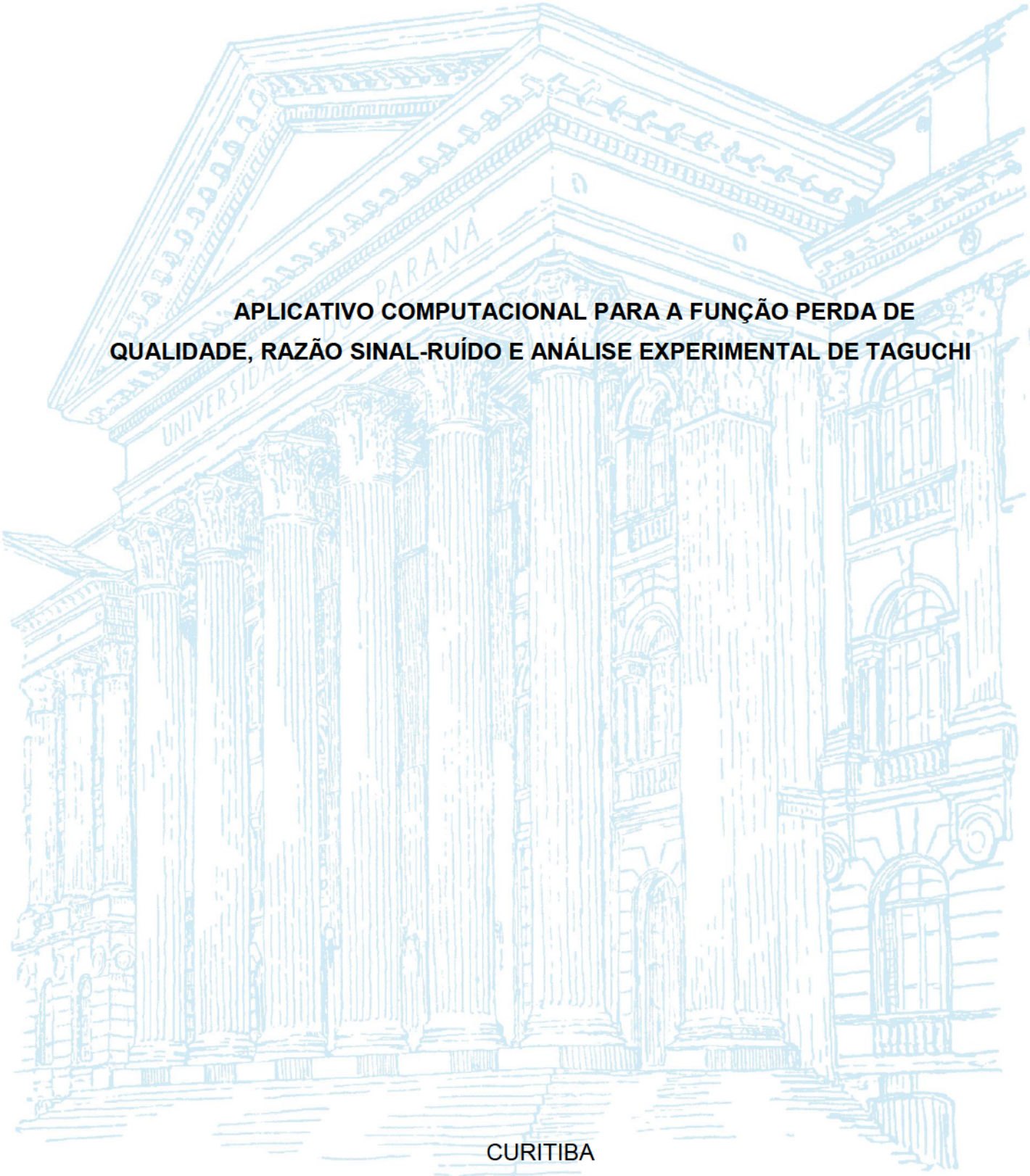
UNIVERSIDADE FEDERAL DO PARANÁ

LUCAS LAMY



APLICATIVO COMPUTACIONAL PARA A FUNÇÃO PERDA DE QUALIDADE, RAZÃO SINAL-RUÍDO E ANÁLISE EXPERIMENTAL DE TAGUCHI

CURITIBA

2018

LUCAS LAMY

APLICATIVO COMPUTACIONAL PARA A FUNÇÃO PERDA DE QUALIDADE, RAZÃO SINAL-RUÍDO E ANÁLISE EXPERIMENTAL DE TAGUCHI

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciências, no Curso de Pós-Graduação em Métodos Numéricos em Engenharia, Setor de Ciências Exatas e Setor de Tecnologia, da Universidade Federal do Paraná.

Orientador: Prof. Dr. Anselmo Chaves Neto

CURITIBA

2018

FICHA CATALOGRÁFICA ELABORADA PELO SISTEMA DE BIBLIOTECAS/UFPR
BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

L242a Lamy, Lucas
Aplicativo computacional para a função perda de qualidade, razão sinal-ruído e análise experimental de taguchi / Lucas Lamy. – Curitiba, 2018.
282 p. : il. color. ; 30 cm.

Dissertação - Universidade Federal do Paraná, Setor de Ciências Exatas e Tecnologia, Programa de Pós-Graduação em Métodos Numéricos em Engenharia, 2018.

Orientador: Anselmo Chaves Neto.

1. Técnica Taguchi. 2. Função perda de qualidade. 3. Aplicativo computacional. I. Universidade Federal do Paraná. II. Chaves Neto, Anselmo. III. Título.

CDD: 519.7

Bibliotecária: Romilda Santos - CRB-9/1214

MINISTÉRIO DA EDUCAÇÃO
SETOR CIÊNCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO MÉTODOS NUMÉRICOS EM ENGENHARIA

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em MÉTODOS NUMÉRICOS EM ENGENHARIA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **LUCAS LAMY** intitulada: **APLICATIVO COMPUTACIONAL PARA A FUNÇÃO PERDA DE QUALIDADE, RAZÃO SINAL-RUÍDO E ANÁLISE EXPERIMENTAL DE TAGUCHI**, após terem inquirido o aluno e realizado a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 02 de Março de 2018.

ANSELMO CHAVES NETO
Presidente da Banca Examinadora (UFPR)

GUSTAVO VALENTIM LOCH
Avaliador Interno (UFPR)

MARCOS AUGUSTO MENDES MARQUES
Avaliador Externo (UFPR)

GERALDO CARVALHO BRITO JUNIOR
Avaliador Externo (UFPR)

AGRADECIMENTOS

Ao professor Dr. Anselmo Chaves Neto, pela orientação, disposição, paciência e empenho para com este trabalho.

Aos meus familiares e minha namorada, por sempre estarem ao meu lado, pelo apoio e incentivo.

Ao Programa de Pós-Graduação em Métodos Numéricos em Engenharia (PPGMNE) da Universidade Federal do Paraná.

Aos colegas do programa de mestrado pela ajuda prestada no decorrer do curso, pelas horas de trabalho e principalmente pelas horas ociosas.

À CAPES pelo apoio financeiro.

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

“Os homens devem moldar o seu caminho. A partir do momento em que você vir o caminho em tudo o que fizer você se tornará o caminho. ”

(MIYAMOTO MUSASHI)

.

RESUMO

Genichi Taguchi ajudou o crescimento da economia japonesa por meio de suas contribuições na Engenharia da Qualidade, mostrando a Função Perda de Qualidade quadrática, Relação Sinal-Ruído e um delineamento de experimento avançado adequado às indústrias. A Função Perda de Qualidade quadrática é uma técnica da Engenharia da Qualidade que procura exprimir conceitos da qualidade em função de perda monetária no que se refere a produtos conformes e não conformes. Já a Razão Sinal-Ruído permite analisar a resposta esperada contra os ruídos presentes e é a medida principal do delineamento de experimento proposto por Taguchi. O seu delineamento de experimento é composto por três etapas: Projeto de Sistema, Projeto de Parâmetros e Projeto de Tolerância. É no Projeto de Parâmetros que é realizado grande parte da otimização do produto, fornecendo a robustez procurada. Um produto robusto é de grande valia, ele apresenta menos variabilidade em sua resposta e maior qualidade com menor custo. Como, existe a dificuldade de obter-se amostras para a estimação dos parâmetros, a técnica computacional intensiva *Bootstrap* permite estimar dos parâmetros, a distribuição, o desvio padrão e intervalos de confiança, a partir de uma única amostra. Este trabalho faz um estudo dos conceitos de Taguchi e utiliza seus parâmetros estatísticos junto com a técnica de reamostragem *Bootstrap* para estimar seus intervalos de confiança. Faz uma análise de um caso na indústria de alimentos, comparando a qualidade de cinco produtos similares. As comparações feitas foram entre turnos de trabalho nos produtos similares e no mesmo processo de fabricação. Um aplicativo "Experimento Taguchi" foi desenvolvido em linguagem *MATLAB* para executar e mostrar todas as etapas deste processo, facilitando o trabalho posterior do experimentador. A comparação do aplicativo foi realizada através da comparação dos resultados com outros softwares que já se encontram no mercado. O aplicativo Experimento Taguchi obteve sucesso na comparação, mostrando resultados precisos e manejo prático.

Palavras-chave: Técnica Taguchi. Função Perda de Qualidade. Razão Sinal-Ruído. Aplicativo Computacional.

ABSTRACT

Genichi Taguchi helped the growth of the Japanese economy through his contributions in quality engineering, showing the quadratic loss function, the signal-to-noise ratio and an advanced experiment design appropriate for industries. The quadratic quality loss function is a quality engineering technique that seeks to express quality concepts in terms of monetary loss for conforming and nonconforming products. The signal-to-noise ratio enables to analyses the expected response with the noises presente and it is the main parameter in experimental design proposed by Taguchi. The experimental design is composed of three stages: system design, parameter design and tolerance design. It is in parameter design which is accomplished the biggest part of the product optimization, providing the robustness. A robust product is of great value, it presents less variability in its response and higher quality with less cost. As there is the difficulty to obtain samples for the estimation of the parameters, the intensive computational technique Bootstrap allows estimate the parameters, distribution, standard deviation and confidence intervals, from a single sample. This work presents some of Taguchi's concepts and provide an evaluation of his statistical with Bootstrap resampling method. Realize an analyze in food industry comparing the quality of five similar products. The comparisons were made between similar products and between work shifts in the same manufacturing process. An application "*Experimento Taguchi*" was developed in MATLAB language to execute and show all the steps of this process, facilitating the posterior work for the experimentalist. The comparison of the app was realized through comparison of the results with other software already on the market. The app "*Experimento Taguchi*" was successful in comparison, showing precise results and practical handling.

Key-words: Taguchi Technique. Quality Loss Function. Signal-to-noise ratio. Computer Application.

LISTA DE FIGURAS

FIGURA 2.1 – CONCENTRAÇÃO DE CORES DE TELEVISORES DOS E.U.A E DO JAPÃO 25
FIGURA 2.2 – PROCESSO CENTRADO 30
FIGURA 2.3 – PROCESSO NÃO CENTRADO 30
FIGURA 2.4 – DIFERENTES PROCESSOS DE C_{pk} 31
FIGURA 2.5 – FUNÇÃO DEGRAU 32
FIGURA 2.6 – FUNÇÃO PERDA DE QUALIDADE QUADRÁTICA DE TAGUCHI ... 34
FIGURA 2.7 – FUNÇÃO PERDA DE QUALIDADE MENOR-É-MELHOR 38
FIGURA 2.8 – FUNÇÃO PERDA DE QUALIDADE MAIOR-É-MELHOR 39
FIGURA 2.9 – FUNÇÃO PERDA DE QUALIDADE ASSIMÉTRICA 41
FIGURA 2.10 – CURVA DA BANHEIRA 48
FIGURA 2.11 – CONTROLE DE QUALIDADE *ON-LINE* E *OFF-LINE* 49
FIGURA 2.12 – ETAPAS DO PROJETO DE PARÂMETRO 53
FIGURA 2.13 – OTIMIZAÇÃO EM DOIS PASSOS 55
FIGURA 2.14 – P-DIAGRAMA 58
FIGURA 2.15 – REPRESENTAÇÃO DE GRÁFICO LINEAR 71
FIGURA 2.16 – GRÁFICO LINEAR $L_4(2^3)$ 72
FIGURA 2.17 – DIFERENTES GRÁFICOS LINEARES PARA $L_8(2^7)$ 72
FIGURA 2.18 – MÉTODO *BOOTSTRAP* 76
FIGURA 4.1 – FLUXOGRAMA DE FUNCIONAMENTO DO PROGRAMA 88
FIGURA 4.2 – JANELA PRINCIPAL 89
FIGURA 4.3 – ALERTA DE ESCOLHA DE NÍVEL E FATORES 90
FIGURA 4.4 – ALERTA ESCOLHA DA MATRIZ EXPERIMENTAL 90
FIGURA 4.5 – JANELA DE OPÇÕES PARA A MATRIZ EXPERIMENTAL 91
FIGURA 4.6 – JANELA PARA ESCOLHA DE INTERAÇÕES 92
FIGURA 4.7 – ALERTA DE IMPOSSIBILIDADE DE CALCULAR AS INTERAÇÕES ENTRE FATORES 92
FIGURA 4.8 – ALERTA DE CONFUSÃO ENTRE FATORES E INTERAÇÃO ENTRE FATORES 92
FIGURA 4.9 – MATRIZ EXPERIMENTAL 93
FIGURA 4.10 – ALERTA DE PARA NÚMERO DE AMOSTRAS 94
FIGURA 4.11 – JANELA DE OPÇÕES PARA ANÁLISE DA VARIÂNCIA 95

FIGURA 4.12 – ANOVA 95
FIGURA 4.13 – ANOVA S/N 95
FIGURA 4.14 – OPÇÕES PARA ANÁLISE GRÁFICA 96
FIGURA 4.15 – ANÁLISE DOS EFEITOS PRINCIPAIS 96
FIGURA 4.16 – ANÁLISE DA INTERAÇÃO DOS EFEITOS PRINCIPAIS 97
FIGURA 4.17 – JANELA DE OPÇÕES PARA ESCOLHA DE CARACTERÍSTICAS DE ANÁLISE 98
FIGURA 4.18 – ALERTA DE SELEÇÃO DA RAZÃO S/N 98
FIGURA 4.19 – JANELA EXPORTAÇÃO DE DADOS 99
FIGURA 4.20 – JANELA EXIBIÇÃO DE RESULTADOS 99
FIGURA 4.21 – ANOM 100
FIGURA 4.22 – ANOM S/N 100
FIGURA 4.23 – JANELA DE MENU DE “AJUDA” 101
FIGURA 4.24 – EXEMPLO DE “AJUDA” 101

LISTA DE GRÁFICOS

GRÁFICO 2.1 – EXEMPLO DO GRÁFICO DE EFEITOS PRINCIPAIS.....67
GRÁFICO 2.2 – INTERAÇÃO ANTISSIMÉTRICA ENTRE FATORES 69
GRÁFICO 2.3 – INTERAÇÃO SINÉRGICA ENTRE FATORES.....70
GRÁFICO 2.4 – INTERAÇÃO PARALELA ENTRE FATORES.....70
GRÁFICO 4.1 – EFEITOS PRINCIPAIS DA RAZÃO S/N – PEACE 110
GRÁFICO 4.2 – INTERAÇÕES ENTRE FATORES DA RAZÃO S/N – PEACE.....110
GRÁFICO 4.3 – EFEITOS PRINCIPAIS RAZÃO S/N – EXPERIMENTO TAGUCHI 111
GRÁFICO 4.4 – INTERAÇÃO RAZÃO S/N FATOR 1 x FATOR 2 – EXPERIMENTO TAGUCHI 111
GRÁFICO 4.5 – INTERAÇÃO RAZÃO S/N FATOR 1 x FATOR 3 – EXPERIMENTO TAGUCHI 112
GRÁFICO 4.6 – INTERAÇÃO RAZÃO S/N FATOR 2 x FATOR 3 – EXPERIMENTO TAGUCHI 112
GRÁFICO 4.7 – ANÁLISE DAS MÉDIAS – EXPERIMENTO TAGUCHI.....113
GRÁFICO 4.8 – INTERAÇÃO FATOR 1 x FATOR 2 – EXPERIMENTO TAGUCHI113
GRÁFICO 4.9 – INTERAÇÃO FATOR 1 x FATOR 3 – EXPERIMENTO TAGUCHI114
GRÁFICO 4.10 – INTERAÇÃO FATOR 2 x FATOR 3 – EXPERIMENTO TAGUCHI 114
GRÁFICO 4.11 – EFEITOS PRINCIPAIS RAZÃO S/N – MINITAB 17.....115
GRÁFICO 4.12 – INTERAÇÕES RAZÃO S/N – MINITAB 17 115
GRÁFICO 4.13 – EFEITOS PRINCIPAIS – MINITAB 17 116
GRÁFICO 4.14 – INTERAÇÕES DAS MÉDIAS – MINITAB 17 116
GRÁFICO 4.15 – EFEITOS PRINCIPAIS RAZÃO S/N – R.....117
GRÁFICO 4.16 – INTERAÇÃO RAZÃO S/N FATOR A x FATOR B – R.....118
GRÁFICO 4.17 – INTERAÇÃO RAZÃO S/N FATOR B x FATOR C – R 118
GRÁFICO 4.18 – INTERAÇÃO RAZÃO S/N FATOR A x FATOR C – R 119
GRÁFICO 4.19 – EFEITOS PRINCIPAIS – R.....120
GRÁFICO 4.20 – INTERAÇÃO FATOR A x FATOR B – R.....121
GRÁFICO 4.21 – INTERAÇÃO FATOR A x FATOR C – R.....121
GRÁFICO 4.22 – INTERAÇÃO FATOR B x FATOR C – R.....122

LISTA DE TABELAS

TABELA 2.1 – CLASSIFICAÇÃO DOS PROCESSOS PELOS ÍNDICES C_p 28
TABELA 2.2 – COMPARAÇÃO DE NÚMERO DE CORRIDAS 59
TABELA 2.3 – MATRIZ $L_8(2^7)$. 60
TABELA 2.4 – MATRIZ $L_4(2^3)$ GENÉRICA COM DUAS OBSERVAÇÕES 62
TABELA 2.5 – EXEMPLO DE ANOM PARA $L_4(2^3)$ 62
TABELA 2.6 – QUADRO DA ANOVA 64
TABELA 2.7 – MATRIZ TRIANGULAR $L_8(2^7)$ 73
TABELA 3.1 – ESPECIFICAÇÕES DOS PRODUTOS 82
TABELA 3.2 – OBSERVAÇÕES DO PRODUTO A 82
TABELA 3.3 – OBSERVAÇÕES DO PRODUTO B 82
TABELA 3.4 – OBSERVAÇÕES DO PRODUTO C 83
TABELA 3.5 – OBSERVAÇÕES DO PRODUTO D 83
TABELA 3.6 – OBSERVAÇÕES DO PRODUTO E 83
TABELA 3.7 – FATORES ANALISADOS DO DISPOSITIVO ELÉTRICO 85
TABELA 3.8 – TABELA EXPERIMENTAL DO DISPOSITIVO ELÉTRICO 85
TABELA 3.9 – NOMENCLATURA DOS FATORES PARA DIFERENTES SOFTWARES 86
TABELA 4.1 – LISTA DE MATRIZES ORTOGONAIS NO PROGRAMA EXPERIMENTO TAGUCHI 90
TABELA 4.2 – RESULTADO DA RAZÃO S/N 102
TABELA 4.3 – RESULTADO DA FUNÇÃO PERDA MÉDIA DE QUALIDADE 102
TABELA 4.4 – RESULTADO DO ÍNDICE C_p 102
TABELA 4.5 – RESULTADO DO ÍNDICE C_{pk} 102
TABELA 4.6 – RESULTADO DO ÍNDICE C_{pm} 103
TABELA 4.7 – RESULTADO DO INTERVALO DE CONFIANÇA PARA A RAZÃO S/N 95% 103
TABELA 4.8 – RESULTADO DO INTERVALO DE CONFIANÇA PARA A RAZÃO S/N 99% 103
TABELA 4.9 – RESULTADO DO INTERVALO DE CONFIANÇA DA FUNÇÃO PERDA MÉDIA DE QUALIDADE 95% 104

TABELA 4.10 – RESULTADO DO INTERVALO DE CONFIANÇA DA FUNÇÃO PERDA DE QUALIDADE MÉDIA 99% 104
TABELA 4.11 – RESULTADO DO INTERVALO DE CONFIANÇA C_p 95% 104
TABELA 4.12 – RESULTADO DO INTERVALO DE CONFIANÇA C_{pk} 95% 104
TABELA 4.13 – RESULTADO DO INTERVALO DE CONFIANÇA C_{pm} 95% 105
TABELA 4.14 – RESULTADO DO INTERVALO DE CONFIANÇA C_p 99% 105
TABELA 4.15 – RESULTADO DO INTERVALO DE CONFIANÇA C_{pk} 99% 105
TABELA 4.16 – RESULTADO DO INTERVALO DE CONFIANÇA C_{pm} 99% 105
TABELA 4.17 – COMPARAÇÃO DOS RESULTADOS DA RAZÃO S/N 107
TABELA 4.18 – COMPARAÇÃO DA ANOM S/N 107
TABELA 4.19 – COMPARAÇÃO DA ANOM 108
TABELA 4.20 – COMPARAÇÃO DA ANOM PARA FATORES DE RUÍDO 109

LISTA DE ABREVIATURAS E SIGLAS

ANOVA	– *Analysis of Variance*
ASI	– *American Supplier Institute*
ASQ	– *American Society for Quality*
BC	– *Bias Corrected*
BCa	– *Bias Corrected and Accelerated*
CEP	– Controle Estatístico de Processo
dB	– Decibel
ECL	– *Electrical Communication Laboratory*
LI	– Limite Inferior
LIE	– Limite Inferior de Especificação
LS	– Limite Superior
LSE	– Limite Superior de Especificação
LTB	– *Larger-the-Better*
MSD	– *Mean Square Deviation*
NTB	– *Nominal-the-Best*
OFF-LINE	– *Off the Line*
ON-LINE	– *On the Line*
PPM	– Parte por Milhão
S/N	– Sinal-Ruído
STB	– *Smaller-the-Better*

LISTA DE SÍMBOLOS

$\hat{\theta}_i^*$ – Estátística *Bootstrap*

x_i^* – Amostra *Bootstrap*

$\hat{\theta}$ – Estimador da estatística genérica

σ^2 – Variância

ϕ^{-1} – Distribuição normal inversa

P – Percentílico

t – t de Student

T – Alvo de projeto

Z – Normal

Δ – Tolerância do valor alvo

P – Valor de substituição de peça

k – Constante de proporcionalidade

ε – Erro experimental

θ – Estatística genérica

μ – Média amostral

σ – Desvio padrão

ϕ – Distribuição normal padrão

SUMÁRIO

1 INTRODUÇÃO 18
1.1 OBJETIVOS 19
1.1.1 Objetivo Geral 19
1.1.2 Objetivos Específicos 19
1.2 JUSTIFICATIVA 20
1.3 ESTRUTURA DA DISSERTAÇÃO 20
2 REVISÃO DE LITERATURA 21
2.1 O QUE É QUALIDADE 21
2.1.1 Uma breve história sobre Taguchi 23
2.1.2 Taguchi e a Qualidade 24
2.2 CAPACIDADE POTENCIAL DO PROCESSO 26
2.2.1 Processo Centrado 27
2.2.2 Processo Unilateral 28
2.2.3 Centralidade de um Processo 29
2.2.4 Processo Não-Centrado 30
2.3 FUNÇÃO PERDA DE QUALIDADE 32
2.3.1 Função Perda de Qualidade Nominal-é-Melhor (NTB) 35
2.3.2 Função Perda de Qualidade Menor-é-Melhor (STB) 37
2.3.3 Função Perda de Qualidade Maior-é-Melhor (LTB) 38
2.3.4 Função Perda Assimétrica 40
2.4 RAZÃO SINAL-RUÍDO 41
2.4.1 Razão Sinal-Ruído para o caso Menor-é-Melhor. 43
2.4.2 Razão Sinal-Ruído para o caso Maior-é-Melhor 44
2.4.3 Razão Sinal-Ruído para o caso Nominal-é-Melhor 44
2.5 PROJETO ROBUSTO 45
2.5.1 Fatores de Controle 45
2.5.2 Fatores de Ruído 46
2.5.3 Controle de Qualidade *on-line* e *off-line* 48
2.5.4 Projeto de Sistema 51
2.5.5 Projeto de Parâmetros 51
2.5.6 Otimização em Dois Passos 54
2.5.7 Projeto de Tolerância 56

2.5.8 Vantagens do Projeto Robusto 56
2.6 PROJETO DE EXPERIMENTO 57
2.6.1 Matriz Ortogonal 59
2.6.2 Modelo Aditivo 61
2.6.3 Análise das Médias (ANOM) 61
2.6.4 Análise da Variância (ANOVA) 63
2.6.5 Modelo Preditivo 65
2.6.6 Teste de Verificação 66
2.6.7 Gráfico dos Efeitos Principais 67
2.6.8 Interações entre fatores 67
2.6.9 Gráficos Lineares 71
2.6.10 Matriz triangular 72
2.7 BOOTSTRAP 73
2.7.1 A técnica *Jackknife* 73
2.7.2 O Método *Bootstrap* 75
2.7.3 Intervalo de Confiança 77
2.7.4 Intervalo de Confiança *Bootstrap* 78
3 MATERIAL E MÉTODOS 82
3.1 PROBLEMA COM DADOS REAIS 82
3.2 COMPARAÇÃO DO APLICATIVO 84
3.2.1 Problema da Literatura 84
3.2.2 Nomenclatura entre Programas 86
4 RESULTADOS E DISCUSSÃO 87
4.1 DESCRIÇÃO DO PROGRAMA EXPERIMENTO TAGUCHI 87
4.1 RESULTADO DOS DADOS REAIS 102
4.1.1 Discussão dos Resultados 106
4.2 COMPARAÇÃO DO APLICATIVO EXPERIMENTO TAGUCHI 107
4.1.2 Resultado do Problema do Tempo de Mudança de Estado. 107
4.1.3 Resultado Final 122
4.1.4 Consideração sobre os Softwares utilizados. 122
5 CONCLUSÃO E CONSIDERAÇÕES FINAIS 124
5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS 125
REFERÊNCIAS 126
APÊNDICE A – PROGRAMA MATLAB 131

ANEXO A – MATRIZES ORTOGONAIS, MATRIZES TRIANGULARES E GRÁFICOS LINEARES 259

1 INTRODUÇÃO

Nas últimas décadas, a importância da qualidade tem crescido e não só a importância da qualidade em si, mas os consumidores começaram a levar em consideração se a empresa possui algum engajamento ambiental/social. Isto é, se o produto que consomem não traz prejuízo para a sociedade em termos de desperdício de matéria prima devido à falta de qualidade na produção dos itens.

Ainda focando no lado financeiro de um produto, existem várias empresas que fornecem o mesmo produto no mercado, com diferentes de preços. Assim, a competitividade entre as empresas é muito alta. Consequentemente, para as empresas que desejam se destacar e ganhar vantagem entre suas similares, é necessário um investimento maior na qualidade de seus produtos, pois isso implicará em redução de custo.

Então, se pode observar a qualidade como uma lista de propriedades que um produto deveria ter, como Garvin (1987) ou Feigenbaum (1991), que entendem que a qualidade baseia-se apenas na determinação e na experiência dos consumidores. De outro modo, Taguchi apresenta a ideia de qualidade como a perda que um produto causa à sociedade depois de ser produzida. A ideia de qualidade de Taguchi é diferente da tradicional e ainda vai ao encontro com o conceito de qualidade da atualidade.

Na mesma linha de pensamento de Deming (1982), que dá ênfase em métodos estatísticos, Taguchi criou novas formas de medir, comparar e melhorar a qualidade. Assim, Taguchi separa os processos de controle de qualidade em duas áreas: "*on-line*", isto é, na linha de produção onde são aplicadas as técnicas de controle estatístico de processo (CEP) e "*off-line*", fora da linha de produção, onde há o projeto de experimento e a fase otimização de parâmetros. É na área *off-line* que Genichi Taguchi propôs a aplicação da Função Perda de Qualidade quadrática, da Razão Sinal-Ruído e do Projeto Robusto.

O planejamento de experimento de Taguchi, é mais adequado à realidade industrial, pois existe a dificuldade de se obter dados amostrais razoáveis da linha de produção. Em experimentos industriais, a exigência dos gerentes de produção é de poucas observações quase sempre. Então, para o Projeto Robusto, o planejamento experimental é uma fase essencial para qualquer projeto de produto ou processo. É utilizado para saber quais variáveis possuem maior influência no resultado e na

decisão de valores dos parâmetros estudados, de modo a deixar o produto com uma qualidade maior e mais adequado à demanda do produtor e cliente. Taguchi utiliza as matrizes ortogonais em sua análise experimental, pois são matrizes mais eficientes, em relação às matrizes fatoriais completas.

Considerando a limitação de amostragem, é de grande interesse possuir uma técnica de reamostragem que permita a estimação de distribuição dos parâmetros e intervalos de confiança, para que assim se possa inferir de modo acurado nas decisões da produção.

O *Bootstrap* é uma técnica estatística não paramétrica computacionalmente intensiva de reamostragem, introduzida em 1979 por Badley Efron, que tinha como objetivo inicial estudar a variabilidade de estatísticas com base em uma única amostra existente (CHAVES NETO, 1991). A técnica tem como vantagem, em relação a outras técnicas de reamostragem, a grande generalidade em sua aplicação.

Dessa forma, para aplicar a técnica *Bootstrap* é necessário gerar um número muito grande de reamostras. De modo que, o avanço tecnológico de *softwares* e computadores cada vez mais potentes, facilitou a aplicação da técnica.

Então, quanto ao ponto de vista da exigência em qualidade e pelo período atual onde, a tecnologia é mais evoluída e com maior acessibilidade, o desenvolvimento de um aplicativo possibilita uma execução precisa dos cálculos exigidos junto à disseminação dos temas abordados nesse trabalho.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um aplicativo computacional na linguagem *MATLAB* que possibilite aplicar as técnicas da Função Perda de Qualidade, Razão Sinal-Ruído e análise experimental de Taguchi; que gere os intervalos de confiança desses parâmetros por meio da técnica computacionalmente intensiva *Bootstrap*.

1.1.2 Objetivos Específicos

Os objetivos específicos do trabalho são:

1. Elaborar um estudo sobre os conceitos de: Capacidade de Processo, Técnica *Bootstrap*, Função Perda de Qualidade de Taguchi, Razão Sinal-Ruído e Projeto Robusto.
2. Desenvolver um programa computacional que facilite à aplicação das técnicas abordadas neste estudo.
3. Utilizar dados reais na comparação da qualidade de produtos obtidos de turnos diferentes de produção e na comparação de qualidade entre produtos similares da mesma.
4. Comparar os resultados do aplicativo desenvolvido com outros *softwares* que já existem no mercado.

1.2 JUSTIFICATIVA

A aplicação das técnicas da Função Perda de Qualidade de Taguchi, da Razão Sinal-Ruído e, consequentemente, do Projeto Robusto ainda é incipiente na indústria brasileira. Um programa computacional que facilite os cálculos da aplicação virá facilitar o emprego desse conhecimento pela indústria e consequentemente sua disseminação.

1.3 ESTRUTURA DA DISSERTAÇÃO

Além desta introdução, esta dissertação apresenta uma fundamentação teórica no capítulo 2, que é composta de uma revisão de literatura onde são utilizados os principais livros da área e vários artigos. Os temas abordados são: Qualidade, Função Perda de Qualidade, Razão Sinal-Ruído, Capacidade Potencial do Processo, Projeto Robusto, Projeto de Experimento e a Técnica *Bootstrap*. No capítulo 3 são descritos o material e os métodos a serem usados. No capítulo 4 são discutidos os resultados referentes aos objetivos propostos, incluindo o programa desenvolvido No capítulo 5, denominado de considerações finais, é apresenta a conclusão e, adiante, se encontra as referências bibliográficas, um apêndice e um anexo.

2 REVISÃO DE LITERATURA

2.1 O QUE É QUALIDADE

Para Ghobadian (1994) o termo "qualidade" tem significado diferente para cada pessoa. Isto é, por que definir "qualidade" é muitas vezes o primeiro passo para uma possível melhoria na qualidade. Um entendimento e uma visão comum do que se entende por "qualidade" ajuda organizações a focar esforços na sua melhoria. Assim, a definição de "qualidade" não só é importante do ponto de vista semântico, mas é importante e necessária para direcionar os esforços para uma determinada causa comum.

Para se entender o que é qualidade, é possível colocar o foco no produto/serviço e listar as propriedades que são desejáveis de se possuir, como Garvin (1987), que dimensionou a qualidade em 8 propriedades: performance, confiabilidade, durabilidade, manutenção, estética, características, qualidade percebida e conformidade com as normas. Então, as descrições dessas propriedades são:

- Performance: diz respeito à característica principal do produto, ou para o que ele foi feito. Frequentemente está relacionada à prontidão de realizar o serviço, as propriedades estão relacionadas a preferências subjetivas;
- Características: são muitas vezes propriedades secundárias, onde se faz chamar atenção, ou não, nos produtos e serviços; características que complementam seu funcionamento básico e função principal;
- Confiabilidade: diz sobre a probabilidade de o produto apresentar um mau funcionamento em um período específico de tempo;
- Durabilidade: uma média de duração da vida do produto no qual não pode ser mais reparado;
- Manutenção: refere-se à velocidade, cortesia, competência e facilidade de reparação. Os consumidores estão preocupados não só com o produto, mas também com o tempo em que o serviço será restabelecido, com que os compromissos de serviço são mantidos, as relações com a empresa que oferece o produto ou serviço e com a frequência que o fabricante não consegue corrigir problemas;

- Qualidade percebida: é a qualidade que o cliente irá perceber. Como os consumidores nem sempre possuem informações completas sobre os atributos técnicos de um produto ou serviço, esse tipo de medida indireta pode constituir a sua única base para a comparação de marcas;
- Estética: são aspectos mais subjetivos, seria como o produto aparente ser, como é sentido e etc. – muito ligada a percepções pessoais;
- Conformidade com as normas: está relacionado com a parte técnica e legal que um produto deve cumprir, ou seja, o grau em que o projeto do produto cumpre as características operacionais atendendo aos padrões e normas estabelecidos.

Também é possível definir qualidade com uma visão direcionada ao cliente, Feigenbaum (1991) diz:

> [...] qualidade é uma determinação do cliente, não uma determinação da engenharia, não uma determinação do marketing ou da gerência. Está baseado na experiência do cliente com o produto ou serviço, avaliado pelo requerimento dele – declarado ou não, consciente ou meramente sentido, tecnicamente operacional ou inteiramente subjetivo – e sempre representando um alvo em movimento em um mercado competitivo. [...] (FEIGENBAUM, 1991).

Juran e Godfrey (1998) complementam dizendo que qualidade significa obter as características dos produtos que atendem às necessidades dos clientes e assim, proporcionar satisfação. Portanto, o objetivo de se obter uma qualidade superior é fornecer maior satisfação ao cliente. Nesse sentido, a definição de qualidade é relacionada à renda, de tal modo que, aumentar a qualidade e as características desejadas, significa maior custo.

Um segundo ponto de vista colocado por Juran e Godfrey (1998) é qualidade livre de erros, isso é, livre de retrabalho, falhas e insatisfação de clientes. Nesse sentido qualidade significa menor custo.

Montgomery (2009) diz que a maioria das pessoas que não sabem o conceito de qualidade a relaciona com uma ou mais características que o produto deveria ter. Assim, ele define a qualidade de duas formas: “qualidade é aptidão para uso” e “qualidade é inversamente proporcional à variabilidade”.

Almeida e Toledo (1992) conseguem agrupar as ideias apresentadas anteriormente identificando que existem três vertentes quando se fala sobre

qualidade: Qualidade como adequação para uso, qualidade enquanto conformidade com especificações e qualidade associada às perdas que um produto impõe à sociedade após a sua venda. Esses autores afirmam:

> [...] poderíamos dizer que sob a denominação "qualidade" estão abrigadas duas grandes vertentes: uma mercadológica, que corresponde à qualidade de adequação do produto ao seu uso; e uma outra técnica, que associa qualidade à conformidade do produto com especificações. A primeira está diretamente associada à qualidade ao mercado e a segunda à qualidade na produção. Assim, ao mesmo tempo em que se busca o atendimento às necessidades da demanda da maneira mais plena possível, procura-se também garantir que os produtos atenderão essas necessidades com eficiência na produção, compondo então a qualidade do produto propriamente dita e do processo que o produz. [...] (ALMEIDA; TOLEDO, 1992, p. 25).

Para se falar sobre a terceira vertente tem-se que abordar Taguchi.

2.1.1 Uma breve história sobre Taguchi

Uma bibliografia com base em Creveling e Fowlkes (1995) e ASQ (2017) afirma que Genichi Taguchi nasceu em 1924 em Takamachi, Japão, uma cidade famosa pela indústria de quimono. Logo era esperado que ele assumisse a responsabilidade dos negócios familiares de quimono. Entretanto, em 1942, durante a II Guerra Mundial, Taguchi foi convocado a participar do Departamento Astronômico do Instituto de Navegação da Marinha Imperial Japonesa, onde apareceu o seu interesse em estatística. No período do pós-guerra ele foi orientado pelo professor Matosaburo Masuyama e trabalhou para o Instituto de Estatística Matemática do Japão.

Em 1949, o *Electrical Communication Laboratory* (ECL) da empresa japonesa Telephone & Telegraph queria modernizar o sistema de telecomunicações do país. Conhecido por suas habilidades de engenharia e seus métodos envolvendo planejamento de experimentos, Genichi Taguchi tornou-se responsável pela produtividade do Departamento de Pesquisa e Desenvolvimento da ECL. Neste período Taguchi começou a desenvolver suas ideias sobre Engenharia da Qualidade.

Em 1962 obteve seu doutorado, e escreveu uma segunda edição do livro "*Design of Experiments*" onde introduz sua pesquisa industrial sobre a Razão Sinal-Ruído.

Taguchi recebeu o *Deming Award* quatro vezes, um importante prêmio japonês na área da qualidade. O imperador do Japão o reconheceu por suas contribuições à produtividade japonesa e entrou no *Hall of Fame* dos engenheiros prestigiados.

A Universidade de Aoyama Gakuin no Japão convidou Taguchi para lecionar, em 1965. Taguchi trabalhou 17 anos na universidade e ajudou a desenvolver o Departamento de Engenharia.

Taguchi visitou os Estados Unidos da América (E.U.A) em 1980 para ajudar a indústria de telecomunicações americana *AT&T Bell Laboratories,* em New Jersey, a aplicar seu método. Logo em seguida várias empresas se interessaram pelos seus métodos e começaram a aplicá-lo. Os casos mais notáveis foram os das indústrias Ford Motor e Xerox.

Taguchi ajudou a fundar o *American Supplier Institute (ASI)*, uma corporação criada para promover o desenvolvimento da indústria americana.

Taguchi foi diretor executivo do ASI, lecionou, prestou consultorias, escreveu artigos e livros até o seu falecimento em 2012.

2.1.2 Taguchi e a Qualidade

"Qualidade é a perda imposta à sociedade a partir do momento em que o produto é expedido", Taguchi *et al.* (2005, p.171)

Essa definição apresentada por Taguchi é diferente das outras apresentadas anteriormente, pois é comum associar a palavra qualidade a uma característica positiva. Taguchi interpreta a qualidade de uma maneira mais operacional e atribui a ela a característica de perda.

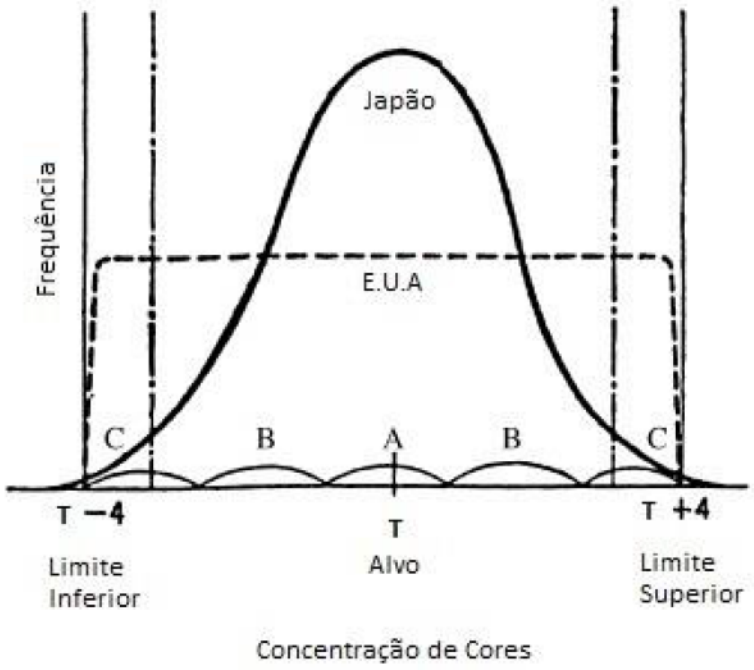
Outro aspecto inovador é que a definição engloba toda a sociedade como sendo a atingida pela perda de qualidade de um produto. Um exemplo disso é o apresentado por Paladini (2010): um consumidor está insatisfeito com o seu carro, pois ele apresenta baixo rendimento. Assim o prejuízo imposto ao grupo social onde ele vive, que é sua vizinhança, é o barulho que o carro apresenta e os danos à sociedade é a poluição causada ao meio ambiente. O dano causado ao fabricante é o desgaste da imagem da empresa produtora do carro pela má qualidade do produto, isso pode impor perdas em futuros negócios.

Além de incorporar a perda no conceito de qualidade, Taguchi percebeu que poderia avaliar a qualidade de um produto considerando, para determinada característica observada x, o desvio d em relação ao alvo projetado (T), $d = x - T$.

Considere o caso da concentração de cores das televisões fabricadas no Japão e nos E.U.A, FIGURA 2.1. Os consumidores preferiam os televisores fabricados no

Japão, ao invés dos americanos, embora os projetos de fabricação fossem os mesmos. Isso se deve ao fato que os televisores fabricados no Japão possuíam menor variabilidade, isto é, os televisores apresentavam melhor concentração de cores, ou seja, os desvios dos alvos projetados eram menores.

FIGURA 2.1 – CONCENTRAÇÃO DE CORES DE TELEVISORES DOS E.U.A E DO JAPÃO



FONTE: Adaptado de Taguchi *et al.* (2005).

Para Taguchi, a qualidade deve ser incorporada ao produto na fase de projeto. A falta de qualidade de um produto não pode ser melhorada ou arrumada através de inspeções após ou durante a produção. Dessa forma o produto é planejado para ser robusto, isto é, imune aos fatores externos não controláveis como, por exemplo, a temperatura, a umidade relativa do ar, o calor, a vibração ou qualquer outro fator a que o produto estará exposto no decorrer do seu uso.

Uma tentativa eficiente para reduzir custos é produzir um produto que possui variabilidade reduzida, medida pelo desvio padrão, que está centrado no seu valor nominal e que, ainda, seja insensível aos fatores externos a que esteja sujeito no uso. A aplicação das técnicas desenvolvidas por Taguchi consegue um produto com qualidade superior e com custos reduzidos, como aponta Marques (2015):

> Sendo assim, de acordo com a filosofia de Taguchi um produto com qualidade superior é barato, com alta confiabilidade e desempenho consistente, pois os fatores causadores da variação funcional de alguma característica do mesmo são minimizados. (MARQUES, 2015, p. 10).

Assim ele dá um passo atrás e recua da produção para a concepção, separa o controle de qualidade em duas fases: *off-line* e *on-line*. A fase *on-line* ocorre na linha de produção e, então, aplica-se o Controle Estatístico de Processo (CEP). Já a fase *off-line* é aquela na qual é realizado todo o projeto do produto para torná-lo robusto, otimizando-o contra a variação tanto do sistema produtivo quanto das condições de uso.

Durante o desenvolvimento do projeto de produto, Taguchi propõe as seguintes etapas, que serão descritas com detalhes adiante:

- Projeto de Sistema;
- Projeto de Parâmetros;
- Projeto de Tolerância.

2.2 CAPACIDADE POTENCIAL DO PROCESSO

Na atualidade é mais comum avaliar o desempenho de um processo de produção, quanto a determinada característica de qualidade, pelo índice de Capacidade Potencial do Processo (C_p), que é a razão entre a amplitude de especificação de projeto e a amplitude do processo.

A quantificação da "capacidade" de um processo de fabricação é um passo básico importante em qualquer programa de melhoria de qualidade. As empresas que não possuem algum controle mínimo de qualidade, como a utilização da Capacidade Potencial do Processo, não têm o controle do seu processo de produção, ou seja, não sabem se a produção está dentro dos padrões desejados e, consequentemente, geram desperdício de matéria prima e mão de obra. Neste contexto os produtos são mais caros e a perda de competitividade no mercado é a consequência. Como aponta Taguchi (1990) “Como as peças não conformes não são expedidas para o cliente, o consumidor não é diretamente afetado a não ser pelo aumento do preço”.

Observando essa necessidade da indústria de controle básico de qualidade os índices de Capacidade Potencial do Processo foram introduzidos por Juran *et al.* em 1974 no seu livro *Quality Control Handbook*.

Bayeux (2001) descreve a Capacidade Potencial do Processo como:

> Os índices de capacidades de processos medem o quanto o processo em questão é capaz de atender as especificações estabelecidas. Quanto maior for o índice, mais o processo é capaz de satisfazer as especificações. Em outras palavras, isso implica na análise da estabilidade e variabilidade do processo, bem como o exame de sua posição relativa aos limites e centro do campo de tolerância da característica de interesse. Assim, o estudo da capacidade do processo visa a determinar o comportamento (existente ou desejável) do processo. (BAYEUX, 2001).

Em outras palavras Costa *et al.* (2010) explicam que essa capacidade depende apenas das próprias especificações e da variabilidade do processo; portanto, ela não está apenas vinculada à presença ou ausência de causas especiais (uma vez que o processo, mesmo na ausência de causas especiais, já possui uma variabilidade natural), embora seja evidente que o desajuste e/ou falta de estabilidade do processo (provocado por causas especiais) reduzem sua capacidade e aumentam o número de itens não conformes produzidos.

Desse modo, a utilização desses índices deve ser acompanhada de alguns cuidados estatísticos. Deve-se tomar cuidado com as condições em que se toma a amostra para que não seja comprometida a representatividade e aleatoriedade do processo de amostragem.

Além disso, deve-se ter evidências a respeito da estabilidade (ausência de causas especiais) do processo produtivo e saber se o processo apresenta uma distribuição que se configura segundo a curva normal.

Como consequência do não cumprimento dessas premissas, os índices não são representativos da capacidade do processo em questão. Essa análise levará a resultados incoerentes ao comportamento real do processo. Logo, deve-se obter as amostras de análise sob três importantes princípios:

- O processo deve estar sob controle estatístico;
- A característica de qualidade observada deve ser unidimensional;
- A característica de qualidade deve possuir distribuição normal.

2.2.1 Processo Centrado

O índice C_p, dado pela equação (2.1), é definido pela razão entre medidas de dispersão, são elas: a dispersão de especificação de projeto e a dispersão real do processo. O numerador deste indicador é a diferença entre os limites de

especificação, isso é, a diferença entre o limite superior de especificação (LSE) e o limite inferior de especificação (LIE). O denominador é a diferença entre a estimativa do alvo do processo (T) mais três desvios padrões com a estimativa do alvo do processo (T) menos três desvios padrões, ou seja, seis desvios padrões.

Dessa forma, o índice C_p possui especificação bilateral, a média está centrada no meio dos seus limites de especificação.

$$C_p = \frac{LSE - LIE}{6\sigma}. \tag{2.1}$$

Como aponta Chaves Neto (2017) à medida que o processo aumenta a sua variabilidade o índice C_p sofre a influência por meio do aumento do desvio padrão, diminuindo o valor da capacidade do processo. O inverso, evidentemente, também acontece e à medida que se elimina as fontes de variabilidade do processo este se torna mais uniforme (padronizado), com desvio padrão menor e, consequentemente, a capacidade aumenta fazendo com que a empresa se torne mais competitiva.

A TABELA 2.1 mostra a classificação dos processos em função da Capacidade Potencial do Processo.

TABELA 2.1 – CLASSIFICAÇÃO DOS PROCESSOS PELOS ÍNDICES C_p

Classificação do Processo	Valor de C_p	Itens não-conformes Parte por milhão (ppm)
Capaz	$C_p > 1{,}33$	70
Razoavelmente capaz	$1 \leq C_p \leq 1{,}33$	Entre 70 e 2700
Incapaz	$C_p < 1$	Mais de 2700

FONTE: Costa *et al.* (2010).

2.2.2 Processo Unilateral

Quando um processo possui apenas uma especificação lateral tem-se a chamada razão unilateral da capacidade potencial ou coeficiente unilateral potencial, que é causada pela natureza da característica de qualidade.

Assim quando o processo possui apenas a especificação superior tem-se o índice C_{p_s} e quando o processo possui apenas a especificação inferior tem-se o índice C_{p_i}. Os índices são representados respectivamente pelas equações (2.2) e (2.3), sendo os parâmetros: $\mu = T$ e σ estimado pelos dados amostrais.

$$C_{p_s} = \frac{LSE - \mu}{3\sigma}; \tag{2.2}$$

$$C_{p_i} = \frac{\mu - LIE}{3\sigma}. \tag{2.3}$$

2.2.3 Centralidade de um Processo

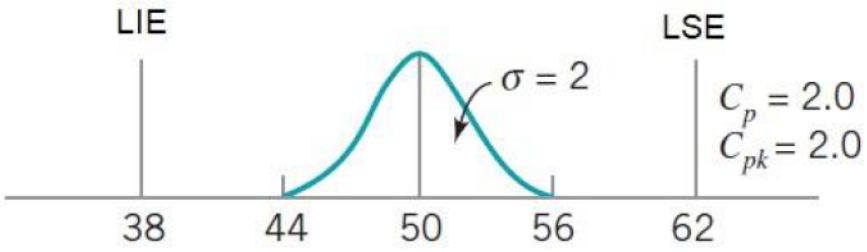
A desvantagem do índice C_p é que ele não considera onde a média do processo está localizada em relação às especificações desejadas. Pode ocorrer casos em que dois processos diferentes tenham o mesmo índice C_p (por exemplo, $C_p = 2$), porém devido à descentralidade de um dos processos, a verdadeira capacidade seja uma inferior à outra, como mostrado na FIGURA 2.2 e 2.3. Por esse motivo é necessário a utilização de outro índice o C_{pk}, quantificado pela equação (2.4).

$$C_{pk} = \min\left(\frac{\mu - LIE}{3\sigma}, \frac{LSE - \mu}{3\sigma}\right). \tag{2.4}$$

O índice C_{pk} é utilizado para verificar a centralidade do processo, para essa verificação tem-se :

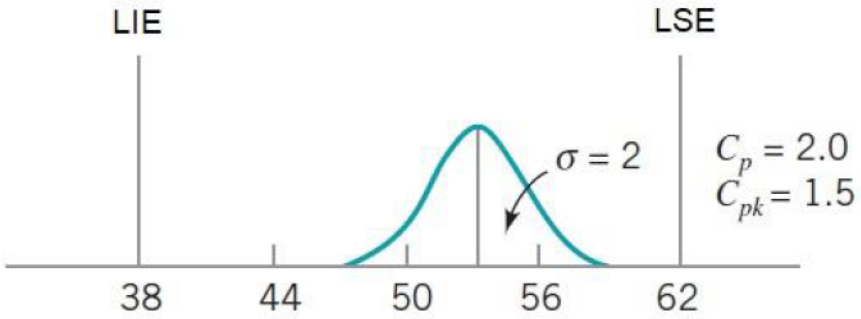
- Se $C_p > C_{pk}$ o processo se encontra descentralizado;
- Caso $C_p = C_{pk}$ o processo está centralizado.

FIGURA 2.2 – PROCESSO CENTRADO



FONTE: Adaptado de Montgomery (2009).

FIGURA 2.3 – PROCESSO NÃO CENTRADO



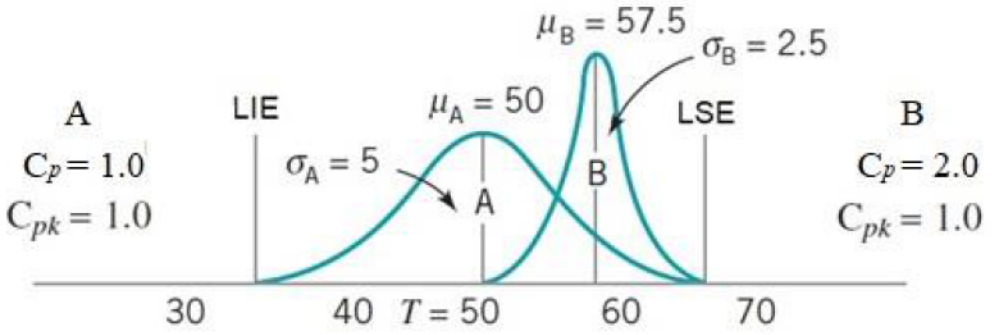
FONTE: Adaptado de Montgomery (2009).

