas computacionais – ANAREDE e ORGANON (Chaves, 2008; Jardim & Stott, 2005, CEPEL, 2007), para validar os resultados dos experimentos numéricos e realizar comparações, principalmente em termos de robustez e eficiência computacional. O ORGANON é um sistema utilizado para avaliação da segurança de redes elétricas de potência que emprega computação de alto desempenho utilizando técnicas numéricas avançadas, algoritmos especializados, arquitetura construtiva flexível e processamento distribuído (Chaves, 2008). No ORGANON, o problema de fluxo de carga é solucionado de duas formas: usando-se os modelos estáticos da rede elétrica e o algoritmo de Newton-Raphson; ou usando-se modelos dinâmicos da rede elétrica e o método de dinâmica sintética (Jardim & Stott, 2005). Em (Chaves, 2008) são feitas diversas e diferentes aplicações com este programa em estudos de fluxo de potência através do método convencional de Newton-Raphson completo ou com o método de Fluxo de Potência de Dinâmica Sintética (FPDS). Este último obtém a solução para o problema de fluxo de potência (caso essa exista) mesmo para casos em que os métodos convencionais falhem, e ainda, apenas soluções estáveis.

Já o programa de análise de redes (ANAREDE – versão 09.04.03) é um conjunto de aplicações computacionais, algoritmos e métodos eficientes, adequados à realização de estudos nas áreas de operação e de planejamento de sistemas elétricos de potência. No ANAREDE, a solução das equações da rede elétrica é realizada pelo algoritmo Newton-Raphson e o reordenamento da matriz Jacobiana, para preservar a esparsidade, é efetuado utilizando a técnica MD (CEPEL, 2007).

#### 4.2.2

### Solucionadores Iterativos Aplicados em Problemas de Sistemas de Energia Elétrica

Na Tabela 4.2 apresenta-se uma síntese do estado da arte dos solucionadores iterativos quando aplicados ao problema de fluxo de carga. A maior parte das informações associadas com os métodos iterativos e estratégias de pré-condicionamento de cada um destes solucionadores já foi apresentada nas seções 2.4.1, 2.4.2 e 3.4 dos capítulos anteriores. Algumas informações associadas com outras estratégias e técnicas numéricas, consideradas em alguns destes solucionadores e que ainda não foram mencionadas, são apresentadas nas últimas duas colunas da Tabela 4.2, sendo estas:

Em (Flueck e Chiang, 1998) combina-se a técnica de reordenamento RCM com o método GMRES e o pré-condicionador fixo FastD. Todas as simulações foram realizadas usando RCM e um sistema de 8027 barras, sendo o solucionador proposto, 50% mais rápido que o tradicional solucionador direto de fatoração LU.

Em (Dag e Semlyen, 2003) o solucionador proposto usa técnicas de escalonamento na matriz  $B'$  do método desacoplado rápido, obtendo-se bons resultados em termos de robustez. Porém em (de Leon, 2003), a proposta de Dag é fortemente criticada pelo uso dos métodos desacoplados junto aos métodos iterativos.

Em (Pessanha et al., 2009) avaliam-se os efeitos do reordenamento RCM e da técnica de máximo produto transversal com escalonamento na qualidade do pré-condicionador  $ILUT(\tau, \rho)$  quase-fixo. Os resultados mostram que este tipo de preprocessamento da matriz Jacobiana melhora significativamente a qualidade do pré-condicionador, conseqüentemente a robustez do solucionador.

Nas últimas linhas da Tabela 4.2 apresentam-se as características básicas do solucionador proposto nesta tese destacando-se o pré-condicionador  $ILU(\xi)$  com regra de preenchimento baseada no erro. Nas próximas seções, todos os componentes do solucionador proposto são tratados com maiores detalhes.

Tabela 4.2 – Características dos solucionadores iterativos usados em aplicações de fluxo de carga

Autor	Ano	Método Iterativo	Estratégias Usadas			
			Pré-condicionador	Regra de Eliminação	Reordenamento	Escalonamento
Galiana	1994	G.Conjugado	Cholesky Incomp., $M=L \cdot L^T(0)$	Sem novos elementos não-nulos	—	—
Mori	1996	Tchebychev	Cholesky Incompleto	Sem novos elementos não-nulos	—	—
Semlyen	1996	GMRES(m)	—	—	—	—
Borges	1996(97)	BiCGStab	$M = L \cdot U$ , da primeira iteração	—	—	—
Pai	1997	GMRES	ILU(k)	Nível de novos elementos não-nulos	—	—
Flueck	1998	GMRES(m)	FastD: $M = B'$	—	RCM	—
Alves	1999	G.Conjugado	Cholesky completo	—	MD, MST	—
De León	2002	BiCGStab, GMRES, QMR, CGS, BiCG + ILU( $\tau$ )	Com parâmetro limitante $\tau$	—	—	—
Dag	2003	G.Conjugado	Aproximação da Inversa	Polinômio de Chebyshev	—	$D_1 \cdot B' \cdot D_2$
Chen	2006	GMRES(m)	Aprox. da Inversa Adaptativa	—	—	—
Mori	2007	BiCGStab	ILU(k)	Nível de novos elementos não-nulos	—	—
Pessanha	2009	GMRES	ILUT( $\tau$ , $\rho$ ) quase-fixo	Duplo parâmetro limitante $\tau$ e $\rho$	RCM	—
Khaitan	2010	GMRES	$M = L \cdot U$ , Multifrontal quase-fixo	—	RCM	—
Solucionador Proposto		<b>GMRES</b>	<b>ILU(<math>\xi</math>)</b>	<b>Baseada no Erro</b>	<b>MD</b>	—

### 4.3 Solucionador Iterativo Para Fluxo de Carga

O solucionador desenvolvido é composto basicamente pelo método iterativo de subespaços de Krylov GMRES (vide Capítulo 2) e pela estratégia de pré-condicionamento de fatoração incompleta ILU( $\xi$ ), com suporte da técnica de reordenamento Mínimo Grau (MD). A estratégia de reinicialização GMRES(m) (apresentada em 2.6), usada para limitar o tamanho dos subespaços de Krylov a um valor máximo “m”, não faz parte do solucionador, uma vez que a estratégia de pré-condicionamento força a convergência do GMRES em poucas iterações lineares (uma ou duas iterações). Portanto, já que poucas iterações estão associadas a subespaços pequenos, então não é necessário estabelecer um valor máximo “m”.

Nesta tese adota-se a técnica de reordenamento MD, porque a mesma acarreta a redução significativa do número de novos elementos não-nulos gerados, reduzindo o esforço computacional realizado para construir e usar o pré-condicionador. Além disso, este tipo de reordenamento aumenta a taxa de convergência do método iterativo GMRES. O solucionador é denominado de GMRES+ILU( $\xi$ )+MD e foi implementado em um programa computacional de fluxo de carga desenvolvido em linguagem Fortran, da forma apresentada no fluxograma da Figura 4.1. Seus componentes estão destacados na cor vermelha, numerados por algarismos romanos na mesma ordem em que são aplicados.