2.2.4 Processo Não-Centrado

Na FIGURA 2.4 os processos A e B têm $C_{pk} = 1$ e o centro dos processos são diferentes. No processo A tem-se $C_p = C_{pk} = 1$ enquanto no processo B tem-se $C_p = 2 > C_{pk} = 1$; mostrando que o processo B está fora de centro. Este é o defeito de C_{pk} como medida da centralidade (MONTGOMERY, 2009).

Como mostrado anteriormente, um grande valor de C_{pk} não reflete de maneira satisfatória a localização da média μ dentro dos intervalos de especificação (LIE e LSE). Devido essa falha é necessário um índice que melhor indique a centralidade, o índice C_{pm}.

FIGURA 2.4 – DIFERENTES PROCESSOS DE C_{pk}



FONTE: Adaptado de Montgomery (2009).

O índice C_{pm} tem o seu foco no valor alvo desejado e de quanto ele se afasta deste valor nominal. O índice concentra-se em medir a competência do processo em agrupar suas respostas em torno do alvo, o que reflete a centralização da capacidade do processo (PEARN; KOTZ, 2006).

A expressão do índice C_{pm} envolve a razão entre os limites de especificação do projeto e a dispersão dos itens de produção em relação ao valor alvo. Então é possível escrever a equação da forma:

$$C_{pm} = \frac{LSE - LIE}{6\tau}, \tag{2.5}$$

onde, τ é a raiz quadrada do desvio quadrático esperado do alvo T (ou ponto médio). Assim temos:

$$T = \frac{LSE - LIE}{2}. \tag{2.6}$$

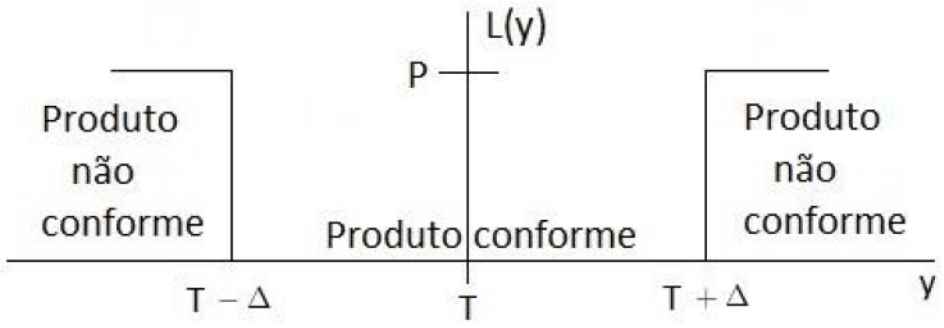
E como $\tau^2 = E[(x-T)^2] = E[(x-\mu)^2] + (\mu - T)^2 = \sigma^2 + (\mu - T)^2$. Pode-se escrever a equação do índice como:

$$C_{pm} = \frac{LSE - LIE}{6\sqrt{\sigma^2 + (\mu - T)^2}}. \tag{2.7}$$

2.3 FUNÇÃO PERDA DE QUALIDADE

É comum as instituições utilizarem a forma clássica de se classificar um produto como conforme ou não conforme, como é representado na FIGURA 2.5.

FIGURA 2.5 – FUNÇÃO DEGRAU



FONTE: O autor (2018).

Sendo o valor central T (valor nominal), com uma tolerância bilateral $T \pm \Delta$, onde Δ equivalente ao desvio do valor nominal e P o custo máximo com substituição do item ou retrabalho.

Essa concepção leva à interpretação de que todos os valores dentro dos limites de tolerância $T \pm \Delta$ são igualmente adequados, e que todos os valores fora dos limites são igualmente inadequados. A função degrau é uma descrição que não representa muito bem a qualidade.

Para entender porque a conformidade de um produto não pode ser simplesmente classificada como conforme ou não conforme, um exemplo é apresentado por Creveling e Fowlkes (1995): considere uma sala com pessoas, em que ocorre uma votação para ajustar o termostato que controla a temperatura do ambiente, para cada vez que a temperatura é aumentada. Para cada grau aumentado no termostato, ou seja, um desvio da média, isso pode se tornar desconfortável para alguma pessoa e assim, ela queira ajustar o termostato para compensar. Note que se subir gradativamente a temperatura do ambiente, cada vez mais pessoas estarão inclinadas a ajustar a temperatura. Portanto, quanto mais se afastar da temperatura inicial, o valor médio, mais pessoas ficarão incomodadas com a temperatura, representando o aumento da “perda”.

A diferença das concepções de qualidade é que no conceito clássico procura-se concentrar a fabricação de produtos cuja qualidade deve estar dentro dos limites de tolerância, ao passo que Taguchi considera a fabricação de produtos cuja qualidade esteja o mais próximo possível do valor nominal projetado. Quando a produção se concentra apenas no cumprimento dos limites de tolerância, os produtos, em média, estarão mais distantes do valor alvo do que os produtos usando a metodologia de Taguchi.

Dessa forma na concepção clássica os procedimentos de melhoria terminam quando se atinge a condição de produzir todas as unidades dentro das especificações. Na visão de Taguchi, considerando a Função Perda, os procedimentos de melhoria irão continuar até que se atinja a perfeição, isto é, quando se alcança o processo exatamente centrado e com variabilidade mínima. O uso da Função Perda implica em uma filosofia de melhoria contínua da qualidade.

Segundo Chaves Neto (2017), a questão primordial, quando se aborda os custos com a qualidade, é aquela de se procurar saber qual o custo que a empresa teria na ausência de qualidade. É importante também saber se o aperfeiçoamento contínuo do produto é eficaz na redução dos custos. Assim, a chamada função perda vem medir a qualidade em termos monetários e pressupõe que o objetivo seja produzir itens de tal forma que atenda a expectativa do consumidor com a menor variabilidade possível.

Com essa filosofia a Função Perda de Qualidade Quadrática foi desenvolvida por Taguchi para fornecer uma estimativa melhor da qualidade, ou ainda, para que a importância da qualidade seja de melhor entendimento para as pessoas. Ele relaciona a performance do produto com o desvio do valor alvo apresentado.

A formulação da Função Perda é realizada por meio da aproximação da série de Taylor em torno de um valor alvo T. Dessa forma:

$$L(y) = L(T) + \frac{L'(T)}{1!}(y - T) + \frac{L''(T)}{2!}(y - T)^2 + \cdots \tag{2.8}$$

e, quando $y = T$ se tem, por definição, que a perda de qualidade é zero quando se está no alvo, e o valor mínimo da função é atingido nesse ponto (sua primeira derivada em relação a T, $L'(T) = 0$). Assim, o primeiro e o segundo termo da equação (2.8),

são iguais a zero (TAGUCHI *et al.*, 1990). Então, ao excluir os termos maiores de segundo grau se obtém:

$$L(y) = \frac{L''(T)}{2!}(y - T)^2 \tag{2.9}$$

ou

$$L(y) = k(y - T)^2. \tag{2.10}$$

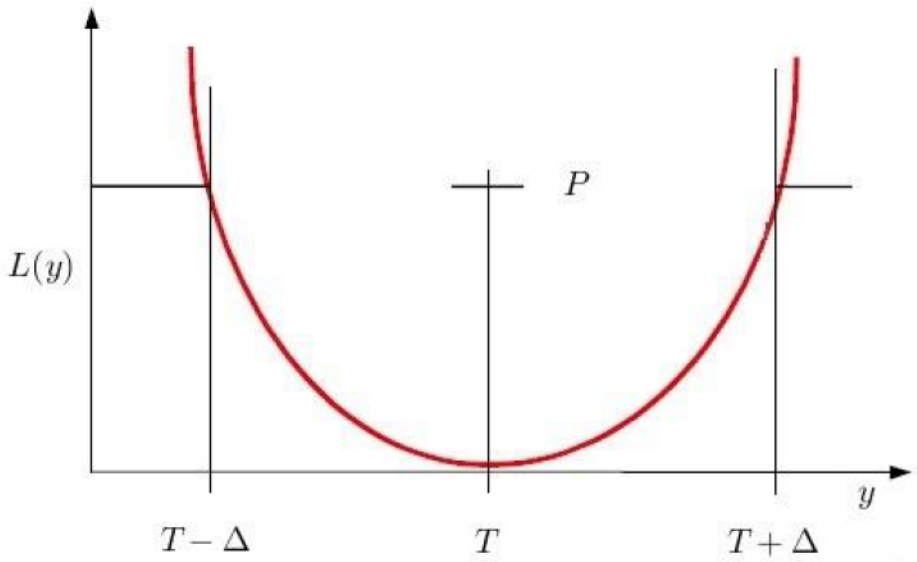
Onde k é uma constante de proporcionalidade definida a seguir. Então, a Função Perda é dada pela equação (2.10) e é representada na FIGURA 2.6.

Assim, quando y supera as tolerâncias devido ao desvio do alvo T, a perda do consumidor devido a um item não conforme é dita como P unidades monetárias. Nesse caso, se for substituído y pelo intervalo $(T + \Delta)$, na equação (2.10), tem-se:

$$P = k(T + \Delta - T)^2; \tag{2.11}$$

$$k = \frac{\mathrm{P}}{\Delta^2}. \tag{2.12}$$

FIGURA 2.6 – FUNÇÃO PERDA DE QUALIDADE QUADRÁTICA DE TAGUCHI



FONTE: O Autor (2018).

Novamente o valor P é o custo para substituição do item ou retrabalho do produto, incluindo os custos atribuídos ao fabricante e consumidor. Custos comuns atribuídos a esse valor são: o custo de parte do produto, a mão de obra do trabalhador para fazer o reparo, o custo de transporte para a assistência, entre outros.

Então, finalmente tem-se a equação para a Função Perda de Qualidade quadrática:

$$L(y) = \frac{P}{\Delta^2}(y - T)^2. \tag{2.13}$$

Outro ponto inovador de Taguchi com a Função Perda de Qualidade foi que ele junta aspectos da engenharia com aspectos econômicos, permitindo uma melhor compreensão da importância do aprimoramento da qualidade (CREVELING; FOWLKES, 1995).

Taguchi definiu três tipos de Função Perda de Qualidade, que estão de acordo com o tipo das características analisadas no produto. Tais funções perda são denominadas de:

1) Função Perda de Qualidade Nominal-é-Melhor (NTB – *Nominal-the-best*);
2) Função Perda de Qualidade Menor-é-Melhor (STB – *Smaller-the-Better*);
3) Função Perda de Qualidade Maior-é-Melhor (LBT – *Larger-the-Better*);
4) Função Perda de Qualidade Assimétrica.

2.3.1 Função Perda de Qualidade Nominal-é-Melhor (NTB)

Esse tipo de função é usado quando a característica de qualidade possui a especificação de projeto com tolerância bilateral, ou seja, quando possui limite superior e inferior nas especificações do seu produto. Adicionalmente, as tolerâncias positiva e negativa a partir do alvo devem ter valores iguais. A Função Perda Nominal-é-Melhor é representada pela equação (2.13) e pela FIGURA 2.6.

Um exemplo para esse tipo de característica é um componente que tenha como alvo uma espessura específica, como: valor alvo de $T = 10\ cm$ e tolerância de $1\ cm$ superior e inferior.

2.3.1.1 Função Perda de Qualidade Média Nominal-é-Melhor

A formulação da Função Perda de Qualidade é realizada para cada característica de qualidade. Na maioria dos casos trabalha-se com amostras de valores observados (vários itens) e, portanto, com o valor médio da perda. Então, tem-se a Função Perda de Qualidade Média. A Função Perda de Qualidade Média é calculada a partir do *mean square deviation* (MSD), isto é, o desvio quadrático médio das observações amostrais e é representada por $\bar{L}(y)$.

Para isso é utilizado uma média de medidas fora do alvo da Função Perda de Qualidade. Essa média dos valores, isso é média de valores de $(y-T)^2$, é o desvio quadrático médio que é utilizado para o cálculo. Dessa forma a Função Perda de Qualidade quadrática para o caso nominal-é-melhor é dada por

$$\bar{L}(y) = k(MSD). \tag{2.14}$$

O cálculo do MSD é:

$$MSD = \frac{1}{n}[(y_1 - T)^2 + (y_2 - T)^2 + \cdots + (y_n - T)^2]; \tag{2.15}$$

$$MSD = \frac{1}{n}\sum_{i=1}^{n}(y_i - T)^2\,; \tag{2.16}$$

$$MSD = \frac{1}{n}\sum_{i=1}^{n}(y_i^2 - 2y_iT + T^2). \tag{2.17}$$

Onde, o segundo termo da equação (2.17) é a média aritmética, $\bar{y}$, da característica funcional dos valores y_i. Ao adicionar e subtrair termos iguais na equação anterior pode-se expressar o MSD como:

$$MSD = \frac{1}{n}\sum_{i=1}^{n}(y_i^2) - \bar{y}^2 + \bar{y}^2 - 2\bar{y}T + T^2\,; \tag{2.18}$$

$$MSD = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 + (\bar{y} - T)^2. \quad (2.19)$$

Note que o primeiro termo é a variância e o segundo termo é o desvio da média $\bar{y}$ do alvo T (CREVELING; FOWLKES, 1995). Então:

$$MSD = \sigma^2 + (\bar{y} - T)^2. \quad (2.20)$$

Portanto, a Função Perda de Qualidade Média com σ^2 estimada por S^2 é:

$$\bar{L}(y) = \frac{P}{\Delta^2}[S^2 + (\bar{y} - T)^2]. \quad (2.21)$$

2.3.2 Função Perda de Qualidade Menor-é-Melhor (STB)

Para esse tipo de característica, as observações nunca assumem valores negativos e seu valor ideal é igual a 0. Assim de acordo com a tolerância o valor nominal é $T = 0$ e o limite superior da tolerância continua sendo $T + \Delta$, como mostra a FIGURA 2.7. Então a Função Perda de Qualidade Menor-é-Melhor é:

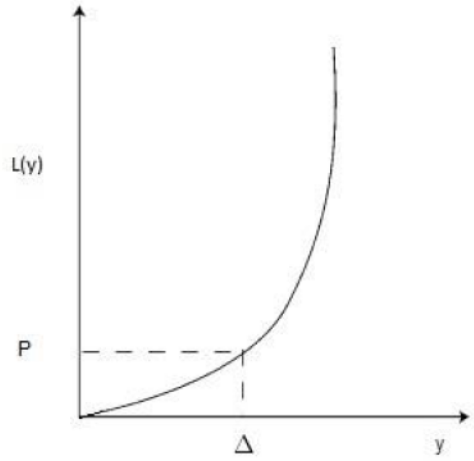
$$L(y) = ky^2 \quad (2.22)$$

ou

$$L(y) = \frac{P}{\Delta^2}y^2. \quad (2.23)$$

Como exemplo de aplicação para essa função, a característica analisada pode ser a poluição do ar ou o coeficiente de atrito de um lubrificante, entre outros.

FIGURA 2.7 – FUNÇÃO PERDA DE QUALIDADE MENOR-É-MELHOR



FONTE: Adaptado de Taguchi *et al.* (2005).

2.3.2.1 Função Perda de Qualidade Média Menor-é-Melhor

Com a mesma linha de raciocínio do caso nominal-é-melhor se pode calcular a Função Perda de Qualidade Média Menor-é-Melhor,

O MSD para essa característica funcional é dado por:

$$MSD = \frac{1}{n}\sum_{i=1}^{n} y_i^2 \approx [S^2 + \bar{y}^2]. \tag{2.24}$$

Logo a Função Perda de Qualidade Média Menor-é-Melhor é:

$$\bar{L}(y) = \frac{P}{\Delta^2}[S^2 + \bar{y}^2\,]. \tag{2.25}$$

2.3.3 Função Perda de Qualidade Maior-é-Melhor (LTB)

Neste caso não existe valores nominais pré-determinados e quanto maior for o valor da característica, melhor. Desse modo, com esse tipo de característica de qualidade, os valores observados são da forma $y \geq 0$ e possuem limite inferior de tolerância $T - \Delta$. A perda é causada por estar fora do limite inferior da tolerância.

A Função Perda de Qualidade Maior-é-Melhor é representada pela equação:

$$L(y) = k\left[\frac{1}{y^2}\right]. \tag{2.26}$$

Para calcular o valor de k, obtém-se o valor da função $L(y)$ quando está no seu limite de tolerância Δ. Assim o resultado da função é P, ou seja, $L(\Delta) = P$. Assim se forem substituídos, esses valores na equação (2.27), obtém-se como resultado:

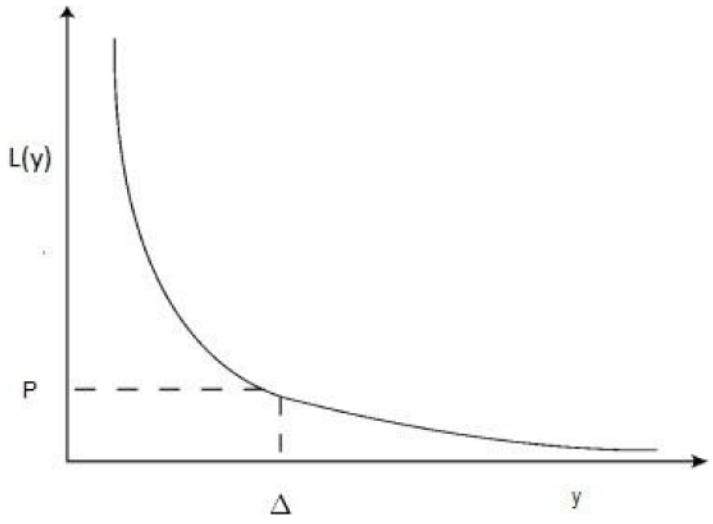
$$P = k\left[\frac{1}{\Delta^2}\right]; \tag{2.27}$$

$$k = P\Delta^2. \tag{2.28}$$

Então finalmente se obtém a equação da Função Perda de Qualidade Maior-é-Melhor, equação (2.29), que é representada pela FIGURA 2.8.

$$L(y) = P\Delta^2\left[\frac{1}{y^2}\right]. \tag{2.29}$$

FIGURA 2.8 – FUNÇÃO PERDA DE QUALIDADE MAIOR-É-MELHOR



FONTE: Adaptado de Taguchi *et al.* (2005).

Um bom exemplo para aplicação desse tipo de característica é a resistência de um cabo ou eficiência de um combustível.

2.3.3.1 Função Perda de Qualidade Média Maior-é-Melhor

Assim como realizado anteriormente para os casos nominal-é-melhor e menor-é-melhor, o desvio quadrático médio para o caso maior-é-melhor é:

$$MSD = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{(y_1)^2} + \frac{1}{(y_2)^2} + \cdots + \frac{1}{(y_n)^2}\right]. \tag{2.30}$$

Portanto, a função com o MSD é dada pela equação

$$L(y) = k(MSD) = k\left[\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{y_i^2}\right)\right]. \tag{2.31}$$

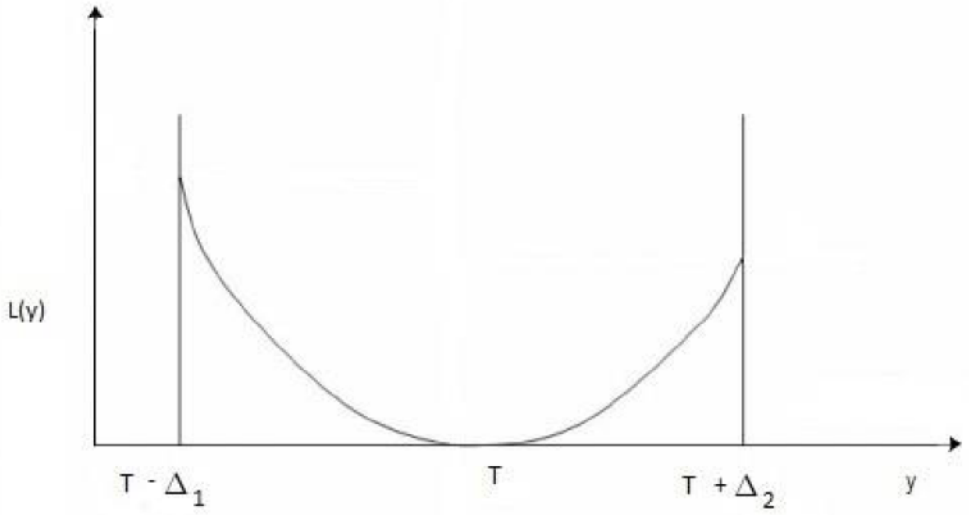
Por fim a equação da Função Perda Média Maior-é-Melhor é:

$$\bar{L}(y) = P\Delta^2\left[\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{y_i^2}\right)\right]. \tag{2.32}$$

2.3.4 Função Perda Assimétrica

Existem casos em que a performance de um produto é menos prejudicial para um lado do que para o outro, nesses casos são necessários dois coeficientes distintos, um para cada lado fora do alvo. Para esses casos utiliza-se a Função Perda Assimétrica, FIGURA 2.9.

FIGURA 2.9 – FUNÇÃO PERDA DE QUALIDADE ASSIMÉTRICA



FONTE: Adaptado de Taguchi *et al.* (2005).

A função possui duas formas:

$$L(y)^{+} = k_1(y - T)^2, \qquad y > T \tag{2.33}$$

$$L(y)^{-} = k_2(y - T)^2, \qquad y \leq T \tag{2.34}$$

onde

$$k_1 = \frac{P_1}{\Delta_1^2}, k_2 = \frac{P_2}{\Delta_2^2}. \tag{2.35}$$

2.4 RAZÃO SINAL-RUÍDO

A Função Perda descrita na seção anterior olha para o desempenho, mas é um desempenho já existente no processo, isto é, ela não necessariamente prediz a performance final do sistema. Isto ocorre porque a Função Perda não é independente dos ajustes da média após reduzir a variabilidade, isso é, se o sistema possui pouca variabilidade, mas está fora do seu alvo, ainda existe perda. Então, a Função Perda não é uma medida adequada para se usar na otimização de parâmetros do projeto,

onde se procura reduzir a variabilidade do processo independentemente do alvo do processo (CREVELING; FOWLKES, 1995).

Taguchi *et al.* (2005) fala sobre a ideia da Razão Sinal-Ruído (S/N):

> A Razão Sinal-Ruído (S/N) é uma escala de medição que tem sido utilizada no setor de comunicação há quase um século. Um rádio mede o sinal ou a onda de voz transmitida a partir de uma estação de transmissão e converte a onda em som. Quanto maior a voz enviada, maior a voz recebida. [...] na verdade, a entrada é misturada com o ruído audível no espaço. Bons equipamentos capturam o sinal e não são influenciados pelo ruído. [...]
>
> Taguchi generalizou o conceito de relação S/N, como usado no setor de comunicação e aplicou-o para a avaliação de sistemas de medição, bem como para a função de produtos e processos. (TAGUCHI *et al.*, 2005, p. 224).

A Razão S/N funciona como medida de robustez de um produto/processo e um aumento na Razão S/N reflete uma melhoria no processo. Consequentemente, ocorre a relação de menor custo com uma melhoria da qualidade.

Por definição a Razão S/N pode ser representada por uma função logarítmica utilizada para mensurar e otimizar o processo ou produto, quanto à sua variabilidade. A Razão S/N também pode ser entendida como o inverso da variância e maximizar essa razão permite a redução da variabilidade do processo contra mudanças indesejáveis (SILVA *et al.*, 2014).

Para Peace (1993), a importância da Razão S/N está ligada diretamente a interpretação dela, como a capacidade de um processo realizar com maior eficiência a fabricação de um produto.

Como aponta Marques (2015), a construção da Razão S/N é baseada no desvio quadrático médio das funções Perda Média de Qualidade, apresentadas na seção anterior, de suas respectivas características de qualidade. Assim, o desvio quadrático médio é modificado com o objetivo de tornar a Razão S/N independente dos ajustes do valor alvo de projeto.

Desse modo, é utilizado o logaritmo para a modificação da Razão S/N. As propriedades da função logarítmica permitem promover a aditividade, isso é, a redução da interação entre fatores. Além disso, o MSD é expresso em decibel (dB).

A Razão S/N tem algumas propriedades (CREVELING; FOWLKES, 1995) que são:

- Reflete a variabilidade da resposta do sistema;
- É independente dos ajustes da média;

- Mede a qualidade relativa, na comparação de propostas;
- Não induz a complicações desnecessárias, como as interações entre fatores de controle quando há muitos fatores a serem analisados.

Então, a Razão S/N é classificada em dois tipos, dinâmica e estática. A Razão S/N dinâmica é utilizada para estudo com múltiplos alvos ou, para intervalos de valores para o alvo, onde se espera que a resposta do sistema siga a entrada do sistema, ou seja, quanto maior a entrada do sinal melhor a resposta (PHADKE, 1989). Esse tipo de característica é mais associado a desenvolvimento de tecnologias como, por exemplo, a eficiência energética de um compressor, sistema de entrega de combustível ou acurácia de um sensor (TAGUCHI *et al.*, 2000).

Já a Razão S/N estática é utilizada em problemas com um único alvo fixo. Esse tipo de característica é comum para a indústria, como, por exemplo, o valor alvo para a espessura de uma peça ou a rigidez de uma mola. Dentro da Razão S/N estática podemos destacar três tipos de característica de qualidade, já apresentadas na seção anterior, a Nominal-é-Melhor, Menor-é-Melhor e Maior-é-Melhor.

Esse trabalho se concentra apenas nas características mais usuais para a indústria, ou seja, na Razão S/N estática.

2.4.1 Razão Sinal-Ruído para o caso Menor-é-Melhor.

Para a Razão Sinal-Ruído Menor-é-Melhor, assim como na Função Perda Menor-é-Melhor, as respostas continuam sendo não negativas e o alvo T da característica de qualidade a ser analisada deseja-se ser igual a 0.

Minimizar o MSD da equação é o mesmo que maximizar a Razão S/N, que é definido pela equação:

$$S/N_{STB} = -10\log[MSD]\,; \tag{2.36}$$

$$S/N_{STB} = -10\log\left[\frac{1}{n}\sum_{i=1}^{n} y_i^2\right]; \tag{2.37}$$

$$S/N_{STB} = -10\log[S^2 + \bar{y}^2]. \tag{2.38}$$

2.4.2 Razão Sinal-Ruído para o caso Maior-é-Melhor

Para a Razão Sinal-Ruído Maior-é-Melhor, assim como na Função Perda Maior-é-Melhor, as respostas da característica do processo continuam sendo não negativas partindo do 0 até o infinito, a resposta desejada é o maior valor possível.

A equação da Razão S/N para o caso maior-é-melhor é:

$$S/N_{LTB} = -10\log[MSD]\,; \tag{2.39}$$

$$S/N_{LTB} = -10\log\left[\frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{y_i^2}\right)\right]. \tag{2.40}$$

2.4.3 Razão Sinal-Ruído para o caso Nominal-é-Melhor

Para o caso nominal-é-melhor são associados dois parâmetros da Razão S/N, Nominal-é-Melhor tipo I e Nominal-é-Melhor tipo II.

Independentemente do tipo da Razão S/N Nominal-é-Melhor existem algumas características em comum. Uma delas é que a Razão S/N seja sempre maximizada, outra é que a Razão S/N é baseada na variabilidade da Função Perda ou no desvio quadrático médio dela. Então, o MSD do caso Nominal-é-Melhor pode ser escrito como, $MSD_{NTB} = [S^2 + (\bar{y} - T)^2]$.

Mas o MSD_{NTB} não descreve o MSD de forma independente dos ajustes, por isso existem os casos que exigem modificações no MSD.

2.4.3.1 Razão Sinal-Ruído Tipo I

As seguintes características são designadas para a Razão S/N Nominal-é-Melhor tipo I: os valores de resposta são contínuos e não negativos, indo de zero até o infinito; esse tipo de problema tem alvos não nulos e variância zero quando a resposta da média é zero. Esse tipo da Razão S/N é dado pela equação:

$$S/N_{NTB-Type\ I} = 10\log\left(\frac{\bar{y}^2}{S^2}\right). \tag{2.41}$$

2.4.3.2 Razão Sinal-Ruído Tipo II

As seguintes características são designadas para a Razão S/N Nominal-é-Melhor tipo II: as respostas são contínuas e podem assumir valores positivos ou negativos; o valor alvo pode ser zero. A característica que difere a Razão S/N Nominal-é-Melhor tipo I da mesma razão do tipo II é que as respostas do alvo podem ser e frequentemente são nulas. Esse tipo de Razão S/N é dado pela equação:

$$S/N_{NTB-Type\ II} = -10\log(S^2). \tag{2.42}$$

2.5 PROJETO ROBUSTO

Genichi Taguchi foi o criador do chamado Projeto Robusto, que é um método de engenharia que usa as técnicas estatísticas de Planejamento de Experimentos.

A ideia fundamental em um Projeto Robusto é aproximar a realidade por meio da resposta da característica funcional observada. São procedimentos aplicados durante as fases de pesquisa e desenvolvimento, que têm como principal objetivo a produção de um produto de alta qualidade, com custo mínimo.

Creveling e Fowlkes (1995) afirmam: “Um produto ou processo é dito ser robusto quando é insensível aos efeitos de variabilidade, mesmo quando as fontes de variabilidade não tenham sido eliminadas. ”

A finalidade é desenvolver um produto ou processo que apesar das condições em que seja usado ou fabricado continue a executar a sua função primária com mínima variabilidade e que o produto possua uma vida longa. Em outras palavras, Projeto Robusto é um processo para obter o desempenho ótimo de um produto e que é minimamente afetado por ruídos ou completamente insensível a ruídos externos.

De acordo com Taguchi *et al.* (2005) “[..] Projeto Robusto é o projeto de produto que será menos afetado pelas condições do usuário. As condições de ruído devem, portanto, ser consideradas no desenvolvimento. ”

2.5.1 Fatores de Controle

São os fatores que podem ser controlados e mantidos, ou seja, se pode determinar os seus níveis tais que as características de interesse permaneçam

consistentes, apesar das mudanças nos fatores de ruído. É aquele fator que vai ser utilizado para fazer combinações para alcançar o melhor desempenho possível de uma característica funcional.

Como exemplo, é possível citar a escolha de matéria-prima para confecção de um produto, tais como, os tipos de madeira, a espessura do material usado, a marca de subproduto entre outros.

2.5.2 Fatores de Ruído

A percepção do usuário sobre qualidade de um produto é diretamente relacionada com a percepção de como o produto se comporta na presença de ruídos. Assim, a variabilidade da função principal do produto durante a sua utilização são causadas pelos fatores de ruído.

Desta forma, os fatores de ruído são definidos como qualquer coisa que cause um desvio do valor alvo na característica funcional do produto. Portanto, são necessários esforços para minimizar os efeitos que os ruídos possuem sobre o desempenho do projeto. Existem duas maneiras de reduzir esses efeitos:

- Eliminar a atual fonte de ruído;
- Eliminar a sensibilidade do produto às fontes de ruído.

Nem sempre a melhor estratégia é eliminar as fontes dos fatores de ruído. Muitas vezes é muito caro e consome muito tempo para eliminar os fatores de ruído, porque alguns fatores não podem ser controlados ou são de difícil controle.

Como o objetivo é desenvolver um processo ou produto que fornece um bom desempenho com variabilidade mínima, é fundamental saber quais as fontes dos ruídos. Os fatores de ruído são classificados em três tipos, dependendo da sua origem:

- Ruídos externos;
- Ruídos de produto-para-produto;
- Ruídos de deterioração.

Ruídos externos são definidos como fonte de variabilidade que vem de fora do produto, tais como: o teor de umidade, a temperatura, o efeito da maresia sobre produtos eletrônicos, pressão acima do esperado, etc.

Ruído de produto-para-produto é a variabilidade inerente dentro de um processo de produção, isto é, dificilmente se consegue obter dois produtos perfeitamente iguais. Este tipo de ruído é monitorado pelo CEP (Controle Estatístico de Processo). Geralmente, o processo de fabricação e a matéria-prima são as maiores causas desse tipo de ruído.

Ruídos de deterioração são geralmente atribuídos a ruídos internos. Esse tipo de ruído acontece porque alguma coisa mudou internamente no produto ou no processo de uso. Ocorrem com produtos que ficam muito tempo armazenados e, também, ocorrem com a idade do produto. Exemplo: uma máquina fotográfica cujo obturador falhou após muitos anos de uso, pois já havia atingido o limite de operações de abertura e fechamento.

O efeito do ruído sobre o produto é usualmente associado de alguma forma com problemas de confiabilidade. Além disso, os defeitos apresentados nos produtos podem ser caracterizados de acordo com o tempo. Três categorias são identificadas e que podem ser representados na chamada Curva da Banheira, mostrada na FIGURA 2.10:

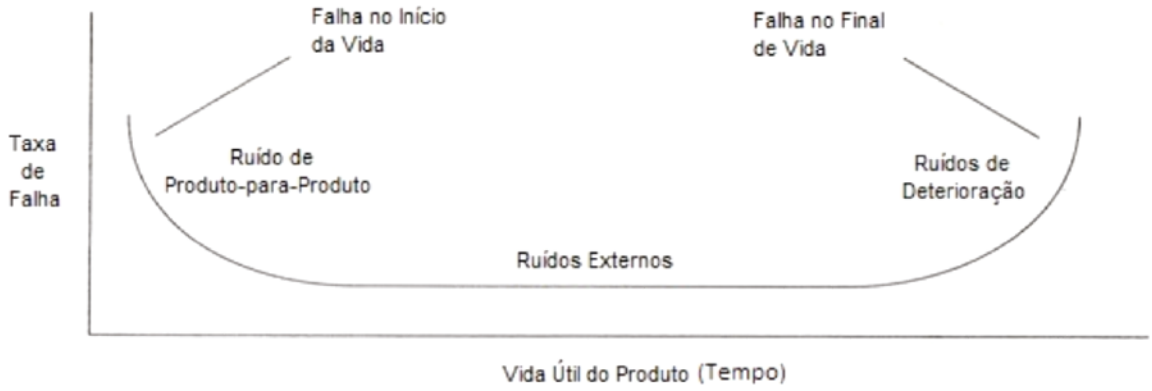
- Falha no início da vida útil;
- Falha durante o tempo de vida normal;
- Falha no final da vida útil.

Pode-se então associar os tipos de ruídos mais comuns para cada tipo de falha característica no tempo de vida do produto, plotando os tipos de falhas contra o tempo de vida. O gráfico da curva da banheira mostra que a probabilidade de falha é maior no começo e no final de vida útil do produto, enquanto a taxa de falha durante o período normal é baixa.

Como Creveling e Fowlkes (1995) apontam, as falhas no começo da vida útil são relacionadas a defeitos não identificados durante o processo de fabricação e que não apareceram na inspeção.

Uma das formas de evitar as falhas no meio da vida útil do produto é restringir o modo de uso do produto. Eliminar as causas dessas falhas é muito custoso porque os ruídos externos estão presentes nas mais variadas formas e intensidades.

FIGURA 2.10 – CURVA DA BANHEIRA



FONTE: Adaptado de Creveling e Fowlkes (1995).

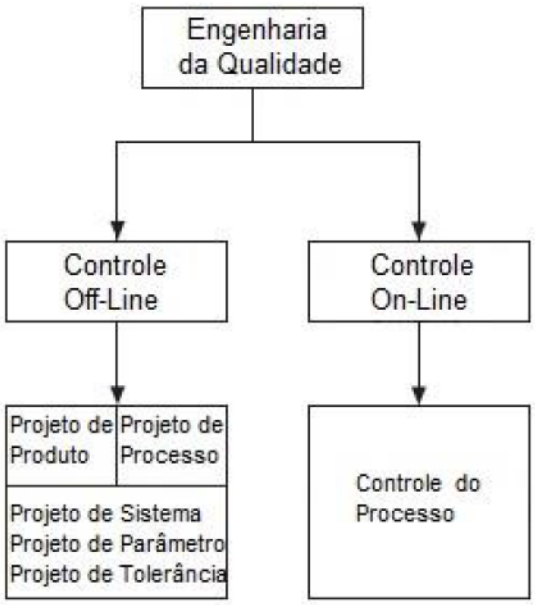
No final da vida útil do produto os efeitos dos ruídos externos se acumulam causando deterioração. Como consequência, a taxa de falha aumenta após um tempo de uso.

2.5.3 Controle de Qualidade *on-line* e *off-line*

Quando Genishi Taguchi propôs o seu método foi preciso dividir o controle de qualidade em momentos diferentes. Assim, surgiram o controle de qualidade *off-line* e o controle de qualidade *on-line*, como mostrado na FIGURA 2.11.

Essa separação ocorreu devido Taguchi propor seu método fora da linha de produção, isso é, na área *off-line*. Em outras palavras, o controle de qualidade *off-line* é feito durante o desenvolvimento do projeto do produto e do projeto de processo. O controle de qualidade *on-line* é feito na linha de produção.

FIGURA 2.11 – CONTROLE DE QUALIDADE *ON-LINE* E *OFF-LINE*



FONTE: Adaptado de Hassan (2000).

2.5.3.1 Controle de Qualidade *off-line*

Qualidade *off-line* (também chamada de engenharia robusta) fornece os principais passos nas primeiras fases de construção de projeto. Uma vez definido o objetivo do projeto há necessidade de definir requisitos fundamentais e desenvolver os conceitos do produto e processo robustos necessários à otimização.

Taguchi *et al.* (2005) descreve a engenharia de qualidade *off-line* como “as operações fora da linha de produção, mas diretamente ligadas à produção, ou ainda, um método sistemático para otimizar o projeto do produto e o projeto do próprio processo produtivo. ”

Essas operações fora da linha de produção são divididas em três etapas:

- Projeto de Sistema;
- Projeto de Parâmetros;
- Projeto de Tolerância.

O Projeto de Sistema é fundamental para que as duas etapas seguintes ocorram da melhor maneira, para isso é preciso escolher o sistema com muito cuidado. Porém, é difícil avaliar se o sistema selecionado é o melhor sem conduzir o Projeto de Parâmetros. O Projeto de Sistema é a fase em que é definido um sistema que funciona sobre um conjunto de condições iniciais.

O Projeto de Parâmetros é o centro da Engenharia da Qualidade. Geralmente como leva tempo e tem alto custo, ele deve resultar em especificações eficientes. Dessa forma é usado para melhorar um sistema e encontrar a funcionalidade máxima de um sistema, para fazê-lo menos sensível aos ruídos.

Projeto de Tolerância é a fase do projeto em que se especifica a variabilidade permitida nos valores dos parâmetros, para balancear o custo de produção com a qualidade de produção, satisfazendo a demanda do produtor e do cliente.

2.5.3.2 Controle de Qualidade *on-line*

Controle de qualidade *on-line* envolve a linha de produção do produto. São técnicas para monitorar produção, medição de qualidade em curso, modos de providenciar sinais para potenciais problemas e direcionamento de soluções. Para esse fim, informações de resposta do sistema são fundamentais para o operador e para o gerenciamento do processo. Além disso, as informações são obtidas em tempo real a partir dos pontos críticos da produção. (PEACE, 1993)

As informações em tempo real são analisadas para determinar se a configuração atual do processo precisa de manutenção. Isso quer dizer que os dados da produção têm que ser comparados com um valor alvo o qual se quer atingir.

Se a análise indica que a atual situação não é aceitável, então ajustes devem ser feitos no processo para que ele volte ao normal, ou seja, reduzindo a variabilidade a níveis aceitáveis.

Hassan *et al.* (2000) concluem:

> Como nenhum processo é perfeito (livre de qualquer variação) mesmo com automação, monitoramento de processos e ferramentas de controle, é sempre necessário detectar, reduzir e prevenir distúrbios (variação). [...] As técnicas do CEP visam a melhoria de qualidade *on-line* e só devem ser implementados após esforços suficientes e recursos terem sido direcionados para "criar" a qualidade da fase de projeto (controle de qualidade *off-line*). (HASSAN *et al.,* 2000).

2.5.4 Projeto de Sistema

O Projeto do Sistema, ou projeto de conceitos, é o primeiro passo e a etapa fundamental no desenvolvimento do produto. Ele tem como objetivo o desenvolvimento de conceitos inovadores que correspondem aos requisitos do cliente, da empresa e da regulamentação.

É nessa fase que é escolhido o melhor sistema para executar a função desejada, isto é, escolher a melhor tecnologia e o melhor conceito a partir de todas as alternativas possíveis que possam desempenhar tal função. Então, se o objetivo é uma reação química, deve haver químicos especializados pesquisando quais os melhores métodos, quais os melhores materiais e qual a tecnologia mais adequada para tal processo.

Como vários sistemas são possíveis para o mesmo objetivo Taguchi *et al.* (2005) apontam "A seleção do sistema é muitas vezes determinada em uma reunião das pessoas responsáveis, onde as opiniões são trocadas sobre as vantagens e desvantagens de cada item nas funções. "

Ainda existem os aspectos sociais embutidos em um Projeto de Sistema. Em caso de falha de algum componente é necessário garantir a segurança. O projeto de segurança é um fator importante na seleção do sistema. Um dos aspectos do Projeto de Sistemas é incluir um mecanismo de segurança que não ponha em risco a vida humana ou que cause perda financeira significativa, se a função falhar. Esses mecanismos são denominados *poka-yokes*.

A taxa de melhoria que pode ser alcançada, por um produto e o seu processo de manufatura não depende apenas do projeto em si, mas da complexidade do sistema. Se um sistema não é suficientemente complicado é difícil melhorar a sua funcionalidade através do Projeto de Parâmetros (TAGUCHI e*t al.*, 2005).

2.5.5 Projeto de Parâmetros

Uma vez concluído o Projeto do Sistema, o próximo passo é determinar o nível ótimo de parâmetros individuais do sistema. O Projeto de Parâmetros é um método utilizado para reduzir a influência das fontes de variação e otimizar o sistema.

A concepção dos parâmetros é o passo mais importante no desenvolvimento de produtos estáveis e confiáveis no processo de fabricação. Nesta fase encontramos

uma combinação de níveis de parâmetros que são capazes de amortecer as influências não só dos ruídos internos, mas de todas as fontes de ruído, mantendo a melhor resposta, durante o maior período de tempo possível.

Há uma diferença entre a abordagem de melhoria de qualidade tradicional e a abordagem do Projeto de Parâmetros. Na abordagem tradicional, o primeiro passo é tentar atingir o alvo, em seguida, reduzir a variabilidade. Assim as medidas para reduzir a variabilidade dependem da abordagem do Projeto de Tolerância, isto é, melhorar a qualidade com o aumento do custo.

Para Taguchi *et al.* (2005) a abordagem anterior é equivocada, pois se tenta primeiro atingir o alvo e depois reduzir a variabilidade. Para Taguchi, o objetivo deve ser aumentar a robustez do produto logo após o Projeto do Sistema, ou seja, primeiro reduzir a variabilidade.

No Projeto de Parâmetros a qualidade é melhorada através da seleção de combinações de fatores de controle e fatores de ruído. Tendo um protótipo, ele deve ser testado sob as condições de uso de alguns clientes para mensurar a sua qualidade relativa.

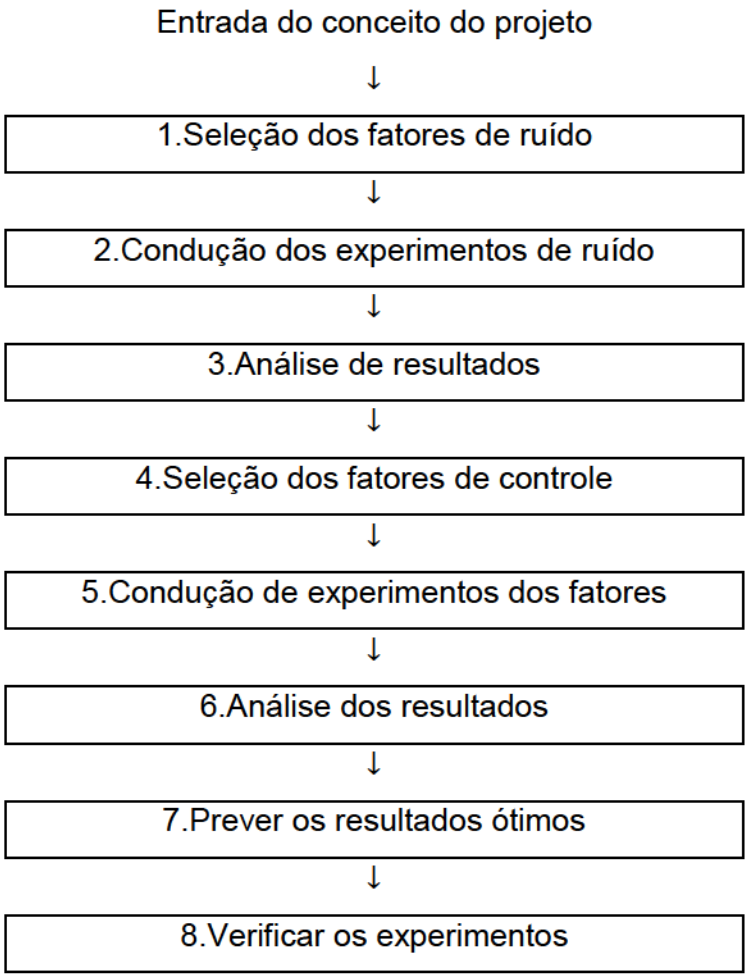
Caso existam várias condições extremas de uso (fatores de ruído), elas devem ser representativas sob o modo de uso de um usuário genérico. Os fatores de controle devem ser os fatores que mais influenciam na resposta do produto.

As principais tarefas do desenvolvimento do Projeto de Parâmetros na Engenharia de Qualidade são:

- Introduzir a Razão S/N;
- Usar matrizes ortogonais para verificar a significância das interações entre os fatores de controle.

Assim como Taguchi *et al.* (2005), outros autores, como Unal e Dean (1991), Zambanini (1992), Creveling e Fowlkes (1995), Athreya e Venkatesh (2012) e Karna e Sahai (2012), cada um com pequenas diferenças em suas definições, indicam que é útil fazer um passo-a-passo como guia do Projeto de Parâmetros. Todo o processo do Projeto de Parâmetros foi dividido em 8 etapas como mostra a FIGURA 2.12.

FIGURA 2.12 – ETAPAS DO PROJETO DE PARÂMETRO



FONTE: O Autor (2018).

Entrada do conceito do projeto é o passo inicial, ou seja, são as características que foram planejadas anteriormente no Projeto de Sistema, como a característica de qualidade.

- Etapa 1 (Seleção dos fatores de ruído) – Identificar quais os fatores de ruído que possuem impacto negativo sobre o desempenho do produto e quais seus níveis, se são difíceis de controlar, controláveis ou incontroláveis e a qual tipo de ruído pertence;
- Etapa 2 (Condução dos experimentos de ruído) – Selecionar a matriz experimental adequada para a execução do experimento dos ruídos e executar o experimento;
- Etapa 3 (Análise de resultados) – Observar o resultado da análise das médias e gráfico dos efeitos principais;

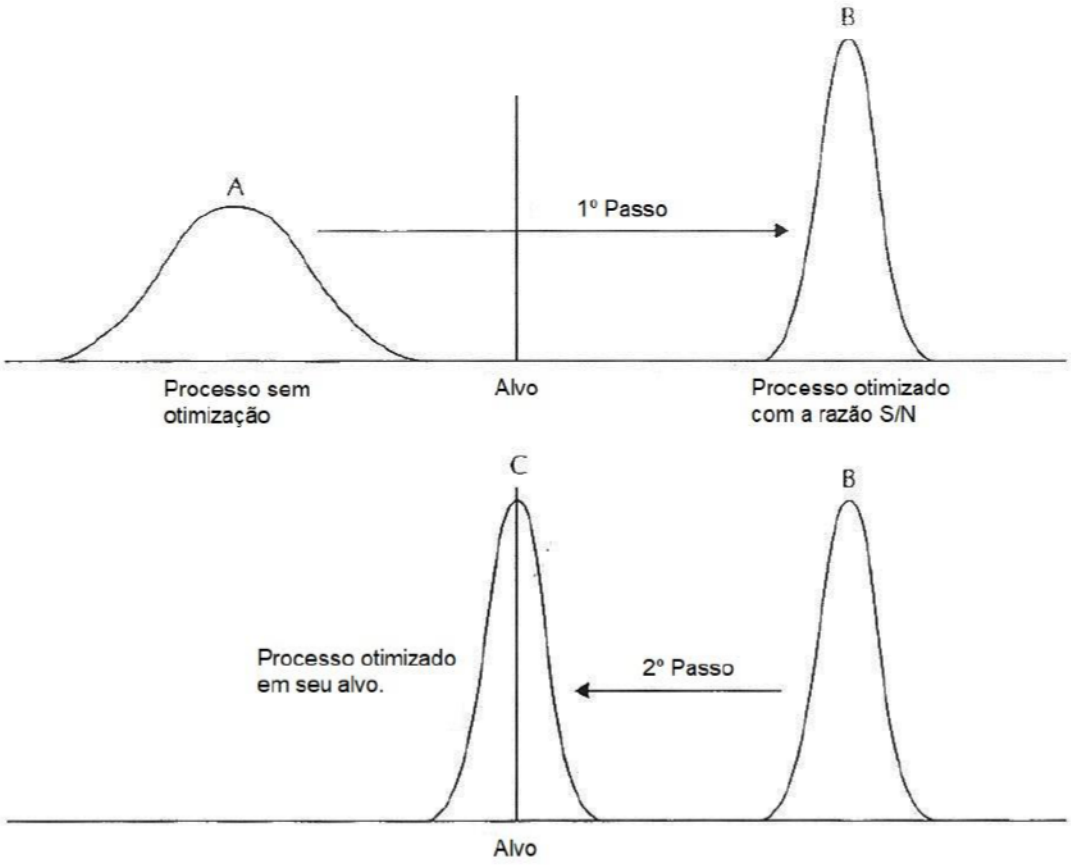
- Etapa 4 (Seleção dos fatores de controle) – Definir quais são os fatores de controle, ou seja, os principais fatores que influenciam a característica funcional e os respectivos níveis;
- Etapa 5 (Condução de experimentos dos fatores) – Selecionar a matriz experimental adequada para a execução do experimento e executar o experimento;
- Etapa 6 (Análise dos resultados) – Observar a resposta da Razão S/N do experimento com os fatores, matriz ortogonal e característica de qualidade selecionados anteriormente. Observar o gráfico de efeitos principais da Razão S/N e análise das médias da Razão S/N. Caso necessário observar o gráfico de efeitos principais da resposta média;
- Etapa 7 (Prever os resultados ótimos) – Construir o modelo com os valores ótimos (modelo preditivo);
- Etapa 8 (Verificar os experimentos) – Executar o experimento de verificação com os valores ótimos e não ótimos. Comparar as respostas dos modelos preditivos com os do experimento de verificação para obter a confirmação da robustez do processo.

2.5.6 Otimização em Dois Passos

Uma das técnicas para se obter o produto robusto é, após a otimização da variabilidade de um produto, ajustar o alvo do produto através da seleção de um fator de controle cuja alteração do nível afeta a média, mas, afeta minimamente a variabilidade. Isto é, primeiro ajusta-se a variabilidade e depois se coloca o processo ao alvo. Essa operação é chamada de otimização em dois passos, sendo esquematizado na FIGURA 2.13.

Assim, de acordo com Chaves Neto (2017), a otimização em dois passos quebra o complexo problema de alcançar um desempenho sobre o alvo em dois problemas mais simples. O primeiro, com o objetivo de produzir dados que permitam ao engenheiro achar pontos dos fatores de controle que produzam uma distribuição da resposta a mais justa possível na presença de ruído. Já o segundo, com o objetivo de identificar o fator que permita que o desempenho médio seja ajustado sobre o alvo, enquanto afeta minimamente a distribuição que está sendo ajustada.

FIGURA 2.13 – OTIMIZAÇÃO EM DOIS PASSOS



FONTE: Adaptado de Creveling e Fowlkes (1995).

Dessa forma, o processo de otimização em dois passos consiste de duas etapas:

- Passo 1: é maximizado a Razão S/N para minimizar a sensibilidade aos efeitos dos ruídos. Para conseguir isto, primeiro identificam-se os fatores de controle que têm maior significância no seu gráfico de efeitos principais, e então seleciona-se os fatores de controle que correspondem aos valores máximos de S/N;
- Passo 2: ajustar o seu sistema na resposta média adequada. Para conseguir isto, identificar o fator de controle que tem ambos: maior elevação nos gráficos da média e menor elevação nos gráficos da Razão S/N. Este é o fator de controle que é usado para ajustar a média.

Assim, os fatores cujos efeitos na Razão S/N são relativamente fortes podem ser usados para reduzir a variabilidade; os fatores cujos efeitos na média são relativamente fortes podem ser usados para ajustar o valor alvo.

2.5.7 Projeto de Tolerância

Uma vez que o Projeto de Parâmetro esteja concluído, tem-se os valores médios dos fatores que compreendem o Projeto de Tolerância dos elementos do sistema. O próximo passo é determinar as tolerâncias desses fatores ou selecionar, da melhor forma, as peças e materiais de modo a otimizar as tolerâncias, o que não significa necessariamente restringi-las de forma excessiva. O que isso significa é, restringir as tolerâncias que precisam ser limitadas e permitir que as tolerâncias que devem ser mais livres sejam capazes de possuir essa flexibilidade, maximizando assim a qualidade, a eficiência e a economia do projeto.

Então, o próximo passo é determinar as tolerâncias desses fatores. Para fazer essa determinação, se considera algumas condições, tais como as ambientais, bem como os elementos do sistema.

Os valores médios e seus intervalos são considerados como fatores de ruído e são dispostos em uma matriz ortogonal, de modo que as significâncias dos fatores no resultado final possam ser determinadas.

Taguchi *et al.* (2005) indicam seis passos para fazer o Projeto de Tolerância:

1. Conduzir um experimento com as tolerâncias já existentes;
2. Usar a análise da variância (*analysis of variance* – ANOVA) para determinar a contribuição de cada fator;
3. Determinar a Função Perda de Qualidade para todas as respostas do sistema;
4. Para cada fator, calcular a Função Perda de Qualidade;
5. Para cada fator calcular a nova perda com a Função Perda de Qualidade, comparar o aumento de custo para decidir qual fator deve ter a tolerância otimizada;
6. Para cada fator aprimorado, se necessário, determinar o ganho.

2.5.8 Vantagens do Projeto Robusto

A utilização da metodologia do Projeto Robusto traz vantagens, como: a prontidão tecnológica, a flexibilidade e a reprodutibilidade (CREVELING; FOWLKES, 1995).

A prontidão tecnológica diz respeito a problemas relacionados a demandas de cliente. Quando o produto está na mão do cliente já é tarde demais para a melhoria

da qualidade. Portanto, um produto robusto possui menos desvios da sua função principal, logo a taxa de reclamações de clientes é menor. Uma vez que o produto está otimizado a produção está pronta para entregar aquilo que foi prometido.

A flexibilidade se refere à utilização de uma função genérica. Uma vez caracterizado um tipo de processo, pode-se reproduzir a mesma análise em outras linhas de produção. Claro que se deve usar com cuidados observando sempre as causas particulares de cada caso.

Quando o produto é robusto ele possui reprodutibilidade, isto é, o produto é reprodutível da forma:

1. Os resultados do projeto de produto são reproduzíveis no próprio produto;
2. Os resultados do estudo de projeto em pequena escala são reproduzíveis na produção em larga escala;
3. Os resultados obtidos em condições limitadas do projeto são reproduzíveis sob várias condições.

2.6 PROJETO DE EXPERIMENTO

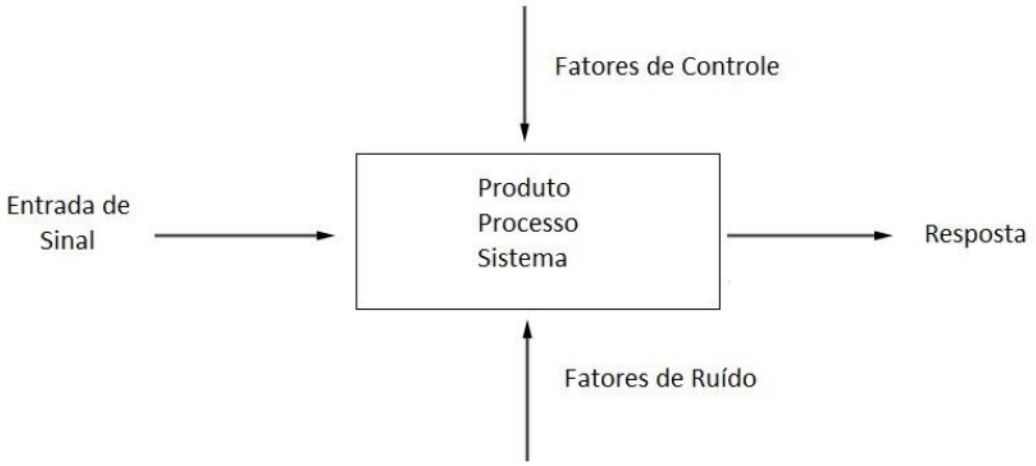
Os objetivos do Projeto de Experimentos, ou delineamento de experimento, é explicar a relação entre os fatores, determinar os fatores que possuem maior e menor influência na resposta, avaliar diferentes configurações possíveis para o processo e aprimorar a resposta desejada. No Projeto de Experimentos é importante escolher a melhor relação entre seus componentes para que a análise seja precisa e eficiente.

O planejamento do Projeto de Experimento requer:

- Definir a matriz experimental;
- Definir a variável resposta (o que será medido ou observado no Projeto de Experimento);
- Definir os fatores de controle e os fatores de ruído e seus respectivos níveis;
- Designar os fatores a matriz experimental.

O P-diagrama, FIGURA 2.14, representa as entradas e as respostas de um sistema, produto ou processo, bem como os fatores de controle e os fatores de ruído envolvidos em um Projeto de Experimento.

FIGURA 2.14 – P-DIAGRAMA



FONTE: O autor (2018).

Então, como o objetivo do Projeto de Experimentos é obter respostas estatísticas, a coleta de dados para a análise experimental deve ocorrer com base em alguns princípios, sempre que possível, para que as conclusões não sejam de caráter duvidoso. Os princípios são: aleatorização, replicação e blocagem.

Como Lira (2017) descreve, “A sequência de experimentos e/ou a atribuição de amostras a diferentes combinações de tratamento de maneira puramente casual é denominada aleatorização”. Dessa forma, os erros experimentais observados atendem ao requisito de apresentar um caráter aleatório, isso é, com essa técnica procura-se diminuir tendências (KRISHNAIAH; SHAHABUDDEN, 2012).

A replicação, segundo Chaves Neto (2017), “A necessidade de repetições ou réplicas em um experimento ocorre devido à falta de confiança nos resultados de um experimento que contenha apenas uma unidade experimental. ” As replicações permitem que o experimentador obtenha uma estimativa precisa do efeito de um fator estudado no experimento.

A blocagem, pode ser utilizada para a melhoria na precisão do experimento, reduzindo a influência de variáveis incontroláveis ou para uma melhor homogeneidade dos dados. Um bloco é uma porção do material experimental ou do meio experimental que apresenta uma probabilidade maior de homogeneidade em si mesma do que entre todo material estudado (LIRA, 2017).

2.6.1 Matriz Ortogonal

As matrizes ortogonais são parte fundamental do Projeto de Experimento de Taguchi. As matrizes ortogonais utilizam apenas uma fração das corridas utilizadas pelos projetos de experimentos fatoriais tradicionais, isto é, dos experimentos fatoriais completos, como mostra a TABELA 2.2. É possível ver que em um experimento com a matriz ortogonal que utiliza 7 fatores de dois níveis é necessário a realização de apenas 8 corridas. Na contramão, para a matriz experimental fatorial completa são necessárias 128 corridas.

TABELA 2.2 – COMPARAÇÃO DE NÚMERO DE CORRIDAS

		Matriz Ortogonal	Fatorial completo
Número de fatores com 2 níveis	Número de fatores com 3 níveis	Número de corridas	Número de corridas
3	0	4	8
7	0	8	128
15	0	16	32768
31	0	32	2147483648
0	4	9	81
0	13	27	1594323
11	0	12	2048
1	7	18	4374
11	12	36	1088391168

FONTE: Adaptado de Mori (2011).

Ao contrário do que pode parecer a utilização de menos corridas não prejudica o resultado do experimento, o uso das matrizes ortogonais é uma utilização da matriz experimental de forma mais eficiente. E a sua construção é baseada dentro do rigor matemático como mostram Kacker *et al.* (1991).

Creveling e Fowlkes (1995) apontam:

> Ortogonal se refere ao balanceamento das várias combinações de fatores, de modo que a nenhum fator é dado mais ou menos peso do que outros no experimento. Ortogonal também se refere ao fato de que o efeito de cada fator pode ser matematicamente gerado independentemente dos efeitos dos outros fatores (CREVELING; FOWLKES, 1995).

A matriz experimental ortogonal é representada por $L_i(n^k)$, onde i é o número de corridas da matriz experimental, n é o número de níveis dos fatores e k é o número de fatores. A TABELA 2.3 mostra um exemplo de matriz ortogonal, na qual pode-se acomodar até sete fatores de dois níveis. Ela descreve as configurações dos níveis de cada um dos sete fatores que serão usadas nas oito corridas, a chamamos de matriz $L_8(2^7)$.

TABELA 2.3 – MATRIZ $L_8(2^7)$.

$L_8(2^7)$	A	B	C	D	E	F	G
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

FONTE: Taguchi *et al.* (2005).

Essa matriz pode ser usada para ilustrar o conceito de ortogonalidade da matriz experimental. Observe que:

- Os níveis 1 e 2 ocorrem o mesmo número de vezes em cada uma das colunas da matriz experimental;
- Para as quatro linhas com nível 1 na coluna A, tem-se duas linhas com nível 1 e duas linhas com nível 2 na coluna B. O mesmo pode ser observado sobre as quatro linhas com nível 2 da coluna A. Esse balanceamento se estende para todas as colunas.

Apesar da ortogonalidade existem alguns cuidados que se deve tomar, como Creveling e Fowlkes (1995) afirmam:

> [...] Ortogonalidade é uma propriedade de emparelhamento das colunas da matriz. Isso significa que as colunas podem ser deixadas em branco sem destruir a ortogonalidade dessa matriz. Entretanto, todos os tratamentos devem ser efetuados para que as condições de balanceamento sejam compridas e mantenham a ortogonalidade. (CREVELING; FOWLKES, 1995).

2.6.2 Modelo Aditivo

Taguchi pontua que muitas situações podem ser representadas pelo modelo aditivo, isto é, apenas pelos seus fatores principais (KRISHNAIAH; SHAHABUDDEN, 2012).

O modelo aditivo é dado da forma:

$$Y_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij}; \tag{2.43}$$

onde, Y é a resposta, μ é a média geral do experimento, α_i e β_j os efeitos dos fatores A e B respectivamente, ε o erro experimental.

No modelo aditivo os efeitos das interações não são desejáveis. O modelo aditivo também assume que os fatores são separáveis, ou seja, independentes. Caso as interações sejam levadas em conta, isso torna o modelo multiplicativo (não aditivo). Se o erro experimental for pequeno isso quer dizer que o modelo é aditivo, caso o erro experimental for muito grande isso pode implicar em alguma interação entre fatores, consequentemente, na não aditividade do modelo.

2.6.3 Análise das Médias (ANOM)

Uma das técnicas usadas na análise de Taguchi é a análise das médias (*analysis of mean* - ANOM). Ela pode ser visualizada no gráfico dos efeitos principais, é um meio de medir o efeito médio de cada fator e assim classificar as suas influências em um dado experimento.

O efeito do fator pode ser mensurado por:

$$efeito\ A = \bar{A}_{maior} - \bar{A}_{menor}; \tag{2.44}$$

onde, $\bar{A}_{maior}$ e $\bar{A}_{menor}$ são a maior e a menor média, respectivamente, do seu fator em algum dos seus níveis. Isto é, $\bar{A}_{maior}$ é a média das observações, do fator A no nível que apresentou a maior média. Assim como o $\bar{A}_{menor}$ é a média das observações do fator A no nível que apresentou a menor média.

Para mostrar esse conceito vai ser adotado uma matriz $L_4(2^3)$ com duas observações, TABELA 2.4. Então, para o cálculo se obtém as médias do fator A em

seus níveis. O $\bar{A}_1$ é a média das observações do fator A quando ele está no nível 1, equação (2.45). Da mesma maneira que $\bar{A}_2$ é a média das observações do fator A quando está no nível 2, equação (2.46).

TABELA 2.4 – MATRIZ $L_4(2^3)$ GENÉRICA COM DUAS OBSERVAÇÕES

Nº	A	B	C	Y_1	Y_2
1	1	1	1	$x_{1,1}$	$x_{1,2}$
2	1	2	2	$x_{2,1}$	$x_{2,2}$
3	2	1	2	$x_{3,1}$	$x_{3,2}$
4	2	2	1	$x_{4,1}$	$x_{4,2}$

FONTE: O autor (2018).

$$\bar{A}_1 = \frac{x_{1,1} + x_{1,2} + x_{2,1} + x_{2,2}}{4} \tag{2.45}$$

$$\bar{A}_2 = \frac{x_{3,1} + x_{3,2} + x_{4,1} + x_{4,2}}{4} \tag{2.46}$$

Assumindo hipoteticamente que $\bar{A}_2, \bar{B}_2$ e $\bar{C}_2$ são as maiores médias entre os níveis dos seus respectivos fatores, tem-se a ANOM para a matriz $L_4(2^3)$, mostrada na TABELA 2.5.

TABELA 2.5 – EXEMPLO DE ANOM PARA $L_4(2^3)$

	Fator A	Fator B	Fator C
Nível 1	$\bar{A}_1$	$\bar{B}_1$	$\bar{C}_1$
Nível 2	$\bar{A}_2$	$\bar{B}_2$	$\bar{C}_2$
Delta	$\bar{A}_2 - \bar{A}_1$	$\bar{B}_2 - \bar{B}_1$	$\bar{C}_2 - \bar{C}_1$
Rank	2	1	3

FONTE: O Autor (2018).

É usual classificar os efeitos dos fatores através do "*Rank*", ou seja, por meio do valor "Delta" (diferença entre a maior e menor média). Assim o efeito com maior Delta ganha a primeira classificação, ou seja, o *rank* número 1. O segundo maior valor

de Delta ganhará o *rank* número 2 e assim por diante. Desse modo, pode-se determinar quais os fatores que possuem maior influência no experimento.

2.6.4 Análise da Variância (ANOVA)

O principal objetivo da Análise da Variância (*analysis of variance* - ANOVA) é extrair dos resultados o quanto varia cada fator (ou interação) relativo à variação total observada nos resultados. Ela também revela a influência dos fatores e das interações entre fatores nos resultados, as respectivas significâncias e avalia o erro experimental.

Quando se aplica a ANOVA geralmente já se está em um estágio mais avançado da análise do Projeto de Experimento de Taguchi. Mesmo em um momento avançado, a análise da variância apresenta muitas informações essenciais que podem fazer necessária a reformulação do Projeto de Experimento. Como aponta Roy (2001) sobre a avaliação do erro "A avaliação do erro é uma parte importante, ela nos informa sobre fatores que podem não estar sendo considerados".

Assim, a sua função da ANOVA dentro do Projeto de Experimento, mais especificamente no Projeto de Parâmetro, é identificar os fatores com fortes contribuições e então utilizar o modelo preditivo e dar continuidade no teste de verificação. Já no Projeto de Tolerância, é utilizada para avaliar os fatores que valem a pena serem aprimorados junto às medidas relativas a custo.

O quadro da análise da variância, com dois fatores e uma interação, é dado pela TABELA 2.6 e tem como modelo:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \alpha_i\beta_j + \epsilon_{ijk} \tag{2.47}$$

onde, μ representa a média global, α_i representa o efeito do nível i do fator A ($i = 1, 2, \dots, a$), β_j representa o efeito do nível j do fator B ($j = 1, 2, \dots, b$), $\alpha_i\beta_j$ representa o efeito da interação entre o fator A e o fator B e ϵ_{ijk} representa um erro aleatório de cada observação ($k = 1, 2, \dots, n$).

TABELA 2.6 – QUADRO DA ANOVA

Fonte de variação	Soma de quadrados	Graus de liberdade	Quadrado médio	Estatística F
$Fator\ A$	SQ_A	$a-1$	$QM_A = \frac{SQ_A}{a-1}$	$\frac{QM_A}{QM_{resíduo}}$
$Fator\ B$	SQ_B	$b-1$	$QM_B = \frac{SQ_B}{b-1}$	$\frac{QM_B}{QM_{resíduo}}$
$Fator\ A * Fator\ B$	SQ_{AB}	$(a-1)(b-1)$	$QM_{AB} = \frac{SQ_{AB}}{(a-1)(b-1)}$	$\frac{QM_{AB}}{(a-1)(b-1)}$
$Resíduo$	$SQ_{resíduo}$	$ab(n-1)$	$QM_{resíduo}$	
$Total$	SQ_{total}	$abn-1$		

FONTE: O Autor (2018).

As equações para o cálculo do quadro da ANOVA são mostradas nas equações (2.48) a (2.56):

$$SQ_A = bn\sum_{i=1}^{a}(\bar{y}_{i..} - \bar{y}_{...})^2; \quad (2.48)$$

$$SQ_B = an\sum_{j=1}^{b}(\bar{y}_{.j.} - \bar{y}_{...})^2; \quad (2.49)$$

$$SQ_{AB} = n\sum_{i=1}^{a}\sum_{j=1}^{b}(\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}_{...})^2; \quad (2.50)$$

$$SQ_{resíduo} = \sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n}(\bar{y}_{ij.} - \bar{y}_{i..} - \bar{y}_{.j.} + \bar{y}_{...}); \quad (2.51)$$

$$\bar{y}_{i..} = \frac{\left(\sum_{j=1}^{b}\sum_{k=1}^{n} y_{ijk}\right)}{bn}; \quad 2.52)$$

$$\bar{y}_{.j.} = \frac{\left(\sum_{i=1}^{a}\sum_{k=1}^{n} y_{ijk}\right)}{an}; \quad (2.53)$$

$$\bar{y}_{ij.} = \frac{\left(\sum_{k=1}^{n} y_{ijk}\right)}{n}; \tag{2.54}$$

$$\bar{y}_{...} = \frac{\left(\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{n} y_{ijk}\right)}{abn}; \tag{2.55}$$

$$SQ_{total} = SQ_A + SQ_B + SQ_{AB} + SQ_{resíduo}. \tag{2.56}$$

A estatística $F_{\upsilon_1,\upsilon_2}$ de Snedecor é obtida com o nível de significância desejado (sendo usual adotar o nível de 5% ou 1%), sendo υ_1 igual aos graus de liberdade do numerador e υ_2 igual aos graus de liberdade do denominador. Se a estatística F do fator for maior que o $F_{\upsilon_1,\upsilon_2}$ tabelado, se diz que o fator é significativo para aquele experimento.

A ANOVA para mais de dois fatores é dada de forma similar a de dois fatores, apenas deve-se adequar a análise para a quantidade de fatores necessários.

2.6.5 Modelo Preditivo

A fórmula geral, supondo 3 fatores, para a equação preditiva é:

$$y = \bar{y}_{exp} + \left(\bar{y}_A - \bar{y}_{exp}\right) + \left(\bar{y}_B - \bar{y}_{exp}\right) + \left(\bar{y}_C - \bar{y}_{exp}\right); \tag{2.57}$$

onde, $\bar{y}_{exp}$ é a média da resposta geral (ou da Razão S/N) para toda a matriz experimental; $\bar{y}_A$, $\bar{y}_B$ e $\bar{y}_C$ são as melhores respostas médias (os maiores valores, entre todos os níveis) dos fatores A, B e C respectivamente.

Com essa simples aproximação não é necessário ajustar nenhuma curva, ou gerar nenhum tipo de equação para descrever os resultados do experimento.

Para esse tipo de previsão é usual utilizar apenas os fatores significantes, pois nem todos os fatores analisados contribuirão significativamente para o resultado do processo ou do sistema. Krishnaiah e Shahabudden (2012) sugerem tomar o número de fatores significativos igual à metade do número de graus de liberdade da matriz ortogonal utilizada para o experimento.

2.6.6 Teste de Verificação

O teste de verificação tem como objetivo verificar a otimização realizada pelo Projeto de Parâmetros, isto é, se as escolhas dos níveis dos parâmetros otimizam o sistema e geram o desempenho ótimo esperado. Também é utilizado para confirmar a não existência de interações, consequentemente tem-se a confirmação do modelo aditivo e a confirmação do modelo preditivo. Assim, o teste de verificação prova que o sistema é insensível aos ruídos, que a performance é repetível e que o projeto é robusto.

O teste de verificação consiste em duas etapas, ou seja, dois experimentos. O primeiro teste usando os fatores de controle em seus níveis ótimos. O segundo teste usando algum dos níveis não ótimos do experimento principal. Cada um desses experimentos sobre as mesmas condições de ruído que o experimento principal. Então os resultados dos experimentos são comparados com seus respectivos resultados preditivos.

Dessa forma é possível ter quatro respostas diferentes do teste de verificação (CREVELING; FOWLKES, 1995):

- Confirmação forte: os dois resultados estão próximos do modelo preditivo. Esse tipo de resultado confirma a repetitividade do sistema. Implica em um sistema robusto;
- Confirmação fraca: os resultados não são muito próximos dos previstos, porém estão errados na mesma proporção. Isso indica que os fatores do experimento não foram compreendidos completamente. A configuração ótima foi encontrada, mas ainda existem fatores importantes a serem encontrados e considerados. Mais experimentos são necessários;
- Confirmação ruim devido à interação: o resultado da configuração não ótima está próximo do resultado previsto, verificando repetitividade. Mas a configuração ótima não está próxima do resultado previsto. É provável que exista um problema com o modelo aditivo, devido a alguma interação forte que não está no modelo. O modelo aditivo pode não representar a realidade;
- Confirmação ruim devido a erro experimental: Nenhum dos resultados corresponde com os modelos previstos. Geralmente esse erro é dado

após várias repetições e é atribuído a problemas desconhecidos e problemas de medição.

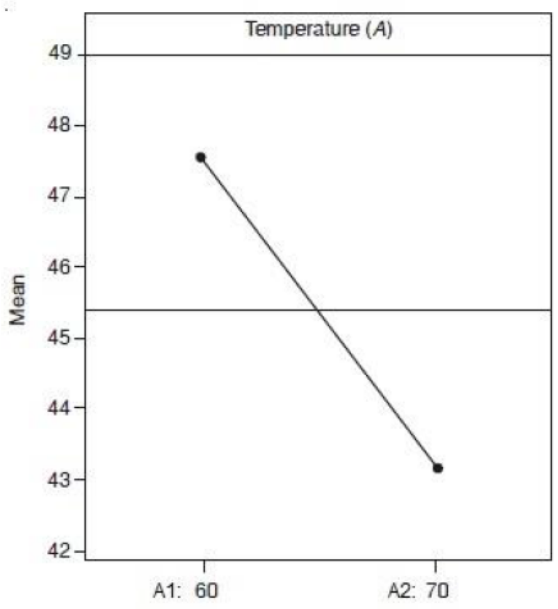
2.6.7 Gráfico dos Efeitos Principais

O gráfico de efeitos principais (*Main effects plot*) mostra o resultado médio de cada grupo (cada nível) de cada variável (fator), assim como é feito numericamente na ANOM descrita anteriormente.

Ao observar o gráfico de efeitos principais é possível ver o comportamento dos fatores quando eles mudam de nível, e assim saber qual o nível do fator irá fornecer melhor resposta.

O gráfico consiste em valores pontuais ligados por uma linha. O eixo x representa os níveis do fator analisado e o eixo y os valores médios da resposta no fator, como ilustra o GRÁFICO 2.1.

GRÁFICO 2.1 – EXEMPLO DO GRÁFICO DE EFEITOS PRINCIPAIS



FONTE: Adaptado de Krishnaiah e Shahabudden (2012).

2.6.8 Interações entre fatores

Segundo Roy (2001) interação quer dizer dependência. Quando se estuda alguns fatores juntos, isto é, um sistema, busca-se descobrir como cada fator afeta o

sistema. Caso os fatores sejam totalmente independentes, um único experimento irá produzir informação sobre o fator que pode ser usado todas as vezes que esse fator aparecer. Mas ocorre que naturalmente existe algum tipo de interação entre os fatores do grupo. Para entender como os fatores se influenciam é utilizado o gráfico de interações.

A questão das interações entre fatores é essencial em qualquer discussão do Projeto de Experimento. A razão para ter um sistema onde os fatores não possuem interação entre si é a reprodutibilidade. Logo o sucesso de um Projeto Robusto é obter um sistema em que a interação dos fatores de controle não exista ou seja muito pequena.

Há duas maneiras de lidar com as interações (CREVELING; FOWLKES, 1995):

- Estudar os efeitos de interação para quantificá-los;
- Construir o projeto de tal forma que se minimize a probabilidade de significantes interações e então evitar ter que estimá-las.

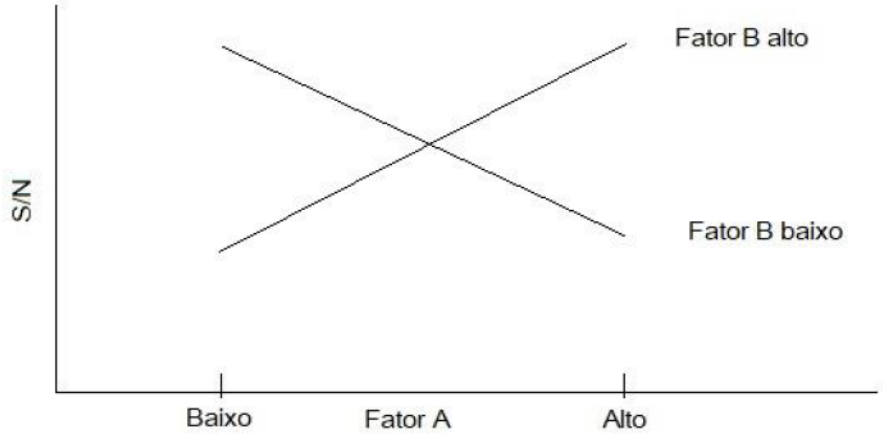
Em outros Projetos de Experimentos é comum mensurar as interações e lidar com elas, ao contrário do que Taguchi pretende com o seu modelo aditivo. Uma abordagem aditiva foca sua atenção nos fatores de controle. O Projeto Robusto assume que os efeitos principais são fortes o suficiente para se destacar dos ruídos aleatórios, isso é, os fatores de controle devem ser independentes e mais significantes que os efeitos de interação.

O objetivo não é ignorar as interações entre fatores, que realmente existem, mas otimizar a performance sem depender das interações. Isso é possível se as interações forem fracas ou possam ser atenuadas.

Quando as interações são analisadas, elas podem apresentar três tipos de comportamento: antissimétrico, monótono ou paralelo.

A interação antissimétrica é dada pelo GRÁFICO 2.2. Quando o fator B é analisado em seu nível baixo, a Razão S/N decai com o aumento do fator A. Por outro lado, com o fator B no nível alto, a Razão S/N aumenta quando o fator A muda de nível baixo para alto. Essa dependência dos fatores é recíproca.

GRÁFICO 2.2 – INTERAÇÃO ANTISSIMÉTRICA ENTRE FATORES



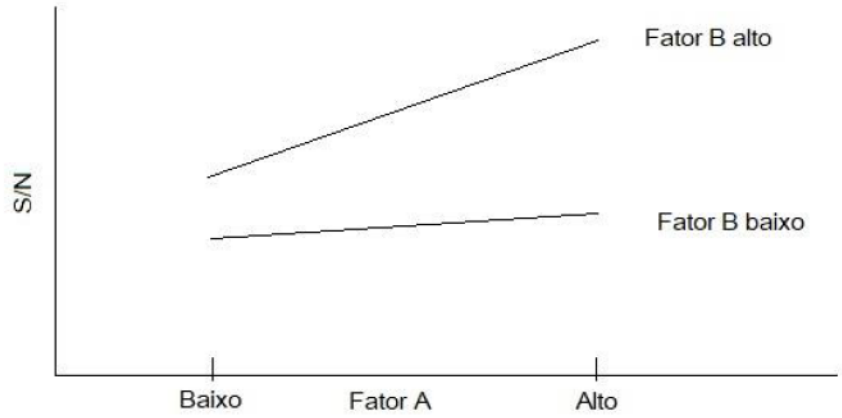
FONTE: O Autor (2018).

Portanto, a relação antissimétrica é obtida quando a inclinação do fator muda o sinal dependendo do nível do outro fator. Isso é uma condição não robusta, pois implica na alta dependência entre os fatores.

Uma interação suave entre os fatores de controle é uma situação tolerável, porém, não desejável. É chamada de interação monótona ou sinérgica, como mostra o GRÁFICO 2.3. Pode-se observar que o ótimo do fator B é no nível alto, independente do nível do fator A; o ótimo do fator A ocorre no nível alto independente do nível do fator B. Para esses casos não é necessário uma reotimização.

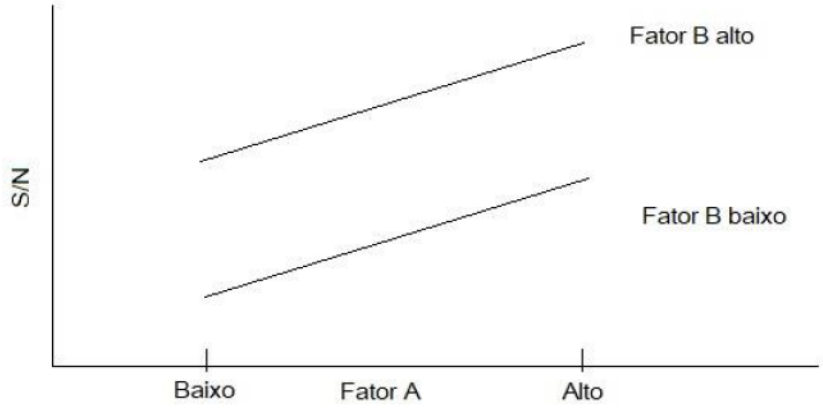
Quando os fatores de controle analisados são independentes, ou seja, quando não existe interação entre eles, o gráfico de interação entre fatores mostra linhas paralelas, como pode ser visto no GRÁFICO 2.4. Isso quer dizer que se mudar os níveis de A, a mudança correspondente na resposta vai ser a mesma, independente do nível do fator B. A relação é recíproca para o fator B em relação ao fator A.

GRÁFICO 2.3 – INTERAÇÃO SINÉRGICA ENTRE FATORES



FONTE: O Autor (2018).

GRÁFICO 2.4 – INTERAÇÃO PARALELA ENTRE FATORES



FONTE: O Autor (2018).

2.6.8.1 Construção dos gráficos de interação

De maneira similar ao que é feito para a construção dos gráficos de efeitos principais e na ANOM, a construção do gráfico das interações é realizada ao analisar-se as médias das respostas dos fatores envolvidos na interação.

Por exemplo, se os fatores A e B têm dois níveis cada (A_1, A_2 e B_1, B_2), então numericamente será necessário analisar $\overline{A_1B_1}$, $\overline{A_1B_2}$, $\overline{A_2B_1}$ e $\overline{A_2B_2}$. Isso quer dizer,

que $\overline{A_1B_1}$ é a média correspondente das respostas quando ambos os fatores A e B estão no nível 1.

Para, finalmente, construir o gráfico basta escolher qual fator será representado no das abscissas e qual fator será dividido em seus níveis, sendo representado por linhas.

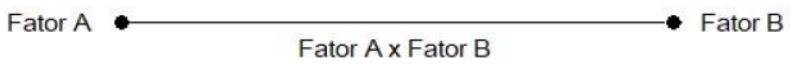
2.6.9 Gráficos Lineares

Atribuir os fatores de controle e interações entre fatores na matriz ortogonal de forma que a matriz experimental continue com todas as propriedades desejadas nem sempre é uma tarefa fácil. Para prevenir erros nessas atribuições e tornar esse trabalho mais rápido Taguchi criou uma representação gráfica, os gráficos lineares, no qual é mapeado onde se deve atribuir todos os fatores e suas interações. (Ver anexo A).

Esses gráficos são práticos pois fornecem uma maneira rápida para configurar os fatores e suas interações, de modo a não permitir confusão entre fator e interação nas colunas da matriz experimental.

Os gráficos lineares são construídos ligando-se pontos com linhas, onde os pontos representam os fatores e as linhas representam as interações entre os fatores que ela conecta, como mostra a FIGURA 2.15.

FIGURA 2.15 – REPRESENTAÇÃO DE GRÁFICO LINEAR



FONTE: O Autor (2018).

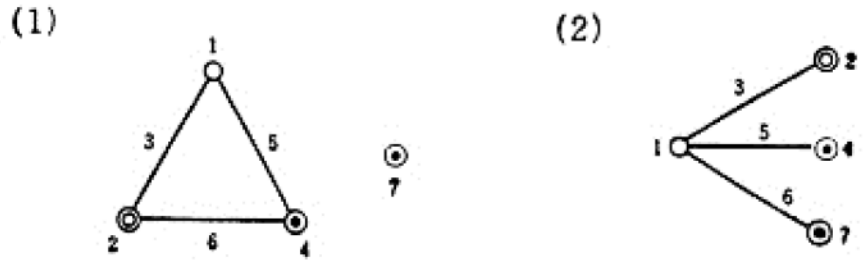
Alternativamente, pode-se utilizar a representação adotada por Taguchi (FIGURA 2.16), na qual são representados os fatores e interações por números. Os números representam em quais colunas da matriz experimental os fatores devem ser atribuídos, assim como ordena os fatores. O mesmo ocorre para as interações.

FIGURA 2.16 – GRÁFICO LINEAR $L_4(2^3)$

FONTE: O Autor (2018).

Os gráficos lineares podem ser simples ou complexos dependendo da matriz ortogonal que ele está representando. Também pode-se ter mais que uma configuração para a mesma matriz ortogonal, como mostrado na FIGURA 2.17, essa flexibilidade está relacionada à percepção de que nem todos os fatores são iguais em termos de "custo". O fator que possui maior custo é aquele em que levam mais tempo e são os mais caros de se mudar. Assim, esses fatores são atribuídos às colunas mais à esquerda da matriz onde os fatores mudam menos frequentemente de uma experiência executada para a próxima.

FIGURA 2.17 – DIFERENTES GRÁFICOS LINEARES PARA $L_8(2^7)$



FONTE: Taguchi *et al.* (2005).

2.6.10 Matriz triangular

Além dos gráficos lineares, Taguchi também utiliza as matrizes triangulares para mapear as interações entre os fatores. (Ver anexo A). A TABELA 2.7 mostra a matriz triangular da matriz ortogonal $L_8(2^7)$.

TABELA 2.7 – MATRIZ TRIANGULAR $L_8(2^7)$

1	2	3	4	5	6	7
(1)	3	2	5	4	7	6
	(2)	1	6	7	4	5
		(3)	7	6	5	4
			(4)	1	2	3
				(5)	3	2
					(6)	1
						(7)

FONTE: Taguchi *et al.* (2005).

A primeira linha da matriz triangular $L_8(2^7)$ representa os fatores, assim como cada linha representa o respectivo fator. Então se procuramos saber em qual coluna devemos atribuir a interação do fator 2 com o fator 4, basta olhar a intersecção da linha 2 com a coluna 4, e o valor 6 irá ser encontrado. Isso quer dizer que a interação "Fator 2 x Fator 4" deve ser atribuída à coluna 6 da matriz experimental.

2.7 BOOTSTRAP

2.7.1 A técnica *Jackknife*

A técnica *Jackknife* foi introduzida por Maurice Quenouille em 1949. O objetivo de Quenouille era melhorar a estimativa do parâmetro ao corrigir seu viés. Mais tarde, John Tukey em 1958 divulgou o método e descobriu que um uso mais importante do *Jackknife* era estimar o erro padrão de uma estimativa. Foi Tukey quem inventou o nome de *Jackknife* porque era uma ferramenta estatística com muitas utilidades (CHERNICK; LABUDDE, 2011).

Efron e Tibshirani (1993) descrevem a técnica do seguinte modo: suponha a amostra $x = (x_1, x_2, \dots, x_n)$ e o estimador $\hat{\theta} = S(x)$ do parâmetro θ, então a técnica consiste em estimar o viés e o erro padrão de $\hat{\theta}$. Funcionalmente a técnica *Jackknife* trabalha deixando uma observação fora de cada vez. Dessa forma, considere uma sub-amostra i, com $n - 1$ observações e com a observação x_i desconsiderada,

$$x_i = (x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n). \tag{2.58}$$

Então, é possível se obter com $i = 1, 2, \ldots, n$ n amostras *Jackknife*. A i –ésima amostra *Jackknife* consiste de um conjunto de dados com a i –ésima observação removida. Dessa forma,

$$\hat{\theta}_i = S(x_i) \tag{2.59}$$

é a i –ésima replicação *Jackknife* de $\hat{\theta}$. Assim a estimativa *Jackknife* do viés é definida por:

$$\widehat{bias}_{jack} = (n-1)\left(\hat{\theta}_{\cdot} - \hat{\theta}\right), \tag{2.60}$$

onde

$$\hat{\theta}_{\cdot} = \sum_{i=1}^{n} \frac{\hat{\theta}_i}{n}. \tag{2.61}$$

A estimativa *Jackknife* para o erro padrão é definido por

$$\widehat{se}_{jack} = \left[\frac{n-1}{n}\sum_{\mathrm{i=1}}^{\mathrm{n}}\left(\hat{\theta}_i - \hat{\theta}_{\cdot}\right)^2\right]^{1/2}. \tag{2.62}$$

2.7.1.1 Relação entre *Jackknife* e *Bootstrap*

Para o cálculo de $\hat{\theta}$ em n amostras *Jackknife*, para um n de no máximo 200 replicações, a técnica *Jackknife* realizará a tarefa de estimar o desvio padrão mais facilmente, do que a técnica *Bootstrap*. Mas ao observar a técnica *Jackknife*, o método utiliza uma quantidade limitada de informações da estatística $\hat{\theta}$, então pode parecer que *Jackknife* é menos eficiente que o *Bootstrap*. Logo, o *Jackknife* pode ser visto como uma aproximação do *Bootstrap* (EFRON; TIBSHIRANI, 1993).

2.7.1.2 Falha da técnica *Jackknife*

O estimador *Jackknife* frequentemente nos fornece uma boa aproximação do estimador *Bootstrap* para o erro padrão e do viés. Entretanto o *Jackknife* falha em estatísticas que não são suaves. Ser suave significa que uma pequena mudança nos dados causa uma pequena mudança na estatística. Um exemplo de uma estatística que não é suave é a mediana (EFRON; TIBSHIRANI, 1993).

2.7.2 O Método *Bootstrap*

Dois tipos de métodos estatísticos são usualmente utilizados em trabalhos científicos, o paramétrico e o não paramétrico. Nos métodos paramétricos se conhece o modelo de probabilidade dos dados e que pode ser identificado com os dados. Os métodos não paramétricos, são utilizados quando não se tem um modelo probabilístico associado aos dados (ABREU, 2011).

O *Bootstrap* é uma técnica estatística não paramétrica computacionalmente intensiva que permite a avaliação da variabilidade de estatísticas com base em dados de uma única amostra existente, a amostra original. Bradley Efron introduziu essa técnica em 1979 e desde ela então faz parte de estudos estatísticos teóricos e aplicados. O *bootstrap* surgiu quando Efron estudou como estimar a distribuição de uma amostra de uma estatística $\theta = S(x)$ com base em dados de amostra de tamanho n, com uma distribuição desconhecida (CHAVES NETO, 1991).

Para completar essa estimação dos parâmetros é necessário um grande número de reamostras, onde o computador passou a ser uma ferramenta essencial. Então, segundo Davison e Hinkley (1997) essa técnica passou a ser chamada de método computacionalmente intensivo.

Hesterberg *et al.* (2003) afirma que a amostra original, ou amostra mestre, deve representar a população da qual foi retirada. E as reamostras da amostra mestre representam o que se deve obter quando se retiram muitas amostras da população original. A distribuição *Bootstrap* da estatística, baseada em muitas reamostras, representa uma aproximação da distribuição amostral da estatística.

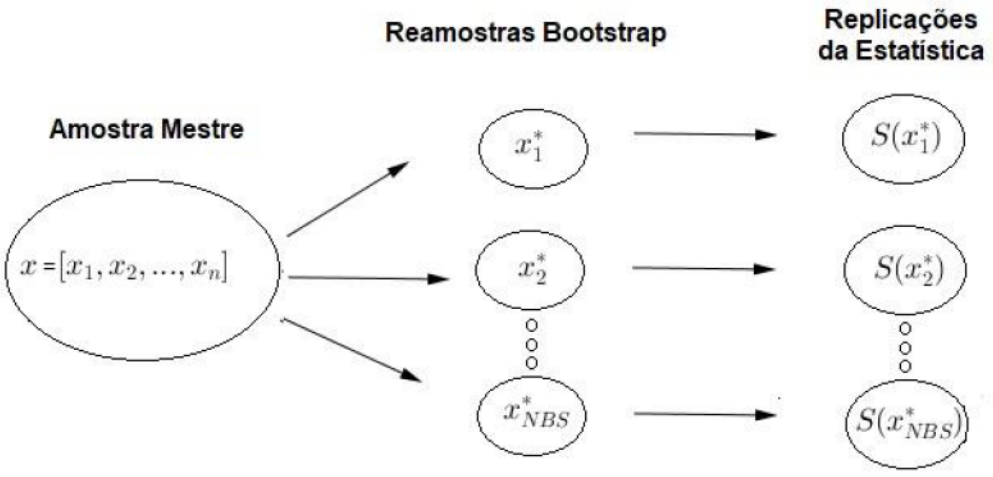
O *Bootstrap* é indicado para problemas em que os métodos estatísticos padronizados não existem ou quando a aplicação desses métodos é difícil. O objetivo do método é obter o desvio padrão, o intervalo de confiança, e a estimativa de

distribuição de probabilidade de estatísticas por meio do uso computacional para o cálculo estatístico.

Operacionalmente, o *Bootstrap* consiste obter um número NBS de reamostras com reposição da amostra original $x = [x_1, x_2, \ldots, x_n]$ de tamanho n. As reamostras (x_i^*) são obtidas com o mesmo tamanho da amostra mestre, n. Então é calculada a estatística de interesse $S(x_i^*)$ (ou de maneira simplificada, $S(x_i^*) = \hat{\theta}_i^*$) para cada reamostra *Bootstrap* e o conjunto de valores *Bootstrap* da estatística corresponde a uma estimativa da distribuição da estatística em questão. A FIGURA 2.18 representa esquematicamente o processo *Bootstrap* para se obter as estatísticas desejadas.

Para que a aplicação da técnica resulte em valores confiáveis deve ser feito, a partir da amostra mestre, centenas ou até milhares de reamostras do mesmo tamanho da amostra mestre. A reamostragem é feita com reposição, sempre selecionando os valores de forma aleatória. Assim, para a geração destas reamostras, as técnicas computacionais são de grande utilidade.

FIGURA 2.18 – MÉTODO *BOOTSTRAP*



FONTE: Adaptado de Efron e Tibshirani (1993).

De acordo com Steffen (2010), é importante salientar que a reamostragem não adiciona informações novas à amostra original. Pode parecer que o *Bootstrap* cria dados do nada, entretanto, os dados criados não são utilizados como se fosse real. O objetivo é verificar como as estatísticas obtidas das reamostras variam em decorrência da amostragem aleatória.

A principal vantagem dessa técnica de reamostragem, além de ser de fácil entendimento, é a generalidade com que se pode aplicá-la, pois a mesma necessita que poucas suposições sejam feitas, fornecendo respostas mais precisas que outros métodos de reamostragem (MARQUES, 2015).

Dois parâmetros são de fundamental importância para o *Bootstrap,* o desvio padrão da estatística e o vício, que são dadas pelas equações (2.63) e (2.64), respectivamente.

$$SE_{boot} = \sqrt{\frac{1}{NBS-1}\sum_{i=1}^{NBS}\left(\hat{\theta}_i^{*} - \frac{1}{NBS}\sum_{j=1}^{NBS}\hat{\theta}_i^{*}\right)^2}\ ; \tag{2.63}$$

onde, $\hat{\theta}_i^{*}$ são as NBS estatísticas *Bootstrap*.

O vício é utilizado para a verificação da validade do intervalo de confiança *Bootstrap*.

$$vício_{boot} = \overline{\hat{\theta}^{*}} - \hat{\theta}\ . \tag{2.64}$$

Onde, $\overline{\hat{\theta}^{*}}$ é a médias da estatísticas *Bootstrap* e $\hat{\theta}$ a estatística da amostra mestre.

2.7.3 Intervalo de Confiança

Montgomery (2009) define intervalo de confiança como “um intervalo estimado de um parâmetro é o intervalo entre duas estatísticas que incluem o valor verdadeiro do parâmetro com alguma probabilidade. ”

Como as estatísticas estimadas são dependentes de amostras e muitas vezes as amostras utilizadas para análise não tem tamanho suficiente para o uso adequado da estimação dos parâmetros, a utilização de intervalos de confiança é uma boa ideia.

Como consequência de uma amostra não ideal pode-se obter intervalos muito amplos e, em alguns casos, em que há maior complexidade do modelo, o intervalo pode estar fora do domínio do parâmetro. Uma alternativa para esses casos é utilizar as técnicas de reamostragem entre as quais se destaca o *Bootstrap* (LOIBEL; ALVES, 2013).

2.7.4 Intervalo de Confiança *Bootstrap*

2.7.4.1 Intervalo de confiança Padrão (Z)

O cálculo do intervalo de confiança *Bootstrap* padrão é dado pela equação:

$$IC_{boot\ Z} = (estatística \pm Z_{\alpha} SE_{boot}), \tag{2.65}$$

onde Z_{α} é o $\alpha - ésimo$ valor do escore da distribuição normal padrão.

2.7.4.2 Intervalo de confiança (t)

O intervalo de confiança *Bootstrap* t é dado pela equação:

$$IC_{boot\ t} = \left(estatística \pm t_{df} SE_{boot}\right). \tag{2.66}$$

sendo $df = (n - 1)$ o número de graus de liberdade, n o tamanho da amostra mestre, e SE_{boot} o desvio padrão *Bootstrap*.

A eficácia deste método para cálculo do intervalo de confiança é confiável desde que a distribuição da estatística de interesse seja aproximadamente normal e pouco viciada (EFRON; TIBSHIRANI, 1993).

2.7.4.3 Intervalo de confiança Percentílico (P)

O intervalo de confiança *Bootstrap* Percentílico é dado pela equação:

$$IC_{boot\ percentil} = \left[\hat{\theta} - P_{(1-\alpha/2)\%} D; \hat{\theta} - P_{(\alpha/2)\%} D\right]; \tag{2.67}$$

onde D é a diferença entre a média das estimativas *Bootstrap* com cada estimativa *Bootstrap* e $\hat{\theta}$ é a própria estimativa *Bootstrap*. A diferença D é dada pela equação seguinte.

$$D = \hat{\theta}_i^{*} - \overline{\hat{\theta}^{*}} \tag{2.68}$$

O intervalo de confiança percentílico é utilizado para verificar a veracidade do intervalo de confiança t. Se ambos os intervalos de confiança possuírem vício pequeno e a distribuição for próxima da normal, os intervalos de confiança *Bootstrap* serão próximos e podem ser ditos confiáveis. Caso não cumpram essas condições, nenhum dos tipos de intervalo deve ser usado.

Segundo Efron e Tibshirani (1993) o vício é considerado pequeno se o mesmo é menor que 25% do seu desvio padrão, isso é

$$DP_{vício} = \sqrt{\sum_{i=1}^{NBS} \frac{\left(\hat{\theta}_i^* - \hat{\theta}\right)^2}{NBS - 1}}. \tag{2.69}$$

2.7.4.4 Intervalo de confiança *Bias Corrected* (BC)

Segundo Efron e Tibshirani (1993), se o vício e a assimetria estão presentes, é recomendável utilizar os métodos de *Bootstrap* com correção, como o método *Bias Corrected* (BC) e *Bias Corrected and Accelerated* (BCa).

O método de estimação para intervalo de confiança BC utiliza como extremos do intervalo os percentis da distribuição *Bootstrap* ajustados para corrigir o vício e a assimetria da distribuição em questão (MARQUES, 2015).

O intervalo de confiança BC é dado pela equação:

$$IC_{BC} = \left[P_{P_I}\left(\hat{\theta}_i^*\right), P_{P_S}\left(\hat{\theta}_i^*\right)\right]. \tag{2.70}$$

Onde p_I e p_S é o percentil inferior e superior selecionado e calculado por meio dos parâmetros de correlação do vício com nível de confiança $(1-\alpha)100\%$. Isso é, para p_I e p_S:

$$P_I = \phi\left(2z_0 - z_{\alpha/2}\right); \tag{2.71}$$

$$P_S = \phi\left(2z_0 + z_{\alpha/2}\right). \tag{2.72}$$

onde z_0, que é parâmetro de correlação do vício, é dado por:

$$z_0 = \phi^{-1}(p_0). \tag{2.73}$$

Com p_0 definido como a probabilidade de uma estimativa ser inferior que a estimativa proveniente da amostra mestre que é dada por

$$p_0 = P\left(\hat{\theta}_i^* \leq \hat{\theta}\right). \tag{2.74}$$

2.7.4.5 Intervalo de confiança *Bias Corrected and accelerated* (BCa)

O método de estimação BCa é conveniente ser usado quando a assimetria estiver presente de uma maneira muito forte. O intervalo é dado por

$$IC_{BCa} = \left[P_{P_I}\left(\hat{\theta}_i^*\right), P_{P_S}\left(\hat{\theta}_i^*\right)\right]. \tag{2.75}$$

O intervalo BCa é semelhante ao método BC, sendo diferenciado por um ajuste de uma constante de aceleração a no cálculo dos percentis P_I e P_s (Andrews e Buchinsky, 2002). Os percentis são dados por

$$P_I = \phi\left(z_0 - \frac{z_0 + z_{\alpha/2}}{1 - a\left(z_0 + z_{\alpha/2}\right)}\right) \tag{2.76}$$

e

$$P_S = \phi\left(z_0 + \frac{z_0 + z_{\alpha/2}}{1 - a\left(z_0 + z_{\alpha/2}\right)}\right). \tag{2.77}$$

E a constante de aceleração a por

$$a = \frac{\sum_{i=1}^{NBS}\left(\hat{\theta}_{(.)} - \hat{\theta}_{(i)}\right)^3}{6\left(\sum_{i=1}^{NBS}\left(\hat{\theta}_{(.)} - \hat{\theta}_{(i)}\right)^2\right)^{3/2}}. \tag{2.78}$$

Como Marques (2015) descreve, “neste caso, $\hat{\theta}_{(i)}$ representa o valor das estimativas do parâmetro estudado para cada amostra i, excetuando-se a observação i da mesma e $\hat{\theta}_{(.)}$ representa o valor da média das estimativas $\hat{\theta}_{(i)}$”.

3 MATERIAL E MÉTODOS

3.1 PROBLEMA COM DADOS REAIS

Para o estudo de caso, foram analisados dados da indústria de alimentos. São cinco alimentos similares (produto A, produto B, produto C, produto D e produto E) que diferem apenas em seus sabores. As observações referem-se a níveis de acidez, em que cada sabor tem a sua especificação. Cada produto é produzido em três turnos de trabalho diferentes: turno 1 (06:00h às 12:00h), turno 2 (13:00h às 21:00h) e turno 3 (23:00h às 05:00h). As medidas foram feitas a cada 2 horas da linha de produção.

TABELA 3.1 – ESPECIFICAÇÕES DOS PRODUTOS

	Alvo	**LIE**	**LSE**
Produto A	0,475	0,420	0,530
Produto B	0,445	0,380	0,510
Produto C	0,405	0,330	0,480
Produto D	0,510	0,430	0,590
Produto E	0,410	0,320	0,500

FONTE: O Autor (2018).

TABELA 3.2 – OBSERVAÇÕES DO PRODUTO A

Produto A						
1º Turno	0,47	0,47	0,46	0,48	0,46	0,49
2º Turno	0,43	0,43	0,46	0,47	0,45	-
3º Turno	0,44	0,46	0,44	0,42	-	-

FONTE: O Autor (2018).

TABELA 3.3 – OBSERVAÇÕES DO PRODUTO B

(Continua)

Produto B							
1º Turno	0,46	0,48	0,59	0,59	0,59	0,57	0,47
2º Turno	0,46	0,41	0,45	0,43	0,46	0,43	0,45
3º Turno	0,46	0,51	0,45	0,44	0,50	0,43	0,46

(Conclusão)

Produto B								
1º Turno	0,45	0,44	0,49	0,51	0,48	0,46	0,44	0,45
2º Turno	0,45	0,47	0,46	0,45	-	-	-	-
3º Turno	-	-	-	-	-	-	-	-

FONTE: O Autor (2018).

TABELA 3.4 – OBSERVAÇÕES DO PRODUTO C

(Continua)

Produto C													
1º Turno	0,41	0,43	0,39	0,44	0,41	0,42	0,41	0,42	0,42	0,44	0,40	0,40	0,42
2º Turno	0,44	0,46	0,39	0,39	0,44	0,33	0,41	0,40	0,42	0,40	0,40	0,46	0,44
3º Turno	0,46	0,41	0,42	0,42	0,41	0,40	0,38	0,40	0,39	0,44	0,39	0,46	0,44

(Conclusão)

Produto C													
1º Turno	0,44	0,43	0,41	0,43	0,43	0,38	0,44	0,38	0,44	0,42	0,42	0,45	0,45
2º Turno	0,38	0,41	0,44	0,43	0,41	0,40	0,45	0,38	0,43	0,41	0,45	0,39	-
3º Turno	0,41	0,39	0,41	0,44	0,48	0,42	0,40	0,36	0,42	0,42	0,43	0,40	0,39

FONTE: O Autor (2018).