Observa-se que, antes de iniciadas as iterações Newton-Raphson, é calculada a seqüência de reordenamento usando-se a técnica MD, matematicamente representada pela matriz de permutações P. Na prática, é gerado um vetor com a seqüência de reordenamento que permitirá montar a matriz Jacobiana já reordenada, em cada iteração k do algoritmo Newton-Raphson. Esse vetor também é usado para reordenar os resíduos de potência ativa e reativa armazenados no vetor  $b_k$ . Cada vez que o teste de convergência do algoritmo Newton-Raphson não for satisfeito, uma nova matriz Jacobiana reordenada é montada. A partir das informações da matriz Jacobiana reordenada, a estratégia de pré-condicionamento ILU( $\xi$ ) constrói um pré-condicionador  $M_k$ , que em seguida será usado para obter explicitamente o novo sistema linear pré-condicionado (4.1) a ser solucionado pelo método iterativo GMRES. Finalmente,

as incógnitas são atualizadas em (4.2). O procedimento é repetido até que o teste de convergência do algoritmo Newton-Raphson seja satisfeito.

$$M_k^{-1} \cdot P \cdot J \cdot P^T \cdot x = M_k^{-1} \cdot P \cdot b_k \quad (4.1)$$

$$\begin{bmatrix} v \\ \theta \end{bmatrix}_{k+1} = \begin{bmatrix} v \\ \theta \end{bmatrix}_k + P^T \cdot x \quad (4.2)$$

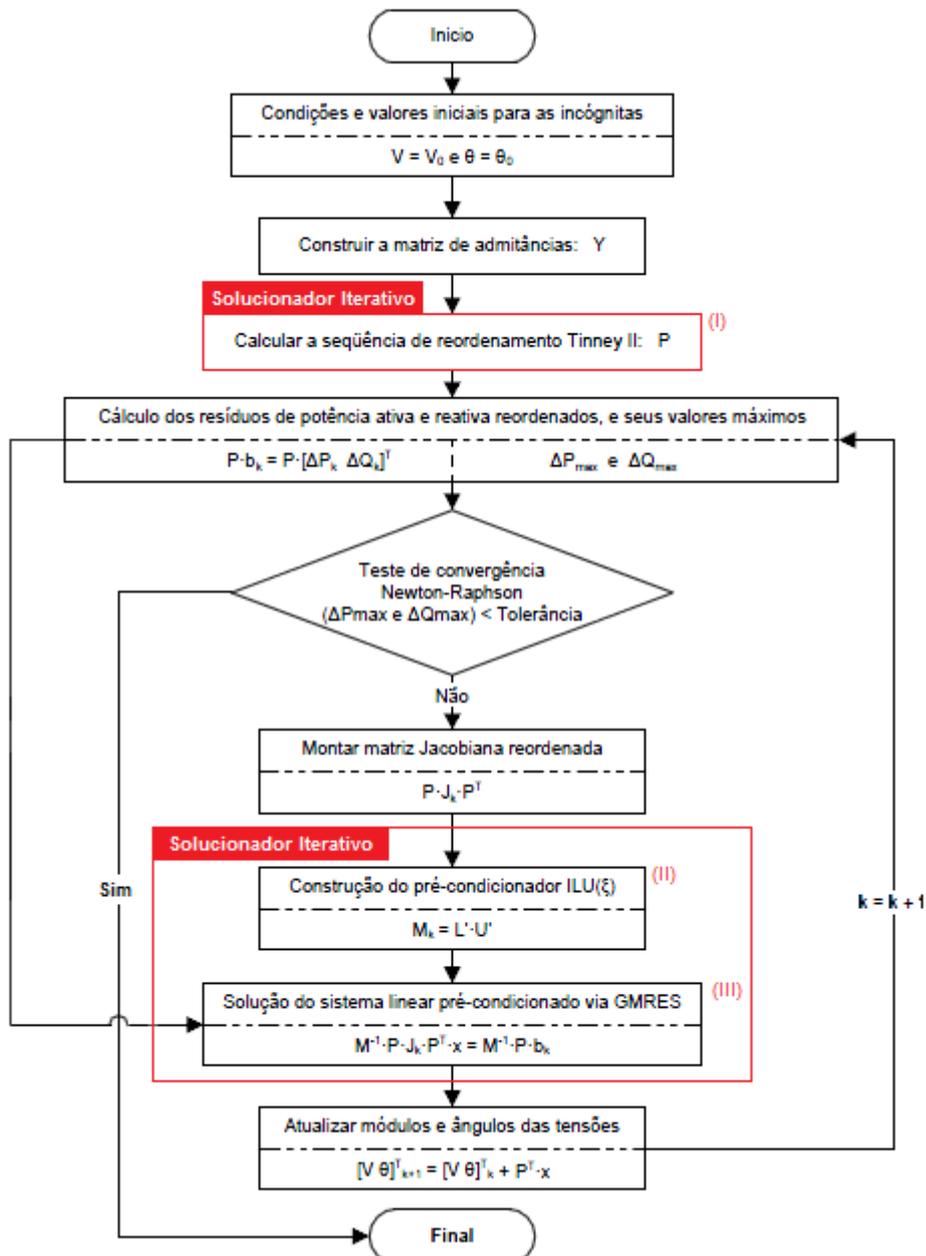


Figura 4.1 – Fluxograma do Solucionador Iterativo para o problema de fluxo de carga

O programa de fluxo de carga modela as barras de acordo como seu tipo (PQ, PV e de referência); as linhas de transmissão, transformadores e

transformadores defasadores são representados pelos seus circuitos  $n$  equivalentes. Os tapes variáveis sob carga podem ainda ter a atuação contínua ou discreta. As cargas das barras são normalmente modeladas como potências ativa e reativa constantes, ou podem ser expressas como uma função da magnitude da tensão da barra. Não são modelados elos de corrente contínua. No algoritmo de fluxo de carga consideram-se o controle de variação automática de tape e os limites de geração de potência reativa.

#### 4.4

#### Regra de Preenchimento Baseada no Cálculo do Erro Associado ao Elemento Descartado

Uma forma simples de estimar o erro total  $\|R\|$ , originado pelos elementos descartados devido às regras de preenchimento da fatoração incompleta, é usando-se (3.10) (Mori et al., 1996; Saad, 2003). Nesta equação, o produto dos fatores triangulares incompletos é igual ao pré-condicionador  $M$  e o produto dos fatores triangulares completos é igual à matriz de coeficientes  $A$ , como mostrado em (3.11). Desta forma, o erro  $\|R\|$  permite vislumbrar quão diferente se tornou o pré-condicionador em relação aos fatores triangulares completos, após o descarte de seus elementos.

Um pré-condicionador sem regra de preenchimento efetua completamente o cálculo dos fatores triangulares. Neste caso, o pré-condicionador  $M$  se iguala à matriz de coeficientes ( $M = L \cdot U = A$ ), o que resulta na convergência do método GMRES em uma única iteração, com erro  $\|R\|$  nulo. Porém, o número de operações de ponto flutuante ainda é considerável. Para se obter esta redução é necessário usar uma regra de preenchimento que descarte o maior número de elementos nos fatores triangulares e sem que isso provoque o aumento do erro  $\|R\|$ , conseqüentemente afetando o desempenho do GMRES.

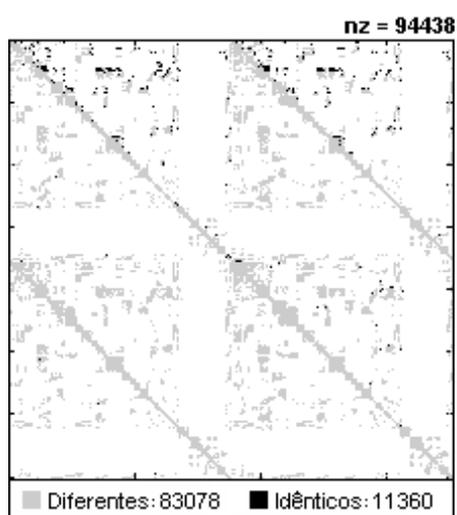
$$\|R\| = \|A - M\| \quad (4.3)$$

$$\|R\| = \|L \cdot U - L' \cdot U'\| \quad (4.4)$$

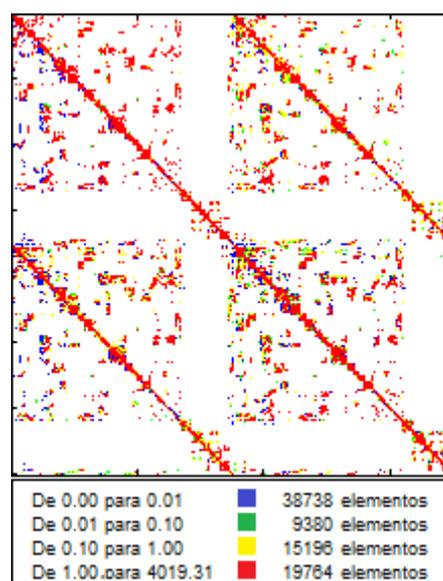
Como exemplo, na Figura 3.6 são apresentados dois gráficos que ilustram como os erros produzidos pelo descarte dos fatores triangulares não-nulos podem se alastrar até em posições de elementos já preenchidos. O experimento foi