TABELA 3.5 – OBSERVAÇÕES DO PRODUTO D

Produto D										
1º Turno	0,50	0,49	0,49	0,48	0,53	0,50	0,55	0,49	0,48	0,52
2º Turno	0,52	0,53	0,51	0,46	0,52	0,54	0,47	0,54	0,54	-
3º Turno	0,57	0,48	0,51	0,48	0,45	0,51	0,52	0,51	0,48	-

FONTE: O Autor (2018).

TABELA 3.6 – OBSERVAÇÕES DO PRODUTO E

Produto E											
1º Turno	0,44	0,45	0,35	0,43	0,47	0,43	0,45	0,44	0,45	0,47	0,47
2º Turno	0,45	0,44	0,44	0,43	0,44	0,43	0,46	0,46	0,46	0,47	-
3º Turno	0,42	0,46	0,45	0,43	0,41	0,41	0,47	0,42	0,43	0,44	0,45

FONTE: O Autor (2018).

Nessa indústria 64% das vezes em que o produto foi retido por não se adequar às especificações, após a reanálise o produto foi totalmente liberado. Nos outros 36% dos casos têm-se algum tipo de descarte do material produzido, parcial ou completo.

Como o custo de produção não envolve apenas os custos da matéria-prima e da embalagem, a análise e a melhoria da produção podem reduzir outras despesas, tais como: gasto de tempo em reanálise, queixas dos consumidores e outros. A melhoria na produção traz ganhos econômicos não só para uma melhor utilização da matéria-prima, consequentemente um produto melhor, mas também traz competitividade para a marca.

A análise foi realizada com o aplicativo "Experimento Taguchi" que será descrito na seção seguinte, com a Razão S/N Nominal-é-Melhor Tipo I, Função Perda de Qualidade Média com valor $P = 6$, Capacidades Potencial do Processo e 10.000 reamostras *Bootstrap* para gerar os intervalos de confiança para os parâmetros analisados.

3.2 COMPARAÇÃO DO APLICATIVO

Para a comparação do aplicativo desenvolvido foi realizada uma comparação de resultados de um problema selecionado (PEACE, 1993). A comparação foi feita com os resultados obtidos pelo aplicativo “Experimento Taguchi”, o resultado da literatura, de um aplicativo que já está no mercado e também com um *software* livre.

Os *softwares* utilizados foram “Experimento Taguchi”, R i386 3.3.1 com os pacotes RcmdrPlugin.DoE, qualityTools e FrF2, além do programa Minitab 17.

3.2.1 Problema da Literatura

Para um dispositivo de controle, o tempo que é necessário para a mudança de estado de “*on*” para “*off*” quando a corrente passa por um dispositivo L.E.D (luz emitida pelo diodo) é crucial. A especificação do produto requer que a mudança de estado não seja maior que 0,005 segundos. Como um teste revelou que um grande número de unidades não cumpre com o tempo da especificação, a fábrica possui o objetivo: minimizar o tempo de mudança de estado de “*on*” para “*off*” (PEACE, 1993).

A matriz experimental utilizada para os fatores de controle neste experimento foi $L_8(2^7)$. Para os fatores de ruído a matriz utilizada é a $L_4(2^3)$.

Os fatores são apresentados na TABELA 3.7.

TABELA 3.7 – FATORES ANALISADOS DO DISPOSITIVO ELÉTRICO

Fator de controle	Nível 1	Nível 2
Tipo do pacote	Pacote 1	Pacote 2
Queima	Não	Sim
Teste de delay	Sem delay	Com delay
Fornecedor de componente	Vendedor X	Vendedor Y

Interação
Tipo do Pacote x Queima
Queima x Teste de delay
Tipo do Pacote x Queima

Fatores de Ruído	Nível 1	Nível 2
Umidade	Baixa	Alta
Secagem	Normal	Pistola de calor
Solda	Solda manual	Solda de onda

FONTE: Adaptado de Peace (1993).

E a matriz experimental com as amostras é exibida na TABELA 3.8.

TABELA 3.8 – TABELA EXPERIMENTAL DO DISPOSITIVO ELÉTRICO

(Continua)

							O	1	2	2	1
							N	1	2	1	2
							M	1	1	2	2
Corrida	**A**	**B**	**AxB**	**C**	**AxC**	**BxC**	**D**				
1	1	1	1	1	1	1	1	2,62	2,66	10,00	10,00
								2,56	2,56	8,24	10,00
2	1	1	1	2	2	2	2	2,27	2,29	5,65	6,07
								2,30	2,29	10,00	10,00
3	1	2	2	1	1	2	2	2,30	2,27	3,73	4,19
								2,33	2,33	8,09	10,00
4	1	2	2	2	2	1	1	2,61	2,53	6,36	8,10
								2,61	2,59	10,00	8,14
5	2	1	2	1	2	1	2	2,36	2,29	2,28	4,24
								2,29	2,25	4,22	2,25

(Conclusão)

6	2	1	2	2	1	2	1	2,54	2,58	6,51	6,33
								2,46	2,47	10,00	6,22
7	2	2	1	1	2	2	1	2,53	2,54	4,40	4,51
								2,51	2,52	6,25	8,12
8	2	2	1	2	1	1	2	2,30	2,27	10,00	4,27
								2,21	2,22	2,20	8,05

FONTE: Adaptado de Peace (1993).

3.2.2 Nomenclatura entre Programas

Como para a comparação do programa desenvolvido foi utilizado um problema proposto (PEACE, 1993) e nem todos os *softwares* apresentam a mesma nomenclatura, a tabela de nomenclaturas equivalente é dada a seguir.

TABELA 3.9 – NOMENCLATURA DOS FATORES PARA DIFERENTES SOFTWARES

Fator de controle	**Peace**	**Experimento Taguchi**	**Minitab 17**	**R**
Tipo do pacote	A	1	A	A
Queima	B	2	B	B
Teste de delay	C	3	C	C
Fornecedor de componente	D	4	D	D
Fatores de Ruído				
Umidade	M	1	A	A
Secagem	N	2	B	B
Solda	O	3	C	C

FONTE: O Autor (2018).

Neste trabalho adotou-se para as tabelas comparativas numéricas a nomenclatura mais comum entre os programas. Para os fatores de controle, como, “Tipo do Pacote” é denominado como fator A, o fator “Queima” é o fator B e assim por diante. Para os fatores de ruído foi adotada a nomenclatura do autor Peace (1993).

4 RESULTADOS E DISCUSSÃO

4.1 DESCRIÇÃO DO PROGRAMA EXPERIMENTO TAGUCHI

O aplicativo “Experimento Taguchi” foi desenvolvido em ambiente *Windows* com o programa *MATLAB 2013a*. É um programa projetado para a execução do Projeto de Experimento de Taguchi que utiliza as matrizes ortogonais, avalia a significância dos fatores e interações entre fatores, assim como executa os cálculos da estimação da Razão S/N, da Função Perda de Qualidade, da Capacidade Potencial do Processo e dos intervalos de confiança *Bootstrap* desses parâmetros.

A intenção ao se desenvolver esse programa é que se possa realizar a maioria da análise em uma única janela para manter uma experiência simples e não induzir erros ao usuário.

O fluxograma da FIGURA 4.1 mostra o funcionamento adequado para o programa em uma aplicação.

A tela inicial do aplicativo, mostrada na FIGURA 4.2, possui os principais campos (círculos numerados de 1 a 9) que são utilizados na análise e que serão descritos detalhadamente adiante. Os elementos são:

- Seleção do número de fatores (“Fatores”) e os respectivos níveis (“Níveis”);
- Seleção das interações entre fatores (“Calcular interações”);
- Escolha da matriz experimental (“Design”);
- Entrada de dados (“Amostra Mestre“) ou importação de dados, via arquivo .xls (“Importar Dados”);
- Entrada do número de reamostras *Bootstrap* (“Nº amostras Bootstrap”);
- Valor alvo T (“Valor nominal”).

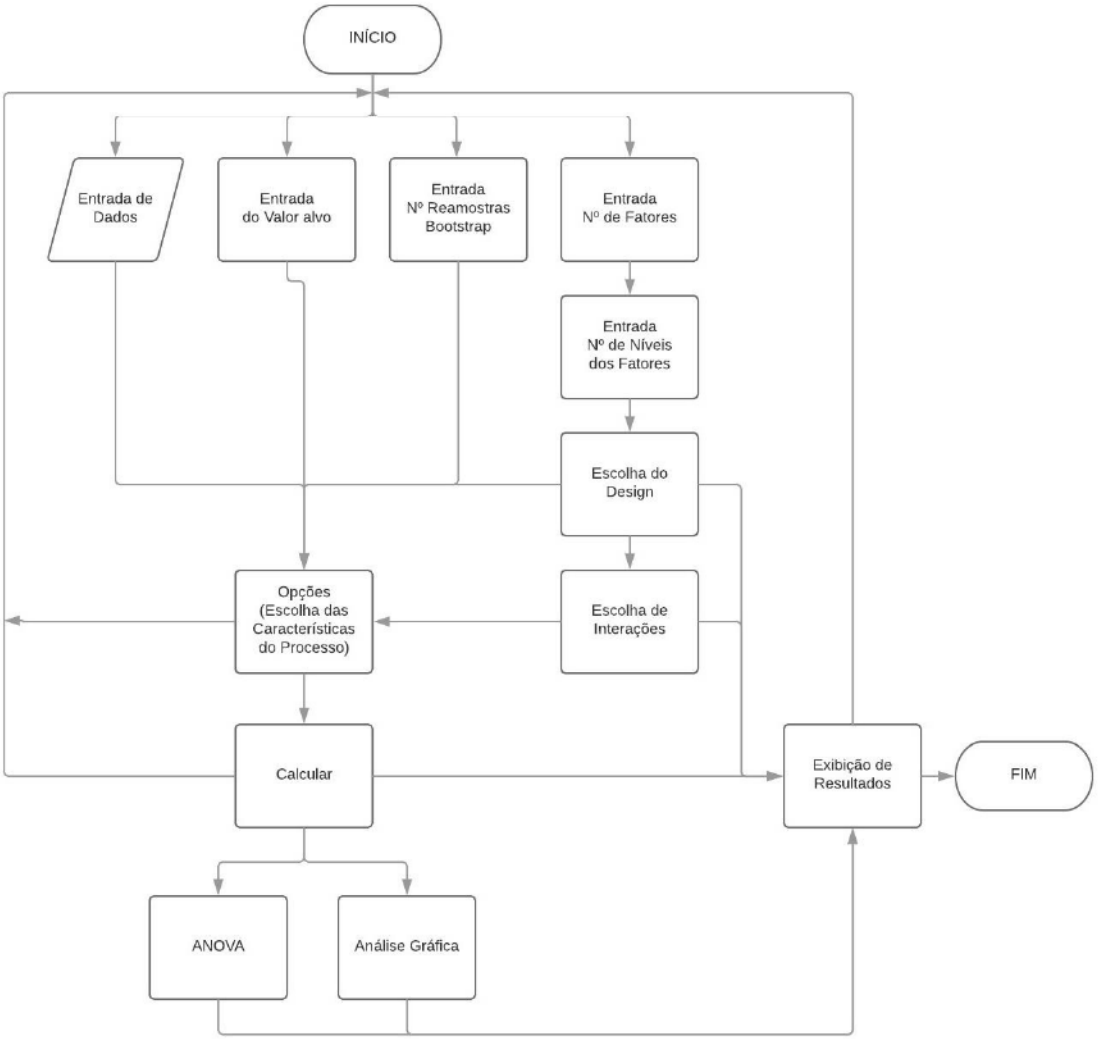
E, também, existem botões para:

- Cálculo da análise da variância (“ANOVA”);
- Gráfico dos efeitos principais (“Gráficos”);
- Seleção das características do processo (“Opções”);
- Exportação de resultados (“Exportar”).

A janela principal ainda possui abas individuais para exibição de dados e resultados de cada etapa efetuada. São elas:

- “Matriz experimental”;
- “Amostra”;
- Análise de Médias (“ANOM”) e análise das médias para a razão S/N (“ANOM-SN”);
- Análise da Variância (“ANOVA”) e análise da variância para a razão S/N (“ANOVA-SN”);
- “Geral”.

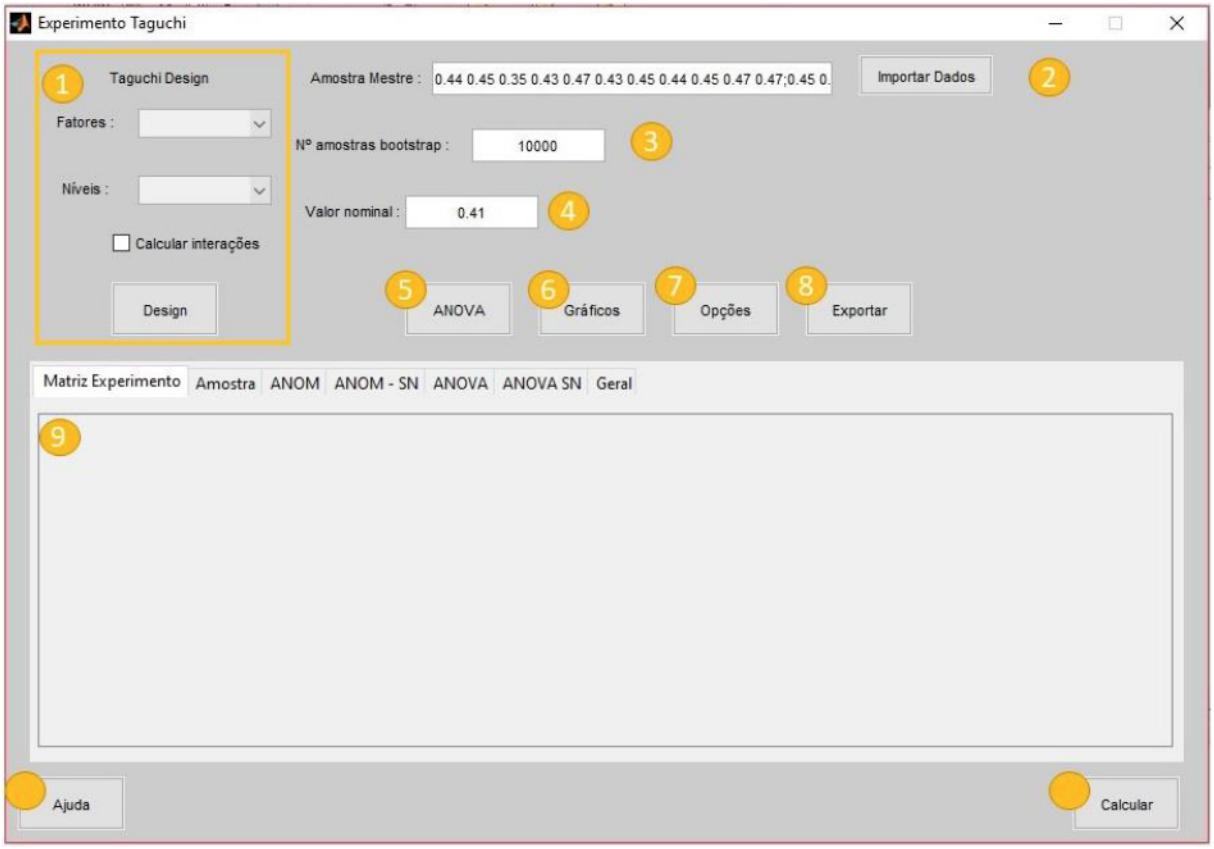
FIGURA 4.1 – FLUXOGRAMA DE FUNCIONAMENTO DO PROGRAMA



FONTE: O Autor (2018).

Em seguida tem-se a descrição dos campos e são divididos como apresenta a FIGURA 4.2.

FIGURA 4.2 – JANELA PRINCIPAL



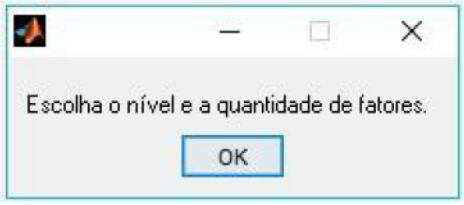
FONTE: O Autor (2018).

Campo 1: na seleção de “Níveis” pode-se escolher as opções “2, 3, 4, 5 ou Mix”, onde a opção “2” representa fatores com 2 níveis e assim sucessivamente para as outras opções até o nível 5. A opção “Mix” refere-se às matrizes experimentais com fatores de níveis diferentes, como a matriz $L_{32}(2x4^9)$.

Na opção “Fatores” existe a seleção de quantos fatores serão analisados, podendo ser criados de 2 a 31 fatores, dependendo do nível desejado dos fatores.

Deve-se escolher primeiro o nível dos fatores e em seguida a quantidade deles, antes de escolher a matriz experimental. Caso as opções de níveis e fatores não forem selecionadas antes da escolha da matriz, o programa irá alertar para fazê-lo, como mostra na FIGURA 4.3.

FIGURA 4.3 – ALERTA DE ESCOLHA DE NÍVEL E FATORES

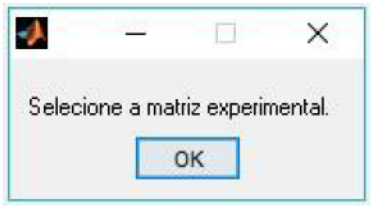


Fonte: O Autor (2018).

O botão “Design” abre a seleção de matrizes experimentais (matrizes ortogonais), como mostra a FIGURA 4.5, que será de acordo com o nível e número de fatores selecionados. As matrizes ortogonais programadas no aplicativo (ver anexo A) foram todas obtidas de Taguchi *et al.* (2005) e são listadas na TABELA 4.1. Após a seleção da matriz experimental ela é exibida no campo 9 na aba “Matriz Experimento”.

Caso o usuário tente escolher as interações antes da escolha da matriz experimental, o programa irá pedir para que ele escolha a matriz primeiro, através da mensagem mostrada na FIGURA 4.4.

FIGURA 4.4 – ALERTA ESCOLHA DA MATRIZ EXPERIMENTAL



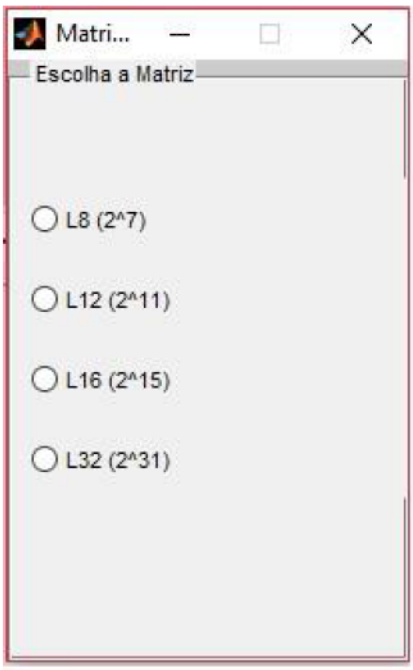
Fonte: O Autor (2018).

TABELA 4.1 – LISTA DE MATRIZES ORTOGONAIS NO PROGRAMA EXPERIMENTO TAGUCHI

Matrizes ortogonais no programa “Experimento Taguchi”	
$L_4(2^3)$	$L_{25}(5^6)$
$L_8(2^7)$	$L_{18}(2x3^7)$
$L_{16}(2^{15})$	$L_{32}(2x4^9)$
$L_{32}(2^{31})$	$L_{36}(2^{11}x3^{12})$
$L_9(3^4)$	$L_{50}(2x5^{11})$
$L_{27}(3^{13})$	$L_{54}(2x3^{25})$
$L_{16}(4^5)$	

FONTE: O Autor (2018).

FIGURA 4.5 – JANELA DE OPÇÕES PARA A MATRIZ EXPERIMENTAL

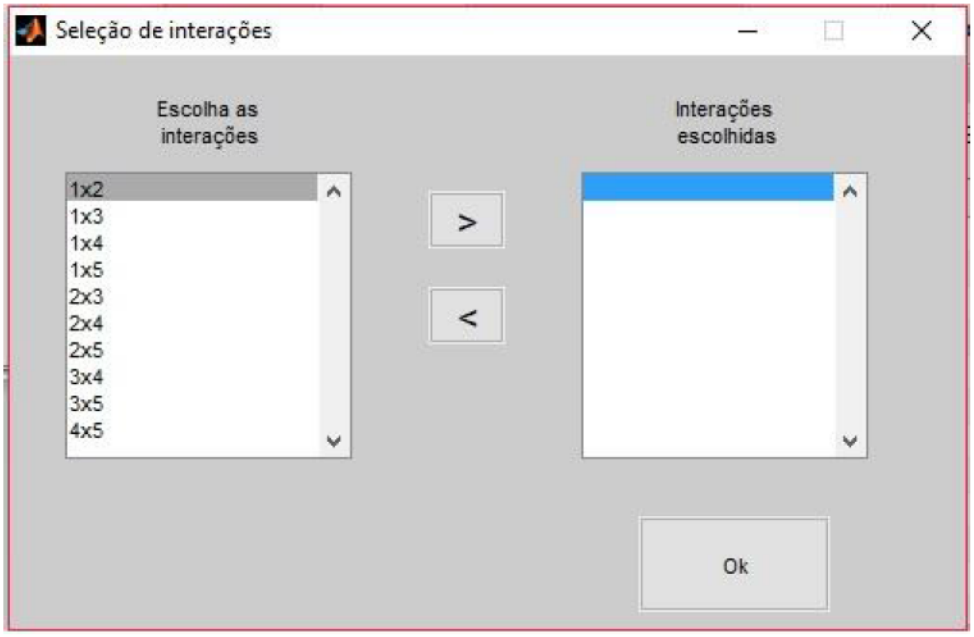


FONTE: O Autor (2018).

Se for de interesse calcular a interação entre os fatores é possível selecionar o campo “Calcular interações”, o programa irá mostrar uma janela para selecionar as interações desejadas, FIGURA 4.6. A janela “seleção de interações” consiste em duas áreas, a do lado esquerdo que contém todas as possíveis interações (levando em consideração o número de fatores selecionados) e a área do lado direito que mostrará todas as interações selecionados pelo usuário. É possível selecionar e retirar as interações com os botões que estão no centro da janela, quando for necessário finalizar a seleção basta apertar o botão “Ok”.

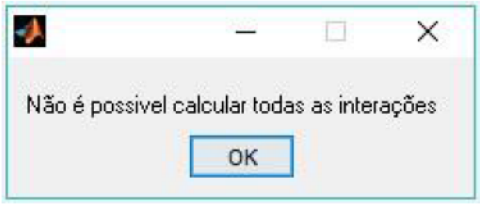
Caso o programa verifique que não é possível calcular todas as interações, ele irá mostrar uma janela de aviso (FIGURA 4.7) ou, se for detectado que existe a confusão na atribuição de fatores e interação entre fatores nas colunas da matriz (fatores e interações atribuídas à mesma coluna da matriz), o programa irá mostrar um aviso deixando que o usuário tome a decisão de continuar ou selecionar novamente as interações (FIGURA 4.8).

FIGURA 4.6 – JANELA PARA ESCOLHA DE INTERAÇÕES



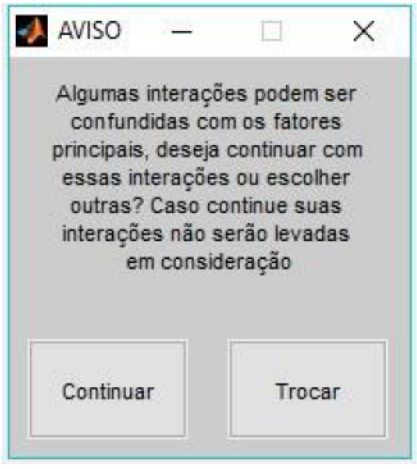
FONTE: O Autor (2018).

FIGURA 4.7 – ALERTA DE IMPOSSIBILIDADE DE CALCULAR AS INTERAÇÕES ENTRE FATORES



Fonte: O Autor (2018).

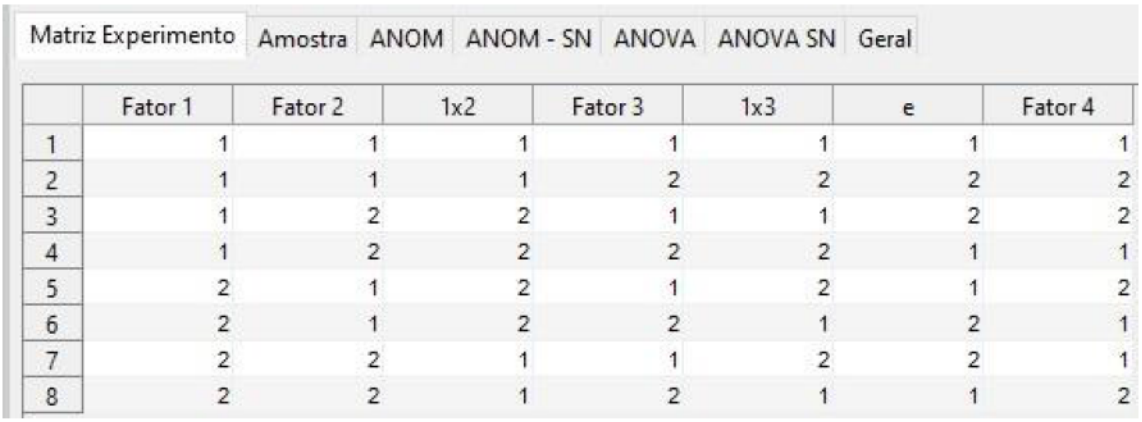
FIGURA 4.8 – ALERTA DE CONFUSÃO ENTRE FATORES E INTERAÇÃO ENTRE FATORES



Fonte: O Autor (2018).

Para as colunas da matriz experimental que são atribuídos fatores, essas ganham o nome “Fator #”, FIGURA 4.9. As colunas das interações ganham o nome com os números dos fatores, separados por um “x”, como a interação do fator 1 com o fator 2, “1x2”. As colunas que não forem utilizadas são nomeadas com “e”.

FIGURA 4.9 – MATRIZ EXPERIMENTAL



Matriz Experimento | Amostra | ANOM | ANOM - SN | ANOVA | ANOVA SN | Geral

	Fator 1	Fator 2	1x2	Fator 3	1x3	e	Fator 4
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

FONTE: O Autor (2018).

Campo 2: é onde se realiza a entrada da amostra mestre. A inserção da amostra pode ser feita de dois modos, manualmente ou via arquivo do *Microsoft Excel* (.xls). Ao inserir a amostra via arquivo .xls, no campo de digitação irá aparecer a mensagem “Amostra carregada”.

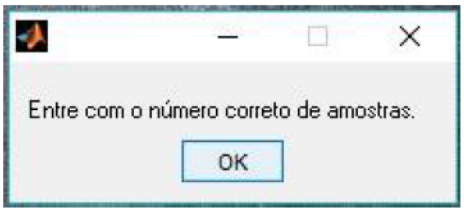
Quando a amostra for inserida manualmente deve-se seguir algumas regras para que o programa possa ler os dados corretamente, são elas:

- Para separar uma observação de outra deixar um espaço em branco. Exemplo: “12 14 12 15 16”.
- Para separar diferentes amostras usar ponto e vírgula. Exemplo: “12 14 12 15 16; 12 11 13 14 13”.
- Para escrever números decimais use “.” Ao invés de “,”. Exemplo: “12.3”.
- Sempre colocar amostras balanceadas, caso as amostras não sejam de mesmo tamanho é usado “NaN” no lugar das observações que estão faltando. Exemplo: “12 14 12 15 16; 12 11 13 NaN NaN”. O programa não levará em consideração os campos “NaN”.

Após a entrada da amostra (manual ou via .xls) ela irá aparecer no campo 9, na aba “Amostra”.

Caso a amostra inserida não esteja de acordo com a matriz experimental, o programa irá mostrar um aviso, conforme FIGURA 4.10.

FIGURA 4.10 – ALERTA DE PARA NÚMERO DE AMOSTRAS



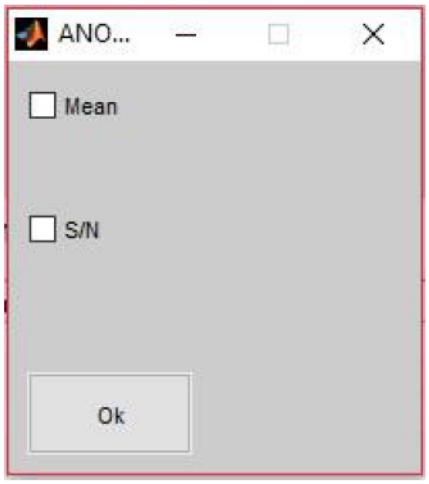
Fonte: O Autor (2018).

Campo 3: campo reservado para o número de reamostras *Bootstrap* desejada para o uso na geração dos intervalos de confiança. Atenção para o número de reamostras, caso for muito grande o programa poderá levar algum tempo para gerar os resultados.

Campo 4: campo reservado para especificar qual o valor alvo. (Caso sua característica de análise não tenha valor alvo o programa não levará em conta esse valor).

Campo 5: para realizar a ANOVA o usuário tem duas opções, FIGURA 4.11, pode escolher realizar a ANOVA para a média ou para a Razão S/N. Para realizar a ANOVA é necessário ter realizado a análise completa pelo menos uma vez (Isso quer dizer ter efetuado pelo menos uma vez a análise experimental). Os resultados vão aparecer no campo 9 nas abas “ANOVA” e “ANOVA-SN”, como apresenta as FIGURAS 4.12 e 4.13.

FIGURA 4.11 – JANELA DE OPÇÕES PARA ANÁLISE DA VARIÂNCIA



FONTE: O Autor (2018).

FIGURA 4.12 – ANOVA

Matriz Experimento | Amostra | ANOM | ANOM - SN | ANOVA | ANOVA SN | Geral

	1	2	3	4	5	6
1	Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
2	Fator 1	0.3973	1	0.3973	0.1851	0.6692
3	Fator 2	51.0289	1	51.0289	23.7794	1.5886e-05
4	Error	90.1290	42	2.1459		
5	Total	141.6444	44			

FONTE: O Autor (2018).

FIGURA 4.13 – ANOVA S/N

Matriz Experimento | Amostra | ANOM | ANOM - SN | ANOVA | ANOVA SN | Geral

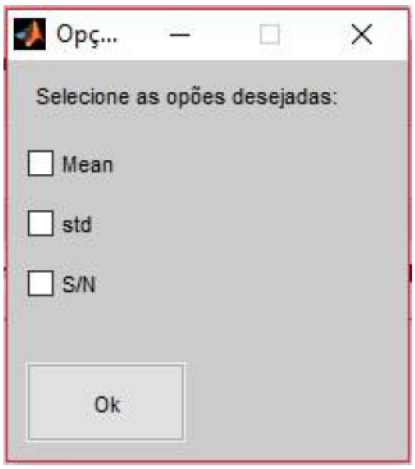
	1	2	3	4	5	6
1	Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
2	Fator 1	0.0012	1	0.0012	0.1731	0.7490
3	Fator 2	0.0067	1	0.0067	1.0000	0.5000
4	Error	0.0067	1	0.0067		
5	Total	0.0147	3			

FONTE: O Autor (2018).

Campo 6: botão para realizar a análise gráfica, isso é, o gráfico de efeitos principais. É possível escolher até 3 opções de análise, são elas: análise das médias,

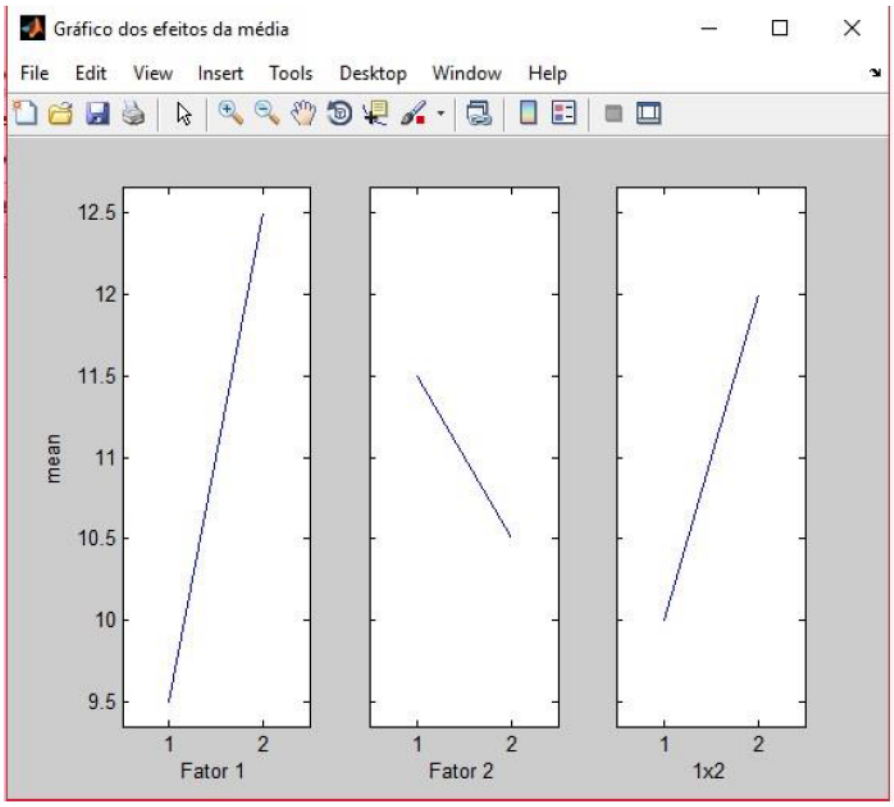
análise das médias do desvio padrão e análise das médias da Razão S/N, FIGURA 4.14. Os resultados são exibidos com os gráficos de efeitos principais, (FIGURA 4.15), ou quando o usuário selecionou alguma interação na janela “seleção de interações” (FIGURA 4.16).

FIGURA 4.14 – OPÇÕES PARA ANÁLISE GRÁFICA



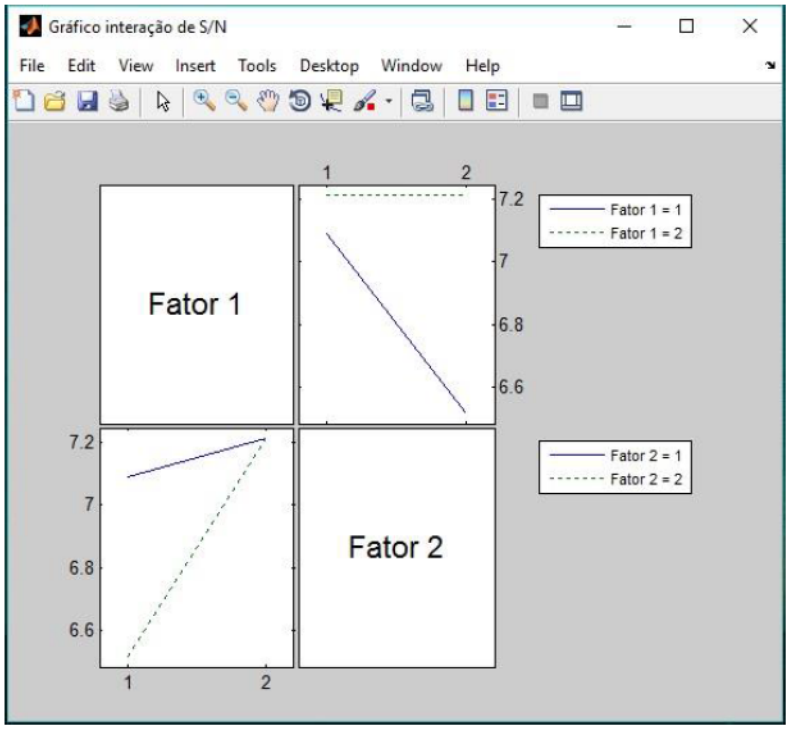
FONTE: O Autor (2018).

FIGURA 4.15 – ANÁLISE DOS EFEITOS PRINCIPAIS



FONTE: O Autor (2018).

FIGURA 4.16 – ANÁLISE DA INTERAÇÃO DOS EFEITOS PRINCIPAIS



FONTE: O Autor (2018).

Campo 7: o botão “opções” é um dos mais importantes, é nele onde se seleciona as características do processo e quais parâmetros se quer calcular (FIGURA 4.17). É obrigatório a entrada da característica do processo, isso é, se o processo é Nominal-é-Melhor (tipo I ou tipo II, para efeitos do cálculo da Função Perda essas duas opções são encaradas apenas como Nominal-é-Melhor), Menor-é-Melhor ou Maior-é-Melhor, para o cálculo da Razão S/N. Caso essa opção não for selecionada o programa irá avisar o usuário, como mostra a FIGURA 4.18.

Os parâmetros Função Perda de Qualidade, Capacidade Potencial do Processo e intervalo de confiança são opcionais para a análise.

Para o cálculo da Função Perda de Qualidade é necessária a entrada da constante de proporcionalidade k ou dos parâmetros P e Δ.

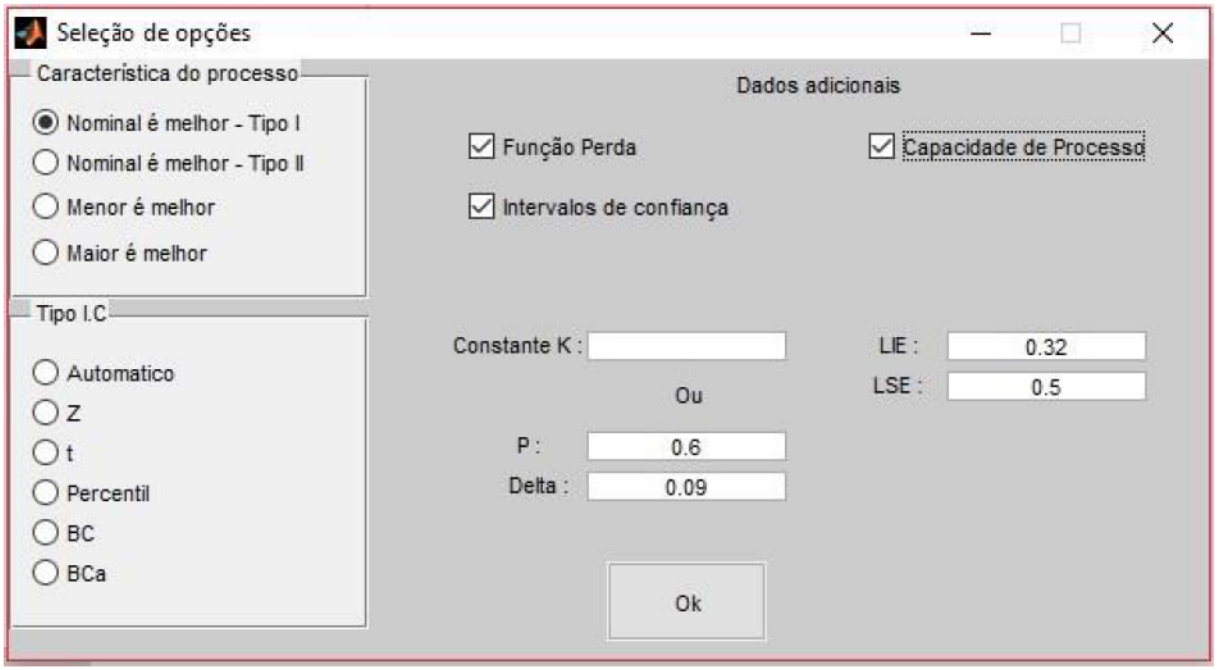
Para o cálculo da Capacidade Potencial do Processo o usuário deve entrar com os limites de especificação (LIE e LSE).

O programa apresenta cinco tipos de intervalos de confiança: Normal, t, Percentílico, *Bias corrected*, *Bias corrected and Accelerated* e uma opção de escolha

automática. A escolha automática escolhe qual o melhor método para gerar o intervalo de confiança.

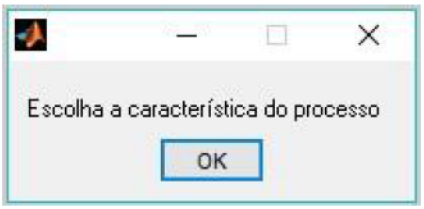
Para finalizar as escolhas basta apertar o botão “Ok”.

FIGURA 4.17 – JANELA DE OPÇÕES PARA ESCOLHA DE CARACTERÍSTICAS DE ANÁLISE



FONTE: O Autor (2018).

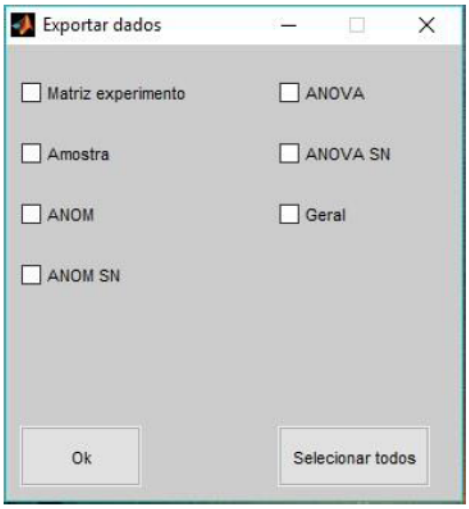
FIGURA 4.18 – ALERTA DE SELEÇÃO DA RAZÃO S/N



Fonte: O Autor (2018).

Campo 8: o botão “Exportar” dá a opção de exportar qualquer janela do campo 9 que seja necessária, como apresenta a FIGURA 4.19. É possível também selecionar todas a janela de uma vez com o botão “Selecionar todos”. O programa irá criar um arquivo “Result.xls” na área do computador que usuário selecionar.

FIGURA 4.19 – JANELA EXPORTAÇÃO DE DADOS



FONTE: O Autor (2018).

Campo 9: a maioria dos resultados do programa será exibida no campo 9. O campo é dividido em sete abas: “Matriz Experimento”, “Amostra”, “ANOM” (FIGURA 4.21), “ANOM-SN” (FIGURA 4.22), “ANOVA”, “ANOVA-SN” e “Geral”.

O resultado da ANOVA só será exibido nas abas “ANOVA” e “ANOVA-SN” caso seja solicitado pelo usuário com o botão “ANOVA”.

O resultado da análise das médias e análise das médias da razão S/N são exibidas automaticamente em seus respectivos campos sem a necessidade de exigir esse resultado.

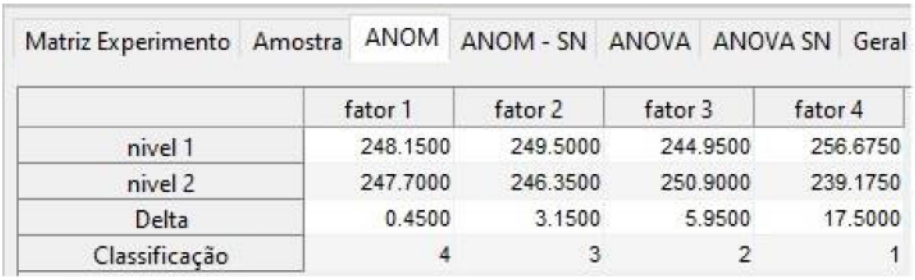
A aba “Geral” é onde aparece todos os parâmetros selecionados para análise no botão “Opções”, também será exibido a matriz experimental, média e variância de cada amostra, como mostra a FIGURA 4.20. As amostras não são exibidas nessa aba.

FIGURA 4.20 – JANELA EXIBIÇÃO DE RESULTADOS

	Fator 1	Fator 2	e	Média	Variância	S/N	SN-LI95%	SN-LS95%	SN-LI99%	SN-LS99%
1	1	1	1	115	3.1667	36.2079	33.3241	44.3631	31.5303	44.3631
2	1	2	2	113	1	41.0616	36.2465	44.3242	36.2465	45.6278
	Função Perda	FP-LI95%	FP-LS95%	FP-LI99%	FP-LS99%	Cp	Cp-LI95%	Cp-LS95%	Cp-LI99%	Cp-LS99%
1	0.7917	0.4808	1.3809	0.3846	1.5385	3.7463	2.9532	5.2874	2.7931	7.0293
2	1.2500	0.7885	1.7308	0.6538	1.9158	6.6667	5.2662	10.2494	5.1247	14.5896
	Cpk	Cpk-LI95%	Cpk-LS95%	Cpk-LI99%	Cpk-LS99%	Cpm	Cpm -LI95%	Cpm -LS95%	Cpm -LI99%	Cpm -LS99%
1	3.7463	2.7180	4.8050	2.7115	5.1640	3.7463	2.9814	5.5635	2.8886	7.6285
2	6	4.5233	7.5580	4.5233	8.1680	2.9814	2.8165	3.1699	2.7912	3.2374

FONTE: O Autor (2018).

FIGURA 4.21 – ANOM

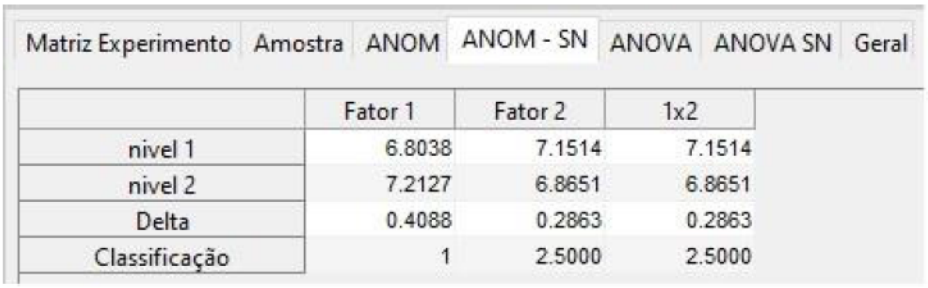


Matriz Experimento | Amostra | ANOM | ANOM - SN | ANOVA | ANOVA SN | Geral

	fator 1	fator 2	fator 3	fator 4
nivel 1	248.1500	249.5000	244.9500	256.6750
nivel 2	247.7000	246.3500	250.9000	239.1750
Delta	0.4500	3.1500	5.9500	17.5000
Classificação	4	3	2	1

FONTE: O Autor (2018).

FIGURA 4.22 – ANOM S/N



Matriz Experimento | Amostra | ANOM | ANOM - SN | ANOVA | ANOVA SN | Geral

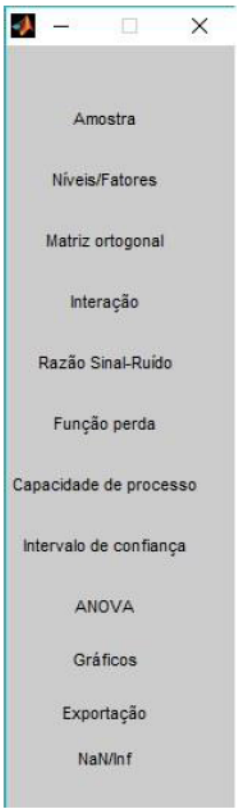
	Fator 1	Fator 2	1x2
nivel 1	6.8038	7.1514	7.1514
nivel 2	7.2127	6.8651	6.8651
Delta	0.4088	0.2863	0.2863
Classificação	1	2.5000	2.5000

FONTE: O Autor (2018).

Campo 10: após todas as seleções de parâmetros para análise, o botão “Calcular” (mostrado na FIGURA 4.2) faz o programa executar a análise. Os resultados irão aparecer no campo 9.

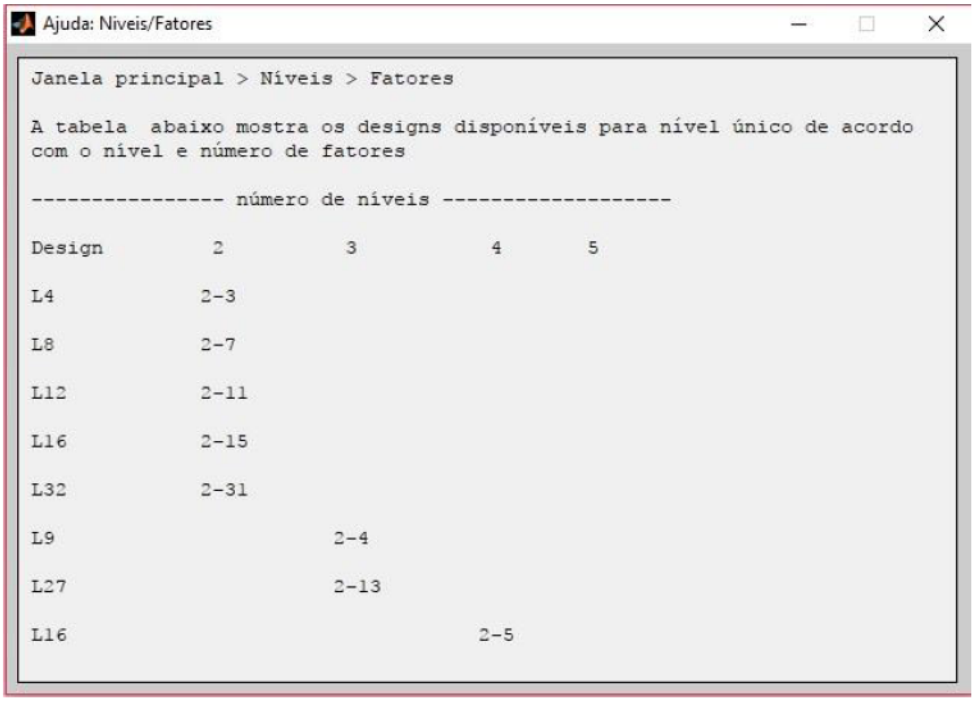
Campo 11: o botão “Ajuda” fornece todas as informações necessárias para o usuário executar as análises desejadas. É estruturada em tópicos, (FIGURA 4.23), que permite ao usuário selecionar qual deseja. O programa apresenta um breve resumo teórico do tópico e como executar o procedimento no programa, (FIGURA 4.24).

FIGURA 4.23 – JANELA DE MENU DE “AJUDA”



FONTE: O Autor (2018).

FIGURA 4.24 – EXEMPLO DE “AJUDA”



FONTE: O Autor (2018).

4.1 RESULTADO DOS DADOS REAIS

TABELA 4.2 – RESULTADO DA RAZÃO S/N

S/N	Produto A	Produto B	Produto C	Produto D	Produto E
1º Turno	32,116	18,870	26,676	26,752	22,359
2º Turno	27,974	28,213	22,723	24,670	30,112
3º Turno	28,609	23,816	23,727	23,349	26,688

FONTE: O Autor (2018).

TABELA 4.3 – RESULTADO DA FUNÇÃO PERDA MÉDIA DE QUALIDADE

$\bar{L}(y)$	Produto A	Produto B	Produto C	Produto D	Produto E
1º Turno	0,293	8,164	0,657	0,546	1,544
2º Turno	2,080	0,435	1,072	0,864	1,214
3º Turno	2,958	1,799	0,885	1,162	0,781

FONTE: O Autor (2018).

TABELA 4.4 – RESULTADO DO ÍNDICE C_p

C_p	Produto A	Produto B	Produto C	Produto D	Produto E
1º Turno	1,568	0,383	1,282	1,153	0,892
2º Turno	1,024	1,247	0,825	0,887	2,145
3º Turno	1,122	0,724	0,925	0,782	1,486

FONTE: O Autor (2018).

TABELA 4.5 – RESULTADO DO ÍNDICE C_{pk}

C_{pk}	Produto A	Produto B	Produto C	Produto D	Produto E
1º Turno	1,473	0,084	1,019	1,052	0,586
2º Turno	0,521	1,203	0,721	0,838	1,239
3º Turno	0,408	0,509	0,801	0,695	1,066

FONTE: O Autor (2018).

TABELA 4.6 – RESULTADO DO ÍNDICE C_{pm}

C_{pm}	Produto A	Produto B	Produto C	Produto D	Produto E
1º Turno	1,508	0,285	1,006	1,104	0,399
2º Turno	0,566	1,236	0,788	0,877	0,717
3º Turno	0,474	0,608	0,867	0,757	0,816

FONTE: O Autor (2018).

TABELA 4.7 – RESULTADO DO INTERVALO DE CONFIANÇA PARA A RAZÃO S/N 95%

S/N	Intervalo de confiança Z 95%									
	Produto A		Produto B		Produto C		Produto D		Produto E	
	LI	LS	LI	LS	LI	LS	LI	LS	LI	LS
1º Turno	-	-	12,793	24,946	22,064	31,289	11,968	41,536	5,842	38,877
2º Turno	-6,303	62,251	18,878	37,549	17,160	28,285	14,065	35,276	20,664	39,560
3º Turno	8,867	48,351	16,184	31,448	18,647	28,808	13,548	33,151	20,768	32,591

FONTE: O Autor (2018).

TABELA 4.8 – RESULTADO DO INTERVALO DE CONFIANÇA PARA A RAZÃO S/N 99%

S/N	Intervalo de confiança Z 99%									
	Produto A		Produto B		Produto C		Produto D		Produto E	
	LI	LS	LI	LS	LI	LS	LI	LS	LI	LS
1º Turno	-	-	10,884	26,856	20,614	32,738	7,322	46,181	0,652	44,067
2º Turno	-17,074	73,022	15,945	40,482	15,413	30,033	10,732	38,609	17,696	42,529
3º Turno	2,663	54,554	13,785	33,846	17,050	30,404	10,468	36,231	18,910	34,449

FONTE: O Autor (2018).

TABELA 4.9 – RESULTADO DO INTERVALO DE CONFIANÇA DA FUNÇÃO PERDA MÉDIA DE QUALIDADE 95%

$\bar{L}(y)$	Intervalo de confiança t 95%									
	Produto A		Produto B		Produto C		Produto D		Produto E	
	LI	LS	LI	LS	LI	LS	LI	LS	LI	LS
1º Turno	0,085	0,500	1,929	14,399	0,403	0,912	0,240	0,853	0,907	2,180
2º Turno	0	4,240	0	2,458	0,490	1,654	0,297	1,432	0,575	1,853
3º Turno	0	6,095	0	5,828	0,330	1,439	0,1587	2,166	0,219	1,343

FONTE: O Autor (2018).

TABELA 4.10 – RESULTADO DO INTERVALO DE CONFIANÇA DA FUNÇÃO PERDA DE QUALIDADE MÉDIA 99%

$\bar{L}(y)$	Intervalo de confiança t 99%									
	Produto A		Produto B		Produto C		Produto D		Produto E	
	LI	LS	LI	LS	LI	LS	LI	LS	LI	LS
1º Turno	0	0,618	0	16,784	0,313	1,002	0,107	0,986	0,638	2,449
2º Turno	0	5,662	0	3,313	0,284	1,861	0,039	1,690	0,296	2,132
3º Turno	0	8,715	0	7,903	0,134	1,635	0	2,623	0	1,581

FONTE: O Autor (2018).

TABELA 4.11 – RESULTADO DO INTERVALO DE CONFIANÇA C_p 95%

C_p	Intervalo de Confiança BCa 95%									
	Produto A		Produto B		Produto C		Produto D		Produto E	
	LI	LS	LI	LS	LI	LS	LI	LS	LI	LS
1º Turno	1,115	2,435	0,306	0,489	0,943	1,549	0,798	1,766	0,531	2,181
2º Turno	0,836	1,296	0,818	1,906	0,527	1,044	0,662	1,377	1,544	2,642
3º Turno	0,793	1,833	0,521	0,942	0,648	1,155	0,475	1,223	1,055	1,854

FONTE: O Autor (2018).

TABELA 4.12 – RESULTADO DO INTERVALO DE CONFIANÇA C_{pk} 95%

C_{pk}	Intervalo de confiança BCa 95%									
	Produto A		Produto B		Produto C		Produto D		Produto E	
	LI	LS	LI	LS	LI	LS	LI	LS	LI	LS
1º Turno	1,115	2,797	0	0,347	0,855	1,413	0,802	1,716	0,402	1,193
2º Turno	0,335	1,341	0,853	2,219	0,523	0,938	0,625	2,115	0,944	1,818
3º Turno	0,166	1,166	0,241	1,685	0,597	1,325	0,43	1,222	0,791	1,883

FONTE: O Autor (2018).

TABELA 4.13 – RESULTADO DO INTERVALO DE CONFIANÇA C_{pm} 95%

C_{pm}	Intervalo de confiança BCa 95%									
	Produto A		Produto B		Produto C		Produto D		Produto E	
	LI	LS	LI	LS	LI	LS	LI	LS	LI	LS
1º Turno	1,093	2,226	0,249	0,322	0,815	1,121	0,781	1,602	0,468	0,888
2º Turno	0,527	0,601	0,815	1,869	0,517	0,972	0,658	1,342	0,701	0,75
3º Turno	0,437	0,503	0,472	0,735	0,627	1,049	0,469	1,132	0,786	0,994

FONTE: O Autor (2018).

TABELA 4.14 – RESULTADO DO INTERVALO DE CONFIANÇA C_p 99%

C_p	Intervalo de confiança BCa 99%									
	Produto A		Produto B		Produto C		Produto D		Produto E	
	LI	LS	LI	LS	LI	LS	LI	LS	LI	LS
1º Turno	1,115	3,550	0,297	0,575	0,932	1,634	0,788	2,025	0,508	2,358
2º Turno	0,836	1,49	0,802	2,091	0,537	1,108	0,632	2,385	1,552	2,831
3º Turno	0,793	-	0,506	1,064	0,586	1,226	0,457	1,532	1,037	2,014

FONTE: O Autor (2018).

TABELA 4.15 – RESULTADO DO INTERVALO DE CONFIANÇA C_{pk} 99%

C_{pk}	Intervalo de confiança BCa 99%									
	Produto A		Produto B		Produto C		Produto D		Produto E	
	LI	LS	LI	LS	LI	LS	LI	LS	LI	LS
1º Turno	1,075	3,946	0	0,617	0,822	1,757	0,769	2,815	0,378	1,312
2º Turno	0,335	2,069	0,775	3,481	0,488	1,022	0,579	2,871	0,895	2,322
3º Turno	0,166	-	0,191	2,202	0,562	1,786	0,380	1,466	0,756	2,739

FONTE: O Autor (2018).

TABELA 4.16 – RESULTADO DO INTERVALO DE CONFIANÇA C_{pm} 99%

C_{pm}	Intervalo de confiança BCa 99%									
	Produto A		Produto B		Produto C		Produto D		Produto E	
	LI	LS	LI	LS	LI	LS	LI	LS	LI	LS
1º Turno	1,093	2,982	0,244	0,343	0,809	1,152	0,772	1,788	0,450	0,899
2º Turno	0,527	0,611	0,799	2,043	0,526	1,022	0,628	2,216	0,703	0,760
3º Turno	0,437	0,503	0,461	0,807	0,571	1,101	0,452	1,369	0,778	1,017

FONTE: O Autor (2018).

4.1.1 Discussão dos Resultados

A análise da Razão S/N revelou que o turno 1 teve três dos melhores resultados (Produto A, C e D), turno 2 apresentou dois melhores resultados (Produto B e E) e o turno 3 não apresentou nenhum melhor resultado. Logo, o turno 3 está com problemas considerando o critério da Razão S/N.

De forma semelhante ocorre quando se analisa a Função Perda Média de Qualidade. Turno 1 teve três melhores resultados (Produto A, C e D), turno 2 apresentou um dos melhores resultados (Produto B) e o turno 3 apresentou um melhor resultado (Produto E).

Quando são analisados o C_p e o C_{pk} os dois índices de capacidade de processo concordam que turno 1 teve três melhores resultados (Produto A, C e D), turno 2 apresentou dois melhores resultados (Produto B e E) e o turno 3 não apresentou nenhum melhor resultado. Porém é importante salientar que os resultados não foram satisfatórios. Em quase todos os turnos a produção apresenta índices baixíssimos, classificados como incapazes na produção. Todos os processos estão descentralizados.

A análise do índice C_{pm} concorda com a Função Perda Média de Qualidade. Turno 1 teve três melhores resultados (Produto A, C e D), turno 2 apresentou um dos melhores resultados (Produto B) e o turno 3 apresentou um melhor resultado (Produto E). Mas novamente os processos apresentam má qualidade.

Os intervalos de confiança gerados estão de acordo com os parâmetros calculados.

Não foi possível realizar nenhum outro tipo de análise, pois faltam informações sobre o processo. Como quais fatores e seus níveis são utilizados para essa produção.

Entretanto, com a análise da Razão S/N é possível ter ideia de qual processo está mais estável (menor variabilidade). Assim como qual processo está gerando menos perdas, qual processo está mais próximo do seu valor de projeto. E finalmente, quais turnos e produtos estão sendo produzidos em processos mais capazes.

Apesar de ser altamente recomendável uma análise completa em todo o processo de produção, com os dados fornecidos é possível começar a investigação do porque a produção está com resultados tão baixos.

4.2COMPARAÇÃO DO APLICATIVO EXPERIMENTO TAGUCHI

4.1.2 Resultado do Problema do Tempo de Mudança de Estado.

Os resultados comparativos da Razão S/N, ANOM e ANOM-S/N e análise gráfica dos *softwares* são apresentados abaixo.

TABELA 4.17 – COMPARAÇÃO DOS RESULTADOS DA RAZÃO S/N

Corrida	Peace	Experimento Taguchi	MiniTab 17	R
1	-16,93	-16,934	-16,9345	-16,9344
2	-15,59	-15,589	-15,5886	-15,5885
3	-14,36	-14,358	-14,3580	-14,3580
4	-15,72	-15,727	-15,7266	-15,7265
5	-9,24	-9,2407	-9,2407	-9,2406
6	-14,88	-14,885	-14,8855	-14,8854
7	-13,27	-13,279	-13,2787	-13,2786
8	-14,15	-14,152	-14,1525	-14,1524

FONTE: O Autor (2018).

TABELA 4.18 – COMPARAÇÃO DA ANOM S/N

(Continua)

		A	**B**	**AxB**	**C**	**AxC**	**BxC**	**D**
Peace	Nível 1	-15,66	-14,16	-14,99	-13,45	-15,08	-14,01	-15,20
	Nível 2	-12,89	-14,38	-13,55	-15,09	-13,46	-14,53	-13,33
	Delta	2,76	0,22	1,44	1,64	1,62	0,52	1,87
	Rank	-	-	-	-	-	-	-

		A	**B**	**AxB**	**C**	**AxC**	**BxC**	**D**
Experimento Taguchi	Nível 1	-15,651	-14,162	-14,988	-13,453	-15,082	-14,013	-15,206
	Nível 2	-12,889	-14,378	-13,552	-15,088	-13,458	-14,527	-13,334
	Delta	2,7626	0,2166	1,4359	1,6353	1,624	0,5141	1,8714
	Rank	1	7	5	3	4	6	2

		A	**B**	**AxB**	**C**	**AxC**	**BxC**	**D**
MiniTab 17	Nível 1	-15,65	-14,16	-	-13,45	-	-	-15,21
	Nível 2	-12,89	-14,38	-	-15,09	-	-	-13,33
	Delta	2,76	0,22	-	1,64	-	-	1,87
	Rank	1	4		3			2

(Conclusão)

		A	B	AxB	C	AxC	BxC	D
R	Nível 1	-15,6519	-14,1622	-	-13,4529	-	-	-15,2063
	Nível 2	-12,8893	-14,3789	-	-15,0882	-	-	-13,3349
	Delta	2,7625	0,2166	-	1,6353	-	-	1,8713
	Rank	-	-	-	-	-	-	-

FONTE: O Autor (2018).

TABELA 4.19 – COMPARAÇÃO DA ANOM

		A	B	AxB	C	AxC	BxC	D
Peace	Nível 1	-	-	-	-	-	-	-
	Nível 2	-	-	-	-	-	-	-
	Delta	-	-	-	-	-	-	-
	Rank	-	-	-	-	-	-	-

		A	B	AxB	C	AxC	BxC	D
Experimento Taguchi	Nível 1	5,2403	4,7125	4,8878	4,3575	4,8909	4,6025	5,1272
	Nível 2	4,0059	4,5338	4,3584	4,8887	4,3553	4,6438	4,1191
	Delta	1,2344	0,1787	0,5294	0,5312	0,5356	0,0413	1,0081
	Rank	1	6	5	4	3	7	2

		A	B	AxB	C	AxC	BxC	D
MiniTab 17	Nível 1	5,24	4,713	-	4,357	-	-	5,127
	Nível 2	4,006	4,534	-	4,889	-	-	4,119
	Delta	1,234	0,179	-	0,531	-	-	1,008
	Rank	1	4	-	3	-	-	2

		A	B	AxB	C	AxC	BxC	D
R	Nível 1	5,2403	4,7125	-	4,3575	-	-	5,1271
	Nível 2	4,0059	4,5337	-	4,8887	-	-	4,1190
	Delta	1,2343	0,1787	-	0,5312	-	-	1,0081
	Rank	-	-	-	-	-	-	-

FONTE: O Autor (2018).

TABELA 4.20 – COMPARAÇÃO DA ANOM PARA FATORES DE RUÍDO

		M	N	O
Peace	Nível 1	2.425	4,590	4,670
	Nível 2	6,830	4,665	4,585
	Delta	4,405	0,075	0,085
	Rank	-	-	-

		M	N	O
Experimento Taguchi	Nível 1	2,4206	4,5853	4,6653
	Nível 2	6,8256	4,6609	4,5809
	Delta	4,4050	0,0756	0,0844
	Rank	1	3	2

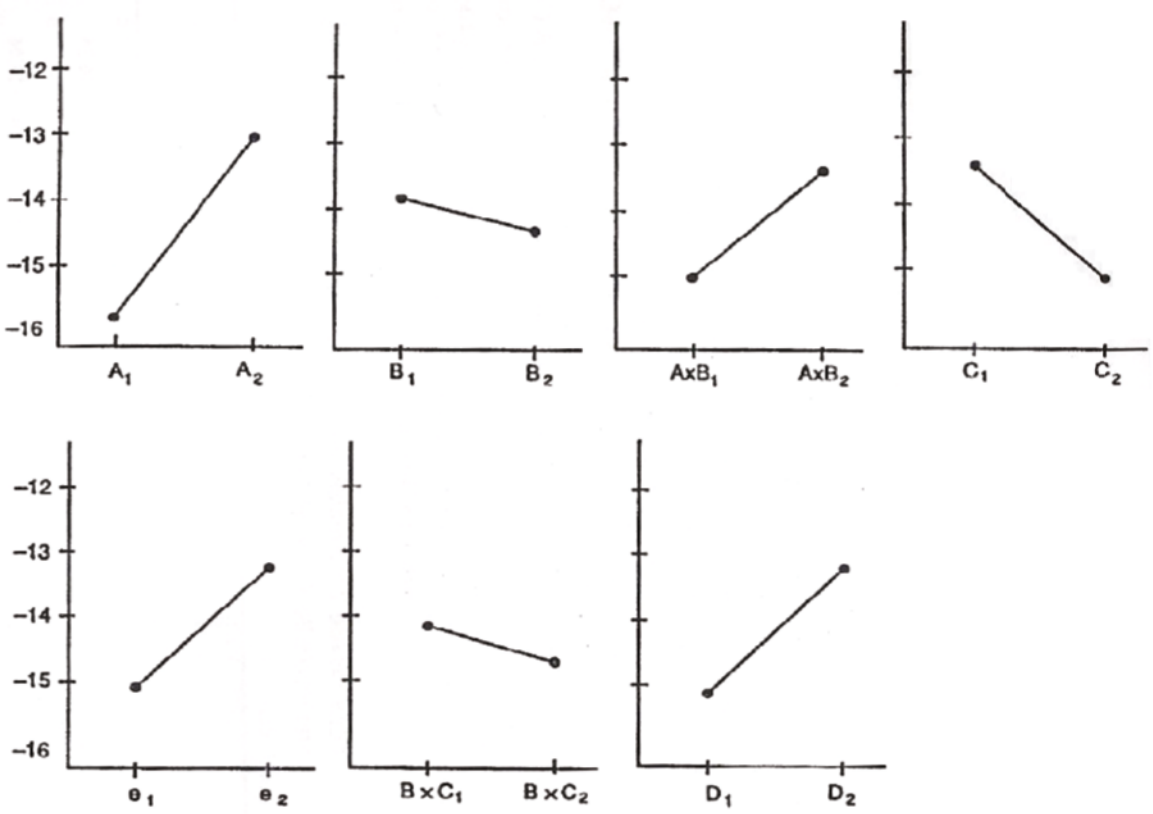
		M	N	O
MiniTab 17	Nível 1	2,421	4,585	4,665
	Nível 2	6,826	4,661	4,581
	Delta	4,405	0,076	0,084
	Rank	1	3	2

		M	N	O
R	Nível 1	2,420625	4,585312	4,665312
	Nível 2	6,825625	4,660938	4,580938
	Delta	4,405	0,0756	0,0843
	Rank	-	-	-

FONTE: O Autor (2018).

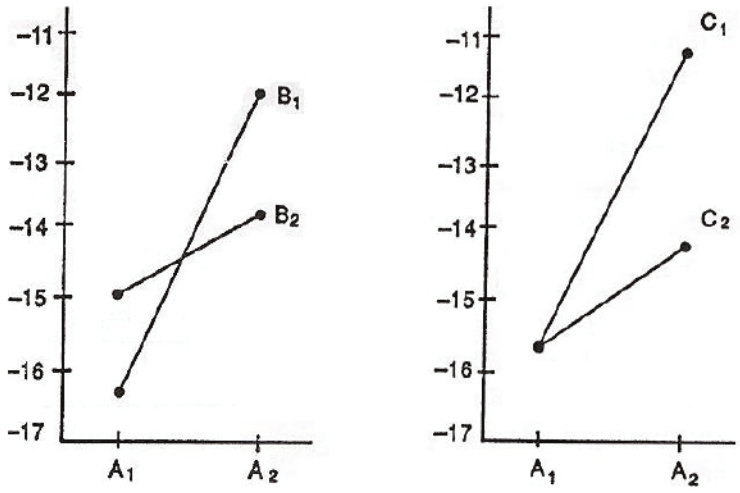
E a análise gráfica dos programas é dada a seguir.

GRÁFICO 4.1 – EFEITOS PRINCIPAIS DA RAZÃO S/N – PEACE



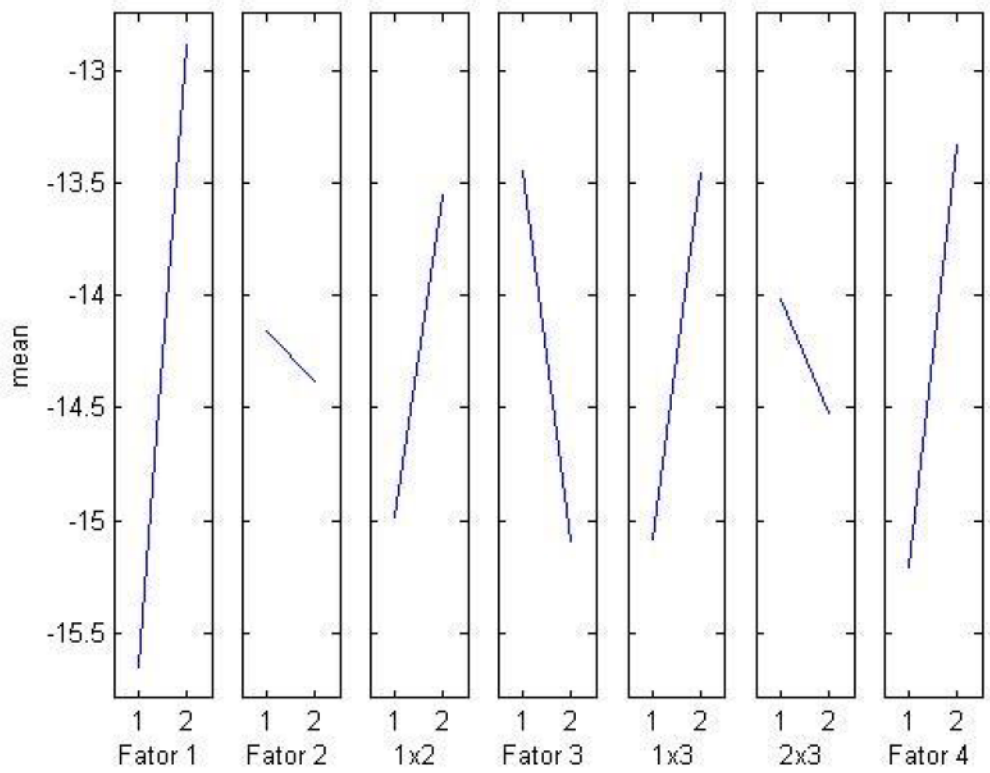
FONTE: Peace (1993).

GRÁFICO 4.2 – INTERAÇÕES ENTRE FATORES DA RAZÃO S/N – PEACE



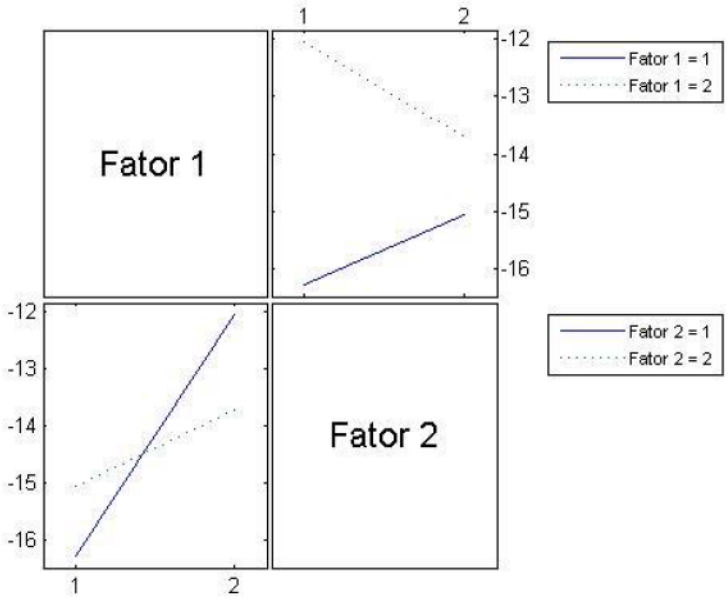
FONTE: Peace (1993).

GRÁFICO 4.3 – EFEITOS PRINCIPAIS RAZÃO S/N – EXPERIMENTO TAGUCHI



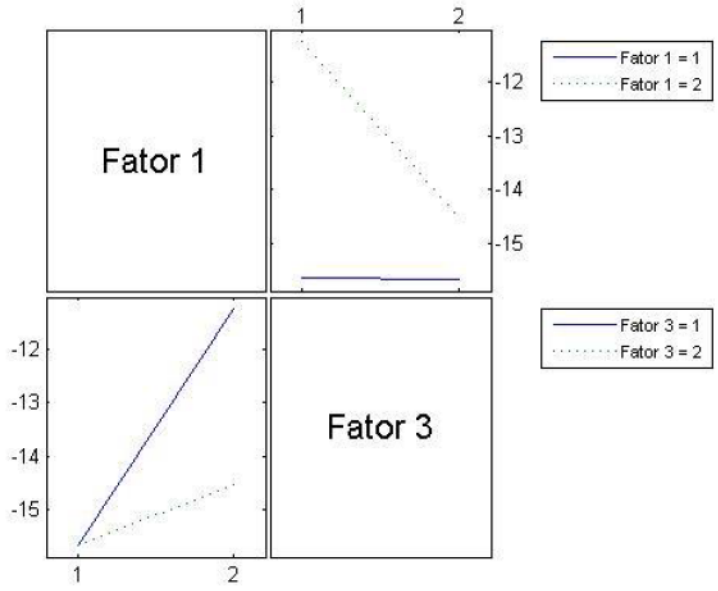
FONTE: O Autor (2018).

GRÁFICO 4.4 – INTERAÇÃO RAZÃO S/N FATOR 1 x FATOR 2 – EXPERIMENTO TAGUCHI



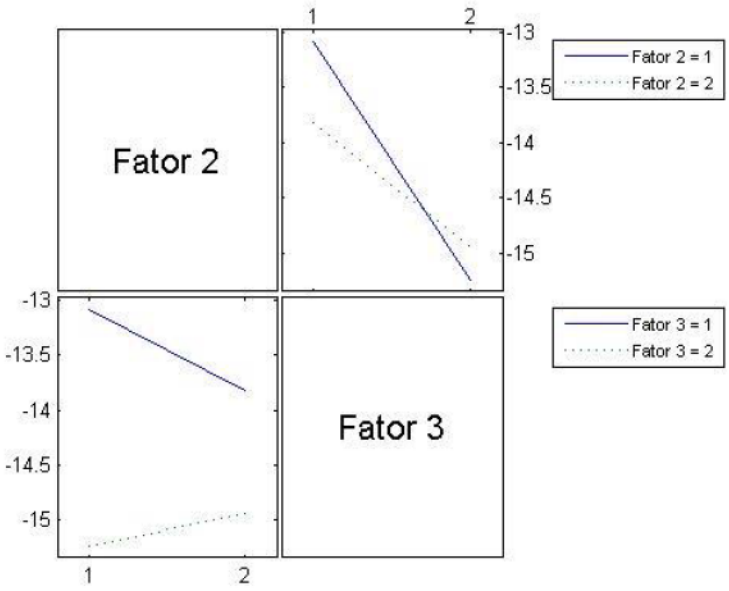
FONTE: O Autor (2018).

GRÁFICO 4.5 – INTERAÇÃO RAZÃO S/N FATOR 1 x FATOR 3 – EXPERIMENTO TAGUCHI



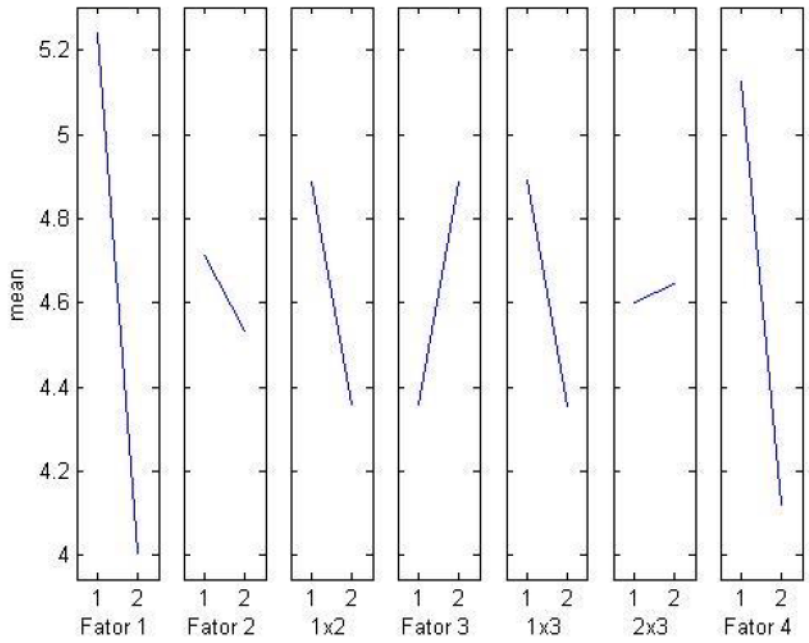
FONTE: O Autor (2018).

GRÁFICO 4.6 – INTERAÇÃO RAZÃO S/N FATOR 2 x FATOR 3 – EXPERIMENTO TAGUCHI



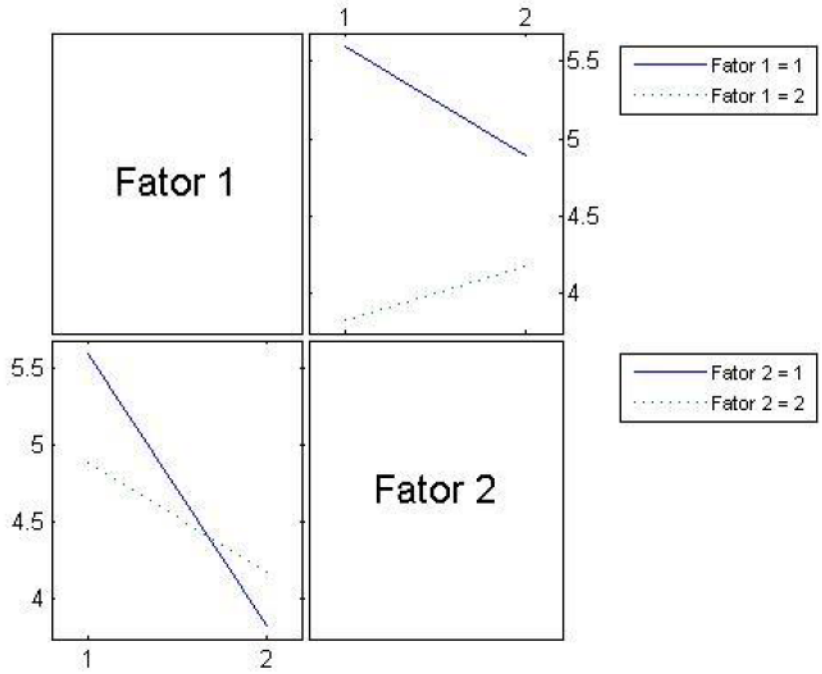
FONTE: O Autor (2018).

GRÁFICO 4.7 – ANÁLISE DAS MÉDIAS – EXPERIMENTO TAGUCHI



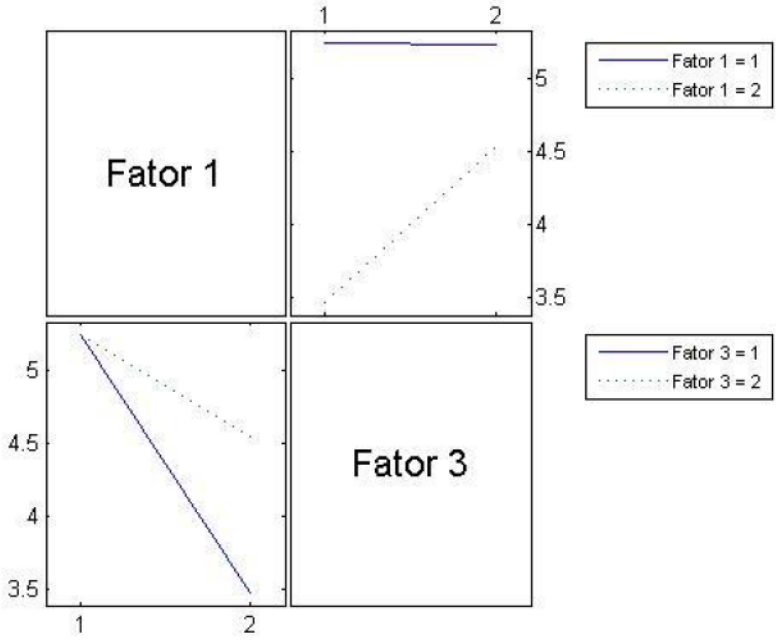
FONTE: O Autor (2018).

GRÁFICO 4.8 – INTERAÇÃO FATOR 1 x FATOR 2 – EXPERIMENTO TAGUCHI



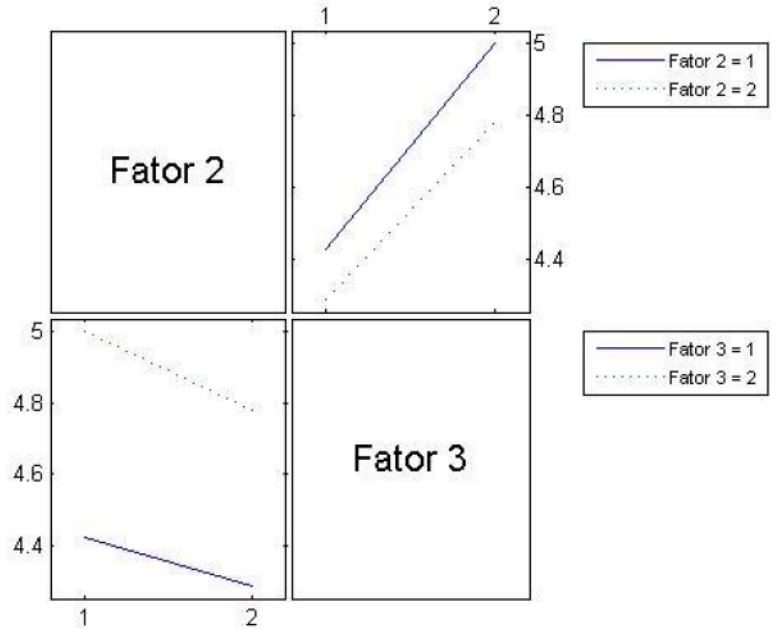
FONTE: O Autor (2018).

GRÁFICO 4.9 – INTERAÇÃO FATOR 1 x FATOR 3 – EXPERIMENTO TAGUCHI



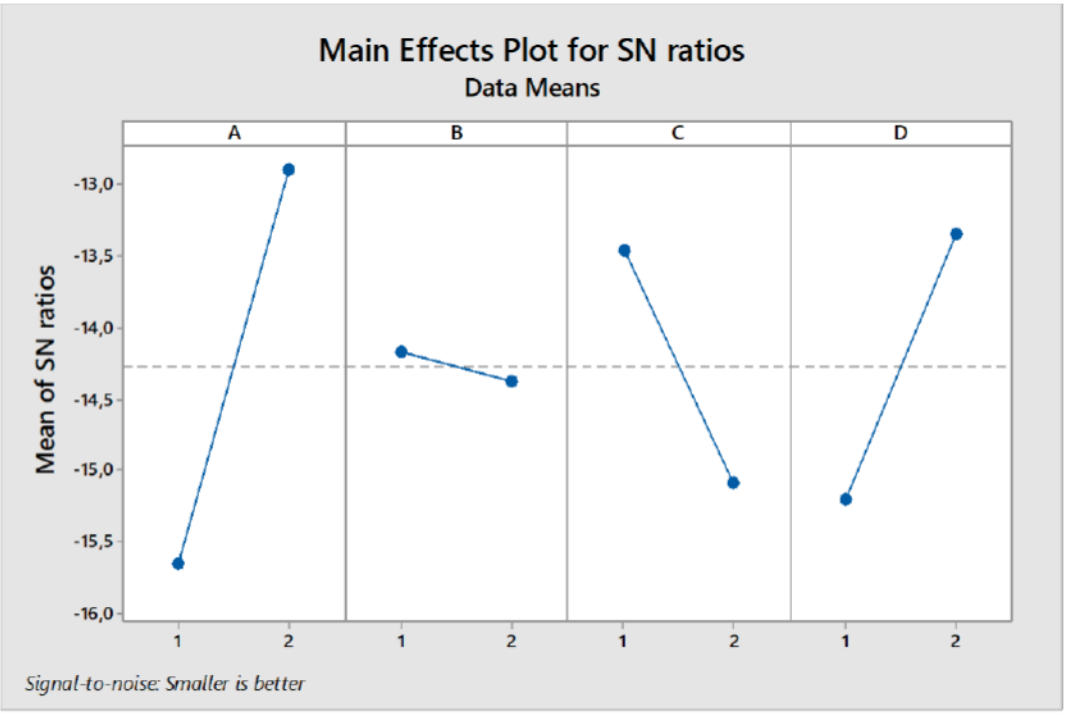
FONTE: O Autor (2018).

GRÁFICO 4.10 – INTERAÇÃO FATOR 2 x FATOR 3 – EXPERIMENTO TAGUCHI



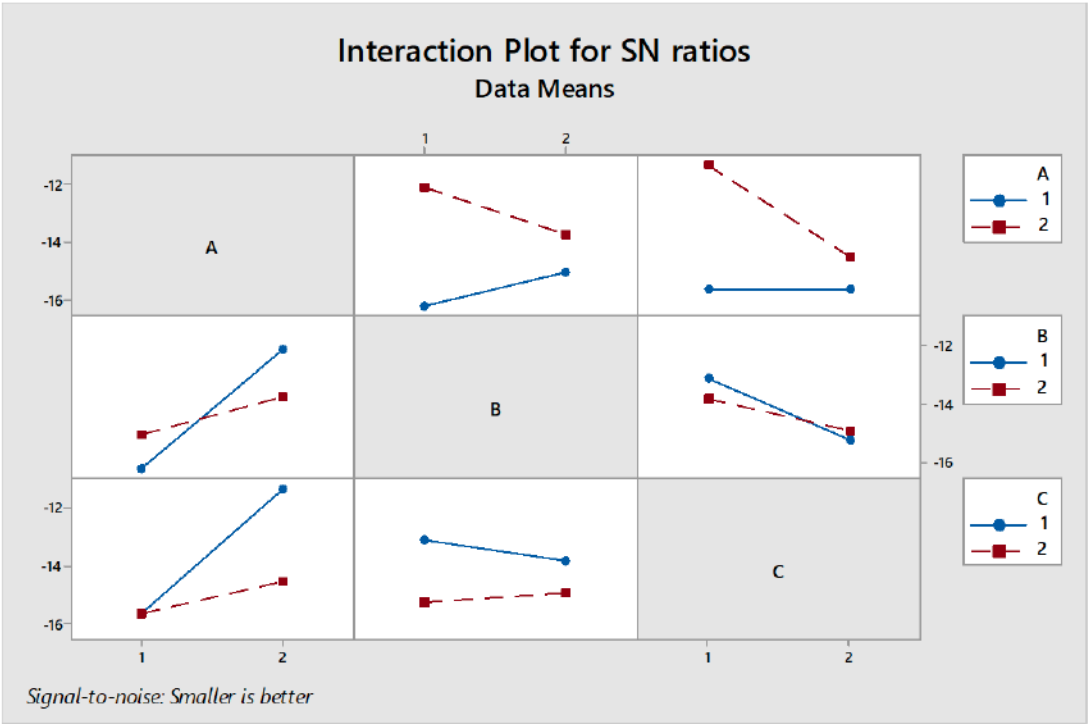
FONTE: O Autor (2018).

GRÁFICO 4.11 – EFEITOS PRINCIPAIS RAZÃO S/N – MINITAB 17



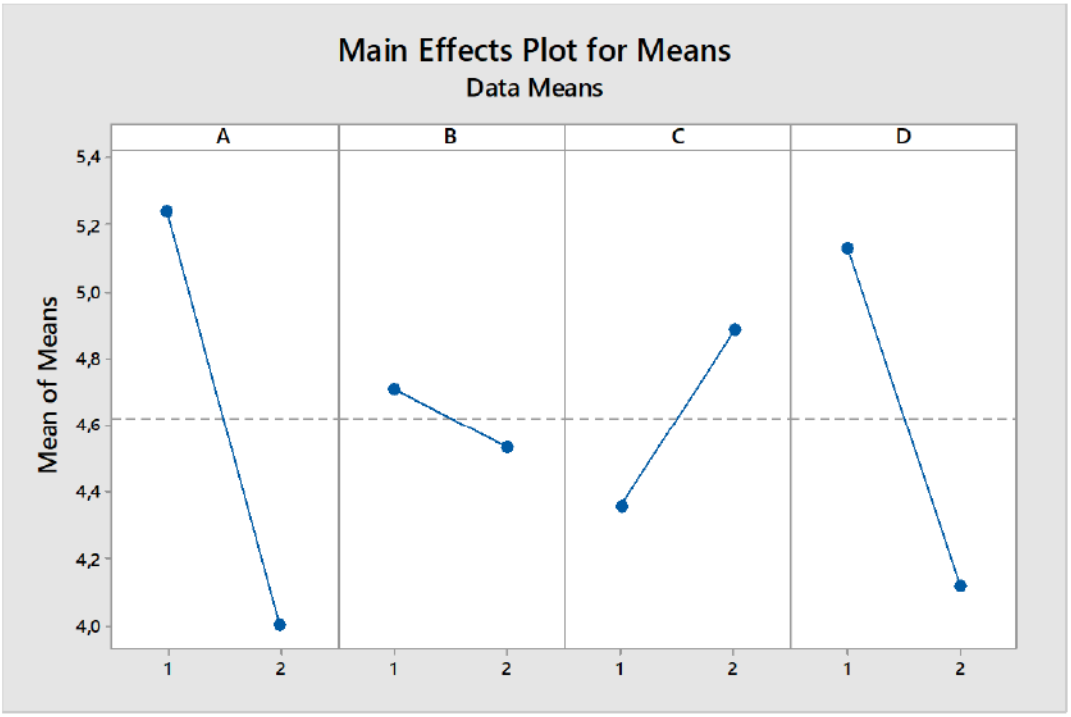
FONTE: O Autor (2018).

GRÁFICO 4.12 – INTERAÇÕES RAZÃO S/N – MINITAB 17



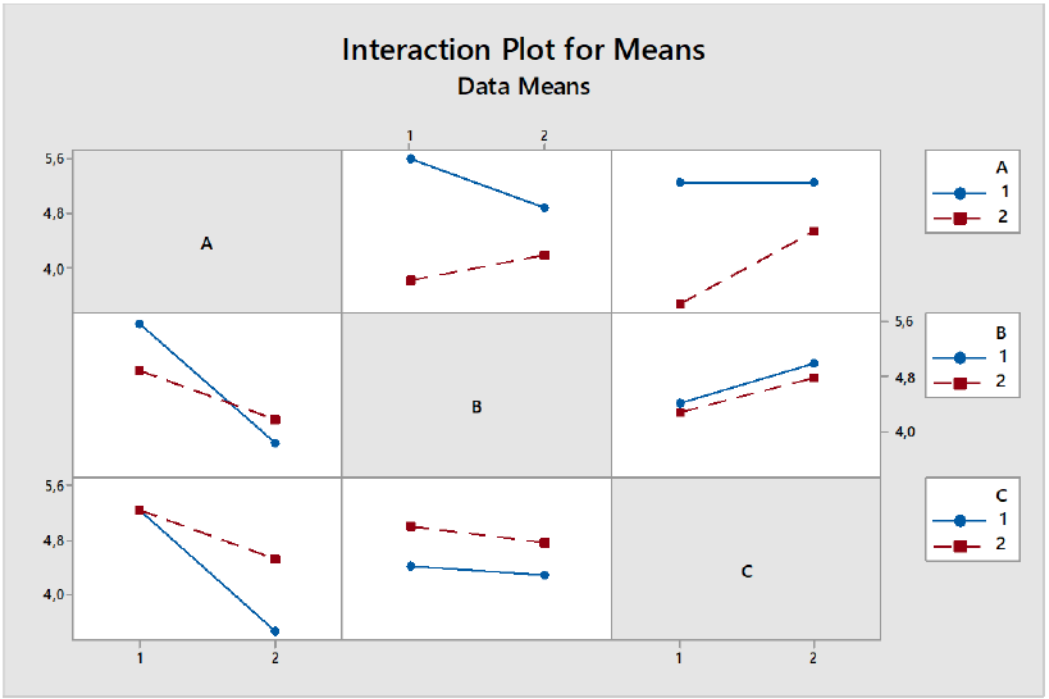
FONTE: O Autor (2018).

GRÁFICO 4.13 – EFEITOS PRINCIPAIS – MINITAB 17



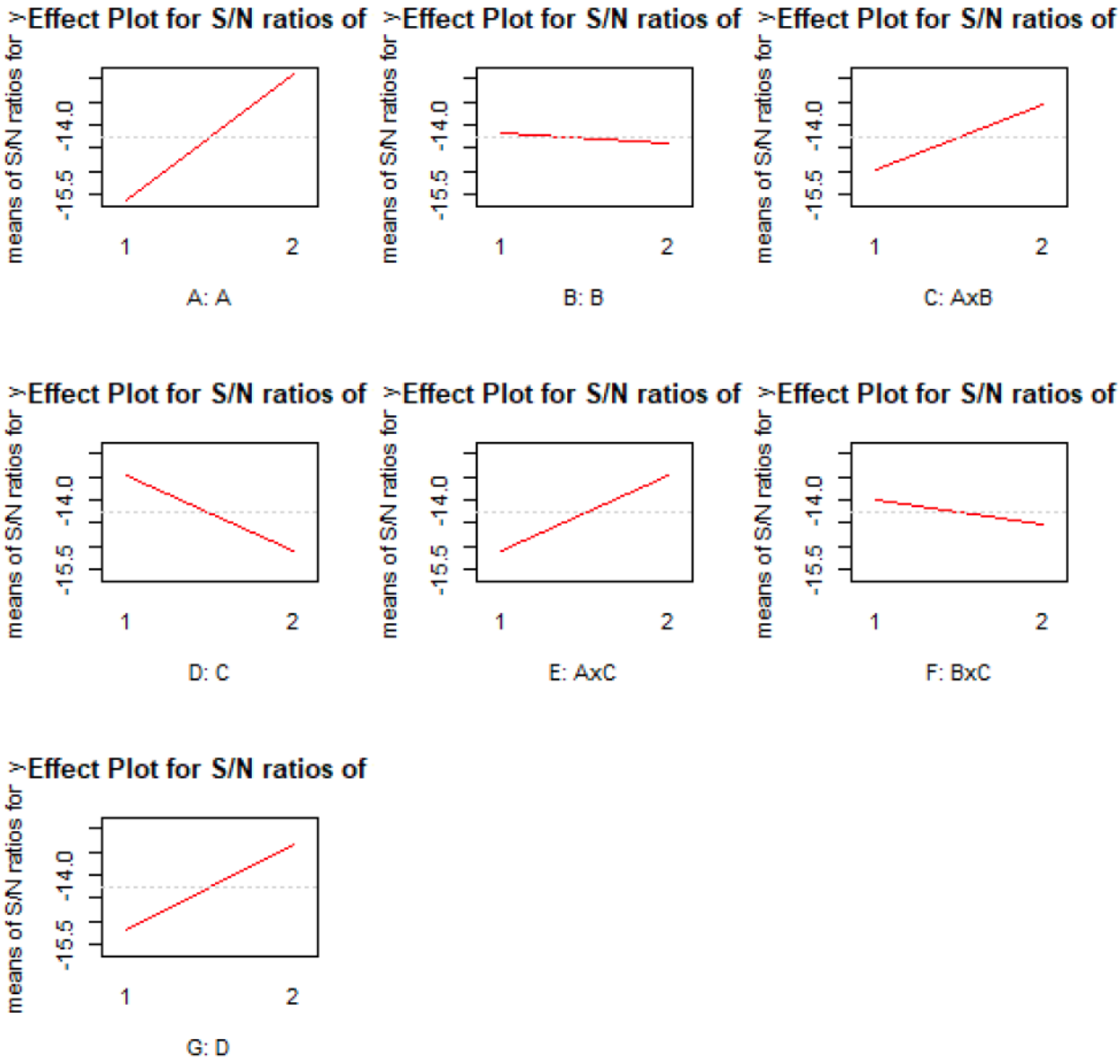
FONTE: O Autor (2018).

GRÁFICO 4.14 – INTERAÇÕES DAS MÉDIAS – MINITAB 17



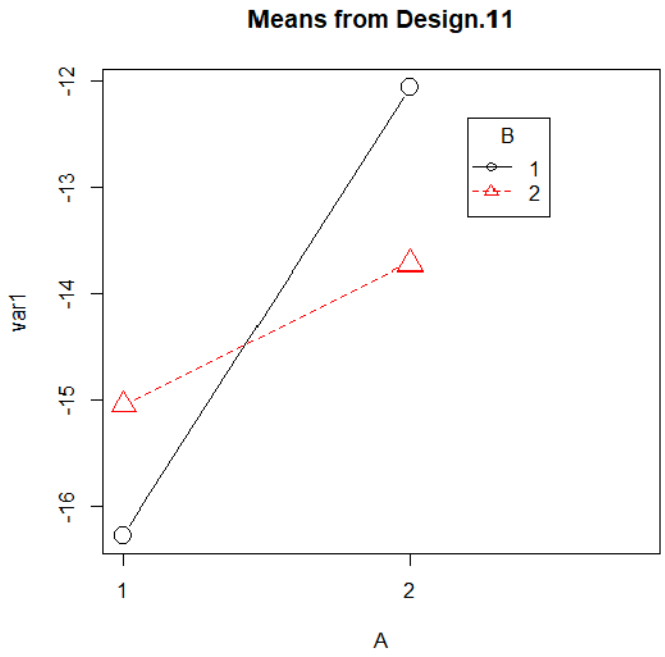
FONTE: O Autor (2018).

GRÁFICO 4.15 – EFEITOS PRINCIPAIS RAZÃO S/N – R



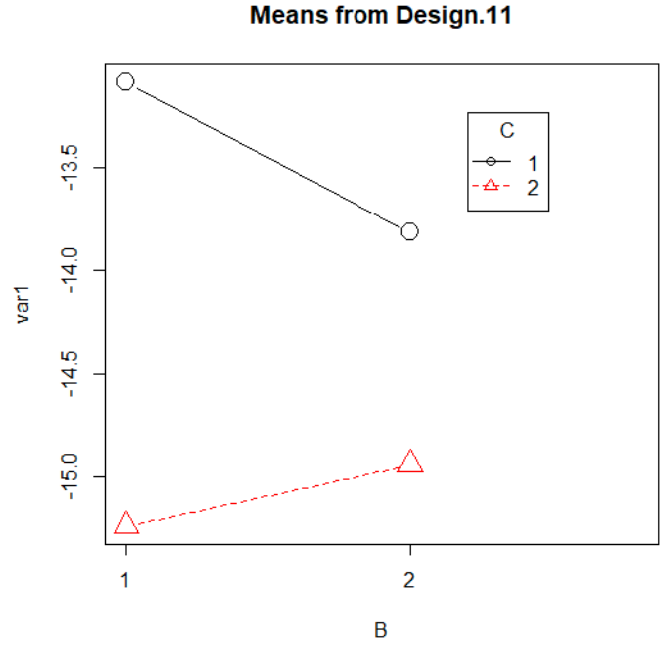
FONTE: O Autor (2018).

GRÁFICO 4.16 – INTERAÇÃO RAZÃO S/N FATOR A x FATOR B – R



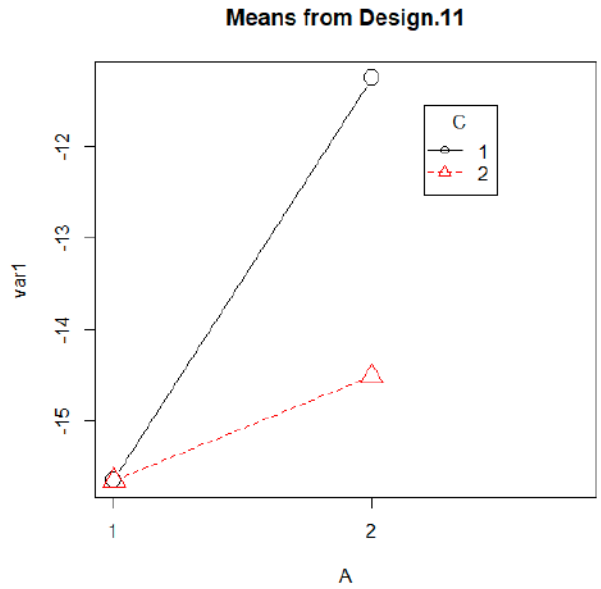
FONTE: O Autor (2018).

GRÁFICO 4.17 – INTERAÇÃO RAZÃO S/N FATOR B x FATOR C – R



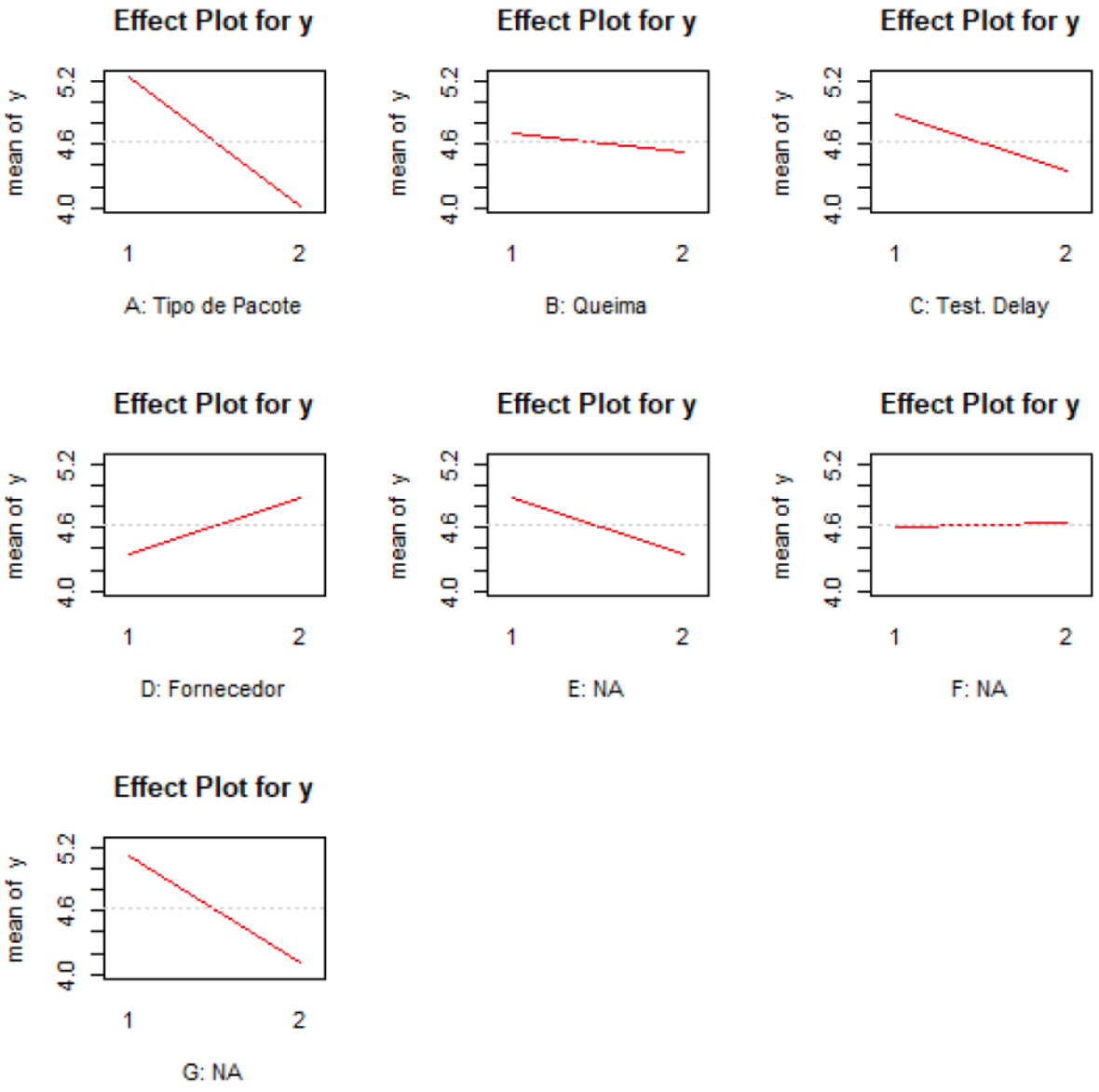
FONTE: O Autor (2018).