apresentado na seção 3.3.2 do capítulo anterior, para o sistema brasileiro de 3513 barras, usando-se o pré-condicionador ILU(0) para o cálculo dos fatores incompletos. No gráfico da Figura 3.6(a) mostra-se que 87,97% dos elementos do pré-condicionador M (83.078) são distintos dos encontrados na matriz Jacobiana. O gráfico do erro, em 4.2(b), permite se ter uma idéia da dimensão destas diferenças. Dos 38.738 elementos com erros entre 0 e 0,01 (pontos azuis), 11.360 têm erro igual a zero (elementos idênticos) e 27.378 têm erros menores que 0,01. Os erros nesta faixa de valores normalmente não prejudicam a convergência do GMRES.

As outras faixas de valores associadas aos pontos de cores verde, amarelo e vermelho normalmente prejudicam a convergência do GMRES. Esses erros devem ser evitados e são gerados quando alguns elementos importantes são descartados durante a fatoração ILU(0). O maior erro provocado entre os elementos de M e da Jacobiana é de 4019,31 e o erro total, para este pré-condicionador, é de  $1,0632 \cdot 10^5$  (norma de Frobenius). Para evitar que os erros se propaguem enquanto é realizada a fatoração, é necessário calcular o erro associado a cada elemento dos fatores triangulares antes de seu preenchimento ou descarte. Este erro deve também considerar os erros propagados nos cálculos futuros onde o elemento seja requerido. Se este valor for zero ou próximo de zero, então o elemento pode ser descartado, caso contrário, deve ser preenchido.



a) Produto dos fatores incompletos ( $L' \cdot U'$ ) do pré-condicionador ILU(0)



b) Matriz R para o pré-condicionador ILU(0)

Figura 4.2 – Erros produzidos pela regra de preenchimento com o pré-condicionador ILU(0), para a matriz Jacobiana do sistema brasileiro de 3513 barras.



$$(U - U') = \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ & & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ & & & 0 & 0 & \dots & 0 & U_{ki} & 0 & \dots & 0 \\ & & & & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ & & & & & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ & & & & & & 0 & 0 & 0 & \dots & 0 \\ & & & & & & & 0 & 0 & \dots & 0 \\ & & & & & & & & 0 & \dots & 0 \\ & & & & & & & & & \ddots & \vdots \\ & & & & & & & & & & 0 \end{bmatrix} \quad (4.10)$$

Pela substituição de (4.9) e (4.10) nos produtos  $(L-L') \cdot U$  como  $L \cdot (U-U')$ , resulta em (4.11) e (4.12).

$$(L - L') \cdot U = \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & 0 & L_{ik} \cdot U_{kk} & L_{ik} \cdot U_{k,k+1} & \dots & L_{ik} \cdot U_{ki} & \dots & L_{ik} \cdot U_{kn} \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 0 \end{bmatrix} \quad (4.11)$$

$$L \cdot (U - U') = \begin{bmatrix} 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & U_{ki} & 0 & \dots & 0 \\ 0 & \dots & 0 & L_{k+1,k} \cdot U_{ki} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & L_{ik} \cdot U_{ki} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & L_{nk} \cdot U_{ki} & 0 & \dots & 0 \end{bmatrix} \quad (4.12)$$

Conforme mencionado no capítulo anterior, uma das dificuldades para propor uma regra de preenchimento está em se reduzir o número de operações associadas com sua execução, dentro do algoritmo de pré-condicionamento.

Neste trabalho propõe-se usar a norma-1 em (4.7) para o cálculo do erro associado a um elemento de L e a norma-∞ em (4.8) para o cálculo do erro associado a um dado elemento de U, como apresentado em (4.13) e (4.14). As referidas normas aproveitam a estrutura das matrizes (4.11) e (4.12), evitando-se a realização de operações adicionais, sendo requeridas, ao invés disto, apenas comparações entre elementos. Portanto, substituindo (4.11) em (4.13) e (4.12) em (4.14) e calculando-se as normas, são obtidas as equações (4.15) e (4.16) que devem ser usadas na regra de preenchimento.

$$\|R(L_{ik})\| = \|(L - L') \cdot U\|_1 \quad (4.13)$$

$$\|R(U_{ki})\| = \|L \cdot (U - U')\|_\infty \quad (4.14)$$

$$\|R(L_{ik})\| = |L_{ik}| \cdot \max\{|U_{km}|\}, m = [k, \dots, n] \quad (4.15)$$

$$\|R(U_{ki})\| = |U_{ki}| \cdot \max\{1, |L_{mk}|\}, m = [k+1, \dots, n] \quad (4.16)$$

Na Figura 4.3 são destacados em vermelho todos os elementos dos fatores triangulares que são requeridos para calcular o erro produzido quando o elemento  $L_{ik}$  (no quadrado) é descartado. Já na Figura 4.4 são destacados em azul todos os elementos necessários para calcular o erro produzido quando o elemento  $U_{ki}$  (encerrado no quadrado) é descartado. Observa-se que, para que esta regra possa ser implementada tanto no fator triangular U como em L, o algoritmo de fatoração deve ser capaz de calcular os elementos  $U_{kk}$ ,  $U_{k,k+1}$ , ...,  $U_{kn}$  (em vermelho) antes de decidir se o elemento  $L_{ik}$  deve ser preenchido ou descartado. Semelhantemente, deve ser capaz de calcular os elementos  $L_{k+1,k}$ , ...,  $L_{nk}$  (em azul) antes de decidir se o elemento  $U_{ki}$  deve ser preenchido ou descartado.

	1	2	...	k	...	i	...	n
1	$U_{11}$	$U_{12}$	...	$U_{1k}$	...	$U_{1i}$	...	$U_{1n}$
2	$L_{21}$	$U_{22}$	...	$U_{2k}$	...	$U_{2i}$	...	$U_{2n}$
...	...	...	...	...	...	...	...	...
k	$L_{k1}$	$L_{k2}$	...	$U_{kk}$	...	$U_{ki}$	...	$U_{kn}$
k+1	$L_{k+1,1}$	$L_{k+1,2}$	...	$L_{k+1,k}$	...	$U_{k+1,i}$	...	$U_{k+1,n}$
...	...	...	...	...	...	...	...	...
i	$L_{i1}$	$L_{i2}$	...	$L_{ik}$	...	$U_{ii}$	...	$U_{in}$
...	...	...	...	...	...	...	...	...
n	$L_{n1}$	$L_{n2}$	...	$L_{nk}$	...	$L_{ni}$	...	$U_{nn}$

Figura 4.3 – Elementos necessários para calcular o erro associado ao elemento  $L_{ik}$ .