GRÁFICO 4.18 – INTERAÇÃO RAZÃO S/N FATOR A x FATOR C – R



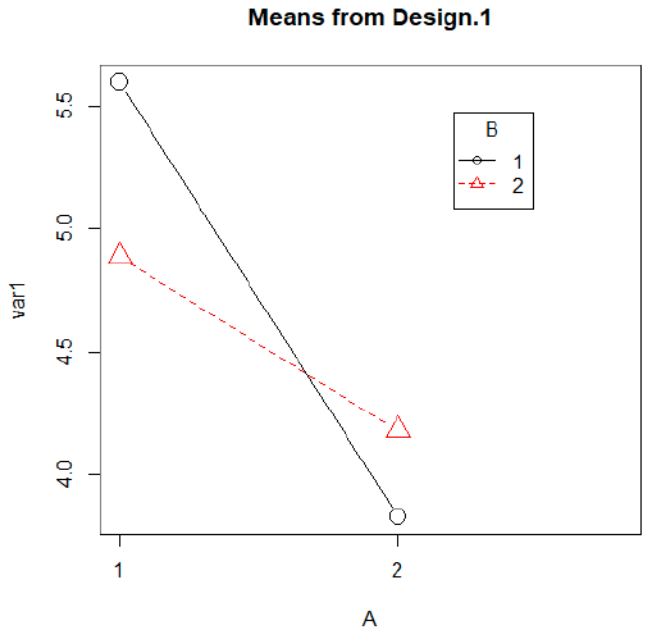
FONTE: O Autor (2018).

GRÁFICO 4.19 – EFEITOS PRINCIPAIS – R



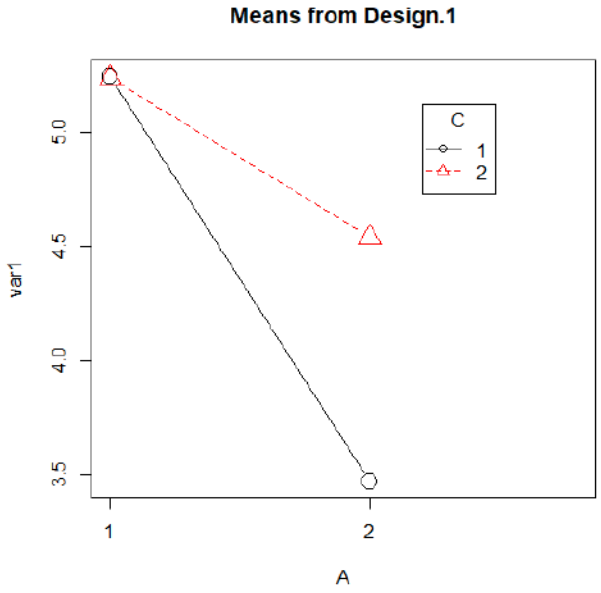
FONTE: O Autor (2018).

GRÁFICO 4.20 – INTERAÇÃO FATOR A x FATOR B – R



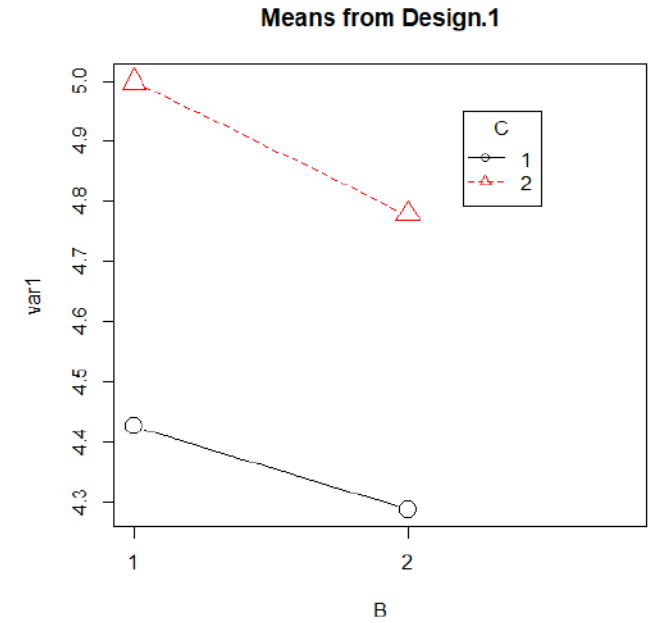
FONTE: O Autor (2018).

GRÁFICO 4.21 – INTERAÇÃO FATOR A x FATOR C – R



FONTE: O Autor (2018).

GRÁFICO 4.22 – INTERAÇÃO FATOR B x FATOR C – R



FONTE: O Autor (2018).

4.1.3 Resultado Final

O aplicativo Experimento de Taguchi apresenta resultados excelentes na análise de dados. Os resultados numéricos obtidos diferem apenas na segunda casa decimal, ou ordem maior, em comparação com os outros *softwares* utilizados e com a literatura. O aplicativo consegue identificar e classificar os fatores mais relevantes para o experimento assim como os outros programas.

Com comportamento da análise numérica, a análise gráfica obteve resultados quase indistinguíveis dos outros *softwares* utilizados na comparação.

4.1.4 Consideração sobre os Softwares utilizados.

Software Minitab 17: o programa permite a atribuição das interações entre fatores em dois momentos distintos da construção do Projeto de Experimento, no início quando se está escolhendo a matriz experimental e no final quando se pede a análise de dados. Caso a atribuição da interação seja feita no início, junto com a escolha da matriz experimental, as atribuições dos fatores nas colunas ficarão em desacordo com as matrizes de Taguchi. Caso a atribuição da interação seja feita quando é requisitada a análise de dados, a atribuição é correta.

Esse tipo de problema é contornável, remete à funcionalidade do programa, porém pode levar a resultados errados para aqueles que não possuem um conhecimento mais aprofundado das matrizes ortogonais de Taguchi.

Software R: para a realização do experimento requer conhecimento de programação na linguagem do R, conhecimento dos pacotes necessários para realizar a análise assim como um pouco de familiaridade com suas funções respectivas.

Software Experimento Taguchi: possui uma interface amigável como o Minitab 17, porém é mais intuitivo. Não causa problemas quando efetuada a análise, isso é, não leva a erros desnecessários por parte do usuário por não conhecer o programa ou a matrizes de Taguchi.

Por poder calcular a Função Perda de Qualidade quadrática, índice de Capacidade Potencial do Processo e intervalos de confiança na mesma análise, mostrou-se uma ferramenta mais completa para a análise de Taguchi em comparação com os outros *softwares* utilizados. Porém, não apresenta outros recursos adicionais para outros tipos de análises complementares que os *softwares* R e Minitab apresentam.

5 CONCLUSÃO E CONSIDERAÇÕES FINAIS

Os objetivos deste trabalho são: desenvolver um estudo sobre as técnicas criadas por Genichi Taguchi que são contribuições para a Engenharia da Qualidade, utilizar a técnica *Bootstrap* para geração de intervalos de confiança dos parâmetros do Taguchi, desenvolver um aplicativo para a aplicação de seus conceitos, compará-lo com outros *softwares* que já existem e realizar o estudo de caso real.

Pode-se dizer que os objetivos foram alcançados com sucesso. Os estudos relacionados com essa dissertação até o presente momento possibilitaram a apresentação de três artigos de congressos, são eles:

1. Simpósio de Métodos Numéricos em Engenharia (SMNE) - com o artigo "Aplicativo computacional para delineamento de experimento de Taguchi";
2. XXXVIII Ibero-Latin American Congress on Computational Methods in Engineering (CILAMCE) – com o artigo "Taguchi's quality measures (loss function and signal to noise ratio) using na application with a graphical interface";
3. 24 th ABCM International Congress of Mechanical Engineering COBEM – com o artigo "Application of Taguchi's loss function and signal to noise ratio".

Quanto ao aplicativo desenvolvido, ele conseguiu entregar resultados excelentes comparáveis a outros *softwares* que já existem no mercado, as respostas do programa são quase indistinguíveis quando comparados aos outros programas. Assim, foi capaz de gerar todos os parâmetros e seus respectivos intervalos de confiança *Bootstrap*. Gerou os gráficos de efeitos principais para os fatores e interação entre fatores de maneira precisa. Também utilizou as matrizes ortogonais de forma correta.

O programa Experimento de Taguchi fornece uma maneira simples de efetuar todos os cálculos que o Projeto de Parâmetros e Projeto de Tolerância exigem. A interface desenvolvida é eficiente e não induz a erros por parte do usuário.

No estudo de caso real foi possível identificar os melhores e piores turnos de trabalho, assim como qual produto é melhor produzido, mesmo não podendo aplicar todos os recursos disponíveis no app desenvolvido. Os parâmetros foram calculados

e os intervalos de confiança foram gerados com sucesso. Efetuando todas as possíveis análises com os dados existentes.

Como contribuição deste trabalho é possível apontar o aplicativo desenvolvido para usuário, possibilitando a utilização da metodologia de Taguchi. Assim como, o acréscimo das matrizes ortogonais, para planejamento de experimentos, na linguagem *MATLAB*.

5.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Aprimorar o aplicativo Experimento de Taguchi, tornando-o uma ferramenta mais completa para análise de dados. Como, por exemplo, adicionar opções para que o usuário possa efetuar gráfico de Pareto e histograma, teste de assimetria para as distribuições, testes de normalidade para amostras e outras ferramentas comuns no controle da qualidade. Adicionar o resto das funções de Taguchi. Assim como efetuar a otimização do código do programa.

Efetuar outro estudo de caso que permita utilizar toda a metodologia disponível no app desenvolvido.

REFERÊNCIAS

ABREU, A. L. E. **Intervalo de confiança bootstrap para valores da função de confiabilidade estimados pelo método de Kaplan-Meier**. 2011. 108p. Dissertação (Mestrado em) – Universidade Federal do Paraná – Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Curitiba, 2011.

ALMEIDA, H. S.; TOLEDO, J. C. **Qualidade total do produto**, São Paulo, v. 2, n. 1, p. 21-37, jun. 1992. Disponível em: <http://www.scielo.br>. Acesso em: 16 jun. 2017.

ASQ. **Genishi Taguchi, Innovative methods and techniques**. Disponível em: <https://asq.org/about-asq/honorary-members/taguchi/>. Acesso em: 10 de nov., 2017.

ATHREYA, S.; VENKATESH, Y. D. Application of Taguchi method for optimization of process parameters in improving the surface roughness of lathe facing operation, **International Refereed Journal of Engineering and Science (IRJES),** v. 1, n. 3, p.13-19, 2012. Disponível em: < http://irjes.com/Papers/vol1-issue3/Version%201/C131319.pdf>. Acesso em: 16 jun. 2017.

BAYEUX, C. Análise da capacidade de processo (parte I), **Banas Qualidade**. p. 57-60, jun. 2001. Disponível em: <https://www.qsp.org.br/biblioteca/pdf/analise.pdf>. Acesso em: 16 jun. 2017.

CHAVES NETO, A. **Bootstrap em séries temporais**. 1991. 225p. Tese (Doutorado em) – PUC/Rio de Janeiro. – Programa de Pós-Graduação em Engenharia Elétrica, 1991.

CHAVES NETO, A. **Notas de aula: técnicas estatísticas aplicadas a engenharia da qualidade**, 2017.

CHERNICK, M. R.; LABUDDE, R. A. **An introduction to bootstrap methods with applications to R.** New Jersey: John Wiley & Sons, Hoboken, 2011.

COSTA, A. F. B.; EPPRECHT, E. K.; CARPINETTI, L. C. R. **Controle estatístico de qualidade.** 2. ed. São Paulo: Atlas, 334p., 2010.

CREVELING, C. M.; FOWLKES, W. Y. **Engineering Methods for Robust Product Design.** Massachusetts: Addison-Wesley Publishing Company, 1995.

DAVISON, A. C.; HINKLEY, D. V. **Bootstrap methods and their application**. Cambridge University Press, 1997.

DEMING, W. E. **Out of the crisis**. Cambridge University Press, University of Cambridge, 1982.

EFRON, B.; TIBSHIRANI, R. J. **An introduction to the bootstrap**, New York: Chapman & Hall, NY, 1993.

FEIGENBAUM, A. V. **Total quality control**, 3. ed., rev. International edition - McGraw-Hill international edition, industrial Engineering Series, 1991.

GARVIN, D. A. **Competing on the eight dimensions of quality**, Harvard Business Review 65, n. 6, nov. – Dec., 1987. Disponível em: <https://hbr.org/1987/11/competing-on-the-eight-dimensions-of-quality>. Acesso em: 23 jul. 2017.

GHOBADIAN, A. Service quality: concepts and models. **International Journal of Quality and Reliability Management**, v.11, n.9, p.43-66, 1994.

HASSAN, A.; BAKSH, M. S. N.; SHAHAROUN, A. M. Issues in quality engineering research, **International Journal of Quality & Reliability Management**, v. 17, n. 8, p. 858-875, 2000.

HESTERBERG, T.; MONAGHAN, S.; MOORE, D. S.; CLIPSON, A.; EPSTEIN, R. **Bootstrap methods and permutation tests,** Companion Chapter 18 To The Practice Of Business Statistics, New York: W. H. Freeman and Company, NY, 2003.

JURAN, J. M.; GODFREY, A. B.; **Juran's quality handbook**, 5. Ed., New York: McGraw-Hill, NY, 1998.

KACKER, R. N.; LAGERGREN, E. S.; FILLIBEN, J. J. Taguchi's orthogonal arrays are classical designs of experiments, **Journal of Research of the National Institute of Standards and Technology**, v. 96, n. 5, set. - oct., 1991. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4927234/>. Acesso em: 20 jul. 2017.

KARNA, S. K.; SAHAI, R. An overview in taguchi method, **International Journal of Engineering and Mathematical Sciences**, v. 1, p.11-18, jan. – jun., 2012. Disponível em: <https://www.researchgate.net/ >. Acesso em: 25 jul. 2017.

KRISHNAIAH, K.; SHAHABUDDEN, P. **Applied design of experiments and taguchi methods,** New Delhi: PHI Learning Private Limited, 2012.

LIRA, S. A. **Notas de aula: planejamento de experimentos**, 2017.

LOIBEL, S.; ALVES, E. J. **Comparação entre intervalos de confiança calculados com métodos bootstrap e intervalos assintóticos**, Anais do Congresso de Matemática Computacional – CMAC Sudeste, 2013. Disponível em: <http://www.sbmac.org.br/cmacs/cmac-se/2013/trabalhos/PDF/4338.pdf>. Acesso em: 20 jul. 2017.

MARQUES, M. A. M. **Estimação dos intervalos de confiança para a função perda de Taguchi, razão sinal ruído e dos índices de capacidade potencial do processo, por meio de método de reamostragem**. 2015. 180p. Tese (Doutorado em) – Universidade Federal do Paraná – Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Curitiba, 2015.

MONTGOMERY, D. C. **Introduction to statistical quality control**. 6. Ed. New York: John Wiley & Sons, Hoboken, 2009.

MORI, T. **Taguchi methods: benefits, impacts, mathematics, statistics, and applications,** New York: ASME, NY, 2011.

PALADINI, E. P. **Gestão da qualidade, teoria e prática**, 2. Ed. – 8. Reimp. – São Paulo: Atlas, 2010.

PEACE, G. S. **Taguchi methods: a hands-on approach**, addison-Wesley Publishing Company, 1993.

PEAR, W. L.; KOTZ, S. **Encyclopedia and handbook of process capability indices: a comprehensive exposition of quality control measures,** Singapore: World Scientific Publishing, 2006.

PHADKE, M. S. **Quality engineering using robust design**. New Jersey: Prentice-Hall, 1989.

ROY, R. K. **Design of experiments using the taguchi approach: 16 steps to product and process improvement.** New York: John Wiley & Sons, 2001.

SILVA, M. B.; Carneiro, L. M.; Silva, J. P. S.; Oliveira, I. S. O.; Filho, H. J. I.; Almeida, C. R. O. An application of the Taguchi method (robust design) to environmental engineering: evaluating advanced oxidative processes in polyester, resin wastewater treatment. **American Journal of Analytical Chemistry**, v. 5, n. 13, p. 828-837, 2014.

STEFFEN, D. **Intervalo "bootstrap" para previsões de séries temporais obtidas pelo método theta**. 2010. 116p. Tese (Mestrado em) – Universidade Federal do Paraná – Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Curitiba, 2010.

TAGUCHI, G.; CHOWDHURY, S.; WU, Y.**Taguchi's quality engineering handbook**, New Jersey: John Wiley & Sons, Hoboken, 2005.

TAGUCHI, G.; CHOWDHURY, S.; WU, Y. **Robust engineering**, New York: McGraw-Hill, NY, 2000.

TAGUCHI G.; ELSAYED, A. E.; THOMAS C. H. **Engenharia da qualidade e sistemas de produção.** Tradução Regina Cláudia Loverri; revisão técnica José Carlos de Castro Waeny. – São Paulo: McGraw-Hill, 1990.

UNAL, R.; DEAN, E. B. **Taguchi approach to design optimization for quality and cost: an overview**, Annual Conference of the International Society of Parametric Analysts, 1991. Disponível em: < http://citeseerx.ist.psu.edu/ >. Acesso em: 23 jul. 2017.

ZAMBANINI, R. A. **The application of Taguchi's method of parameter design to the design of mechanical systems**. 1992. 131p. Tese (Mestrado em) – University Lehigh – Applied Mathematics. Disponível em: < https://preserve.lehigh.edu/etd/131/>. Acesso em: 24 jul. 2017.