	1	2	...	k	...	i	...	n
1	$U_{11}$	$U_{12}$	...	$U_{1k}$	...	$U_{1i}$	...	$U_{1n}$
2	$L_{21}$	$U_{22}$	...	$U_{2k}$	...	$U_{2i}$	...	$U_{2n}$
...	...	...	...	...	...	...	...	...
k	$L_{k1}$	$L_{k2}$	...	$U_{kk}$	...	$U_{ki}$	...	$U_{kn}$
k+1	$L_{k+1,1}$	$L_{k+1,2}$	...	$L_{k+1,k}$	...	$U_{k+1,i}$	...	$U_{k+1,n}$
...	...	...	...	...	...	...	...	...
i	$L_{i1}$	$L_{i2}$	...	$L_{ik}$	...	$U_{ii}$	...	$U_{in}$
...	...	...	...	...	...	...	...	...
n	$L_{n1}$	$L_{n2}$	...	$L_{nk}$	...	$L_{ni}$	...	$U_{nn}$

Figura 4.4 – Elementos necessários para calcular o erro associado ao elemento  $U_{ki}$ .

## 4.5 Adequação da Regra Baseada no Erro

Nesta seção, a regra de preenchimento baseada no erro é adequada para ser usada no pré-condicionador  $ILUT(\tau, \rho)$ , analisado no capítulo anterior. O pré-condicionador resultante será chamado de pré-condicionador de duplo parâmetro limitante  $ILUT(\xi, \tau)$ . Neste pré-condicionador não é mais necessário usar o parâmetro  $\rho$ .

Conforme visto no Capítulo 3, o algoritmo de fatoração usado neste pré-condicionador é o IKJ e deriva da eliminação de Gauss (Saad, 2003). Este tipo de algoritmo permite calcular os elementos  $U_{kk}, U_{k,k+1}, \dots, U_{kn}$  antes de decidir se o elemento  $L_{ik}$  deve ser preenchido ou descartado. Porém, o algoritmo IKJ não permite calcular os elementos  $L_{k+1,k}, \dots, L_{nk}$  antes de decidir se o elemento  $U_{ki}$  deve ser preenchido ou descartado, porque no mesmo os elementos de  $L$  e  $U$

são calculados por linhas, uma de cada vez. Portanto, a regra baseada no erro pode ser aplicada apenas nos elementos do fator L, enquanto que uma variante da regra convencional usada no  $ILUT(\tau, \rho)$  é usada para os elementos de U.

O cálculo da norma-2 usada na regra de preenchimento convencional, dada pela raiz quadrada da soma dos quadrados dos elementos da linha, pode exigir considerável esforço computacional dependendo da dimensão da matriz de coeficientes. Portanto, ao invés de se usar esta norma para descartar os elementos em U, recomenda-se realizar a média aritmética dos elementos da i-ésima linha da matriz de coeficientes original. Este procedimento reduz o número de operações de ponto flutuante, sem comprometer a qualidade do pré-condicionador. No entanto, a eficiência de ambas as heurísticas é inferior à da regra baseada no erro para preenchimento dos elementos, proposta no presente trabalho. O resultado é um pré-condicionador que aplica dois critérios para definir sua regra de preenchimento, da seguinte forma:

---

De acordo com a relação dada por (4.17), um elemento  $L_{ik}$  é descartado se o erro resultante for menor que a tolerância  $\xi$  usada para todas as linhas de L. Caso contrário, calcula-se  $L_{ik}$  usando-se o elemento  $a_{ik}$  da matriz resultante do processo de eliminação de Gauss até a correspondente etapa. Já para um elemento  $U_{ik}$ , este é descartado se for menor que uma certa tolerância relativa  $\tau_i$ , obtida multiplicando-se a tolerância absoluta  $\tau$  pela média aritmética dos elementos da i-ésima linha da matriz de coeficientes original. O cálculo de  $w$  é efetuado de acordo com o algoritmo de fatoração IKJ apresentado no capítulo anterior.

---

$$L_{ik} = \begin{cases} 0, & \|R(L_{ik})\| < \xi \\ \frac{a_{ik}}{U_{kk}}, & \|R(L_{ik})\| \geq \xi \end{cases} \quad (4.17)$$

$$U_{ik} = \begin{cases} 0, & w_k < \tau_i \\ w_k, & w_k \geq \tau_i \end{cases} \quad (4.18)$$

A adaptação da regra baseada no erro no pré-condicionador  $ILUT(\tau, \rho)$  falha devido ao fato do algoritmo de fatoração IKJ não permitir a aplicação da regra

em ambos os fatores triangulares L e U, dificultando a exploração das vantagens do uso desta regra. A seguir, mostra-se uma proposta para superar estes problemas, utilizando-se outro algoritmo de fatoração que permite aplicar a regra em ambos os fatores triangulares.

#### 4.6 Pré-condicionador ILU( $\xi$ ), Baseado no Cálculo do Erro

Nesta seção propõe-se uma eficiente e robusta estratégia de pré-condicionamento da fatoração incompleta ILU( $\xi$ ), derivada de uma versão do algoritmo de eliminação de Gauss conhecido como Doolittle. A maior vantagem deste algoritmo é que permite usar de forma efetiva a regra de preenchimento baseada no erro em ambos os fatores triangulares L e U.

Os fatores L e U são apresentados em (4.19) e (4.20). Os elementos da diagonal de L são iguais a um e a primeira linha de U é igual à primeira linha da matriz de coeficientes A. As equações usadas pelo algoritmo de Doolittle para calcular os elementos dos fatores triangulares são apresentadas em (4.22) e podem ser obtidas ao se igualar cada elemento do produto dos fatores triangulares L·U de (4.19) e (4.20) aos elementos da matriz de coeficientes A, apresentada em (4.21)

Estas equações podem ser implementadas computacionalmente de diferentes formas, todas baseadas na eliminação de Gauss. Teoricamente, os diferentes tipos de eliminação de Gauss realizam o mesmo número de operações de ponto flutuante. No entanto, no contexto da fatoração incompleta, estas variantes resultam em importantes diferenças práticas quando o interesse é utilizar a regra de preenchimento baseada no erro.

$$L = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ L_{k-1,1} & \cdots & 1 & 0 & \cdots & 0 & \cdots & 0 \\ L_{k1} & \cdots & L_{k,k-1} & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ L_{i1} & \cdots & L_{i,k-1} & L_{ik} & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ L_{n1} & \cdots & L_{n,k-1} & L_{nk} & \cdots & L_{ni} & \cdots & 1 \end{bmatrix} \quad (4.19)$$

$$U = \begin{bmatrix} U_{11} & \cdots & U_{1,k-1} & U_{1k} & \cdots & U_{1i} & \cdots & U_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & U_{k-1,k-1} & U_{k-1,k} & \cdots & U_{k-1,i} & \cdots & U_{k-1,n} \\ 0 & \cdots & 0 & U_{kk} & \cdots & U_{ki} & \cdots & U_{kn} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & U_{ii} & \cdots & U_{in} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & U_{nn} \end{bmatrix} \quad (4.20)$$

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1,k-1} & a_{1k} & \cdots & a_{1i} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{k-1,1} & \cdots & a_{k-1,k-1} & a_{k-1,k} & \cdots & a_{k-1,i} & \cdots & a_{k-1,n} \\ a_{k1} & \cdots & a_{k,k-1} & a_{kk} & \cdots & a_{ki} & \cdots & a_{kn} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & \cdots & a_{i,k-1} & a_{ik} & \cdots & a_{ii} & \cdots & a_{in} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{n,k-1} & a_{nk} & \cdots & a_{ni} & \cdots & a_{nn} \end{bmatrix} \quad (4.21)$$

$$U_{ki} = a_{ki} - \sum_{j=1}^{k-1} [L_{kj} \cdot U_{ji}], \quad i=k, \dots, n$$

$$L_{ik} = \frac{a_{ik} - \sum_{j=1}^{k-1} [L_{ij} \cdot U_{jk}]}{U_{kk}}, \quad i=k+1, \dots, n \quad (4.22)$$

As equações de Doolittle podem gerar algoritmos de eliminação de Gauss que calculam os elementos dos fatores triangulares por linhas, de forma similar à fatoração IKJ, ou por colunas (Bodewig, 1956; Wilkinson, 1965; Dahlquist & Björk, 1974; Hoffman, 2001). Em ambos algoritmos não seria possível usar a regra de preenchimento baseada no erro. A partir de (4.22), propõe-se usar o algoritmo de eliminação de Gauss apresentado na Figura 4.5, que permite calcular os elementos dos fatores triangulares simultaneamente, por linhas em U e por colunas em L, como mostrado na Figura 4.6.