APÊNDICE A – PROGRAMA MATLAB

```
%--------------------------------
%Janela principal
%--------------------------------
function []=Experimento_Taguchi
%VARIAVEIS
% popup1 - Número de fatores
% popup2 - Nível dos fatores
% check5 - Calcular interaçoes
% ed1 - Dados amostrais
% ed5 - nº reamostras bootstrap
% ed6 - Valor nominal
% tbl3 - Matriz experimental
% tbl99 - Amostra
% tbl5 - ANOM
% tbl52 - ANOM S/N
% tbl8 - ANOVA
% tbl9 - ANOVA S/N
% tbl6 - Geral

global tbl3 tbl5 tbl6 tbl52 tbl8 tbl9 tbl99 color ed1 ed5 ed6 popup1
popup2 check5

s.fig(3)=figure('position',[200 50 900 600],...
    'menubar','none',...
    'name','Experimento Taguchi',...
    'numbertitle','off',...
    'resize','off');

color=get(s.fig(3),'Color');

s.txt1=uicontrol(s.fig(3),...
    'style','text',...
    'units','pixels',...
    'HandleVisibility','off',...
    'position',[65 550 100 30],...
    'string','Taguchi Design',...
    'BackgroundColor', color);

s.txt2=uicontrol(s.fig(3),...
    'style','text',...
    'units','pixels',...
    'HandleVisibility','off',...
    'position',[10 517 100 30],...
    'string','Fatores :',...
    'BackgroundColor', color);

popup1 = uicontrol(s.fig(3),...
    'Style', 'popup',...
    'String', {''},...
    'Position', [100 500 100 50],...
    'callback',@callback_popup1);

s.txt3=uicontrol(s.fig(3),...
    'style','text',...
    'units','pixels',...
```

```
    'HandleVisibility','off',...
    'position',[10 467 100 30],...
    'string','Níveis :',...
    'BackgroundColor', color);

popup2 = uicontrol(s.fig(3),...
    'Style', 'popup',...
    'String', {'','2','3','4','5','Mix'},...
    'Position', [100 450 100 50],...
    'Callback', @callback_popup2);

check5=uicontrol(s.fig(3),...
    'style','checkbox',...
    'position',[80 430 150 40],...
    'string','Calcular interações',...
    'BackgroundColor', color,...
    'callback',@callback_interacao);

s.pb1=uicontrol('style','pushbutton',...
    'units','pixels',...
    'HandleVisibility','off',...
    'position',[80 380 80 40],...
    'string','Design',...
    'callback',{@callback_design,s});

s.pb2=uicontrol('style','pushbutton',...
    'units','pixels',...
    'HandleVisibility','off',...
    'position',[400 380 80 40],...
    'string','Gráficos',...
    'callback',{@callback_graficos,s});

s.pb3=uicontrol('style','pushbutton',...
    'units','pixels',...
    'HandleVisibility','off',...
    'position',[500 380 80 40],...
    'string','Opções',...
    'callback',{@callback_op,s});

s.pb4=uicontrol('style','pushbutton',...
    'units','pixels',...
    'HandleVisibility','off',...
    'position',[600 380 80 40],...
    'string','Exportar',...
    'callback',{@callback_OpExport,s});

s.pb5=uicontrol('style','pushbutton',...
    'units','pixels',...
    'HandleVisibility','off',...
    'position',[640 560 100 30],...
    'string','Importar Dados',...
    'callback',{@callback_import,s});

s.pb6=uicontrol('style','pushbutton',...
    'units','pixels',...
    'HandleVisibility','off',...
    'position',[800 10 80 40],...
    'string','Calcular',...
    'callback',{@callback_calc,s});
```

```
      pb7=uicontrol('style','pushbutton',...
          'units','pixels',...
          'HandleVisibility','off',...
          'position',[300 380 80 40],...
          'string','ANOVA',...
          'callback',@callback_OpANOVA);

      pbHelp=uicontrol('style','pushbutton',...
          'units','pixels',...
          'HandleVisibility','off',...
          'position',[10 10 80 40],...
          'string','Ajuda',...
          'callback',@ajuda);

      s.txt(1)=uicontrol('style','text',...
          'units','pixels',...
          'HandleVisibility','off',...
          'position',[220 550 100 30],...
          'string','Amostra Mestre :',...
          'BackgroundColor', color);

      ed1=uicontrol('style','edit',...
          'units','pixels',...
          'HandleVisibility','off',...
          'position',[320 560 300 25],...,
          'string','0.44  0.45  0.35  0.43  0.47  0.43  0.45  0.44  0.45  0.47
0.47;0.45  0.44  0.44  0.43  0.44  0.43  0.46  0.46  0.46  0.47  nan;0.42  0.46  0.45
0.43  0.41  0.41  0.47  0.42  0.43  0.44  0.45;0.42  0.46  0.45  0.43  0.41  0.41  0.47
0.42  0.43  0.44  0.45',...
          'BackgroundColor', 'White' );

      s.txt(5)=uicontrol('style','text',...
          'units','pixels',...
          'position',[200 500 150 30],...
          'HandleVisibility','off',...
          'string','N° amostras bootstrap :',...
          'BackgroundColor', color);

      ed5=uicontrol('style','edit',...
          'units','pixels',...
          'HandleVisibility','off',...
          'position',[350 510 100 25],...,
          'string','10000',...
          'BackgroundColor', 'White' );

      s.txt(6)=uicontrol('style','text',...
          'units','pixels',...
          'HandleVisibility','off',...
          'position',[210 450 100 30],...
          'string','Valor nominal :',...
          'BackgroundColor', color);

ed6=uicontrol('style','edit',...
                    'units','pixels',...
                    'HandleVisibility','off',...
                    'position',[300 460 100 25],...,
                    'string','0.41',...
                    'BackgroundColor', 'White' );

      s.tg = uitabgroup(s.fig(3),...
```

```
    'position',[0.02 0.1 0.96 0.5]);

s.tg1= uitab(s.tg,...
    'title','Matriz Experimento');

s.tg99 = uitab(s.tg,...
    'title','Amostra');

s.tg3 = uitab(s.tg,...
    'title','ANOM');

s.tg32 = uitab(s.tg,...
    'title','ANOM - SN');

s.tg5 = uitab(s.tg,...
    'title','ANOVA');

s.tg6 = uitab(s.tg,...
    'title','ANOVA SN');

s.tg4 = uitab(s.tg,...
    'title','Geral');

%----------abas--------------
%matrix exp
tbl3=uitable(s.tg1,...
    'position',[5 10 850 250]);
%Amostra
tbl99=uitable(s.tg99,...
    'position',[5 10 850 250]);
%ANOM
tbl5=uitable(s.tg3,...
    'position',[5 10 850 250]);
%ANOM - SN
tbl52=uitable(s.tg32,...
    'position',[5 10 850 250]);
%ANOVA
tbl8=uitable(s.tg5,...
    'position',[5 10 850 250]);
%ANOVA SN
tbl9=uitable(s.tg6,...
    'position',[5 10 850 250]);
%Geral
tbl6=uitable(s.tg4,...
    'position',[5 10 850 250]);

end

%-----------------------
%condição para interação
%-----------------------
function []=callback_popup1(varargin)
global intOK check5
      intOK=3;
      set(check5,'value',0);
end

%-----------------------------------------------------
%Exibi a quantidade de fatores disponiveis por nivel
%-----------------------------------------------------
```

```
      function []= callback_popup2(varargin)
      global popup1 intOK check5

      fa=get(gcbo,'value');

      switch fa
          case 2

set(popup1,'string',{'','2','3','4','5','6','7','8','9','10','11','12','13'
,'14','15','16','17','18','19','20','21','22','23','24','25','26','27','28'
,'29','30','31'});

          case 3
set(popup1,'string',{'','2','3','4','5','6','7','8','9','10','11','12','13'
});
          case 4
              set(popup1,'string',{'','2','3','4','5'});
          case 5
              set(popup1,'string',{'','2','3','4','5','6'});
          case 6

set(popup1,'string',{'','2','3','4','5','6','7','8','9','10','11','12','13'
,'14','15','16','17','18','19','20','21','22','23','24','25','26'});
      end

      set(popup1,'value',1);
      intOK=3;
      set(check5,'value',0);
      end

      %-----------------------------
      %Janela de opções dos gráficos
      %-----------------------------
      function []=callback_graficos(varargin)

      %VARIAVEIS
      %figplotOP -janela de opções
      %ch1graf - opção do gráfico da média
      %ch2graf - opção do gráfico do std
      %ch3graf - opção do gráfico da s/n

      global color ch1graf ch2graf ch3graf figplotOP

      figplotOP=figure('position', [300 300 200 200],...
          'menubar','none',...
          'name','Opções de gráfico',...
          'numbertitle','off',...
          'resize','off');

      uicontrol(figplotOP,...
          'style','text',...
          'position',[0 150 180 40],...
          'string','Selecione as opões desejadas:',...
          'BackgroundColor', color);

      ch1graf=uicontrol(figplotOP,...
          'style','checkbox',...
          'units','pixels',...
          'HandleVisibility','off',...
```

```
    'position',[10 130 80 40],...
    'string','Mean',...
    'BackgroundColor', color);

ch2graf=uicontrol(figplotOP,...
    'style','checkbox',...
    'units','pixels',...
    'HandleVisibility','off',...
    'position',[10 100 80 40],...
    'string','std',...
    'BackgroundColor', color);

ch3graf=uicontrol(figplotOP,...
    'style','checkbox',...
    'units','pixels',...
    'HandleVisibility','off',...
    'position',[10 70 80 40],...
    'string','S/N',...
    'BackgroundColor', color);

push1=uicontrol(figplotOP,...
    'style','pushbutton',...
    'units','pixels',...
    'HandleVisibility','off',...
    'position',[10 10 80 40],...
    'string','Ok',...
    'callback',@callback_OPgrafic);
End

%-------------------------------------------
%Check Interação: Cria janela opção interação
%-------------------------------------------
function []=callback_interacao(varargin)
global color vetint intList2 intList1  check5 opint intOK catInt unlock
namefatshow2 namefatshow3 popup1...
    popup2 L_name design conflito MIX

%VARIAVEIS:
%opint - janela de escolha de interações
%intList1 - exibição da lista de interações possíveis
%vetint - lista de interações possíveis
%intList2 - lista de interações escolhidas
%unlock - variavel para marcar e desmarcar o checkbox

% Desmarca quando estiver marcado

if unlock == 1
    set(check5,'value',0)
    unlock=0;
    return
end

%Verifica se foi escolhia a matriz
if isempty(intOK) == 1 | intOK~=1
    msgbox('Selecione a matriz experimental.');
    set(check5,'value',0);
    return
end

%avalia se a matriz escolhida permite interação
```

```
      valpopup1=get(popup1,'value');
      [lin,col]=size(design);

      conflito=0;
      %CONTROLE
      CONFLITOCHECK5=conflito

      switch L_name
          case 'L4 (2^3)'
              if valpopup1 == col
                  msgbox('O numero de fatores nao permite o cálculo de
interações');
                  set(check5,'value',0);
                  return
              end
          case 'L8 (2^7)'
              if valpopup1 == col
                  msgbox('O numero de fatores nao permite o cálculo de
interações');
                  set(check5,'value',0);
                  return
              end
          case 'L12 (2^11)'
                  msgbox('Essa matriz não permite o cálculo de interações');
                  set(check5,'value',0);
                  return
          case 'L16 (2^15)'
              if valpopup1 == col
                  msgbox('O numero de fatores nao permite o cálculo de
interações');
                  set(check5,'value',0);
                  return
              end

          case 'L32 (2^31)'
              if valpopup1 == col
                  msgbox('O numero de fatores nao permite o cálculo de
interações');
                  set(check5,'value',0);
                  return
              end
          case 'L9 (3^4)'
              if valpopup1 >= col-1
                  msgbox('O numero de fatores nao permite o cálculo de
interações');
                  set(check5,'value',0);
                  return
              end
          case 'L27 (3^13)'
              if valpopup1 >= col-1
                  msgbox('O numero de fatores nao permite o cálculo de
interações');
                  set(check5,'value',0);
                  return
              end
          case 'L16 (4^5)'
              if valpopup1 >= col-2
                  msgbox('O numero de fatores nao permite o cálculo de
interações');
                  set(check5,'value',0);
                  return
```

```
            end
        case 'L25 (5^6)'
            if valpopup1 >= col-3
                msgbox('O numero de fatores nao permite o cálculo de
interações');
                set(check5,'value',0);
                return
            end

        case 'L36 (2^11 x 3^12)'
            msgbox('Essa matriz não permite o cálculo de interações');
                set(check5,'value',0);
                return

    end

    %criar as interações
    vetor_interacaoMODIFICADO

    %construção da interface
    opint=figure('position', [100 100 500 300],...
        'menubar','none',...
        'name','Seleção de interações',...
        'numbertitle','off',...
        'resize','off');

    intText1=uicontrol('parent',opint,...
        'style','text',...
        'position',[55 250 100 30],...
        'string','Escolha as interações',...
        'BackgroundColor', color);

    intList1=uicontrol('parent',opint,...
        'style','listbox',...
        'position',[30 90 150 150],...
        'string',vetint,... %vetint
        'BackgroundColor', 'White' );

    intText2=uicontrol('parent',opint,...
        'style','text',...
        'position',[325 250 100 30],...
        'string','Interações escolhidas',...
        'BackgroundColor', color);

    intList2=uicontrol('parent',opint,...
        'style','listbox',...
        'position',[300 90 150 150],...
        'string',' ',...
        'BackgroundColor', 'White' );

    intPb1=uicontrol('parent',opint,...
        'style','pushbutton',...
        'position',[220 200 40 30],...
        'string','>',...
        'fontsize', 14,...
        'callback',@callback_transfereD);

    intPb2=uicontrol('parent',opint,...
        'style','pushbutton',...
```

```
    'position',[220 150 40 30],...
    'string','<',...
    'fontsize', 14,...
    'callback',@callback_transfereE);

intPb5=uicontrol('parent',opint,...
    'style','pushbutton',...
    'position',[330 10 100 50],...
    'string','Ok',...
    'callback',@callback_calculaInt);

catInt=1;

unlock=1;

end

%-------------------------------------------------
%Transfere para a direita a interação selecionada
%-------------------------------------------------
function [] = callback_transfereD(varargin)
global  intList2  intList1  vetint  catInt  listv  listI  namefatshow3
namefatshow3L namefatshow4

%VARIAVEIS
%vetint - Vetor de todas as interação
%intList1 - uicontrol listbox
%intList2 - uicontrol listbox
%listv - Lista das interações selecionadas
%listI - Lista do indice dos grupos das interações selecionadas
%catInt - Contador

if catInt ==1
    v1=get(intList1,'value');
    v2=vetint(v1);
    listI=v1;
    listv=v2;
    set(intList2,'string',v2); %v22 (como mascara)
else
    v3=get(intList1,'value');
    v4=vetint(v3);

    %teste para ver se ja possui o elemento na listI
    testSame3=intersect(listI,v3);
    testSame4=isempty(testSame3);

    %teste para ver se ja possui o elemento na listv
    testSame1=intersect(listv,v4);
    testSame2=isempty(testSame1);

    if testSame4 == 1
        listI=vertcat(listI,v3);
    end

    if testSame2 == 1
        listv=vertcat(listv,v4);
        set(intList2,'string',listv);
    end
end
```

```
% --------- mecanismo de selecionar a próxima interação -----------------
      tamlist=length(vetint);
      selec=get(intList1,'value');
      if selec<tamlist
          set(intList1,'value',selec+1);
      end

      catInt=catInt+1;
      end


      %-------------------------------------------------
      %Transfere para a esquerda a interação selecionada
      %-------------------------------------------------
      function [] = callback_transfereE(varargin)
      global intList2 listv listI namefatshow3L

      %VARIÁVEIS
      %intList2 - uicontrol listbox
      %listv - lista das interações selecionadas
      %listI - lista do indice dos grupos das interações selecionadas

      outV1=get(intList2,'value');
      if outV1 ==1
          set(intList2,'value',1);
      else
          set(intList2,'value',outV1-1);
      end
      listv(outV1)=[];
      listI(outV1)=[];
      %namefatshow3L(outV1)=[];
      set(intList2,'string',listv);
      end

      %---------------------------------
      %Opções de Design para as matrizes
      %---------------------------------
      function [] = DESIGNOPCOES (varargin)
      %VARIAVEIS
      %L_name - Nome da matriz selecionada
      %vetint - Lista de todas as interações
      %listv - Lista das interações selecionadas
      %valpopup1 - Numero de fatores
      %IntOri - Interação da matriz original
      %IntPro - Interação adaptada para o número de fatores
      %intera - Colunas das interações (respectivo a matriz original)
      %fac - Colunas dos fatores (respectivo a matriz original)
      %escolha - opção das colunas na matriz para os fatores
      %interacao - coluna atribuída a interação
      %alternativa - contador

      global listv L_name popup1 escolha interacao IntOri IntPro alternativa
      aux90 MIX

      valpopup1=get(popup1,'value');

      switch L_name
          case 'L4 (2^3)'
              IntOri{1}={'1x2'};
```

```
            IntPro{1}={'1x2'};
            Intera{1}=[3];
            Fac{1}=[1 2];
            %sem tri

            alternativa =1;
            %verificação de qual caso se encaixa
            for aux4=1:2
                alternativa

            %pula para a proxima alternativa caso o número de fatores não
            %seja compatível
                if  aux4<=1
                    n_fat=length(Fac{alternativa});
                    if n_fat < valpopup1
                        alternativa=alternativa+1;
                        continue
                    end
                end

                if alternativa > 1 %Ultimo caso possível

                    escolha=[1 2 3]
                end

                %conpara as interações possiveis com as selecionadas
                aux0=length(IntPro{alternativa});
                aux1=length(listv);
                conta=0;

                for par1=1:aux1
                    for par2=1:aux0

teste=strcmp(listv(par1),IntPro{alternativa}(par2));
                        if teste == 1
                            conta=conta+1;
                        end
                    end
                end

          %atribui as interações OU manda procurar na próxima alternativa
                if conta == aux1
                    escolha = Fac{alternativa}
                    interacao=Intera{alternativa}
                    return
                else
                    alternativa=alternativa+1;
                end
            end

        case 'L8 (2^7)'
            IntOri{1}={'1x2' '1x4' '2x4'};
            IntPro{1}={'1x2' '1x3' '2x3'};
            Intera{1}=[3 5 6];
            Fac{1}=[1 2 4 7];

            IntOri{2}={'1x2' '1x4' '1x7'};
            IntPro{2}={'1x2' '1x3' '1x4'};
            Intera{2}=[3 5 6];
            Fac{2}=[1 2 4 7];
```

```
            alternativa =1;
            %verificação de qual caso se encaixa
            for aux4=1:3
                alternativa

            %pula para a próxima alternativa caso o numero de fatores não
            %seja compatível
                if  aux4<=2
                    n_fat=length(Fac{alternativa});
                    if n_fat < valpopup1
                        alternativa=alternativa+1;
                        continue
                    end
                end

                if alternativa > 2 %Ultimo caso possível
                    if valpopup1 <= 4
                        escolha=[1 2 4 7]
                    else
                        escolha=[1 2 3 4 5 6 7]
                    end

                    h=Matriz_Triangular
                    if h == 0
                        IntPro{alternativa}=listv;
                        return
                    else
                        aviso_interacao
                        return
                    end
                end

                %compara as interações possíveis com as selecionadas
                aux0=length(IntPro{alternativa});
                aux1=length(listv);
                conta=0;

                for par1=1:aux1
                    for par2=1:aux0

teste=strcmp(listv(par1),IntPro{alternativa}(par2));
                        if teste == 1
                            conta=conta+1;
                        end
                    end
                end

          %atribui as interações OU manda procurar na próxima alternativa
                if conta == aux1
                    escolha = Fac{alternativa}
                    interacao=Intera{alternativa}
                    return
                else
                    alternativa=alternativa+1;
                end
            end

        case 'L16 (2^15)'
            %casos possíveis
```

```
            IntOri{1}={'1x2' '1x4' '1x8' '1x15' '2x4' '2x8' '2x15' '4x8'
'4x15' '8x15'};
            IntPro{1}={'1x2' '1x3' '1x4' '1x5' '2x3' '2x4' '2x5' '3x4'
'3x5' '4x5'};
            Intera{1}=[3 5 9 14 6 10 13 12 11 7];
            Fac{1}=[1 2 4 8 15];

            IntOri{2}={'1x2'  '1x4'  '1x8'  '1x10'  '1x12'  '1x15'  '2x4'
'8x15'};
            IntPro{2}={'1x2' '1x3' '1x4' '1x5' '1x6' '1x7' '2x3'  '4x7'};
            Intera{2}=[3 5 9 11 13 14 6 7];
            Fac{2}=[1 2 4 8 10 12 15];

            IntOri{3}={'1x2' '1x6' '1x8' '1x12' '2x8' '4x11' '5x11'};
            IntPro{3}={'1x2' '1x5' '1x6' '1x8' '2x6' '3x7' '4x7'};
            Intera{3}=[3 7 9 13 10 15 14];
            Fac{3}=[1 2 4 5 6 8 11 12];

            IntOri{4}={'1x2' '1x4' '1x6' '1x8' '1x10' '1x12' '1x14'};
            IntPro{4}={'1x2' '1x3' '1x4' '1x5' '1x6' '1x7' '1x8'};
            Intera{4}=[3 5 7 9 11 13 15];
            Fac{4}=[1 2 4 6 8 10 12 14];

            IntOri{5}={'1x2' '4x8' '5x10' '7x9' '6x11'};
            IntPro{5}={'1x2' '3x7' '4x9' '6x8' '5x10'};
            Intera{5}=[3 12 15 14 13];
            Fac{5}=[1 2 4 5 6 7 8 9 10 11];

            IntOri{6}={'1x6' '4x6' '5x6' '5x8' '5x10' '5x12' '5x14'};
            IntPro{6}={'1x4' '2x4' '3x4' '3x5' '3x6' '3x7' '3x8'};
            Intera{6}=[3 5 7 9 11 13 15];
            Fac{6}=[1 4 5 6 8 10 12 14];

            alternativa =1;
            %verificação de qual caso se encaixa
            for aux4=1:7
                alternativa

           %pula para a próxima alternativa caso o numero de fatores não
           %seja compatível
                if  aux4<=6
                    n_fat=length(Fac{alternativa});
                    if n_fat < valpopup1
                        alternativa=alternativa+1;
                        continue
                    end
                end

                if alternativa > 6 %Ultimo caso possível
                    if valpopup1 <= 4
                        escolha=[1 2 4 8]
                    elseif valpopup1 == 5
                        escolha=[1 2 4 8 15]
                    elseif (valpopup1 > 5) & (valpopup1 <= 8)
                        escolha=[1 2 4 8 7 11 13 14]
                    elseif valpopup1 >8
                        escolha=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]
                    end

                    h=Matriz_Triangular;
```

```
                    if h == 0
                        IntPro{alternativa}=listv;
                        return
                    else
                        aviso_interacao
                        return
                    end
                end

                %compara as interações possíveis com as selecionadas
                aux0=length(IntPro{alternativa});
                aux1=length(listv);
                conta=0;

                for par1=1:aux1
                    for par2=1:aux0

teste=strcmp(listv(par1),IntPro{alternativa}(par2));
                        if teste == 1
                            conta=conta+1;
                        end
                    end
                end

            %atribui as interações OU manda procurar na próxima alternativa
                    if conta == aux1
                        escolha = Fac{alternativa}
                        interacao=Intera{alternativa}
                        return
                    else
                        alternativa=alternativa+1;
                    end
              end

          case 'L32 (2^31)'
                IntOri{1}={'1x2' '1x4' '1x8' '1x15' '1x16' '1x22' '2x4' '2x8'
'2x15' '2x16' '2x25' '4x8' '4x15' '4x16' '4x26' '8x15'...
                    '8x16' '8x21' '15x16' '15x19'};
                IntPro{1}={'1x2' '1x3' '1x4' '1x5' '1x6' '1x9' '2x3' '2x4'
'2x5' '2x6' '2x10' '3x4' '3x5' '3x6' '3x11' '4x5'...
                    '4x6' '4x8' '5x6' '5x7'};
                Intera{1}=[3 5 9 14 17 23 6 10 13 18 27 12 11 20 30 7 24 29 3
28];
                Fac{1}=[1 2 4 8 15 16 19 21 22 25 26];

                IntOri{2}={'1x2' '1x4' '1x8' '1x15' '1x16' '1x18' '1x20' '1x22'
'1x24' '1x26' '1x28' '1x30' '2x4' '2x8' '2x15'...
                    '4x8' '4x15' '8x15'};
                IntPro{2}={'1x2' '1x3' '1x4' '1x5' '1x6' '1x7' '1x8' '1x9'
'1x10' '1x11' '1x12' '1x13' '2x3' '2x4' '2x5'...
                    '3x4' '3x5' '4x5'};
                Intera{2}=[3 5 9 14 17 19 21 23 25 27 29 31 6 10 13 12 11 7];
                Fac{2}=[1 2 4 8 15 16 18 20 22 24 26 28 30];

                IntOri{3}={'1x2' '1x4' '1x10' '1x12' '1x18' '1x20' '1x26'
'1x28' '2x4' '8x15' '8x16' '8x25' '14x16'...
                    '15x16' '15x25' '16x25'};
                IntPro{3}={'1x2' '1x3' '1x4' '1x5' '1x6' '1x7' '1x8' '1x9'
'2x3' '4x8' '4x9' '4x13' '7x9' '8x9' '8x13' '9x13'};
                Intera{3}=[3 5 11 13 19 21 27 29 6 7 24 17 30 31 22 9];
```

```
            Fac{3}=[1 2 4 8 10 12 14 15 16 18 20 23 25 26 28];

            IntOri{4}={'1x2' '1x4' '1x18' '1x20' '2x4' '8x15' '8x16' '8x25'
'10x16' '11x16' '12x16' '13x16' '14x16'...
                '15x16' '15x25' '16x25' };
            IntPro{4}={'1x2' '1x3' '1x12' '1x13' '2x3' '4x10' '4x11' '4x15'
'5x11' '6x11' '7x11' '8x11' '9x11'...
                '10x11' '10x15' '11x15'};
            Intera{4}=[3 5 21 19 21 6 7 24 17 26 27 28 29 30 31 22 9];
            Fac{4}=[1 2 4 6 8 10 11 12 13 14 15 16 18 20 23 25];

            IntOri{5}={'1x2' '1x4' '1x12' '1x26' '2x4' '8x15' '8x16' '8x25'
'10x20' '11x23' '14x19' '15x16' '15x25' '16x25'};
            IntPro{5}={'1x2' '1x3' '1x7' '1x17' '2x3' '4x9' '4x10' '4x16'
'5x13' '6x15' '8x12' '9x10' '9x16' '10x16'};
            Intera{5}=[3 5 13 27 6 7 24 17 30 28 29 31 22 9];
            Fac{5}=[1 2 4 8 10 11 12 14 15 16 18 19 20 21 23 25 26];

            IntOri{6}={'1x2' '1x4' '1x10' '1x12' '1x20' '1x26' '1x28' '2x4'
'8x15' '8x16' '8x25' '10x20' '15x16' '15x25' '16x25'};
            IntPro{6}={'1x2' '1x3' '1x5' '1x6' '1x11' '1x14' '1x15' '2x3'
'4x8' '4x9' '4x13' '5x11' '8x9' '8x13' '9x13'};
            Intera{6}=[3 5 11 13 19 21 27 29 6 7 24 17 30 31 22 9];
            Fac{6}=[1 2 4 8 10 12 14 15 16 18 20 23 25 26 28];

            IntOri{7}={'1x2' '1x4' '1x10' '1x20' '2x4' '8x15' '8x16' '8x18'
'8x25' '10x20' '12x23' '14x19' '15x16' '15x25' '16x25'};
            IntPro{7}={'1x2' '1x3' '1x5' '1x13' '2x3'  '4x9' '4x10' '4x11'
'4x15' '5x13' '6x14' '8x12' '9x10' '9x15' '10x15'};
            Intera{7}=[3 5 11 21 6 7 24 26 17 30 27 29 31 22 9];
            Fac{7}=[1 2 4 8 10 12 13 14 15 16 18 19 20 23 25 28];

            IntOri{8}={'1x2'  '1x4'  '2x4'  '8x15'  '8x25'  '9x29'  '10x16'
'10x30' '11x24' '12x23' '13x18' '15x25' '16x30'};
            IntPro{8}={'1x2' '1x3' '2x3' '4x10' '2x15' '5x17' '6x11' '6x18'
'7x14' '8x13' '9x12' '10x15' '11x18'};
            Intera{8}=[3 5 6 7 17 21 26 20 19 27 31 22 34];
            Fac{8}=[1 2 4 8 9 10 11 12 13 15 16 18 23 24 25 28 29 30];

            IntOri{9}={'1x2'  '1x4' '1x6' '1x8' '1x10' '1x16' '1x20' '1x22'
'1x24' '1x26' '2x16' '4x8' '6x26' '10x20' '13x16' '15x16' '22x24'  };
            IntPro{9}={'1x2' '1x3' '1x4' '1x5' '1x6' '1x9' '1x11' '1x12'
'1x13' '1x14' '2x9' '3x5' '4x14' '6x11' '7x9' '8x9' '12x13'};
            Intera{9}=[1 5 7 9 10 17 21 23 25 27 18 12 28 30 29 31 14];
            Fac{9}=[1 2 4 6 8 10 13 15 16 19 20 22 24 26];

            IntOri{10}={'1x2'  '1x4'  '2x4'  '8x16'  '9x19'  '10x20'  '11x23'
'12x17' '13x18' '14x21' '15x22'};
            IntPro{10}={'1x2'  '1x3'  '2x3'  '5x13'  '6x16'  '7x17'  '8x20'
'9x14' '10x15' '11x18' '12x19'};
            Intera{10}=[3 5 6 24 26 30 28 29 31 27 25];
            Fac{10}=[1 2 4 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
23];

            IntOri{11}={'1x2'  '1x4'  '1x24'  '1x26'  '1x28'  '1x30'  '2x4'
'2x16' '2x17' '2x20' '2x21' '4x8' '4x9' '4x10' '4x11'};
            IntPro{11}={'1x2' '1x3' '1x13' '1x14' '1x15' '1x16' '2x3' '2x9'
'2x10' '2x11' '2x12' '3x5' '3x6' '3x7' '3x8'};
            Intera{11}=[3 5 25 27 29 31 6 18 19 22 23 12 13 14 15];
            Fac{11}=[1 2 4 7 8 9 10 11 16 17 20 21 24 26 28 30];
```

```
            IntOri{12}={'1x2'  '1x6'  '1x14'  '1x28'  '1x30'  '2x9'  '2x17'
'2x21' '2x25' '4x8' '4x16' '4x18' ...
                '5x8' '8x16' '10x16'};
            IntPro{12}={'1x2' '1x5' '1x9' '1x15' '1x16' '2x7' '2x11' '2x13'
'2x14' '3x6' '3x10' '3x12'...
                '4x6' '6x10' '8x10'};
            Intera{12}=[3 7 15 28 31 11 19 23 27 12 20 22 13 24 26];
            Fac{12}=[1 2 4 5 6 8 9 10 14 16 17 18 21 25 28 30];

            IntOri{13}={'1x2' '1x4' '1x6' '1x8' '1x10' '1x12' '1x14' '2x16'
'2x17' '2x20' '2x21' '2x24' '2x25' '2x28' '2x29'};
            IntPro{13}={'1x2'  '1x3'  '1x4'  '1x5'  '1x6'  '1x7'  '1x8'  '2x9'
'2x10' '2x11' '2x12' '2x13' '2x14' '2x15' '2x16'};
            Intera{13}=[3 5 7 9 11 13 15 18 19 22 23 26 27 30 31];
            Fac{13}=[1 2 4 6 8 10 12 14 16 17 20 21 24 25 28 29];

            alternativa =1;
            %verificação de qual caso se encaixa
            for aux4=1:14
                alternativa

            %pula para a próxima alternativa caso o numero de fatores não
            %seja compatível
                if  aux4<=13
                    n_fat=length(Fac{alternativa});
                    if n_fat < valpopup1
                        alternativa=alternativa+1;
                        continue
                    end
                end

                if alternativa > 13 %Ultimo caso possível
                    if valpopup1 <= 4
                        escolha=[1 2 4 8]
                    elseif valpopup1 == 5
                        escolha=[1 2 4 8 16]
                    elseif valpopup1 == 6
                        escolha=[1 2 4 8 16 31]
                    elseif (valpopup1 > 6) & (valpopup1 <= 16)
                        escolha=[1 2 4 8 16 7 11 13 14 19 21 22 25 26 28
31]
                    elseif valpopup1 >16
                        escolha=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27 28 29 30 31]
                    end

                    h=Matriz_Triangular
                    if h == 0
                        IntPro{alternativa}=listv;
                        return
                    else
                        aviso_interacao
                        return
                    end

                end

                %compara as interações possíveis com as selecionadas
                aux0=length(IntPro{alternativa});
```

```
                aux1=length(listv);
                conta=0;

                for par1=1:aux1
                    for par2=1:aux0

teste=strcmp(listv(par1),IntPro{alternativa}(par2));
                        if teste == 1
                            conta=conta+1;
                        end
                    end
                end

          %atribui as interações OU manda procurar na próxima alternativa
                if conta == aux1
                    escolha = Fac{alternativa}
                    interacao=Intera{alternativa}
                    return
                else
                    alternativa=alternativa+1;
                end
            end

        case 'L9 (3^4)'
            IntOri{1}={'1x2'};
            IntPro{1}={'1x2'};
            Intera{1}=[3 4];
            Fac{1}=[1 2];

            alternativa =1;
            %verificação de qual caso se encaixa
            for aux4=1:2
                alternativa
            %pula para a próxima alternativa caso o número de fatores não
            %seja compatível
                if  aux4<=1
                    n_fat=length(Fac{alternativa});
                    if n_fat < valpopup1
                        alternativa=alternativa+1;
                        continue
                    end
                end

                if alternativa > 1 %Ultimo caso possível
                    escolha=[1 2 3 4]
                end

                %compara as interações possíveis com as selecionadas
                aux0=length(IntPro{alternativa});
                aux1=length(listv);
                conta=0;

                for par1=1:aux1
                    for par2=1:aux0

teste=strcmp(listv(par1),IntPro{alternativa}(par2));
                        if teste == 1
                            conta=conta+1;
                            aux90{conta}=par2;
                        end
```

```
            end
        end

  %atribui as interações OU manda procurar na próxima alternativa
        if conta == aux1
            escolha = Fac{alternativa}
            interacao=Intera{alternativa}
            return
        else
            alternativa=alternativa+1;
        end
    end

case 'L27 (3^13)'
    IntOri{1}={'1x2' '1x5' '2x5'};
    IntPro{1}={'1x2' '1x3' '2x3'};
    Intera{1}=[3 4 6 7 8 11];
    Fac{1}=[1 2 5 9 10 12 13];

    IntOri{2}={'1x2' '1x3' '1x4' '1x5'};
    IntPro{2}={'1x2' '1x3' '1x4' '1x5'};
    Intera{2}=[3 4 6 7 9 10 12 13];
    Fac{2}=[1 2 5 8 11];

    alternativa =1;
    %verificação de qual caso se encaixa
    for aux4=1:3
        alternativa
    %pula para a próxima alternativa caso o número de fatores não
    %seja compatível
        if  aux4<=2
            n_fat=length(Fac{alternativa});
            if n_fat < valpopup1
                alternativa=alternativa+1;
                continue
            end
        end

        if alternativa > 2 %Ultimo caso possível
            if valpopup1 <=7
                escolha=[1 2 5 9 10 12 13]
            else
                escolha =[1 2 3 4 5 6 7 8 9 10 11 12 13]
            end

            h=Matriz_Triangular
            if h == 0
                IntPro{alternativa}=listv;
                return
            else
                aviso_interacao
                return
            end
        end

        %compara as interações possíveis com as selecionadas
        aux0=length(IntPro{alternativa});
        aux1=length(listv);
        conta=0;
```

```
                for par1=1:aux1
                    for par2=1:aux0

teste=strcmp(listv(par1),IntPro{alternativa}(par2));
                        if teste == 1
                            conta=conta+1;
                            aux90{conta}=par2;
                        end
                    end
                end

          %atribui as interações OU manda procurar na próxima alternativa
                if conta == aux1
                    escolha = Fac{alternativa}
                    interacao=Intera{alternativa}
                    return
                else
                    alternativa=alternativa+1;
                end
            end

        case 'L16 (4^5)'
            IntOri{1}={'1x2'};
            IntPro{1}={'1x2'};
            Intera{1}=[3 4 5];
            Fac{1}=[1 2];

            alternativa =1;
            %verificação de qual caso se encaixa
            for aux4=1:2
                alternativa

            %pula para a próxima alternativa caso o número de fatores não
            %seja compatível
                if  aux4<=1
                    n_fat=length(Fac{alternativa});
                    if n_fat < valpopup1
                        alternativa=alternativa+1;
                        continue
                    end
                end

                if alternativa > 1 %Último caso possível

                    escolha=[1 2 3 4 5]
                end

                %compara as interações possíveis com as selecionadas
                aux0=length(IntPro{alternativa});
                aux1=length(listv);
                conta=0;

                for par1=1:aux1
                    for par2=1:aux0

teste=strcmp(listv(par1),IntPro{alternativa}(par2));
                        if teste == 1
                            conta=conta+1;
                            aux90{conta}=par2;
                        end
```

```
                    end
                end

            %atribui as interações OU manda procurar na próxima alternativa
                if conta == aux1
                    escolha = Fac{alternativa}
                    interacao=Intera{alternativa}
                    return
                else
                    alternativa=alternativa+1;
                end
            end

        case 'L25 (5^6)'
            IntOri{1}={'1x2'};
            IntPro{1}={'1x2'};
            Intera{1}=[3 4 5 6];
            Fac{1}=[1 2];

            alternativa =1;
            %verificação de qual caso se encaixa
            for aux4=1:2
                alternativa

            %pula para a proxima alternativa caso o numero de fatores não
            %seja compativel
                if  aux4<=1
                    n_fat=length(Fac{alternativa});
                    if n_fat < valpopup1
                        alternativa=alternativa+1;
                        continue
                    end
                end

                if alternativa > 1 %Ultimo caso possível
                    escolha=[1 2 3 4 5 6]
                end

                %compara as interações possíveis com as selecionadas
                aux0=length(IntPro{alternativa});
                aux1=length(listv);
                conta=0;

                for par1=1:aux1
                    for par2=1:aux0

teste=strcmp(listv(par1),IntPro{alternativa}(par2));
                        if teste == 1
                            conta=conta+1;
                            aux90{conta}=par2;
                        end
                    end
                end

            %atribui as interações OU manda procurar na próxima alternativa
                if conta == aux1
                    escolha = Fac{alternativa}
                    interacao=Intera{alternativa}
                    return
                else
```

```
                    alternativa=alternativa+1;
                end
            end

        case 'L18 (2x3^7)'
            IntOri{1}={'1x2'};
            IntPro{1}={'1x2'};
            Intera{1}=[];
            Fac{1}=[1 2];

            MIX=1;
            alternativa =1;
            %verificação de qual caso se encaixa
            for aux4=1:2
                alternativa

            %pula para a próxima alternativa caso o número de fatores não
            %seja compatível
                if  aux4<=1
                    n_fat=length(Fac{alternativa});
                    if n_fat < valpopup1
                        alternativa=alternativa+1;
                        continue
                    end
                end

                if alternativa > 1 %Ultimo caso possível
                    escolha=[1 2 3 4 5 6 7 8]
                end

                %compara as interações possíveis com as selecionadas
                aux0=length(IntPro{alternativa});
                aux1=length(listv);
                conta=0;

                for par1=1:aux1
                    for par2=1:aux0

teste=strcmp(listv(par1),IntPro{alternativa}(par2));
                        if teste == 1
                            conta=conta+1;
                            aux90{conta}=par2;
                        end
                    end
                end

          %atribui as interações OU manda procurar na próxima alternativa
                if conta == aux1
                    escolha = Fac{alternativa}
                    interacao=Intera{alternativa}
                    return
                else
                    alternativa=alternativa+1;
                end
            end

        case 'L32 (2x4^9)'
            IntOri{1}={'1x2'};
            IntPro{1}={'1x2'};
            Intera{1}=[];
```

```
            Fac{1}=[1 2];

            MIX=1;
            alternativa =1;
            %verificação de qual caso se encaixa
            for aux4=1:2
                alternativa

            %pula para a próxima alternativa caso o número de fatores não
            %seja compatível
                if  aux4<=1
                    n_fat=length(Fac{alternativa});
                    if n_fat < valpopup1
                        alternativa=alternativa+1;
                        continue
                    end
                end

                if alternativa > 1 %Ultimo caso possível
                    escolha=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
19 20 21 22 23 24 25 26]
                end

                %compara as interações possíveis com as selecionadas
                aux0=length(IntPro{alternativa});
                aux1=length(listv);
                conta=0;

                for par1=1:aux1
                    for par2=1:aux0

teste=strcmp(listv(par1),IntPro{alternativa}(par2));
                        if teste == 1
                            conta=conta+1;
                            aux90{conta}=par2;
                        end
                    end
                end

          %atribui as interações OU manda procurar na próxima alternativa
                if conta == aux1
                    escolha = Fac{alternativa}
                    interacao=Intera{alternativa}
                    return
                else
                    alternativa=alternativa+1;
                end
            end

        case 'L50 (2x5^11)'
            IntOri{1}={'1x2'};
            IntPro{1}={'1x2'};
            Intera{1}=[];
            Fac{1}=[1 2];

            MIX=1;
            alternativa =1;
            %verificação de qual caso se encaixa
            for aux4=1:2
                alternativa
```

```
            %pula para a próxima alternativa caso o número de fatores não
            %seja compatível
                if  aux4<=1
                    n_fat=length(Fac{alternativa});
                    if n_fat < valpopup1
                        alternativa=alternativa+1;
                        continue
                    end
                end

                if alternativa > 1 %Ultimo caso possível
                    escolha=[1 2 3 4 5 6 7 8 9 10 11 12]
                end

                %compara as interações possíveis com as selecionadas
                aux0=length(IntPro{alternativa});
                aux1=length(listv);
                conta=0;

                for par1=1:aux1
                    for par2=1:aux0

teste=strcmp(listv(par1),IntPro{alternativa}(par2));
                        if teste == 1
                            conta=conta+1;
                            aux90{conta}=par2;
                        end
                    end
                end

            %atribui as interações OU manda procurar na próxima alternativa
                if conta == aux1
                    escolha = Fac{alternativa}
                    interacao=Intera{alternativa}
                    return
                else
                    alternativa=alternativa+1;
                end
            end

        case 'L54 (2 x 3^25)'
            IntOri{1}={'1x2' '1x9' '2x9' '3x9' '4x9' '5x9' '6x9' '7x9'
'8x9'};
            IntPro{1}={'1x2' '1x9' '2x9' '3x9' '4x9' '5x9' '6x9' '7x9'
'8x9'};
            Intera{1}=[nan nan nan nan nan 15 16 17 18 19 20 21 22 23
24 25 26];
            Fac{1}=[1 2 3 4 5 6 7 8 9];

            alternativa =1;
            %verificação de qual caso se encaixa
            for aux4=1:2
                alternativa

            %pula para a próxima alternativa caso o número de fatores não
            %seja compatível
                if  aux4<=1
                    n_fat=length(Fac{alternativa});
                    if n_fat < valpopup1
```

```
                        alternativa=alternativa+1;
                        continue
                    end
                end

                if alternativa > 1 %Último caso possível
                    if valpopup1 <=3
                        escolha=[1 2 9];
                    else
                        escolha=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26]
                    end
                    IntPro{alternativa}=listv;
                    msgbox('Essa matriz não pode calcular as interações
selecionadas');
                    return
                end

                %compara as interações possíveis com as selecionadas
                aux0=length(IntPro{alternativa});
                aux1=length(listv);
                conta=0;

                for par1=1:aux1
                    for par2=1:aux0

teste=strcmp(listv(par1),IntPro{alternativa}(par2));
                        if teste == 1
                            conta=conta+1;
                            aux90{conta}=par2;
                        end
                    end
                end

            %atribui as interações OU manda procurar na próxima alternativa
                    if conta == aux1
                        escolha = Fac{alternativa}
                        interacao=Intera{alternativa}
                        return
                    else
                        alternativa=alternativa+1;
                    end
              end
      end


      %---------- limpar variáveis -------------------
      clearvars aux0 aux1 conta
      end


      %---------------------------------
      %Janela AVISO de confusão de fator
      %---------------------------------
      function aviso_interacao(varargin)
      global color

      cont_troca= figure('position',[300 300  200 200],...
                         'menubar','none',...
                         'name','AVISO',...
```

```
                            'numbertitle','off',...
                            'resize','off');
                        uicontrol(cont_troca,...
                            'position',[10 90 180 100],...
                            'style','text',...
                            'string','Algumas interações podem ser
confundidas com os fatores principais, deseja continuar com essas interações
ou escolher outras? Caso continue suas interações não serão levadas em
consideração',...
                            'BackgroundColor', color)
                        uicontrol(cont_troca,...
                            'style','pushbutton',...
                            'position',[10 10 80 50],...
                            'string','Continuar',...
                            'callback',@continuar)
                        uicontrol(cont_troca,...
                            'style','pushbutton',...
                            'position',[110 10 80 50],...
                            'string','Trocar',...
                            'callback',@troca)

    end


    %------------------------
    %opção da janela aviso
    %------------------------
    function continuar(varargin)
    global check5
        set(check5,'value',0);
        close(gcf)
    end


    %----------------------
    %opção da janela aviso
    %----------------------
    function troca(varargin)
    global conflito check5
        conflito=0;
        %CONTROLE
        CONFLITOTROCA=conflito
        set(check5,'value',0);
        close(gcbf)
    end


    %------------------------
    %Função Matriz triangular
    %------------------------
    function [achou] = Matriz_Triangular(varargin)
    global escolha L_name listv popup1 interacao alternativa IntPro
    conflito IntOri

    %VARIAVEIS
    %conflito - parâmetro para avisar quando existe conflito entre
    %interação fator principal
    %IntOri - Interação com nomenclatura original
    %IntPro - Interação com nomenclatura do programa
```

```
conflito=0;
LG=0;
L27=0;

switch L_name

    case 'L8 (2^7)'
        Tab_int=[
            nan 3 2 5 4 7 6;
            nan nan 1 6 7 4 5;
            nan nan nan 7 6 5 4;
            nan nan nan nan 1 2 3;
            nan nan nan nan nan 3 2;
            nan nan nan nan nan nan 1;
            nan nan nan nan nan nan nan];
        LG=1;

    case 'L16 (2^15)'
        Tab_int=[
    nan  3  2 5 4 7 6 9 8 11  10  13  12  15  14;
    nan nan 1 6 7 4 5 10  11  8 9 14  15  12  13;
    nan nan nan 7 6 5 4 11  10  9 8 15  14  13  12;
    nan nan nan nan 1 2 3 12  13  14  15  8 9 10  11;
    nan nan nan nan nan 3 2 13  12  15  14  9 8 11  10;
    nan nan nan nan nan nan 1 14  15  12  13  10  11  8 9;
    nan nan nan nan nan nan nan 15  14  13  12  11  10  9 8;
    nan nan nan nan nan nan nan nan 1 2 3 4 5 6 7;
    nan nan nan nan nan nan nan nan nan 3 2 5 4 7 6;
    nan nan nan nan nan nan nan nan nan nan 1 6 7 4 5;
    nan nan nan nan nan nan nan nan nan nan nan 7 6 5 4;
    nan nan nan nan nan nan nan nan nan nan nan nan 1 2 3;
    nan nan nan nan nan nan nan nan nan nan nan nan nan 3 2;
    nan nan nan nan nan nan nan nan nan nan nan nan nan nan 1];
    LG=1;

  case 'L32 (2^31)'
    Tab_int=[
    nan 3 2 5 4 7 6 9 8 11 10 13 12 15 14 17 16 19 18 21 20 23 22 25
24 27 26 29 28 31 30;
    nan nan 1 6 7 4 5 10 11 8 9 14 15 12 13 18 19 16 17 22 23 20 21 26
27 24 25 30 31 28 29;
    nan nan nan 7 6 5 4 11 10 9 8 15 14 13 12 19 18 17 16 23 22 21 20
27 26 25 24 31 30 29 28;
    nan nan nan nan 1 2 3 12 13 14 15 8 9 10 11 20 21 22 23 16 17 18
19 28 29 30 31 24 25 26 27;
    nan nan nan nan nan 3 2 13 12 15 14 9 8 11 10 21 20 23 22 17 16 19
18 29 28 31 30 25 24 27 26;
    nan nan nan nan nan nan 1 14 15 12 13 10 11 8 9 22 23 20 21 18 19
16 17 30 31 28 29 26 27 24 25;
    nan nan nan nan nan nan nan 15 14 13 12 11 10 9 8 23 22 21 20 19
18 17 16 31 30 29 28 27 26 25 24;
    nan nan nan nan nan nan nan nan 1 2 3 4 5 6 7 24 25 26 27 28 29 30
31 16 17 18 19 20 21 22 23;
    nan nan nan nan nan nan nan nan nan 3 2 5 4 7 6 25 24 27 26 29 28
31 30 17 16 19 18 21 20 23 22;
    nan nan nan nan nan nan nan nan nan nan 1 3 7 4 5 26 27 24 25 30
31 28 29 18 19 16 17 22 23 20 21;
    nan nan nan nan nan nan nan nan nan nan nan 7 6 5 4 27 26 25 24 31
30 29 28 19 18 17 16 23 22 21 20;
    nan nan nan nan nan nan nan nan nan nan nan nan 1 2 3 28 29 30 31
24 25 26 27 20 21 22 23 16 17 18 19;
```

```
          nan nan nan nan nan nan nan nan nan nan nan nan 3 2 29 28 31
30 25 24 27 26 21 20 23 22 17 16 19 18;
          nan nan nan nan nan nan nan nan nan nan nan nan nan 1 30 31 28
29 26 27 24 25 22 23 20 21 18 19 16 17;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan 31 30
29 28 27 26 25 24 23 22 21 20 19 18 17 16;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan 1
2 3 4 5 6 7 8 9 10 11 12 13 14 15;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
3 2 5 4 7 6 9 8 11 10 13 12 15 14;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan 1 6 7 4 5 10 11 8 9 14 15 12 13;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan 7 6 5 4 11 10 9 8 15 14 13 12;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan 1 2 3 12 13 14 15 8 9 10 11;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan 3 2 13 12 15 14 9 8 11 10;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan 1 14 15 12 13 10 11 8 9;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan 15 14 13 12 11 10 9 8;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan 1 2 3 4 5 6 7;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan nan 3 2 5 4 7 6;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan nan nan 1 6 7 4 5;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan nan nan nan 7 6 5 4;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan nan nan nan nan 1 2 3;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan nan nan nan nan nan 3 2;
          nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan nan
nan nan nan nan nan nan nan nan nan nan nan nan nan 1];
          LG=1;

        case 'L27 (3^13)'
          Tab_int=[
          nan 3 2 2 6 5 5 9 8 8 12 11 11;
          nan 4 4 3 7 7 6 10 10 9 13 13 12;
          nan nan 1 1 8 9 10 5 6 7 5 6 7;
          nan nan 4 3 11 12 13 11 12 13 8 9 10;
          nan nan nan 1 9 10 8 7 5 6 6 7 5;
          nan nan nan 2 13 11 12 12 13 11 10 8 9;
          nan nan nan nan 10 8 9 6 7 5 7 5 6;
          nan nan nan nan 12 13 11 13 11 12 9 10 8;
          nan nan nan nan nan 1 1 2 3 4 2 4 3;
          nan nan nan nan nan 7 6 11 13 12 8 10 9;
          nan nan nan nan nan nan 1 4 2 3 3 2 4;
          nan nan nan nan nan nan 5 13 12 11 10 9 8;
          nan nan nan nan nan nan nan 3 4 2 4 3 2;
          nan nan nan nan nan nan nan 12 11 13 9 8 10;
          nan nan nan nan nan nan nan nan 1 1 2 3 4;
          nan nan nan nan nan nan nan nan 10 9 5 7 6;
          nan nan nan nan nan nan nan nan nan 1 4 2 3;
          nan nan nan nan nan nan nan nan nan 8 7 6 5;
          nan nan nan nan nan nan nan nan nan nan 3 4 2;
          nan nan nan nan nan nan nan nan nan nan 6 5 7;
          nan nan nan nan nan nan nan nan nan nan nan 1 1;
```

```
    nan nan nan nan nan nan nan nan nan nan nan 13 12;
    nan nan nan nan nan nan nan nan nan nan nan nan 1;
    nan nan nan nan nan nan nan nan nan nan nan nan 11];

    L27=1;
end

%seleciona os fatores
 aux7=length(listv);
aux8=get(popup1,'value');
for par1=1:aux7
  teste_igualdade=strcmp(listv{par1}(3),'x');
  if teste_igualdade == 1
    primeiro=str2num(listv{par1}(1:2));
    teste_vazio=length(listv{par1});
    if teste_vazio == 4
    segundo=str2num(listv{par1}(4));
    else
    segundo=str2num(listv{par1}(4:5));
    end
  else
    primeiro=str2num(listv{par1}(1));
    teste_vazio=length(listv{par1});
    if teste_vazio == 3
    segundo=str2num(listv{par1}(3));
    else
    segundo=str2num(listv{par1}(3:4));
    end
  end

  pprimeiro=escolha(primeiro);
  ssegundo=escolha(segundo);

  if LG == 1
    %criavetor interacao
    aux45=Tab_int(pprimeiro,ssegundo);
    if par1==1
      interacao=aux45;
      IntOri{alternativa}=[num2str(pprimeiro) 'x' num2str(ssegundo)];
    elseif par1 == 2
      hj= [num2str(pprimeiro) 'x' num2str(ssegundo)];
      IntOri{alternativa}=[{IntOri{alternativa}} {hj}];
      interacao=[interacao aux45];
    else
      hj= [num2str(pprimeiro) 'x' num2str(ssegundo)];
      IntOri{alternativa}=[IntOri{alternativa} hj];
      interacao=[interacao aux45];
    end
  end

  if L27 == 1
    %criavetor interacao
    par=mod(primeiro,2);
    if par == 0
      aux45=Tab_int(primeiro-1,segundo);
      aux34=Tab_int(primeiro,segundo);
    else
       aux45=Tab_int(primeiro,segundo);
      aux34=Tab_int(primeiro+1,segundo);
    end
```

```
        if par1==1
           interacao=[aux45 aux34];
          IntOri{alternativa}=[num2str(pprimeiro) 'x' num2str(ssegundo)];
        elseif par1 == 2
          hj= [num2str(pprimeiro) 'x' num2str(ssegundo)];
          IntOri{alternativa}=[{IntOri{alternativa}} {hj}];
          interacao=[interacao aux45 aux34];
        else
          hj= [num2str(pprimeiro) 'x' num2str(ssegundo)];
          IntOri{alternativa}=[IntOri{alternativa} hj];
          interacao=[interacao aux45 aux34];
        end
      end

    end

    IIINTERACAO=interacao
    EESCOLHA=escolha

    if L27 == 1
      aux7=2*aux7
    end
    achou=0;
    for par1=1:aux7
      %achou=0;
      for par2=1:aux8
        teste_igual=(interacao(par1)==escolha(par2));
        if teste_igual == 1
          achou=1;
          conflito=1
        return
        end
      end
    end

    end


    %-----------------------------------------------------------------
    %calcula o número de interações possíveis e chama TODAS as funções
    %-----------------------------------------------------------------
    function []=callback_calculaInt(varargin)
    global opint listv popup2 design3 design L_name des nameExp tbl3

    %Df - numero de colunas
    %colu1 - numero de colunas da matriz no modo padrão
    %colu2 - numero de colunas do design selecionado baseado na matriz
padrão

    tamanho=length(listv);
    valpopup2=get(popup2,'value');
    [linh1,colu1]=size(design);
    [linh2,colu2]=size(design3);

    if valpopup2 ==2
      Df=colu2 + tamanho;
      if Df>colu1
        msgbox('Não é possivel calcular todas as interações');
        return
```

```
  end

  DESIGNOPCOES
  DESIGNN_3
  DESIGNN_4
  nameMODIFICADO
  [r1,c1]=size(des);
  set(tbl3,'columnname',nameExp(1:c1));
  GRUPING

elseif valpopup2==3
  Df=colu2 + 2*tamanho;
  if Df>colu1
    msgbox('Não é possivel calcular todas as interações');
    return
  end

  DESIGNOPCOES
  DESIGNN_3
  DESIGNN_4
  nameMODIFICADO
  [r1,c1]=size(des);
  set(tbl3,'columnname',nameExp(1:c1));
  GRUPING

elseif valpopup2==4
  Df=colu2 + 3*tamanho;
  if Df>colu1
    msgbox('Não é possivel calcular todas as interações');
    return
  end

  DESIGNOPCOES
  DESIGNN_3
  DESIGNN_4
  nameMODIFICADO
  [r1,c1]=size(des);
  set(tbl3,'columnname',nameExp(1:c1));
  GRUPING

elseif valpopup2==5
  Df=colu2 + 4*tamanho
  if Df>colu1
    msgbox('Não é possivel calcular todas as interações');
    return
  end

  DESIGNOPCOES
  DESIGNN_3
  DESIGNN_4
  nameMODIFICADO
  [r1,c1]=size(des);
  set(tbl3,'columnname',nameExp(1:c1));
  GRUPING

elseif valpopup2==6
  switch L_name
    case 'L18 (2 x 3^7)'
    f=strcmp('1x2',listv(1));
    if (tamanho > 1) | (f == 0)
```

```
            msgbox('Não é possivel calcular todas as interações, essa matriz
calcula apenas a interação Fator 1 x Fator 2');
            return
          end

          DESIGNOPCOES
          DESIGNN_3
          nameMODIFICADO
          [r1,c1]=size(des);
          set(tbl3,'columnname',nameExp(1:c1));
          GRUPING

          case 'L32 (2 x 4^9)'
          f=strcmp('1x2',listv(1));
          if (tamanho > 1) | (f == 0)
            msgbox('Não é possivel calcular todas as interações, essa matriz
calcula apenas a interação Fator 1 x Fator 2');
            return
          end

          DESIGNOPCOES
          DESIGNN_3
          nameMODIFICADO
          [r1,c1]=size(des);
          set(tbl3,'columnname',nameExp(1:c1));
          GRUPING

          case 'L50 (2 x 5^11)'
          f=strcmp('1x2',listv(1));
          if (tamanho > 1) | (f == 0)
            msgbox('Não é possivel calcular todas as interações, essa matriz
calcula apenas a interação Fator 1 x Fator 2');
            return
          end

          DESIGNOPCOES
          DESIGNN_3
          nameMODIFICADO
          [r1,c1]=size(des);
          set(tbl3,'columnname',nameExp(1:c1));
          GRUPING

          case 'L54 (2 x 3^25)'

          intA=0;
          for w=1:tamanho
            f=strcmp('1',listv{w}(1));
            if f==1
              intA=intA +1;
            end
            if intA > 2
              msgbox('Não é possivel calcular todas as interações, selecione
no maximo 2 interações com o primeiro fator');
              return
            end
          end
          if valpopup2 > 9
            msgbox('Não é possivel calcular todas as interações');
            return
          end
```

```
    DESIGNOPCOES
    DESIGNN_3
    DESIGNN_4
    nameMODIFICADO
    [r1,c1]=size(des);
    set(tbl3,'columnname',nameExp(1:c1));
    GRUPING
  end

end

close(opint);
end


%---------------------------
%Janela de opções de análise
%---------------------------
function []=callback_op(varargin)

%VARIAVEIS
%bgSN - Tipo de Razão S/N
%edit7 - K
%edit9 - P
%edit10 - Delta
%bgIC - Tipo de intervalo de confiança
%edit11 - LIE
%edit12- LSE

global color bgSN figOp text7 edit7 text8 text9 edit9 text10 edit10
bgIC text11 edit11 text12 edit12

figOp=figure('position',[500 100 600 300],...
  'menubar','none',...
  'name','Seleção de opções',...
  'numbertitle','off',...
  'resize','off');

bgSN = uibuttongroup(figOp,...
      'visible','on',...
      'Title','Característica do processo',...
      'HandleVisibility','off',...
      'Position',[0 .6 .3 .4],...
      'SelectionChangeFcn',@SelectionChangeFcn_typeSN);

    r1 = uicontrol(bgSN,'style','radiobutton',...
          'string','Nominal é melhor - Tipo I',...
          'Position',[10 80 150 15]);

    r2 = uicontrol(bgSN,'style','radiobutton',...
          'string','Nominal é melhor - Tipo II',...
          'Position',[10 60 150 15]);

    r3 = uicontrol(bgSN,'style','radiobutton',...
          'string','Menor é melhor',...
          'Position',[10 38 100 15]);

    r4 = uicontrol(bgSN,'style','radiobutton',...
```

```
            'string','Maior é melhor',...
            'Position',[10 15 100 15]);

bgIC = uibuttongroup(figOp,...
      'visible','off',...
      'Title','Tipo I.C',...
      'HandleVisibility','off',...
      'Position',[0 0.05 .3 .55],...
      'SelectionChangeFcn',@SelectionChangeFcn_typeIC);

    ra1 = uicontrol(bgIC,'style','radiobutton',...
          'string','Automatico',...
          'Position',[10 120 150 15]);

    ra2 = uicontrol(bgIC,'style','radiobutton',...
          'string','Z',...
          'Position',[10 100 150 15]);

    ra3 = uicontrol(bgIC,'style','radiobutton',...
          'string','t',...
          'Position',[10 80 100 15]);

    ra4 = uicontrol(bgIC,'style','radiobutton',...
          'string','Percentil',...
          'Position',[10 60 100 15]);

    ra5 = uicontrol(bgIC,'style','radiobutton',...
          'string','BC',...
          'Position',[10 40 100 15]);

    ra6 = uicontrol(bgIC,'style','radiobutton',...
          'string','BCa',...
          'Position',[10 20 100 15]);

set(bgIC,'selectedObject',[]);
set(bgSN,'selectedObject',[]);

    Text1 = uicontrol(figOp,'style','Text',...
          'string','Dados adicionais',...
          'Position',[230 280 350 15],...
         'backgroundcolor',color);

    ch1 = uicontrol(figOp,'style','checkbox',...
          'string','Função Perda',...
          'Position',[230 250 100 15],...
          'callback',@callback_ch1,...
          'backgroundcolor',color);

    ch2 = uicontrol(figOp,'style','checkbox',...
          'string','Intervalos de confiança',...
          'Position',[230 220 150 15],...
          'callback',@callback_ch2,...
          'backgroundcolor',color);

    ch3 = uicontrol(figOp,'style','checkbox',...
          'string','Capacidade de Processo',...
          'Position',[430 250 150 15],...
          'callback',@callback_ch3,...
          'backgroundcolor',color);
```

```
text7 = uicontrol(figOp,'style','Text',...
      'string','Constante K :',...
      'Position',[220 150 70 15],...
      'visible','off',...
      'backgroundcolor',color);

edit7 = uicontrol(figOp,'style','edit',...
      'Position',[290 150 100 15],...
      'visible','off',...
      'backgroundcolor', 'White'

text8 = uicontrol(figOp,'style','Text',...
      'string','Ou',...
      'Position',[290 125 100 15],...
      'visible','off',...
      'backgroundcolor',color);

text9 = uicontrol(figOp,'style','Text',...
      'string','P :',...
      'Position',[210 100 100 15],...
      'visible','off',...
      'backgroundcolor',color);

edit9 = uicontrol(figOp,'style','edit',...
     'Position',[290 100 100 15],...
     'visible','off',...
     'backgroundcolor', 'White',...
     'string','0.6');

text10 = uicontrol(figOp,'style','Text',...
      'string','Delta :',...
      'Position',[240 80 50 15],...
      'visible','off',...
      'backgroundcolor',color);

edit10 = uicontrol(figOp,'style','edit',...
     'Position',[290 80 100 15],...
     'visible','off',...
     'backgroundcolor', 'White',...
     'string','0.09');

text11 = uicontrol(figOp,'style','Text',...
      'string','LIE :',...
      'Position',[420 150 50 15],...
      'visible','off',...
      'backgroundcolor',color);

edit11 = uicontrol(figOp,'style','edit',...
      'units','pixels',...
      'Position',[470 150 100 15],...
      'visible','off',...
      'backgroundcolor', 'White',...
      'string','0.32');

text12 = uicontrol(figOp,'style','Text',...
      'string','LSE :',...
      'Position',[420 130 50 15],...
      'visible','off',...
```

```
          'backgroundcolor',color);

    edit12 = uicontrol(figOp,'style','edit',...
          'Position',[470 130 100 15],...
          'visible','off',...
          'backgroundcolor', 'White',...
          'string','0.5');

pushb=uicontrol('style','pushbutton',...
  'units','pixels',...
  'HandleVisibility','off',...
  'position',[300 10 80 40],...
  'string','Ok',...
  'callback',@callback_closeOP);