No passo k da eliminação de Gauss, os elementos da k-ésima linha de U e a k-ésima coluna de L (identificados por linhas tracejadas) são calculados utilizando elementos já computados anteriormente nas linhas e colunas de U e L. Por exemplo, para se calcular o elemento  $U_{ki}$  são usados os elementos destacados em azul, enquanto que para calcular o elemento  $L_{ik}$  são usados os elementos

destacados em vermelho. Observa-se que este algoritmo permite calcular elementos de L em linhas posteriores à linha k. Esta característica possibilita o uso em U da regra de preenchimento baseada no erro.

1. Para  $i = 1, \dots, n$  Faça:
2.      $m_{i:n} = J_{i, i:n}$
3.     Para  $k = i, \dots, n$  Faça:
4.         Para  $j = 1, \dots, i-1$  Faça:
5.              $m_k = m_k - L_{ij} \cdot U_{jk}$
6.         Fim – Faça
7.     Fim – Faça
8.     Registrar:  $U_{\max} = \max\{m_{i:n}\}$
9.      $n_{i+1:n} = J_{i+1:n, i}$
10.     Para  $k = i+1, \dots, n$  Faça:
11.         Para  $j = 1, \dots, i-1$  Faça:
12.              $n_k = n_k - L_{kj} \cdot U_{ji}$
13.         Fim – Faça
14.     Fim – Faça
15.      $n = n/U_{ii}$
16.     Registrar:  $L_{\max} = \max\{n_{i+1:n}\}$
17.     Aplicar a regra de preenchimento baseada no erro em: m e n
18.      $U_{i, i:n} = m_{i:n}$
19.      $L_{i+1:n, i} = n_{i+1:n}$
20. Fim – Faça

Figura 4.5 – Algoritmo do pré-condicionador ILU( $\xi$ ).

	1	...	k-1	k	...	i	...	n
1	$U_{11}$	...	$U_{1,k-1}$	$U_{1k}$	...	$U_{1i}$	...	$U_{1n}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
k-1	$L_{k-1,1}$	...	$U_{k-1,k-1}$	$U_{k-1,k}$	...	$U_{k-1,i}$	...	$U_{k-1,n}$
k	$L_{k1}$	...	$L_{k,k-1}$	$U_{kk}$	...	$U_{ki}$	...	$U_{kn}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
i	$L_{i1}$	...	$L_{i,k-1}$	$L_{ik}$	...	$U_{ii}$	...	$U_{in}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
n	$L_{n1}$	...	$L_{n,k-1}$	$L_{nk}$	...	$L_{ni}$	...	$U_{nn}$

Figura 4.6 – Elementos necessários para calcular os elementos  $L_{ik}$  e  $U_{ki}$  durante a fatoração triangular.

No algoritmo da Figura 4.5, observa-se que no passo  $k$  os elementos da linha de  $U$  são armazenados no vetor  $m$ , enquanto que os elementos da coluna de  $L$  são armazenados no vetor  $n$  (passos 5 e 12, respectivamente). Uma vez calculados ambos os vetores, procura-se o maior elemento de  $m$ , sendo armazenado em  $U_{\max}$ , e o maior elemento de  $n$ , armazenado em  $L_{\max}$  (passos 8 e 16, respectivamente). Ambos os valores máximos (em valores absolutos) são usados pela regra de preenchimento, no passo 17, para descartar em  $m$  e  $n$  os elementos associados com os menores erros e preencher a linha de  $U$  e a coluna de  $L$ , da seguinte forma:

---

De acordo com as relações dadas por (4.23) e (4.24), tanto um elemento  $L_{ki}$  como também um elemento  $U_{ik}$  são descartados se seus erros associados foram menores que uma dada tolerância  $\xi$  usada para todas as colunas de  $L$  e linhas de  $U$ . Caso contrário,  $L_{ki}$  é preenchido usando-se o valor armazenado em  $n_k$  e  $U_{ik}$  é preenchido usando-se o valor armazenado em  $m_k$ .

---

$$U_{ik} = \begin{cases} 0, & \|R(U_{ik})\| < \xi \\ m_k, & \|R(U_{ik})\| \geq \xi \end{cases} \quad (4.23)$$

$$L_{ki} = \begin{cases} 0, & \|R(L_{ki})\| < \xi \\ n_k, & \|R(L_{ki})\| \geq \xi \end{cases} \quad (4.24)$$

Como observado na Figura 4.6, o algoritmo da Figura 4.5 permite encontrar os elementos  $U_{kk}$ ,  $U_{k,k+1}$ , ...,  $U_{kn}$ , necessários para se calcular o erro associado ao elemento  $L_{ki}$  e os elementos  $L_{k+1,k}$ , ...,  $L_{nk}$ , que, por sua vez, são necessários para se calcular o erro associado ao elemento  $U_{ik}$ . Nota-se que os elementos necessários para calcular o erro (passo  $k$ ) são calculados neste mesmo passo, portanto, não é necessário realizar grandes buscas de elementos nas matrizes  $L$  e  $U$ . Igualmente, também não é necessário armazenar valores  $U_{\max}$ ,  $L_{\max}$  ou erros do passo  $k$ , porque não serão mais necessários nos próximos passos.

Observe-se que não é necessário multiplicar o parâmetro  $\xi$  por alguma norma ou média aritmética dos elementos da matriz de coeficientes, como realizado no pré-condicionador  $ILUT(\tau, \rho)$ , porque o erro independe dos valores dos elementos

da matriz de coeficientes. Este pré-condicionador usa apenas um único parâmetro  $\xi$ . Esta característica facilita o ajuste dos parâmetros do solucionador, como será visto nas próximas seções e no próximo capítulo.

#### 4.7 Método Convencional Para Ajuste dos Parâmetros dos Solucionadores

O ajuste dos parâmetros de um solucionador iterativo é considerado fundamental para garantir sua robustez e eficiência computacional, principalmente em análises mais complexas, como aquelas que solucionam vários fluxos de carga com diferentes pontos de operação. Nesses casos a escolha de valores apropriados para os parâmetros se torna ainda mais relevante.

A idéia fundamental é escolher os valores dos parâmetros que permitam preencher o menor número de elementos sem prejudicar a qualidade do pré-condicionador, isto é, sem torná-lo muito diferente de  $A$ . Porém, a escolha de valores apropriados para os parâmetros é uma tarefa difícil, já que existe uma interdependência com as tolerâncias relativa e absoluta do GMRES e com o tipo de problema que está sendo solucionado.

Pouco tem sido pesquisado em relação a este tema, apenas em (de Leon & Semlyen, 2002) apresenta-se uma fórmula que permite calcular para cada sistema elétrico o valor do parâmetro limitante  $\tau$  no pré-condicionador ILU( $\tau$ ), dado por (3.22), onde  $\tau$  é igual à metade do número de equações totais  $N_{eq}$  do sistema linear. Na prática, esta fórmula não garante robustez nem o melhor desempenho do solucionador.

$$\tau = \frac{1}{2 \cdot N_{eq}} \quad (4.25)$$

No entanto, o procedimento convencional define primeiro valores arbitrários para as duas tolerâncias absoluta e relativa do GMRES. Nas simulações realizadas nos capítulos anteriores se usou tolerâncias iguais ( $10^{-5}$ ) e menores que a tolerância usada no teste de convergência do algoritmo Newton-Raphson, pois tolerâncias maiores costumam impedir a convergência do Newton-Raphson.

Uma vez definidas as tolerâncias, a escolha dos parâmetros do pré-condicionador é realizada com base em tentativa e erro.

Este método não considera a possibilidade de encontrar melhores desempenhos para outras tolerâncias do GMRES, que não necessariamente devem ser consideradas iguais. Durante o procedimento de tentativa e erro, o método pode encontrar um mínimo número de operações local ( $\text{MFLOPS}_L$ ) associado com  $\tau_L$ , como ilustrado na Figura 4.7, que não necessariamente é o menor possível ( $\text{MFLOPS}_G$ ). Observa-se que para obter o mínimo número de operações global ( $\text{MFLOPS}_G$ ), o procedimento de tentativa e erro deveria começar com um valor de  $\tau$  próximo de  $\tau_G$ .

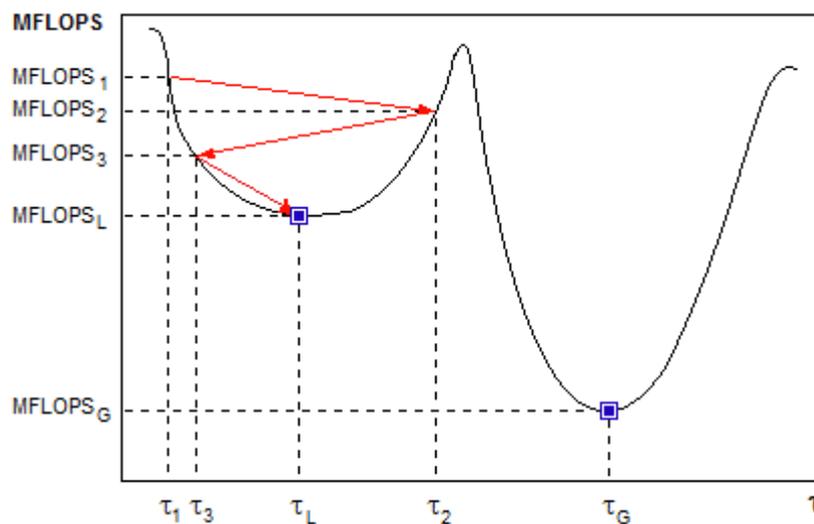


Figura 4.7 – Procedimento de tentativa e erro para busca de valores de um parâmetro do pré-condicionador.

Quando o pré-condicionador possui um segundo parâmetro, o procedimento de tentativa e erro é repetido para este segundo parâmetro fixando-se o primeiro parâmetro.

#### 4.8

#### Método Proposto Para Ajuste dos Parâmetros dos Solucionadores

O método proposto consiste em realizar sucessivas simulações de fluxo de carga variando simultaneamente todos os parâmetros do solucionador e registrando o número de operações de ponto flutuante para cada caso gerado. O objetivo é encontrar regiões de valores para os parâmetros onde o solucionador

apresente alto desempenho, permitindo estabelecer e recomendar faixas para os valores dos parâmetros onde o solucionador garante convergência com a maior eficiência possível. O método também permite identificar as faixas de valores onde o solucionador não converge.

O método é ilustrado para ajustar os parâmetros de duas configurações de solucionador iterativo, sendo estas: o solucionador proposto neste trabalho, identificado como GMRES+ILU( $\xi$ )+MD, e o solucionador GMRES+ILUT( $\tau, \rho$ )+MD, apresentado no capítulo 2. Na Tabela 4.3 especificam-se os parâmetros associados a cada solucionador.

Tabela 4.3 – Parâmetros usados nos solucionadores

Solucionadores	Método Iterativo		Pré-condicionador	
GMRES+ILU( $\xi$ )+MD	atol	rtol	$\xi$	
GMRES+ILUT( $\tau, \rho$ )+MD			$\tau$	$\rho$

#### 4.8.1 Passos Fundamentais

Esta seção apresenta, de maneira sucinta, os passos envolvidos no método para ajuste dos parâmetros dos solucionadores, onde uma descrição geral é mostrada na Figura 4.8. O primeiro passo refere-se à definição do espaço de busca onde são procurados os valores para os parâmetros que garantem alto desempenho do solucionador. O segundo refere-se aos experimentos numéricos realizados para avaliar o desempenho do solucionador considerando-se todas as combinações possíveis dos valores de seus parâmetros, definidos no espaço de busca. O passo seguinte tem como objetivo calcular as faixas de valores recomendáveis para as tolerâncias absoluta e relativa do GMRES. Finalmente, o quarto passo calcula as faixas de valores recomendáveis para os parâmetros do pré-condicionador.

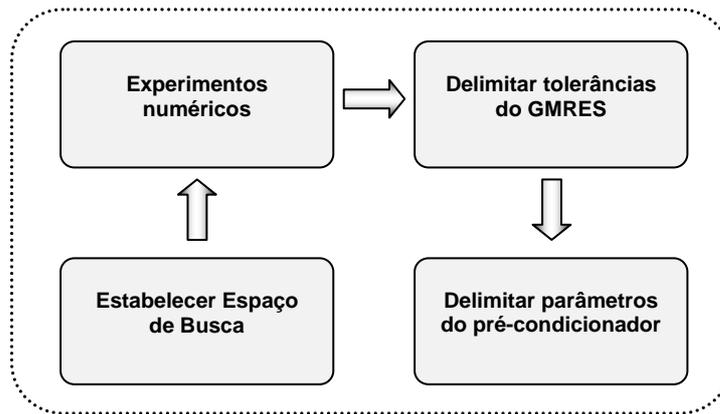


Figura 4.8 – Passos fundamentais para delimitação dos parâmetros do solucionador.

Nas próximas seções, os passos mais relevantes no contexto deste trabalho são discutidos mais detalhadamente.

#### 4.8.2 Estabelecimento do Espaço de Busca

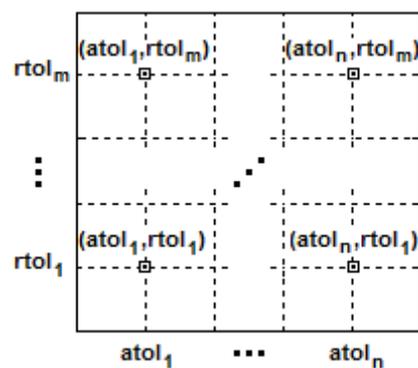
Para definir o espaço de busca, primeiramente deve-se estabelecer o domínio de cada parâmetro, isto é, um valor mínimo e máximo para cada tolerância do GMRES e para cada parâmetro do pré-condicionador, como mostrado na Tabela 4.4 e na Figura 4.9. Depois devem ser definidos valores intermediários para cada parâmetro, sendo que o número de valores totais é definido em função do grau de precisão que se deseja conseguir nos parâmetros.

Finalmente, o espaço de busca é definido por todas as combinações possíveis entre as tolerâncias absoluta e relativa do GMRES, como apresentado na Figura 4.9(a) e por todas as combinações possíveis realizadas entre os parâmetros do pré-condicionador, como mostrado na Figura 4.9(b-c).

Para os valores especificados na Tabela 4.4, existem  $(m \times n)$  combinações possíveis entre a tolerância absoluta e relativa (ver Figura 4.9(a)); e  $(q \times p)$  combinações possíveis entre os parâmetros  $\tau$  e  $\rho$  do pré-condicionador  $ILUT(\tau, \rho)$  (ver Figura 4.9(c)). No caso do pré-condicionador  $ILU(\xi)$  não existem combinações, já que este pré-condicionador possui somente um único parâmetro. Porém, existem  $k$  valores possíveis para  $\xi$  (ver Figura 4.9(b)).

Tabela 4.4 – Domínios dos parâmetros usados nos solucionadores

Parâmetros	Máximo	Mínimo	Número total de valores
atol	$atol_1$	$atol_n$	n
rtol	$rtol_1$	$rtol_m$	m
$\xi$	$\xi_1$	$\xi_k$	k
$\tau$	$\tau_1$	$\tau_q$	q
$\rho$	$\rho_1$	$\rho_p$	P



a) Espaço de busca das tolerâncias do GMRES

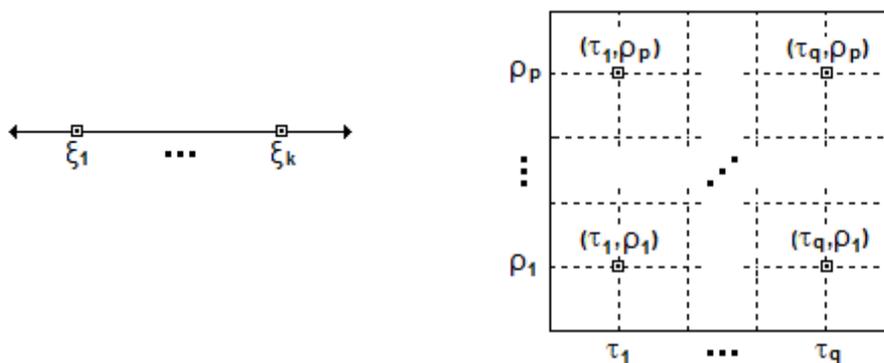
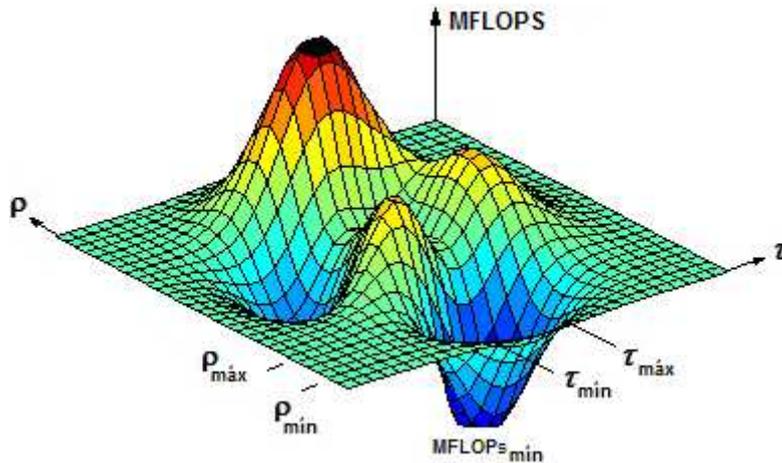
b) Espaço de busca dos parâmetros do pré-condicionador ILUT( $\xi$ )c) Espaço de busca dos parâmetros do pré-condicionador ILUT( $\tau, \rho$ )

Figura 4.9 – Espaço de busca dos parâmetros do solucionador.

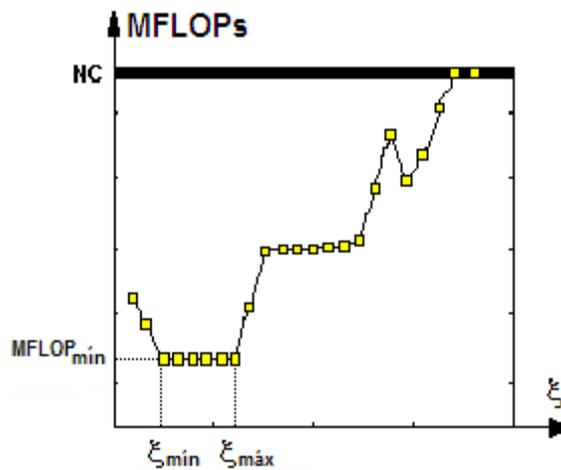
### 4.8.3 Experimentos Numéricos

No segundo passo, para cada combinação das tolerâncias absoluta e relativa ( $atol, rtol$ ) do espaço de busca da Figura 4.9(a) são simulados tantos Fluxos de Carga quanto combinações dos parâmetros do pré-condicionador existam, registrando-se o número de operações de ponto flutuante (MFLOPS) para cada

combinação dos parâmetros do pré-condicionador. Portanto, para cada combinação ( $atol, rtol$ ) é gerado um gráfico de eficiência computacional (em MFLOPS) em função dos parâmetros do pré-condicionador, como mostrado na Figura 4.10(a) para o pré-condicionador ILUT( $\tau, \rho$ ) e, na Figura 4.10(b) para o pré-condicionador ILU( $\xi$ ).



a) Gráfico de eficiência para o pré-condicionador ILUT( $\tau, \rho$ )



b) Gráfico de eficiência para o pré-condicionador ILU( $\xi$ )

Figura 4.10 – Gráficos de eficiência em função dos parâmetros do pré-condicionador

Para cada gráfico gerado é registrado o número de operações de ponto flutuante mínimo ( $MFLOPS_{min}$ ) e este valor é associado à sua correspondente combinação ( $atol, rtol$ ). Após terem sido realizados todos os experimentos e registrados todos os  $MFLOPS_{min}$  associados a todas as combinações ( $atol, rtol$ ) é gerado o gráfico de alta eficiência. Este gráfico apresenta em escala de cores, todos  $MFLOPS_{min}$  em função de suas correspondentes combinações ( $atol, rtol$ ), como mostrado na Figura 4.11, sendo que a cor azul está associada com os menores  $MFLOPS_{min}$ , a

cor vermelha com os maiores  $\text{MFLOPS}_{\min}$  e em negrito os casos não convergentes.

Observa-se que entre todos os valores registrados para  $\text{MFLOPS}_{\min}$  existe uma região que contém os casos com os menores  $\text{MFLOPS}_{\min}$ . Este valor chamado de  $\text{MFLOPS}_{\text{óti}}$  é o número mínimo de operações possível (global) e está associado com a região quadrada em azul escuro, segmentada pelas linhas brancas. Os pontos brancos indicam as posições de todas as combinações que possuem número de operações igual a  $\text{MFLOPS}_{\text{óti}}$ .

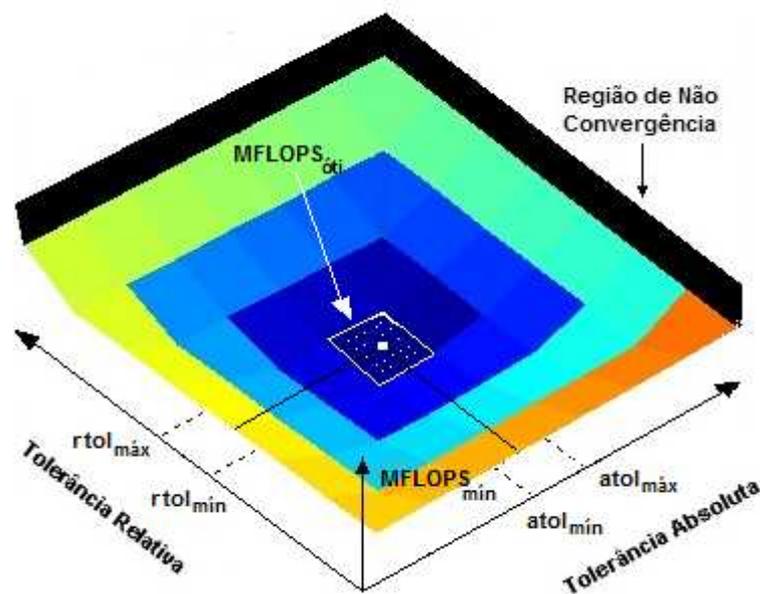


Figura 4.11 – Gráficos de alta eficiência em função das tolerâncias do GMRES

#### 4.8.4 Delimitação das Tolerâncias do GMRES

No terceiro passo, a partir do gráfico de alta eficiência apresentado na sua forma bidimensional na Figura 4.12, é segmentada a região dos  $\text{MFLOPS}_{\text{óti}}$ . Neste caso, esta região é o quadrado azul do centro do gráfico, onde cada ponto branco está associado a uma combinação  $(\text{atol}, \text{rtol})$  que apresenta o menor número de operações possível em todo o espaço de busca ( $\text{MFLOPS}_{\text{óti}}$ ). Este gráfico permite encontrar as faixas de valores  $(\text{atol}_{\min}, \text{atol}_{\max})$  e  $(\text{rtol}_{\min}, \text{rtol}_{\max})$  recomendáveis para as tolerâncias do método GMRES associadas ao melhor desempenho possível.

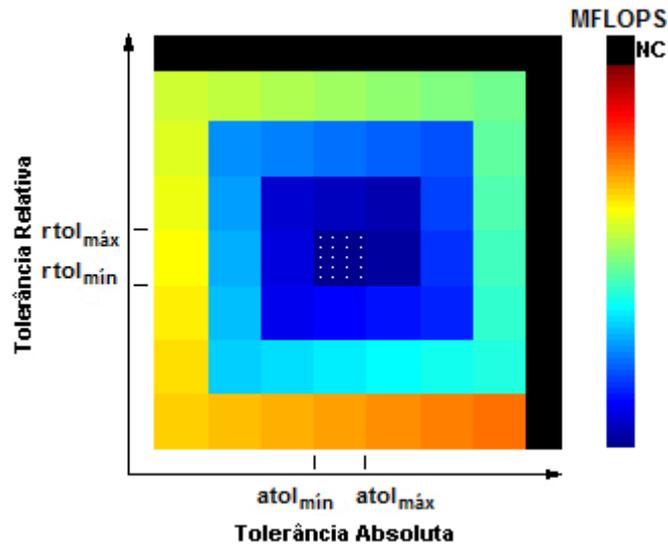


Figura 4.12 – Regiões de alta eficiência em função das tolerâncias do GMRES

#### 4.8.5 Delimitação dos Parâmetros do Pré-condicionador

No quarto e último passo, a partir das faixas  $(atol_{\min}, atol_{\max})$  e  $(rtol_{\min}, rtol_{\max})$  recomendáveis para as tolerâncias do GMRES é escolhido um valor para cada tolerância, isto é,  $atol = atol_{\text{óti}}$  e  $rtol = rtol_{\text{óti}}$  (Figura 4.13). Já que todas as tolerâncias dessas faixas têm o mesmo  $MFLOPS_{\text{óti}}$ , pode-se escolher qualquer um.

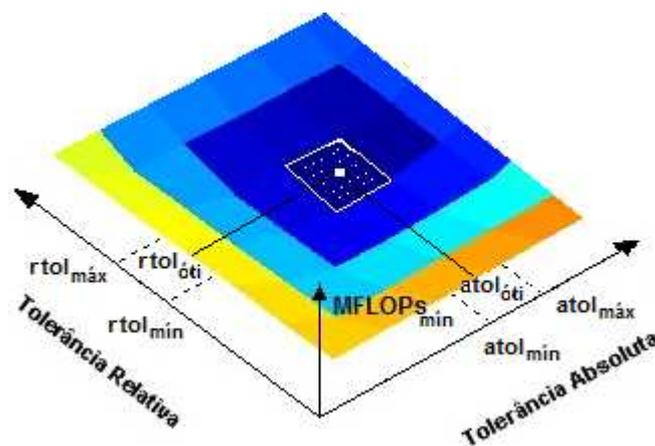


Figura 4.13 – Escolha dos valores  $(atol_{\text{óti}}, rtol_{\text{óti}})$  para as tolerâncias absoluta e relativa do GMRES

Uma vez escolhidos os valores da tolerância absoluta e relativa, utiliza-se o gráfico de eficiência calculado em função dos parâmetros do pré-condicionador (já obtido no passo dois, ver Figura 4.10), associado aos valores  $(atol_{\text{óti}}, rtol_{\text{óti}})$ , escolhidos neste passo. A partir deste gráfico, na sua forma bidimensional

(Figura 4.14) são geradas as regiões de desempenho do solucionador em função dos parâmetros do pré-condicionador, especificamente para os valores das tolerâncias do GMRES escolhidas no início do presente passo ( $\text{atol}_{\text{óti}}, \text{rtol}_{\text{óti}}$ ). As regiões de desempenho são diferenciadas pela escala de cores, sendo que a cor azul está associada com os melhores desempenhos (menores MFLOPS), a cor vermelha com os piores desempenhos (maiores MFLOPS) e em negrito os casos não convergentes.

As regiões de desempenho do solucionador permitem obter as faixas de valores recomendáveis para os parâmetros do pré-condicionador associadas ao melhor desempenho possível  $\text{MFLOPS}_{\text{óti}}$ , sendo estas:  $(\xi_{\text{mín}}, \xi_{\text{máx}})$  para o pré-condicionador  $\text{ILU}(\xi)$ ; e  $(\tau_{\text{mín}}, \tau_{\text{máx}})$  e  $(\rho_{\text{mín}}, \rho_{\text{máx}})$  para o pré-condicionador  $\text{ILUT}(\tau, \rho)$ .

Recomenda-se aplicar este método em diferentes configurações do sistema interligado nacional brasileiro, com diferentes níveis de carga. O método deve gerar, para cada sistema elétrico, faixas de valores para todos os parâmetros do solucionador. A interseção dessas faixas poderá ser considerada confiável para ser usada sempre no solucionador, garantindo robustez e alto desempenho computacional. A eficácia do método pode ser avaliada com experimentos de fluxo de carga e Análise de Contingências. No próximo capítulo, o método proposto é aplicado para a geração de valores recomendáveis dos parâmetros do solucionador iterativo no problema de fluxo de carga.

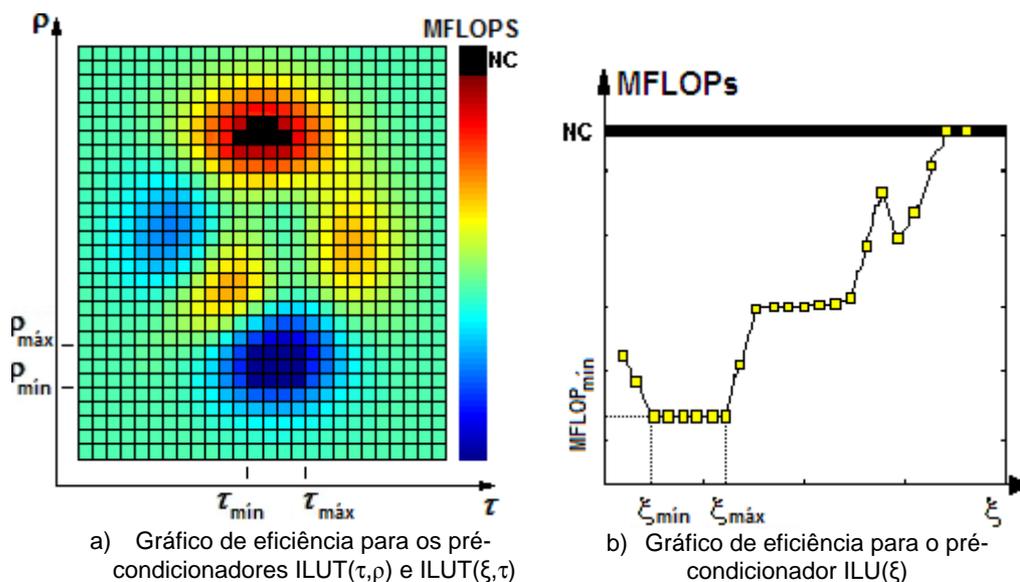


Figura 4.14 – Regiões de eficiência em função dos parâmetros do pré-condicionador