End


%--------------------------
%Fecha e manda variaveis globais
%------------------------------
function [] = callback_closeOP(varargin) da janela de opções
%VARIAVEIS
%edit7 - K
%edit9 - P
%edit10 - Delta
%edit11 - LIE
%edit12- LSE
%LIE - LIE
%LSE - LSE
%K - K
%P - P
%delta - Delta

global figOp sntype k P delta edit7 edit9 edit10 LIE LSE edit11 edit12

k=str2num(get(edit7,'string'));
P=str2num(get(edit9,'string'));
delta=str2num(get(edit10,'string'));
LIE=str2num(get(edit11,'string'));
LSE=str2num(get(edit12,'string'));

verifica=ischar(sntype);
if verifica == 0
  msgbox('Escolha a característica do processo');
  return
else
  close(figOp)
end
end

%-------------------------------------------------------------------
%Função MAINEFFECTPLOT e INTERACTIONPLOT (janela de opções graficas)
%-------------------------------------------------------------------
function [] = callback_OPgrafic(varargin)

%VARIAVEIS
%SNlinha - Resultado S/N
%amostra2 - Amostra
%ch1graf - Opçao de análise: média
%ch2graf - Opçao de análise: std
```

```
      %ch3graf - Opçao de análise: S/N
      %check5 - seleção calcular interaçao
      %listI - Lista do indice dos grupos das interações selecionadas
      %listv - Lista das interações selecionadas

      global figplotOP SNlinha amostra2 ch1graf ch2graf ch3graf check5 intOK
      listI listv g design3...
        namefat namefatshow2 nameANOM design4 conflito MIX

      ch1valueG=get(ch1graf,'value');
      ch2valueG=get(ch2graf,'value');
      ch3valueG=get(ch3graf,'value');

      amostra22=amostra2(:);

      [linhasAm,colunasAm]=size(amostra2);
      [linhasSN,colunasSN]=size(SNlinha);
      if (get(check5,'value') == 0) | (conflito == 1) | (MIX==1)
        [linhasD,colunasD]=size(design3);
      else
        [linhasD,colunasD]=size(design4);
      end

      SNresulta=SNlinha(1:linhasD);
      SNresult=SNresulta(:);

      if (get(check5,'value') == 0) | (conflito == 1) | (MIX==1)
        namefatshow=namefat(1:colunasD)
      end


      %------------------------------mean-------------------------
      if ch1valueG ==1 %mean

        for t=1:colunasD
          if (get(check5,'value') == 0) | (conflito == 1) | (MIX==1)
            fatores=repmat(design3(:,t),colunasAm,1);
          else
             fatores=repmat(design4(:,t),colunasAm,1);
          end
          grupofatmean(:,t)=fatores;
        end

        figplotmean=figure('name','Gráfico dos efeitos da média',...
          'NumberTitle','off');
        if (get(check5,'value') == 0) | (conflito == 1) | (MIX==1)
          op_name=namefatshow;
        else
          op_name=nameANOM;
        end

maineffectsplot(amostra22,grupofatmean,'parent',figplotmean,'varnames',op_n
ame)
      end

%--------------------------------std--------------------------------
      if ch2valueG ==1
        if colunasAm <2
```

```
          msgbox('O resultado do experimento deve ter no minimo 1
replicação');
        else
          figplotstd=figure('name','Gráfico dos efeitos do desvio
padrão',...
          'NumberTitle','off');
          if (get(check5,'value') == 0) | (conflito == 1) | (MIX==1)
            maineffectsplot(amostra2,design3,'statistic','std','parent',fig
      plotstd,'varnames',namefatshow)
          else
            maineffectsplot(amostra2,design4,'statistic','std','parent',fig
      plotstd,'varnames',nameANOM)
          end
        end
      end

%------------------------------s/n-----------------------------------
      if ch3valueG ==1

        for t=1:colunasD
          if (get(check5,'value') == 0) | (conflito == 1) | (MIX==1)
            fatores=repmat(design3(:,t),linhasSN,1); %Aqui pega 'linhasAm'
pela forma que esta escrito o SNlinha
          else
            fatores=repmat(design4(:,t),linhasSN,1);
          end
          grupofatsn(:,t)=fatores;
        end

        figplotsn=figure('name','Gráfico dos efeitos de S/N',...
          'NumberTitle','off');
        if (get(check5,'value') == 0) | (conflito == 1) | (MIX==1)
          op_name=namefatshow;
        else
          op_name=nameANOM;
        end

maineffectsplot(SNresult,grupofatsn,'parent',figplotsn,'varnames',op_name)
      end

      if get(check5,'value') == 1
        oo=length(listI);
        if ch1valueG ==1
          for o=1:oo
            number=listI(o);
             interacplot(o)=figure('name','Gráfico interação das médias',
'NumberTitle','off');

interactionplot(amostra2,{g{number}(:,1),g{number}(:,2)},'varnames',namefat
show2{number})
          end
        end

        if ch3valueG ==1 %S/N
          for o=1:oo
            number=listI(o);
             interacplotSN(o)=figure('name','Gráfico interação de S/N',...
             'NumberTitle','off');
```

```
interactionplot(SNresult,{g{number}(:,1),g{number}(:,2)},'varnames',namefat
show2{number})
          end
        end
      end

      close(figplotOP);

      %set(check5,'value',0);
      intOK=0;

      %---------- Limpa variavel -----------------------
      clearvars SNlinha
      end

      %--------------------
      %Janela opções ANOVA
      %--------------------
      function [] = callback_OpANOVA(varargin)

      %VARIAVEIS
      %chose1 - seleção ANOVA mean
      %chose2 - seleçaõ ANOCA S/N

      global chose1 chose2 color figANOVAtOP

      figANOVAtOP=figure('position', [300 300 200 200],...
        'menubar','none',...
        'name','ANOVA',...
        'numbertitle','off',...
        'resize','off');

      chose1=uicontrol(figANOVAtOP,...
        'style','checkbox',...
        'units','pixels',...
        'HandleVisibility','off',...
        'position',[10 160 80 40],...
        'string','Mean',...
        'BackgroundColor', color);

      chose2=uicontrol(figANOVAtOP,...
        'style','checkbox',...
        'units','pixels',...
        'HandleVisibility','off',...
        'position',[10 100 80 40],...
        'string','S/N',...
        'BackgroundColor', color);

      push1=uicontrol(figANOVAtOP,...
        'style','pushbutton',...
        'units','pixels',...
        'HandleVisibility','off',...
        'position',[10 10 80 40],...
        'string','Ok',...
        'callback',@ANOVA);

      end
```

```
%---------------------------
%Janela opções de exportação
%---------------------------
function [] =callback_OpExport(varargin)
%VARIAVEIS
%chose1exp - seleção opção da janela matriz experimento
%chose3exp - seleção opção da janela amostra
%chose4exp - seleção opção da janela ANOM
%chose5exp - seleção opção da janela ANOM S/N
%chose6exp - seleção opção da janela ANOVA
%chose7exp - seleção opção da janela ANOVA S/N
%chose8exp - seleção opção da janela Geral

global chose1exp chose3exp chose4exp chose5exp chose6exp chose7exp
chose8exp color figExptOp

figExptOp=figure('position', [300 300 300 300],...
  'menubar','none',...
  'name','Exportar dados',...
  'numbertitle','off',...
  'resize','off');

chose1exp=uicontrol(figExptOp,...
  'style','checkbox',...
  'units','pixels',...
  'HandleVisibility','off',...
  'position',[10 250 120 40],...
  'string','Matriz experimento',...
  'BackgroundColor', color);

chose8exp=uicontrol(figExptOp,...
  'style','checkbox',...
  'units','pixels',...
  'HandleVisibility','off',...
  'position',[10 210 120 40],...
  'string','Amostra',...
  'BackgroundColor', color);

chose3exp=uicontrol(figExptOp,...
  'style','checkbox',...
  'units','pixels',...
  'HandleVisibility','off',...
  'position',[10 170 120 40],...
  'string','ANOM',...
  'BackgroundColor', color);

chose4exp=uicontrol(figExptOp,...
  'style','checkbox',...
  'units','pixels',...
  'HandleVisibility','off',...
  'position',[10 130 120 40],...
  'string','ANOM SN',...
  'BackgroundColor', color);

chose5exp=uicontrol(figExptOp,...
  'style','checkbox',...
  'units','pixels',...
  'HandleVisibility','off',...
  'position',[180 250 120 40],...
```

```
  'string','ANOVA',...
  'BackgroundColor', color);

chose6exp=uicontrol(figExptOp,...
  'style','checkbox',...
  'units','pixels',...
  'HandleVisibility','off',...
  'position',[180 210 120 40],...
  'string','ANOVA SN',...
  'BackgroundColor', color);

chose7exp=uicontrol(figExptOp,...
  'style','checkbox',...
  'units','pixels',...
  'HandleVisibility','off',...
  'position',[180 170 120 40],...
  'string','Geral',...
  'BackgroundColor', color);

push1Exp=uicontrol(figExptOp,...
  'style','pushbutton',...
  'units','pixels',...
  'HandleVisibility','off',...
  'position',[10 10 80 40],...
  'string','Ok',...
  'callback',@callback_export);

push2Exp=uicontrol(figExptOp,...
  'style','pushbutton',...
  'units','pixels',...
  'HandleVisibility','off',...
  'position',[180 10 100 40],...
  'string','Selecionar todos',...
  'callback',@callback_SelectAll);
end


%----------------------------------------
%seleciona todas as opções para exportação
%----------------------------------------
function callback_SelectAll(varargin)
global chose1exp chose3exp chose4exp chose5exp chose6exp chose7exp
chose8exp
set ([chose1exp, chose3exp, chose4exp, chose5exp, chose6exp, chose7exp,
chose8exp],'value',1)
end


%---------------------
%análise das médias
%---------------------
function  ANOM(varargin)
global amostra2 tbl5 design3 check5 design4 nameANOM conflito popup2
MIX

[linhasAm,colunasAm]=size(amostra2);
amostraANOM=amostra2(:);

if (get(check5,'value') == 0) || (conflito == 1) || (MIX==1)
  [linhasD,colunasD]=size(design3);
```

```
else
  [linhasD,colunasD]=size(design4);
end

for q=1:colunasD
  if (get(check5,'value') == 0) || (conflito == 1)
    fator=repmat(design3(:,q),colunasAm,1);

  else
    if MIX==1
      fator=repmat(design3(:,q),colunasAm,1);
    else
      fator=repmat(design4(:,q),colunasAm,1);
    end
  end
  statarray{q}=grpstats(amostraANOM,fator);
  Dif(q)=max(statarray{q})-min(statarray{q});
end

[norm,desc]=sort(Dif,'descend');

for h=1:colunasD
 p=desc(1,h);
 classF(1,p)=h;
end

for h=1:colunasD-1 %seleciona o 1 da fila
  h;
  soma=classF(h);
  numero=1;
  for r=h+1:colunasD %compara com os de mais

    if Dif(h) == Dif(r) %se for igual
      soma=soma+classF(r);
      numero=numero+1;
      posi(numero)=r;
    end
  end
  if numero > 1
    cla=soma/numero;
    for w=1:numero %substitui a classificação
      if w == 1
            classF(h)=cla;
      else
            classF(posi(w))=cla;
      end
    end
  end
    soma=0;
end

for i=1:colunasD
  if i==1
    statarrayEx=statarray{i};
  else
    if get(popup2,'value') == 6
      if i==2
            statarrayEx=[{statarrayEx} {statarray{i}}];
       else
            statarrayEx=[statarrayEx statarray{i}];
```

```
      end
    else
    statarrayEx=horzcat(statarrayEx,statarray{i});
    end
  end
end

if get(popup2,'value') == 6
  statarrayEx=statarrayEx(1:colunasD);
end

if get(popup2,'value') == 6
  %acha o maior vetor
  for aux=1:colunasD
    if aux ==1
       aux2=length(statarrayEx{aux});
    else
       aux3=length(statarrayEx{aux});
      teste=(aux3>aux2);
    if teste == 1
      max_val=aux3;
    else
      max_val=aux2;
    end
    end
  end

  %preenche o menor com 'nan'
  for aux4=1:colunasD
    teste_tam=length(statarrayEx{aux4});
    %dife=max_val-teste_tam;
    if teste_tam < max_val
    aux5=teste_tam+1;
    statarrayEx{aux4}(aux5:max_val)=nan;
    end
  end

  statarrayEx=cell2mat(statarrayEx);
end

[ii,jj]=size(statarrayEx);

for g=1:ii
  nunu=num2str(g);
 rown{g}=['nivel' ' ' nunu];
end

rown{ii+1}='Delta';
rown{ii+2}='Classificação';

Exibi=vertcat(statarrayEx,Dif,classF);

for i=1:colunasD
  ind=num2str(i);
  colu{i}=['fator ' '' ind];
end

if (get(check5,'value') == 0) | (conflito == 1) | (MIX ==1)
  set(tbl5,'columnname',colu);
```

```
else
  set(tbl5,'columnname',nameANOM);
end
set(tbl5,'rowname',rown);
set(tbl5,'data',Exibi);

end


%-------------------------
%Análise das médias de S/N
%-------------------------
function ANOMSN
global design3 SNlinha tbl52 design4 check5 nameANOM conflito popup2
MIX

[linhasAm,colunasAm]=size(SNlinha);
if (get(check5,'value') == 0) | (conflito == 1) | (MIX==1)
  [linhasD,colunasD]=size(design3);
else
  [linhasD,colunasD]=size(design4);
end
amostraA=SNlinha(1:linhasD);
amostraANOM=amostraA(:);

for q=1:colunasD
  if (get(check5,'value') == 0) | (conflito == 1) | (MIX==1)
    fator=repmat(design3(:,q),linhasAm,1);
  else
    if MIX==1
    fator=repmat(design3(:,q),linhasAm,1);

    else
    fator=repmat(design4(:,q),linhasAm,1);

    end
    fator=repmat(design4(:,q),linhasAm,1);
  end
  statarray{q}=grpstats(amostraANOM,fator);
  Dif(q)=max(statarray{q})-min(statarray{q});
end

[norm,desc]=sort(Dif,'descend');

for h=1:colunasD
 p=desc(1,h);
 classF(1,p)=h;
end

for h=1:colunasD-1 %seleciona o 1 da fila
  soma=classF(h);
  numero=1;
  for r=h+1:colunasD %compara com os de mais

    if Dif(h) == Dif(r) %se for igual
      soma=soma+classF(r);
      numero=numero+1;
      posi(numero)=r;
    end
  end
```

```
  if numero > 1
    cla=soma/numero;
    for w=1:numero %substitui a classificação
    if w == 1
      classF(h)=cla;
    else
      classF(posi(w))=cla;
    end
    end
  end
    soma=0;
end

for i=1:colunasD
  if i==1
    statarrayEx=statarray{i};
  else
    if get(popup2,'value') == 6
          if i==2

            statarrayEx=[{statarrayEx} {statarray{i}}];
          else
            statarrayEx=[statarrayEx statarray{i}];
          end
    else
    statarrayEx=horzcat(statarrayEx,statarray{i});
    end
  end
end

if get(popup2,'value') == 6
  statarrayEx=statarrayEx(1:colunasD);
end

if get(popup2,'value') == 6
  %acha o maior vetor
  for aux=1:colunasD
    if aux ==1
      aux2=length(statarrayEx{aux});
    else
      aux3=length(statarrayEx{aux});
      teste=(aux3>aux2);
    if teste == 1
      max_val=aux3;
    else
      max_val=aux2;
    end
    end
  end

  %preenche o menor com 'nan'
  for aux4=1:colunasD
    teste_tam=length(statarrayEx{aux4});

    if teste_tam < max_val
       aux5=teste_tam+1;
       statarrayEx{aux4}(aux5:max_val)=nan;
    end
  end
```

```
  statarrayEx=cell2mat(statarrayEx);
end


[ii,jj]=size(statarrayEx);
for g=1:ii
  nunu=num2str(g);
 rown{g}=['nivel' ' ' nunu];
end
rown{ii+1}='Delta';
rown{ii+2}='Classificação';

Exibi=vertcat(statarrayEx,Dif,classF);

for i=1:colunasD
  ind=num2str(i);
  colu{i}=['fator ' '' ind];
end


if (get(check5,'value') == 0) | (conflito == 1) | (MIX==1)
  set(tbl52,'columnname',colu);
else
  set(tbl52,'columnname',nameANOM);
end
set(tbl52,'rowname',rown);
set(tbl52,'data',Exibi);
end


%--------------------
%Analise da variância
%--------------------
function ANOVA (varargin)
global amostra2 tbl8 tbl9 chose1 chose2 figANOVAtOP SNlinha ...
design3listv escolha listI popup1 unlock namefat conflito check5

%VARIAVEIS
% valuechose1 - Opção ANOVA
% valuechose2 - Opção ANOVA SN
% calpopup1 - numero de fatores
% listv lista das interações selecionadas
% listI lista do indice dos grupos das interações selecionadas

valuechose1=get(chose1,'value');
valuechose2=get(chose2,'value');
valpopup1=get(popup1,'value');

listind=listI;
listvec=listv;

aux8=isempty(namefat);
if aux8 == 1
  msgbox('Escolha primeiro a matrix');
  return
else
  [linhasD,colunasD]=size(design3);
  namefatshow=namefat(1:colunasD);
end
```

```
% -----------  Escolha da analise -------------------------
%cria as linhas da matriz dos vetores sozinhos, Ex:[1 0 0 0]
for f=1:valpopup1
  for r=1:valpopup1
    modelsolo{f}(r)=0;
  end
  modelsolo{f}(f)=1;
  if f==2
    modelsolo2=vertcat(modelsolo{f-1},modelsolo{f});
  elseif f >2
    modelsolo2=vertcat(modelsolo2,modelsolo{f});
  end
end

if get(check5,'value') == 1
  %pega os parametros da interaçao Ex:1x10 -> Pega 1 e pega 10
  tam=length(listind);
  for w=1:tam
    eq=strcmp(listvec{w}(3),'x');
    if eq == 1
          param1=str2num(listvec{w}(1:2));
          vazio=length(listvec{w});
          if vazio == 4
            param2=str2num(listvec{w}(4));
          else
            param2=str2num(listvec{w}(4:5));
          end
    else
    param1=str2num(listvec{w}(1));
    vazio=length(listvec{w});
       if vazio == 3
        param2=str2num(listvec{w}(3));
        else
       param2=str2num(listvec{w}(3:4));
       end
    end

%----------cria as linhas da matriz dos vetores interaçao, Ex:[1 1 0 0]
        for r=1:valpopup1
          modeli{w}(r)=0;
        end
        modeli{w}(param1)=1;
        modeli{w}(param2)=1;
        if w==1
          modeli2=modeli{w};
        elseif w==2
           modeli2=vertcat(modeli{w-1},modeli{w});
        elseif w >2
           modeli2=vertcat(modeli2,modeli{w});
        end
      end
      if conflito == 0
        MODEL=vertcat(modelsolo2, modeli2)
      else
        MODEL = modelsolo2
      end
    else
      MODEL = modelsolo2

    end
```

```
% ------- analise -----------------------
[linhasAm,colunasAm]=size(amostra2);
[linhasD,colunasD]=size(design3);
amostraANOVA=amostra2(:);
SNlinhaA=SNlinha(1:linhasD);

if valuechose1 == 1

  for t=1:colunasD
    fator=repmat(design3(:,t),colunasAm,1);
    grupofat(:,t)=fator;
  end
  [p,tblMean] =
anovan(amostraANOVA,grupofat,'display','off','model',MODEL,'varname',namefa
tshow);
  tblMean(:,4)=[];
  set(tbl8,'data',tblMean);
end

if valuechose2 == 1
  [p,tblSN] =
anovan(SNlinhaA,design3,'display','off','model',MODEL,'varname',namefatshow
);
  tblSN(:,4)=[];
  set(tbl9,'data',tblSN);

end

close(figANOVAtOP);
%---------------------- Limpar variaveis ---------------------------
clearvars modelsolo modelsolo2 modeli modeli2 MODEL listind listvec

end


%-------------------------------------------
%função de exibição opção Loss Function e I.C
%-------------------------------------------
function []=callback_ch1(varargin)
global check1 text7 edit7 text8 text9 edit9 text10 edit10
check1=get(gcbo,'value');

if check1==1
  set(text7,'visible','on');
  set(edit7,'visible','on');
  set(text8,'visible','on');
  set(text9,'visible','on');
  set(edit9,'visible','on');
  set(text10,'visible','on');
  set(edit10,'visible','on');
else
  set(text7,'visible','off');
  set(edit7,'visible','off');
  set(text8,'visible','off');
  set(text9,'visible','off');
  set(edit9,'visible','off');
  set(text10,'visible','off');
  set(edit10,'visible','off');
end
```

```
end

%------------------------------
%Verificação para execução do IC
%------------------------------
function []=callback_ch2(varargin)
global check2 bgIC
check2=get(gcbo,'value');

if check2 == 1
  set(bgIC,'visible','on');
else
  set(bgIC,'visible','off');
end

end

%--------------------------
%função de exibição opção Cp
%--------------------------
function[]=callback_ch3(varargin)
global check3 text11 edit11 text12 edit12
check3=get(gcbo,'value');

if check3==1
  set(text11,'visible','on');
  set(edit11,'visible','on');
  set(text12,'visible','on');
  set(edit12,'visible','on');
else
  set(text11,'visible','off');
  set(edit11,'visible','off');
  set(text12,'visible','off');
  set(edit12,'visible','off');
end
end


%-----------------------------------------------------
%change function que seleciona a caracteistica do S/N
%-----------------------------------------------------
function [] = SelectionChangeFcn_typeSN(varargin)
global sntype
sntype=get(gcbo,'string');
end


%-----------------------------------------------------
%change function que seleciona a caracteistica do IC
%-----------------------------------------------------
function [] = SelectionChangeFcn_typeIC(varargin)
global ictype
ictype=get(gcbo,'string');
end


%--------------------------------
%função para exportar resultados
%--------------------------------
```

```
function []=callback_export(varargin)
%VARIAVEIS
% tbl3 - Matriz experimental
% tbl99 - Amostra
% tbl5 - ANOM
% tbl52 - ANOM S/N
% tbl8 - ANOVA
% tbl9 - ANOVA S/N
% tbl6 - Geral
% chose1exp - seleção opção da janela matriz experimento
% chose3exp - seleção opção da janela amostra
% chose4exp - seleção opção da janela ANOM
% chose5exp - seleção opção da janela ANOM S/N
% chose6exp - seleção opção da janela ANOVA
% chose7exp - seleção opção da janela ANOVA S/N
% chose8exp - seleção opção da janela Geral
%dataexport - dados para exportaçao
%design -
%Columnnname - nome das colunas em ANOVA e ANOVA S/N

global dataexport design Columnnname figExptOp chose1exp chose3exp
chose4exp chose5exp chose6exp chose7exp...
chose8exp tbl5 tbl52 tbl8 tbl9 tbl99

path=uigetdir;
filename = 'Result.xlsx';
fullpath=strcat(path,'\',filename);

if get(chose1exp,'value') == 1
  xlswrite(fullpath,{'Matriz Experimento'},1,'A1');
  xlswrite(fullpath,design,1,'A2');
end

if get(chose8exp,'value') == 1
  xlswrite(fullpath,{'Amostra'},2,'A1');
  xlswrite(fullpath,get(tbl99,'data'),2,'A2');
end

if get(chose3exp,'value') == 1
  xlswrite(fullpath,{'ANOM'},3,'B1');
  xlswrite(fullpath,get(tbl5,'columnname')',3,'B2');
  xlswrite(fullpath,get(tbl5,'rowname'),3,'A3');
  xlswrite(fullpath,get(tbl5,'data'),3,'B3');
end

if get(chose4exp,'value') == 1
  xlswrite(fullpath,{'ANOM SN'},4,'B1');
  xlswrite(fullpath,get(tbl52,'columnname')',4,'B2');
  xlswrite(fullpath,get(tbl52,'rowname'),4,'A3');
  xlswrite(fullpath,get(tbl52,'data'),4,'B3');
end

if get(chose5exp,'value') == 1
  xlswrite(fullpath,{'ANOVA'},5,'A1');
  xlswrite(fullpath,get(tbl8,'data'),5,'A2');
end

if get(chose6exp,'value') == 1
  xlswrite(fullpath,{'ANOVA SN'},6,'A1');
  xlswrite(fullpath,get(tbl9,'data'),6,'A2');
```

```
end


if get(chose7exp,'value') == 1
  xlswrite(fullpath,Columnnname,7,'A1');
  xlswrite(fullpath,dataexport,7,'A2');
end

close(figExptOp);
end


%------------------------------
%função para importar resultados
%------------------------------
function []=callback_import(varargin)
%VARIAVEIS
%ed1 - campo de entrada de dados
%import - variavel de sinal para importação
%importdados - leitura de arquivo .xls

global ed1 importdados import
import=1;
[file,path] = uigetfile('*.*');
caminho=strcat(path,file);
importdados=xlsread(caminho);
set(ed1,'string','Dados importados');
end


%-----------------------------------------------
%Função de execução dos cálculos dos parâmetros
%-----------------------------------------------
function [] =callback_calc(varargin)
%Essa função foi adaptada da função
%"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

global sntype ed1 ed5 ed6 tbl4 tbl5 tbl52 tbl99 tbl6 linha ...
coluna design importdados import check1 check2 check3 check5 ...
ictype Lmedia SEBOOTFP LmediaB data SNNTBIB n a aSN ...
SNNTBI SEBOOTSR B Columnnname dataexport SNlinha amostra2 ...
Cp Cpk Cpkm SEBOOTCpkm SEBOOTCpk SEBOOTCp CpkmBOOT CpkBOOT CpBOOT...
acp acpk acpkm k P delta LIE LSE nameExp

%VARIAVEIS:
%condBOOT - condição de tamanho da amostra para executar bootstrap
%testOptions - verifica se foi escolhida caracteristica do processo
%B - numero de amostras bootstrap
%mu - valor nomnial
%opbg11 - caracteristica do processo

testOptions=isempty(sntype);
if testOptions == 1
  msgbox('Escolha a caracteristica do processo nas opções');
end

if strcmp(get(ed1,'string'),'Dados importados') == 1 %Seleciona de qual
entrada vem os dados
```

```
        amostra2=importdados
      else
        amostra2 = str2num(get(ed1,'string'))
      end

      B = str2num(get(ed5,'string'));
      mu = str2num(get(ed6,'string'));
      opbg11=sntype;

      [i,condBOOT]=size(amostra2);

      %condição para continuar o programa, ver se a amostra esta correta
      %para o design
      if i ~= linha
        msgbox('Entre com o numero de amostras correto!')
        set(check5,'value',0);
        return
      end

      for t=1:linha
        %Pega a amostra
        amostra=amostra2(t,:);

        %Retira as 'Nan' para calculos
        amostra(isnan(amostra))=[];

        %tamanho da amostra
        n=length(amostra);

        if condBOOT > 1
        [bootstat,bootsam]=bootstrp(B,@mean,amostra);

        %cria uma matriz Bamostras com as amostras bootstrap
        for i=1:n;
          for j=1:B;
          m=bootsam(i,j);
          Bamostras{t}(i,j)=amostra(m);
          end
        end

      %imprime as B amostras bootstrap ('-> em linhas, não colunas como é o
comum)
        BA{t}=Bamostras{t}';

        end

%*************************************************************************
%Calculo da MEDIA e VARIANCIA da AMOSTRA MESTRE
%*************************************************************************

        ymedia=mean(amostra);
        mediaLinha(t)=ymedia;
        sigma = var(amostra);
        varLinha(t)=sigma;


%*************************************************************************
%*******************************RAZAO SINAL-RUIDO*************************
%*************************************************************************
```

```
        for i=1:n
          v(i)=(amostra(i)-ymedia)^2;
        end

        sigmaSN=sum(v)/(n-1);

        switch opbg11


          case 'Nominal é melhor - Tipo I'
%************************************************************************
%Calculo da razao S/N para o caso nominal-e-melhor tipoI para a amostra mestre
%************************************************************************

          SNNTBI=10*log10((ymedia)^2/sigma); %sigmaSN^2
          SNlinha(t)=SNNTBI;



%************************************************************************
%Calculo da razao S/N nominal-e-melhor tipo I de cada amostra bootstrap
%************************************************************************
          if condBOOT > 1
          for i=1:B
            for j=1:n
              u(i,j)=(BA{t}(i,j)-bootstat(i))^2;
            end
            sigma1(i)=sum(u(i,:))/n;
            SNNTBIB(i)=10*log10((bootstat(i))^2/(sigma1(i))^2);
          end
          SNNTBIB;
          end


          case 'Nominal é melhor - Tipo II'
%************************************************************************
%Calculo da razao S/N de para o caso nominal-e-melhor tipoII para a amostra
mestre
%************************************************************************

          SNNTBI=-10*log10(sigma); %sigmaSN^2
          SNlinha(t)=SNNTBI;

%************************************************************************
%Calculo da razao S/N nominal-e-melhor tipo II de cada amostra boostrap
%************************************************************************

          if condBOOT > 1
          for i=1:B
            for j=1:n
              u(i,j)=(BA{t}(i,j)-bootstat(i))^2;
            end
            sigmaSN(i)=sum(u(i,:))/(n-1);
            SNNTBIB(i)=10*log10((sigmaSN(i))^2);
          end
          SNNTBIB;
          end

          case 'Menor é melhor'
%************************************************************************
```

```
%Calculo da razao S/N para o caso menor-e-melhor para a amostra mestre
%*************************************************************************

          for i=1:n
            y2(i)=amostra(i)^2;
          end
          SNNTBI=-10*log10((1/n)*sum(y2));
          SNlinha(t)=SNNTBI;


%*************************************************************************
%Calculo da razao S/N para o caso menor-e-melhor de cada amostra boostrap
%*************************************************************************

          if condBOOT > 1
          for i=1:B
            S=0;
            for j=1:n
              soma(i,j)=BA{t}(i,j)^2;
            end
            S=sum(soma(i,:));
            SNNTBIB(i)=-10*log10((1/n)*S);
          end
          SNNTBIB;
          End

          case 'Maior é melhor'
%*************************************************************************
%Calculo da razao S/N  para o caso maior-e-melhor para a amostra mestre
%*************************************************************************

          for i=1:n
            y2(i)=1/(amostra(i)^2);
          end

          SNNTBI=-10*log10((1/n)*sum(y2));
          SNlinha(t)=SNNTBI;


%*************************************************************************
%Calculo da razao S/N  para o caso maior-e-melhor de cada amostra boostrap
%*************************************************************************
          if condBOOT > 1
          for i=1:B
            S=0;
            for j=1:n
              soma(i,j)=1/(BA{t}(i,j)^2);
            end
            S=sum(soma(i,:));
            SNNTBIB(i)=-10*log10((1/n)*S);
          end
          SNNTBIB;
          end
        end

%*************************************************************************
%**********************FUNCAO PERDA DE TAGUCHI****************************
%*************************************************************************

        if check1==1
```

```
          teste=isscalar(k);

          switch opbg11

          case 'Nominal é melhor - Tipo I'
            % ***********************************************
            % ***CARACTERISTICA FUNCIONAL NOMINAL-E-MELHOR***
            % ***********************************************
            if teste == 0
              k=P/(delta^2);
            end


%*********************************************************************
%**Calculo da funcao perda media de Taguchi para a amostra mestre**
%*********************************************************************
            %Calcula a perda media para o tipo nominal-e-melhor
            Lmedia=k*(sigma+(ymedia-mu)^2);

            Lmedialinha(t)=Lmedia;

%*************************************************************
%*Funcao perda media nominal-e-melhor para cada amostra bootstrap*
%*************************************************************
            if condBOOT > 1
            for i=1:B
              for j=1:n
                  w(i,j)=(BA{t}(i,j)-bootstat(i))^2;
              end
              sigma(i)=sum(w(i,:))/n; %(Variancia referente a cada B amostra
bootstrap)
              LmediaB(i)=k*(sigma(i)+(bootstat(i)-mu)^2);%Calcula a perda
media para o tipo nominal-e-melhor cada B amostra bootstrap
            end
            end

          case 'Nominal é melhor - Tipo II'
            % ***********************************************
            % ***CARACTERISTICA FUNCIONAL NOMINAL-E-MELHOR***
            % ***********************************************

            if teste == 0
              k=P/delta^2;
            end

%*********************************************************************
%***Calculo da funcao perda media de Taguchi para a amostra mestre
%*********************************************************************
            %Calcula a perda media para o tipo nominal-e-melhor
            Lmedia=k*(sigma+(ymedia-mu)^2);

            Lmedialinha(t)=Lmedia;

%*************************************************************
%*Funcao perda media nominal-e-melhor para cada amostra bootstrap
%*************************************************************

            if condBOOT > 1
```

```
            for i=1:B
              for j=1:n
                  w(i,j)=(BA{t}(i,j)-bootstat(i))^2;
              end
              sigma(i)=sum(w(i,:))/n; %(Variancia referente a cada B amostra
bootstrap)
              LmediaB(i)=k*(sigma(i)+(bootstat(i)-mu)^2);%Calcula  a  perda
media para o tipo nominal-e-melhor cada B amostra bootstrap
            end
            end

          case 'Menor é melhor'

            mu=0;
            if teste == 0
              k=P/delta^2;
            end


%************************************************************************
%***Calculo da funcao perda media de Taguchi para o caso menor-e-melhor
para a amostra mestre
%************************************************************************
            %Calcula a perda media para o tipo menor-e-melhor
            Lmedia=k*(sigma+(ymedia)^2);

            Lmedialinha(t)=Lmedia;

%************************************************************************
%***Calculo da funcao perda media de Taguchi de cada amostra boostrap***
%************************************************************************

            if condBOOT > 1
            for i=1:B
              for j=1:n
                  w(i,j)=(BA{t}(i,j)-bootstat(i))^2;
              end
              sigma(i)=sum(w(i,:))/n; %Variancia referente a cada B amostra
bootstrap
              LmediaB(i)=k*(sigma(i)+(bootstat(i))^2);%Calcula a perda media
para o tipo menor-e-melhor para cada B amostra bootstrap
            end
            end

          case 'Maior é melhor'

            if teste == 0
              k=(P)*(delta^2);
            end

%************************************************************************
%***Calculo  da  funcao  perda  media  de  Taguchi  para  a  amostra  mestre***
%************************************************************************

            for i=1:n
              w(i)=(1/(amostra(i))^2);
            end

            SOMA=sum(w);
```

```
            %Calcula a perda media para o tipo maior-e-melhor
            Lmedia=k*(1/n)*SOMA;

            Lmedialinha(t)=Lmedia;

%*****************************************************************
%*Funcao  perda  media  maior-e-melhor  para  cada  amostra  bootstrap*
%*****************************************************************

            if condBOOT > 1
            for i=1:B
              INVERSOY=0;
              for j=1:n
                  INVERSOYB(j)=1/BA{t}(i,j)^2;
              end
              LmediaB(i)=k*(1/n)*sum(INVERSOYB);
            end
            end
          end
        end

%*************************************************************************
%*************************************************************************
%************************INDICES Cp, Cpk E Cpkm***********************
%*************************************************************************
%*************************************************************************

      if check3 ==1

        if condBOOT > 1
        %**Teste de normalidade para Amostra mestre**
        if n>=4
          hdoCP=lillietest(amostra,0.01);
        else
          AmostraUnica=amostra;
          Count=length(AmostraUnica);
          for i=2:B+1
                for j=1:n
                  Count=Count+1;
                  AmostraUnica(1,Count)=BA{t}(i-1,j);
                end
          hdoCP=lillietest(AmostraUnica);
          end
        end
        end

        %*************************************************************
        %***Calculo do indice Cp, Cpk e Cpkm para a amostra mestre***
        %*************************************************************
        if condBOOT > 1
        if hdoCP==0
          stddev=std(amostra);
          ymedia=mean(amostra);

          switch sntype
          case 'Nominal é melhor - Tipo I'
            Cp=(LSE-LIE)/(6*stddev);
            Cps=(LSE-ymedia)/(3*std(amostra));
            Cpi=(ymedia-LIE)/(3*std(amostra));
            Cpk=min([Cpi Cps]);
```

```
            T=(LSE+LIE)/2;
            V=(T-ymedia)/std(amostra);
            Cpkm=Cp/(sqrt((1+V^2)));

            CpExibi(t)=Cp;
            CpkExibi(t)=Cpk;
            CpkmExibi(t)=Cpkm;

          case 'Nominal é melhor - Tipo II'
            Cp=(LSE-LIE)/(6*stddev);
            Cps=(LSE-ymedia)/(3*std(amostra));
            Cpi=(ymedia-LIE)/(3*std(amostra));
            Cpk=min([Cpi Cps]);
            T=(LSE+LIE)/2;
            V=(T-ymedia)/std(amostra);
            Cpkm=Cp/(sqrt((1+V^2)));

            CpExibi(t)=Cp;
            CpkExibi(t)=Cpk;
            CpkmExibi(t)=Cpkm;

          case 'Menor é melhor'
            mu=0;
            LIE=0;
            Cp=(LSE-LIE)/(6*stddev);
            Cps=(LSE-ymedia)/(3*std(amostra));
            Cpi=(ymedia-LIE)/(3*std(amostra));
            Cpk=min([Cpi Cps]);
            T=(LSE+LIE)/2;
            V=(T-ymedia)/std(amostra);
            Cpkm=Cp/(sqrt((1+V^2)));

            CpExibi(t)=Cp;
            CpkExibi(t)=Cpk;
            CpkmExibi(t)=Cpkm;

          case 'Maior é melhor'
            msgbox('Nao se aplica o calculo de Cp, Cpk e Cpkm pois nao se
conhece o LSE')
          end

          if condBOOT > 1
%*************************************************************'
%*Calculo do indice Cp, Cpk e Cpkm para cada amostra bootstrap*'
%*************************************************************'
          for i=1:B
          media=mean(BA{t}(i,:));
          switch sntype
            case 'Nominal é melhor - Tipo I'
            CpBOOT(i)=(LSE-LIE)/(6*std(BA{t}(i,:)));
            Cpi(i)=(media-LIE)/(3*std(BA{t}(i,:)));
            Cps(i)=(LSE-media)/(3*std(BA{t}(i,:)));
            CpkBOOT(i)=min([Cpi(i) Cps(i)]);
            T=(LSE+LIE)/2;
            V=(T-ymedia)/std(BA{t}(i,:));
            CpkmBOOT(i)=CpBOOT(i)/sqrt((1+V^2));

            case 'Nominal é melhor - Tipo II'
            CpBOOT(i)=(LSE-LIE)/(6*std(BA{t}(i,:)));
            Cpi(i)=(media-LIE)/(3*std(BA{t}(i,:)));
```

```
            Cps(i)=(LSE-media)/(3*std(BA{t}(i,:)));
            CpkBOOT(i)=min([Cpi(i) Cps(i)]);
            T=(LSE+LIE)/2;
            V=(T-ymedia)/std(BA{t}(i,:));
            CpkmBOOT(i)=CpBOOT(i)/sqrt((1+V^2));

            case 'Menor é melhor'
            CpBOOT(i)=(LSE-LIE)/(6*std(BA{t}(i,:)));
            Cpi(i)=(media-LIE)/(3*std(BA{t}(i,:)));
            Cps(i)=(LSE-media)/(3*std(BA{t}(i,:)));
            CpkBOOT(i)=min([Cpi(i) Cps(i)]);
            T=(LSE+LIE)/2;
            V=(T-ymedia)/std(BA{t}(i,:));
            CpkmBOOT(i)=CpBOOT(i)/sqrt((1+V^2));
          end
          end
        end

        else
          CpExibi(t)=NaN;
          CpkExibi(t)=NaN;
          CpkmExibi(t)=NaN;
        end
      end
      end


%************************************************************************
%****CALCULO DE PARAMETROD PARA INTERVALOS DE CONFIANCA BOOTSTRAP********
%************************************************************************

        if condBOOT > 1
        if check2 ==1

          if check1 == 1

%**************BOOTSTRAP APLICADO A FUNCAO PERDA DE TAGUCHI*************

          %Calculo do desvio padrao bootstrap (Erro padrao)
          MediaLmediaB=mean(LmediaB);
          for i=1:B
            SEBOOT(i)=(LmediaB(i)-MediaLmediaB)^2;
          end
          SEBOOTFP=sqrt(sum(SEBOOT)/(B-1));

%------------TESTES DE NORMALIDADE E CURTOSE FUNÇÃO PERDA -------------

          %****Teste de normalidade de Lilliefors****
          h=lillietest(LmediaB); %se igual a 1, rejeita-se que é da familia
normal a 5%

          %****Teste de normalidade de Kolmogorov-Smirnov****
          ks=kstest(LmediaB);

%**CALCULO DA CURTOSE PARA A DISTRIBUICAO BOOTSTRAP DA FUNCAO PERDA**
          CurtoseFperda=skewness(LmediaB);

          %***CALCULO DO VICIO***
          MediaLmediaB=mean(LmediaB);
```

```
            VICIO=abs(MediaLmediaB-Lmedia);
            for i=1:B
              VBOOT(i)=(LmediaB(i)-Lmedia)^2;
            end
            DesvPadVICIO=sqrt(sum(VBOOT)/(B-1));

%--------- calculo de a para 'a' para função perda -------------------
            switch opbg11

              case 'Nominal é melhor - Tipo I'

                Ba=amostra(:,2:n);
                nB=length(Ba);
                ymediaA=mean(Ba);
                for i=1:nB
                    s(i)=(Ba(i)-ymediaA)^2;
                end
                sigmaA=sum(s)/nB;
                LAmediaA(1)=k*(sigmaA+(ymediaA-mu)^2);
                for i=1:n-1
                      Ba(:,i)=amostra(i);
                      ymediaA=mean(Ba);
                for j=1:nB
                  s(j)=(Ba(j)-ymediaA)^2;
                end
                sigmaA=sum(s)/n;
                LAmediaA(i+1)=k*(sigmaA+(ymediaA-mu)^2);
                end
                LamediaAmean=mean(LAmediaA);
                nLa=length(LAmediaA);
                for l=1:nLa
                      num(l)=(LamediaAmean-LAmediaA(l))^3;
                      den(l)=(LamediaAmean-LAmediaA(l))^2;
                end
                a=sum(num)/(6*(sum(den))^(3/2));

              case 'Nominal é melhor - Tipo II'

                Ba=amostra(:,2:n);
                nB=length(Ba);
                ymediaA=mean(Ba);
                for i=1:nB
                      s(i)=(Ba(i)-ymediaA)^2;
                end
                sigmaA=sum(s)/nB;
                LAmediaA(1)=k*(sigmaA+(ymediaA-mu)^2);
                for i=1:n-1
                      Ba(:,i)=amostra(i);
                      ymediaA=mean(Ba);
                for j=1:nB
                  s(j)=(Ba(j)-ymediaA)^2;
                end
                sigmaA=sum(s)/n;
                LAmediaA(i+1)=k*(sigmaA+(ymediaA-mu)^2);
                end
                LamediaAmean=mean(LAmediaA);
                nLa=length(LAmediaA);
                for l=1:nLa
                      num(l)=(LamediaAmean-LAmediaA(l))^3;
                      den(l)=(LamediaAmean-LAmediaA(l))^2;
```

```
        end
        a=sum(num)/(6*(sum(den))^(3/2));

      case 'Menor é melhor'

        Ba=amostra(:,2:n);
        nB=length(Ba);
        ymediaA=mean(Ba);
        for i=1:nB
              s(i)=(Ba(i)-ymediaA)^2;
        end
        sigmaA=sum(s)/nB;
        LAmediaA(1)=k*(sigmaA+(ymediaA-mu)^2);
        for i=1:n-1
              Ba(:,i)=amostra(i);
              ymediaA=mean(Ba);
        for j=1:nB
          s(j)=(Ba(j)-ymediaA)^2;
        end
        sigmaA=sum(s)/n;
        LAmediaA(i+1)=k*(sigmaA+(ymediaA-mu)^2);
        end
        LAmediaA;
        LamediaAmean=mean(LAmediaA);
        nLa=length(LAmediaA);
        for l=1:nLa
              num(l)=(LamediaAmean-LAmediaA(l))^3;
              den(l)=(LamediaAmean-LAmediaA(l))^2;
        end
        a=sum(num)/(6*(sum(den))^(3/2));

      case'Maior é melhor'

        Ba=amostra(:,2:n);
        nB=length(Ba);
        for s=1:nB
              OutraLmedia(s)=k*(1/n)*sum(1/(Ba(s))^2);
        end
        LAmediaA(1)=sum(OutraLmedia);
        for z=1:n-1
        Ba(:,z)=amostra(z);
              for s=1:nB
                OutraLmedia(s)=k*(1/n)*sum(1/(Ba(s))^2);
              end
              LAmediaA(z+1)=sum(OutraLmedia);
        end
        LAmediaA;
        LamediaAmean=mean(LAmediaA);
        nLa=length(LAmediaA);

        for l=1:nLa
              num(l)=(LamediaAmean-LAmediaA(l))^3;
              den(l)=(LamediaAmean-LAmediaA(l))^2;
        end
        a=sum(num)/(6*(sum(den))^(3/2));

    end
    end

%------------------- Calculode a para indices Cp Cpk Cpkm--------
```

```
if check3 == 1
  if hdoCP == 0 %teste normalidade
  switch sntype
    case 'Nominal é melhor - Tipo I'
    %*************************
    %***Calculo de a para Cp***
    %*************************
    BaCp=amostra(:,2:n);
    nCp=length(BaCp);
    ymediaACp=mean(BaCp);
    for i=1:nCp
      sCp(i)=(BaCp(i)-ymediaACp)^2;
    end
    sigmaACp=sqrt(sum(sCp)/nCp);
    Cpa(1)=(LSE-LIE)/6*sigmaACp;
    for i=1:n-1
      BaCp(:,i)=amostra(i);
      ymediaACp=mean(BaCp);
      for j=1:nCp
        sCp(j)=(BaCp(j)-ymediaACp)^2;
      end
      sigmaACp=sum(sCp)/nCp;
      Cpa(i+1)=(LSE-LIE)/6*sigmaACp;
    end
    Cpa;
    Cpamean=mean(Cpa);
    nCpa=length(Cpa);
    for l=1:nCpa
      num(l)=(Cpamean-Cpa(l))^3;
      den(l)=(Cpamean-Cpa(l))^2;
    end
    acp=sum(num)/(6*(sum(den))^(3/2));


    %*************************
    %***Calculo de a para Cpk***
    %*************************
    BaCpk=amostra(:,2:n);
    nCpk=length(BaCpk);
    ymediaACpk=mean(BaCpk);
    for i=1:nCpk
      sCpk(i)=(BaCpk(i)-ymediaACpk)^2;
    end
    sigmaACpk=sqrt(sum(sCpk)/nCpk);
    Cpsa=(LSE-ymediaACpk)/(3*sigmaACpk);
    Cpia=(ymediaACpk-LIE)/(3*sigmaACpk);
    Cpka(1)=min([Cpia Cpsa]);
    for i=1:n-1
      BaCpk(:,i)=amostra(i);
      ymediaACpk=mean(BaCpk);
      for j=1:nCpk
        sCpk(j)=(BaCpk(j)-ymediaACpk)^2;
      end
      sigmaACpk=sum(sCpk)/nCpk;
      Cpsa(i+1)=(LSE-ymediaACpk)/(3*sigmaACpk);
      Cpia(i+1)=(ymediaACpk-LIE)/(3*sigmaACpk);
      Cpka(i+1)=min([Cpia(i+1) Cpsa(i+1)]);
    end
    Cpka;
    Cpkamean=mean(Cpka);
```

```
nCpka=length(Cpka);
for l=1:nLa
  num(l)=(Cpkamean-Cpka(l))^3;
  den(l)=(Cpkamean-Cpka(l))^2;
end
acpk=sum(num)/(6*(sum(den))^(3/2));


%****************************
%***Calculo de a para Cpkm***
%****************************
BaCpkm=amostra(:,2:n);
nCpkm=length(BaCpkm);
ymediaACpkm=mean(BaCpkm);
for i=1:nCpkm
  sCpkm(i)=(BaCpkm(i)-ymediaACpkm)^2;
end
sigmaACpkm=sqrt(sum(sCpkm)/nCpkm);
OutroCp=(LSE-LIE)/(6*sigmaACpkm);
T=(LSE+LIE)/2;
V(1)=(T-ymediaACpkm)/sigmaACpkm;
Cpkma(1)=OutroCp/(sqrt((1+V(1)^2)));
for i=1:n-1
  BaCpkm(:,i)=amostra(i);
  ymediaACpkm=mean(BaCpkm);
  for j=1:nCpkm
    sCpkm(j)=(BaCpkm(j)-ymediaACpkm)^2;
  end
  sigmaACpkm=sum(sCpkm)/nCpkm;
  OutroCp(i+1)=(LSE-LIE)/6*sigmaACpkm;
  T=(LSE+LIE)/2;
  V(i+1)=(T-ymediaACpkm)/sigmaACpkm;
Cpkma(i+1)=OutroCp(i+1)./(sqrt((1+V(i+1).^2)));
end
Cpkma;
Cpkmamean=mean(Cpkma);
nCpkma=length(Cpkma);
for l=1:nCpkma
  num(l)=(Cpkmamean-Cpkma(l))^3;
  den(l)=(Cpkmamean-Cpkma(l))^2;
end
acpkm=sum(num)/(6*(sum(den))^(3/2));
case 'Nominal é melhor - Tipo II'
%**************************
%***Calculo de a para Cp***
%**************************
BaCp=amostra(:,2:n);
nCp=length(BaCp);
ymediaACp=mean(BaCp);
for i=1:nCp
  sCp(i)=(BaCp(i)-ymediaACp)^2;
end
sigmaACp=sqrt(sum(sCp)/nCp);
Cpa(1)=(LSE-LIE)/6*sigmaACp;
for i=1:n-1
  BaCp(:,i)=amostra(i);
  ymediaACp=mean(BaCp);
  for j=1:nCp
    sCp(j)=(BaCp(j)-ymediaACp)^2;
  end
  sigmaACp=sum(sCp)/nCp;
```

```
  Cpa(i+1)=(LSE-LIE)/6*sigmaACp;
end
Cpa;
Cpamean=mean(Cpa);
nCpa=length(Cpa);
for l=1:nCpa
  num(l)=(Cpamean-Cpa(l))^3;
  den(l)=(Cpamean-Cpa(l))^2;
end
acp=sum(num)/(6*(sum(den))^(3/2));


%**************************
%***Calculo de a para Cpk***
%**************************
BaCpk=amostra(:,2:n);
nCpk=length(BaCpk);
ymediaACpk=mean(BaCpk);
for i=1:nCpk
  sCpk(i)=(BaCpk(i)-ymediaACpk)^2;
end
sigmaACpk=sqrt(sum(sCpk)/nCpk);
Cpsa=(LSE-ymediaACpk)/(3*sigmaACpk);
Cpia=(ymediaACpk-LIE)/(3*sigmaACpk);
Cpka(1)=min([Cpia Cpsa]);
for i=1:n-1
  BaCpk(:,i)=amostra(i);
  ymediaACpk=mean(BaCpk);
  for j=1:nCpk
    sCpk(j)=(BaCpk(j)-ymediaACpk)^2;
  end
  sigmaACpk=sum(sCpk)/nCpk;
  Cpsa(i+1)=(LSE-ymediaACpk)/(3*sigmaACpk);
  Cpia(i+1)=(ymediaACpk-LIE)/(3*sigmaACpk);
  Cpka(i+1)=min([Cpia(i+1) Cpsa(i+1)]);
end
Cpka;
Cpkamean=mean(Cpka);
nCpka=length(Cpka);
for l=1:nLa
  num(l)=(Cpkamean-Cpka(l))^3;
  den(l)=(Cpkamean-Cpka(l))^2;
end
acpk=sum(num)/(6*(sum(den))^(3/2));


%****************************
%***Calculo de a para Cpkm***
%****************************
BaCpkm=amostra(:,2:n);
nCpkm=length(BaCpkm);
ymediaACpkm=mean(BaCpkm);
for i=1:nCpkm
  sCpkm(i)=(BaCpkm(i)-ymediaACpkm)^2;
end
sigmaACpkm=sqrt(sum(sCpkm)/nCpkm);
OutroCp=(LSE-LIE)/(6*sigmaACpkm);
T=(LSE+LIE)/2;
V(1)=(T-ymediaACpkm)/sigmaACpkm;
Cpkma(1)=OutroCp/(sqrt((1+V(1)^2)));
for i=1:n-1
```

```
  BaCpkm(:,i)=amostra(i);
  ymediaACpkm=mean(BaCpkm);
  for j=1:nCpkm
    sCpkm(j)=(BaCpkm(j)-ymediaACpkm)^2;
  end
  sigmaACpkm=sum(sCpkm)/nCpkm;
  OutroCp(i+1)=(LSE-LIE)/6*sigmaACpkm;
  T=(LSE+LIE)/2;
  V(i+1)=(T-ymediaACpkm)/sigmaACpkm;
       Cpkma(i+1)=OutroCp(i+1)./(sqrt((1+V(i+1).^2)));
end
Cpkma;
Cpkmamean=mean(Cpkma);
nCpkma=length(Cpkma);
for l=1:nCpkma
  num(l)=(Cpkmamean-Cpkma(l))^3;
  den(l)=(Cpkmamean-Cpkma(l))^2;
end
acpkm=sum(num)/(6*(sum(den))^(3/2));
case 'Menor é melhor'

%**************************
%***Calculo de a para Cp***
%**************************
BaCp=amostra(:,2:n);
nCp=length(BaCp);
ymediaACp=mean(BaCp);
for i=1:nCp
  sCp(i)=(BaCp(i)-ymediaACp)^2;
end
sigmaACp=sqrt(sum(sCp)/nCp);
Cpa(1)=(LSE-LIE)/6*sigmaACp;
for i=1:n-1
  BaCp(:,i)=amostra(i);
  ymediaACp=mean(BaCp);
  for j=1:nCp
    sCp(j)=(BaCp(j)-ymediaACp)^2;
  end
  sigmaACp=sum(sCp)/nCp;
  Cpa(i+1)=(LSE-LIE)/6*sigmaACp;
end
Cpa;
Cpamean=mean(Cpa);
nCpa=length(Cpa);
for l=1:nCpa
  num(l)=(Cpamean-Cpa(l))^3;
  den(l)=(Cpamean-Cpa(l))^2;
end
acp=sum(num)/(6*(sum(den))^(3/2));


%**************************
%***Calculo de a para Cpk***
%**************************
BaCpk=amostra(:,2:n);
nCpk=length(BaCpk);
ymediaACpk=mean(BaCpk);
for i=1:nCpk
  sCpk(i)=(BaCpk(i)-ymediaACpk)^2;
end
sigmaACpk=sqrt(sum(sCpk)/nCpk);
```

```
Cpsa=(LSE-ymediaACpk)/(3*sigmaACpk);
Cpia=(ymediaACpk-LIE)/(3*sigmaACpk);
Cpka(1)=min([Cpia Cpsa]);
for i=1:n-1
  BaCpk(:,i)=amostra(i);
  ymediaACpk=mean(BaCpk);
  for j=1:nCpk
    sCpk(j)=(BaCpk(j)-ymediaACpk)^2;
  end
  sigmaACpk=sum(sCpk)/nCpk;
  Cpsa(i+1)=(LSE-ymediaACpk)/(3*sigmaACpk);
  Cpia(i+1)=(ymediaACpk-LIE)/(3*sigmaACpk);
  Cpka(i+1)=min([Cpia(i+1) Cpsa(i+1)]);
end
Cpka;
Cpkamean=mean(Cpka);
nCpka=length(Cpka);
for l=1:nCpka
  num(l)=(Cpkamean-Cpka(l))^3;
  den(l)=(Cpkamean-Cpka(l))^2;
end
acpk=sum(num)/(6*(sum(den))^(3/2));


%*****************************
%***Calculo de a para Cpkm***
%*****************************
BaCpkm=amostra(:,2:n);
nCpkm=length(BaCpkm);
ymediaACpkm=mean(BaCpkm);
for i=1:nCpkm
  sCpkm(i)=(BaCpkm(i)-ymediaACpkm)^2;
end
sigmaACpkm=sqrt(sum(sCpkm)/nCpkm);
OutroCp=(LSE-LIE)/(6*sigmaACpkm);
T=(LSE+LIE)/2;
V(1)=(T-ymediaACpkm)/sigmaACpkm;
Cpkma(1)=OutroCp/(sqrt((1+V(1)^2)));
for i=1:n-1
  BaCpkm(:,i)=amostra(i);
  ymediaACpkm=mean(BaCpkm);
  for j=1:nCpkm
    sCpkm(j)=(BaCpkm(j)-ymediaACpkm)^2;
  end
  sigmaACpkm=sum(sCpkm)/nCpkm;
  OutroCp(i+1)=(LSE-LIE)/6*sigmaACpkm;
  T=(LSE+LIE)/2;
  V(i+1)=(T-ymediaACpkm)/sigmaACpkm;
        Cpkma(i+1)=OutroCp(i+1)./(sqrt((1+V(i+1).^2)));
end
Cpkma;
Cpkmamean=mean(Cpkma);
nCpkma=length(Cpkma);
for l=1:nCpkma
  num(l)=(Cpkmamean-Cpkma(l))^3;
  den(l)=(Cpkmamean-Cpkma(l))^2;
end
acpkm=sum(num)/(6*(sum(den))^(3/2));

case 'Maior é melhor'
```

```
            end
          end


          if hdoCP == 0

%Calculo do DESVIO PADRÂO boot para os indices Cp, Cpk e Cpkm (Erro padrao)

          %*Desvio padrao bootstrap para Cp *
          MediaCpBOOT=mean(CpBOOT);
          for i=1:B
            SEBOOT1(i)=(CpBOOT(i)-MediaCpBOOT)^2;
          end
          SEBOOTCp=sqrt(sum(SEBOOT1)/(B-1));

          %*Desvio padrao bootstrap para Cpk *
          MediaCpkBOOT=mean(CpkBOOT);
          for i=1:B
            SEBOOT(i)=(CpkBOOT(i)-MediaCpkBOOT)^2;
          end
          SEBOOTCpk=sqrt(sum(SEBOOT)/(B-1));

          %Desvio padrao bootstrap para Cpkm *
          MediaCpkmBOOT=mean(CpkmBOOT);
          for i=1:B
            SEBOOT2(i)=(CpkmBOOT(i)-MediaCpkmBOOT)^2;
          end
          SEBOOTCpkm=sqrt(sum(SEBOOT2)/(B-1));

          %----- CALCULO DO VICIO -------------------

          %***CALCULO DO VICIO Cp***
           VICIOCp=abs(mean(CpBOOT)-Cp);
           for i=1:B
             VBOOT1(i)=(CpBOOT(i)-Cp).^2;
           end
           DesvPadVICIO1=sqrt(sum(VBOOT1)/(B-1));

           %***CALCULO DO VICIO Cpk***
           VICIOCpk=abs(MediaCpkBOOT-Cpk);
           for i=1:B
             VBOOT(i)=(CpkBOOT(i)-Cpk)^2;
           end
           DesvPadVICIOCpk=sqrt(sum(VBOOT)/(B-1));

           %***CALCULO DO VICIO Cpkm***
           VICIOCpk=abs(MediaCpkBOOT-Cpk);
           for i=1:B
             VBOOT(i)=(CpkBOOT(i)-Cpk)^2;
           end
           DesvPadVICIOCpk=sqrt(sum(VBOOT)/(B-1));

          end
          end


%------------TESTES DE NORMALIDADE E ASSIMETRIA s/n ----------------------

            %****Teste de normalidade de Lilliefors****
```

```
        hSN=lillietest(SNNTBIB);

        %****Teste de normalidade de Kolmogorov-Smirnov****
        ksSN=kstest(SNNTBIB);

        %**CALCULO DA CURTOSE PARA A DISTRIBUICAO S/N
        CurtoseSN=skewness(SNNTBIB);

        %Calculo do desvio padrao bootstrap (Erro padrao)
        MediaSNNTBIB=mean(SNNTBIB);
        for i=1:B
          SEBOOT(i)=(SNNTBIB(i)-MediaSNNTBIB)^2;
        end
        SEBOOTSR=sqrt(sum(SEBOOT)/(B-1));

        %***CALCULO DO VICIO***
        VICIOSN=abs(MediaSNNTBIB-SNNTBI);
        for i=1:B
          VBOOT(i)=(SNNTBIB(i)-SNNTBI)^2;
        end
        DesvPadVICIO=sqrt(sum(VBOOT)/(B-1));

%---------calculo de a para 'a' para S/N --------------------

        switch opbg11

          case 'Nominal é melhor - Tipo I'

          BaSN=amostra(:,2:n);
          nBSN=length(BaSN);
          ymediaASN=mean(BaSN);
          for iSN=1:nBSN
              sSN(iSN)=(BaSN(iSN)-ymediaASN)^2;
                          end
          S=sum(sSN);
          sigmaASN=sum(sSN)/nBSN;

                          SNNTBIa(1)=10*log10((ymediaASN)^2/sigmaASN^2);
          for i=1:n-1
            BaSN(:,i)=amostra(i);
            ymediaASN=mean(BaSN);
            for j=1:nBSN
              sSN(j)=(BaSN(j)-ymediaASN)^2;
            end
            sigmaASN=sum(sSN)/n;
            SNNTBIa(i+1)=10*log10((ymediaASN)^2/sigmaASN^2);
          end
          SNNTBIa;
          SNNTBIamean=mean(SNNTBIa);
          nLaSN=length(SNNTBIa);
          for l=1:nLaSN
            numSN(l)=(SNNTBIamean-SNNTBIa(l))^3;
            denSN(l)=(SNNTBIamean-SNNTBIa(l))^2;
          end
          aSN=sum(numSN)/(6*(sum(denSN))^(3/2));

          case'Nominal é melhor - Tipo II'

          BaSN=amostra(:,2:n);
          nBSN=length(BaSN);
```

```
ymediaASN=mean(BaSN);
for iSN=1:nBSN
  sSN(iSN)=(BaSN(iSN)-ymediaASN)^2;
end
S=sum(sSN);
sigmaASN=sum(sSN)/nBSN;
SNNTBIa(1)=10*log10(sigmaASN^2);
for i=1:n-1
  BaSN(:,i)=amostra(i);
  ymediaASN=mean(BaSN);
  for j=1:nBSN
    sSN(j)=(BaSN(j)-ymediaASN)^2;
  end
  sigmaASN=sum(sSN)/n;
  SNNTBIa(i+1)=10*log10(sigmaASN^2);
end
SNNTBIa;
SNNTBIamean=mean(SNNTBIa);
nLaSN=length(SNNTBIa);
for l=1:nLaSN
  numSN(l)=(SNNTBIamean-SNNTBIa(l))^3;
  denSN(l)=(SNNTBIamean-SNNTBIa(l))^2;
end
aSN=sum(numSN)/(6*(sum(denSN))^(3/2));

case 'Menor é melhor'

BaSN=amostra(:,2:n);
nBSN=length(BaSN);
for iSN=1:nBSN
  y2SNSB(iSN)=BaSN(iSN)^2;
end
SSNSB=sum(y2SNSB);
SNNTBIa(1)=-10*log10((1/nBSN)*SSNSB);
for i=1:n-1
  BaSN(:,i)=amostra(i);
  for j=1:nBSN
    y2SNSB(j)= BaSN(j)^2;
  end
  SSNSBa=sum(y2SNSB);
  SNNTBIa(i+1)=-10*log10((1/nBSN)*SSNSBa);
end
SNNTBIa;
SNNTBIamean=mean(SNNTBIa);
nLaSN=length(SNNTBIa);
for l=1:nLaSN
  numSN(l)=(SNNTBIamean-SNNTBIa(l))^3;
  denSN(l)=(SNNTBIamean-SNNTBIa(l))^2;
end
aSN=sum(numSN)/(6*(sum(denSN))^(3/2));

case 'Maior é melhor'

BaSN=amostra(:,2:n);
nBSN=length(BaSN);
for iSN=1:nBSN
  y2SNSB(iSN)=1/(BaSN(iSN)^2);
end
SSNSB=sum(y2SNSB);
SNNTBIa(1)=-10*log10((1/nBSN)*SSNSB);
for i=1:n-1
```

```
            BaSN(:,i)=amostra(i);
            for j=1:nBSN
              y2SNSB(j)=1/(BaSN(iSN)^2);
            end
            SSNSBa=sum(y2SNSB);
            SNNTBIa(i+1)=-10*log10((1/nBSN)*SSNSBa);
          end
          SNNTBIa;
          SNNTBIamean=mean(SNNTBIa);
          nLaSN=length(SNNTBIa);
          for l=1:nLaSN
            numSN(l)=(SNNTBIamean-SNNTBIa(l))^3;
            denSN(l)=(SNNTBIamean-SNNTBIa(l))^2;
          end
          a=sum(numSN)/(6*(sum(denSN))^(3/2));
        end


%*********************************************************************
%**********CALCULO DO INTERVALOS DE CONFIANCA *********************
%*********************************************************************

      switch ictype

      case 'Automatico'

      [ICSNi95(t), ICSNs95(t),ICSNi99(t), ICSNs99(t)]= ICpSN;

      if check1 == 1
            [ICLi95(t), ICLs95(t),ICLi99(t), ICLs99(t)]= ICpLoss;
      end

      if hdoCP==0
            [ICLi95CP(t), ICLs95CP(t), ICLi99CP(t), ICLs99CP(t)]=ICtCP;
            [ICLi95CPk(t), ICLs95CPk(t), ICLi99CPk(t),
      ICLs99CPk(t)]=ICtCPk;
            [ICLi95CPkm(t), ICLs95CPkm(t), ICLi99CPkm(t),
      ICLs99CPkm(t)]=ICtCPkm;
      end


      case 'Z'

      [ICSNi95(t), ICSNs95(t),ICSNi99(t), ICSNs99(t)]=ICZSN;

      if check1 == 1
            [ICLi95(t), ICLs95(t),ICLi99(t), ICLs99(t)]= ICZLoss;
      end

      if check3 == 1
      if hdoCP ==0
            [ICLi95CP(t), ICLs95CP(t), ICLi99CP(t), ICLs99CP(t)]=ICZCP;
            [ICLi95CPk(t), ICLs95CPk(t), ICLi99CPk(t),
      ICLs99CPk(t)]=ICZCPk;
            [ICLi95CPkm(t), ICLs95CPkm(t), ICLi99CPkm(t),
      ICLs99CPkm(t)]=ICZCPkm;
      end
      end
```

```
case 't'

[ICSNi95(t), ICSNs95(t),ICSNi99(t), ICSNs99(t)]=ICtSN;

if check1 == 1
      [ICLi95(t), ICLs95(t),ICLi99(t), ICLs99(t)]= ICtLoss;
end

if check3 == 1
if hdoCP==0
      [ICLi95CP(t), ICLs95CP(t), ICLi99CP(t), ICLs99CP(t)]=ICtCP;
      [ICLi95CPk(t), ICLs95CPk(t), ICLi99CPk(t),
ICLs99CPk(t)]=ICtCPk;
            [ICLi95CPkm(t), ICLs95CPkm(t), ICLi99CPkm(t),
      ICLs99CPkm(t)]=ICtCPkm;
end
end

case 'Percentil'

[ICSNi95(t), ICSNs95(t),ICSNi99(t), ICSNs99(t)]= ICpSN;

if check1 == 1
[ICLi95(t), ICLs95(t),ICLi99(t), ICLs99(t)]= ICpLoss;
end

if check3 == 1
      if hdoCP==0
            [ICLi95CP(t), ICLs95CP(t), ICLi99CP(t),
      ICLs99CP(t)]=ICpCP;
            [ICLi95CPk(t), ICLs95CPk(t), ICLi99CPk(t),
      ICLs99CPk(t)]=ICpCPk;
            [ICLi95CPkm(t), ICLs95CPkm(t), ICLi99CPkm(t),
      ICLs99CPkm(t)]=ICpCPkm;
      end
end

case 'BC'

[ICSNi95(t), ICSNs95(t),ICSNi99(t), ICSNs99(t)]= ICBCSN;

if check1 == 1
      [ICLi95(t), ICLs95(t),ICLi99(t), ICLs99(t)]= ICBCLoss;
end

if check3 == 1
      if hdoCP==0
            [ICLi95CP(t), ICLs95CP(t), ICLi99CP(t),
      ICLs99CP(t)]=ICBCCP;
            [ICLi95CPk(t), ICLs95CPk(t), ICLi99CPk(t),
      ICLs99CPk(t)]=ICBCCPk;
            [ICLi95CPkm(t), ICLs95CPkm(t), ICLi99CPkm(t),
      ICLs99CPkm(t)]=ICBCCPkm;
      end
end

case 'BCa'

[ICSNi95(t), ICSNs95(t),ICSNi99(t), ICSNs99(t)]=ICBCaSN;
```

```
      if check1 == 1
            [ICLi95(t), ICLs95(t),ICLi99(t), ICLs99(t)]= ICBCaLoss;
      end

      if check3 == 1
            if hdoCP==0
                  [ICLi95CP(t), ICLs95CP(t), ICLi99CP(t),
            ICLs99CP(t)]=ICBCaCP;
                  [ICLi95CPk(t), ICLs95CPk(t), ICLi99CPk(t),
            ICLs99CPk(t)]=ICBCaCPk;
                  [ICLi95CPkm(t), ICLs95CPkm(t), ICLi99CPkm(t),
            ICLs99CPkm(t)]=ICBCaCPkm;
            end
      end

      end
      end
      end

      end

%-------------------------------------------------------------------------
%------------------------------ ANOM -------------------------------------
%-------------------------------------------------------------------------
      ANOM

%-------------------------------------------------------------------------
%------------------------------ ANOM SN----------------------------------
%-------------------------------------------------------------------------
      ANOMSN

%----------------- Matrix exibição da amostra Mestre --------------------
      set(tbl99,'data',amostra2);

%---------------- Matriz de exibição de resultaods Final ----------------
      for i=1:linha
        for j=1:(coluna+27)
          if j<=coluna
          data(i,j)=design(i,j);
          elseif j==coluna+1
          data(i,j)=mediaLinha(i);
          elseif j==coluna+2
          data(i,j)=varLinha(i);
          elseif j==coluna+3
          data(i,j)=SNlinha(i);
          elseif (j==coluna+4) & (check2 == 1) & (condBOOT > 1)
          data(i,j)=ICSNi95(i);
          elseif (j==coluna+5) & (check2 == 1) & (condBOOT > 1)
          data(i,j)=ICSNs95(i);
          elseif (j==coluna+6) & (check2 == 1) & (condBOOT > 1)
          data(i,j)=ICSNi99(i);
          elseif (j==coluna+7) & (check2 == 1) & (condBOOT > 1)
          data(i,j)=ICSNs99(i);
          elseif (j==coluna+8) & (check1 == 1)
          data(i,j)=Lmedialinha(i);
          elseif (j==coluna+9) & (check1 == 1) & (check2 == 1) & (condBOOT >
1)
          data(i,j)=ICLi95(i);
```

```
            elseif (j==coluna+10) & (check1 == 1) & (check2 == 1) & (condBOOT
> 1)
            data(i,j)=ICLs95(i);
            elseif (j==coluna+11) & (check1 == 1) & (check2 == 1) & (condBOOT
> 1)
            data(i,j)=ICLi99(i);
            elseif (j==coluna+12) & (check1 == 1)& (check2 == 1) & (condBOOT >
1)
            data(i,j)=ICLs99(i);
            elseif (j==coluna+13) & (check3 == 1) & (condBOOT > 1)
            data(i,j)=CpExibi(i);
            elseif (j==coluna+14) & (check3 == 1)& (check2 == 1) & (condBOOT >
1)
            data(i,j)=ICLi95CP(i);
            elseif (j==coluna+15) & (check3 == 1)& (check2 == 1) & (condBOOT >
1)
            data(i,j)=ICLs95CP(i);
            elseif (j==coluna+16) & (check3 == 1)& (check2 == 1) & (condBOOT >
1)
            data(i,j)=ICLi99CP(i);
            elseif (j==coluna+17) & (check3 == 1)& (check2 == 1) & (condBOOT >
1)
            data(i,j)=ICLs99CP(i);
            elseif (j==coluna+18) & (check3 == 1) & (condBOOT > 1)
            data(i,j)=CpkExibi(i);
            elseif (j==coluna+19) & (check3 == 1)& (check2 == 1) & (condBOOT >
1)
            data(i,j)=ICLi95CPk(i);
            elseif (j==coluna+20) & (check3 == 1)& (check2 == 1) & (condBOOT >
1)
            data(i,j)=ICLs95CPk(i);
            elseif (j==coluna+21) & (check3 == 1)& (check2 == 1) & (condBOOT >
1)
            data(i,j)=ICLi99CPk(i);
            elseif (j==coluna+22) & (check3 == 1)& (check2 == 1) & (condBOOT >
1)
            data(i,j)=ICLs99CPk(i);
            elseif (j==coluna+23) & (check3 == 1) & (condBOOT > 1)
            data(i,j)=CpkmExibi(i);
            elseif (j==coluna+24) & (check3 == 1)& (check2 == 1) & (condBOOT >
1)
            data(i,j)=ICLi95CPkm(i);
            elseif (j==coluna+25) & (check3 == 1)& (check2 == 1) & (condBOOT >
1)
            data(i,j)=ICLs95CPkm(i);
            elseif (j==coluna+26) & (check3 == 1)& (check2 == 1) & (condBOOT >
1)
            data(i,j)=ICLi99CPkm(i);
            elseif (j==coluna+27) & (check3 == 1)& (check2 == 1) & (condBOOT >
1)
            data(i,j)=ICLs99CPkm(i);
            end
          end
        end
        dataexport=data;

        for j=1:coluna
          ColumnName1(j)=nameExp(j);
        end

        set(tbl6,'data',data)
```

```
      set(tbl6,'ColumnName',{ColumnName1{:},'Média','Variância','S/N','SN-
LI95%','SN-LS95%','SN-LI99%','SN-LS99%','Função      Perda','FP-LI95%','FP-
LS95%','FP-LI99%','FP-LS99%','Cp','Cp-LI95%','Cp-LS95%','Cp-LI99%','Cp-
LS99%','Cpk','Cpk-LI95%','Cpk-LS95%','Cpk-LI99%','Cpk-LS99%','Cpkm','Cpkm-
LI95%','Cpkm-LS95%','Cpkm-LI99%','Cpkm-LS99%'});
      aux12=get(tbl6,'ColumnName');
      Columnnname=aux12';

      %-----------------------------------------------------------------
      %-------------------- LIMPAR VARIÁVEIS -------------------------
      %-----------------------------------------------------------------
      clearvars -global data check1 check2 check3 sntype Cp Cpk Cpkm CpBOOT
CpkBOOT CpkmBOOT
      clearvars teste BAExi Bamostras

      %clear global;
      end


      function [LIz95, LSz95, LIz99, LSz99]=ICZLoss(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global Lmedia SEBOOTFP

            %'*********************************************************
            %*I.C. BOOSTRAT PADRAO DE 95% PARA A FUNCAO PERDA MEDIA*
            %'*********************************************************
            z1=norminv(0.025);
            z2=norminv(0.975);
            LIz95=(Lmedia+z1*SEBOOTFP);
            LSz95=(Lmedia+z2*SEBOOTFP);
            if LIz95<0
                  LIz95=0;
            end
            ICz95=[LIz95 LSz95];

            %*********************************************************
            %*I.C. BOOSTRAT PADRAO DE 99% PARA A FUNCAO PERDA MEDIA*
            %*********************************************************
            z1=norminv(0.005);
            z2=norminv(0.995);
            LIz99=(Lmedia+z1*SEBOOTFP);
            LSz99=(Lmedia+z2*SEBOOTFP);
            if LIz99<0
                  LIz99=0;
            end
            ICz99=[LIz99 LSz99];
      end


      function [LIt95, LSt95, LIt99, LSt99]=ICtLoss(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global Lmedia SEBOOTFP n
            %*************************************************
            %*I.C. BOOSTRAT t DE 95% PARA A FUNCAO PERDA MEDIA*
            %*************************************************
            t1=tinv(0.025,n-1);
            t2=tinv(0.975,n-1);
```

```
            LIt95=(Lmedia+t1*SEBOOTFP);
            LSt95=(Lmedia+t2*SEBOOTFP);
            if LIt95<0
                  LIt95=0;
            end
            ICt95=[LIt95 LSt95];

            %*****************************************************
            %*I.C. BOOTSTRAP t DE 99% PARA A FUNCAO PERDA MEDIA*
            %*****************************************************
            t1=tinv(0.005,n-1);
            t2=tinv(0.995,n-1);
            LIt99=(Lmedia+t1*SEBOOTFP);
            LSt99=(Lmedia+t2*SEBOOTFP);
            if LIt99<0
                  LIt99=0;
            end
            ICt99=[LIt99 LSt99];
      end


function [LIp95, LSp95, LIp99, LSp99]=ICpLoss(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global LmediaB
            %*********************************
            %*I.C. BOOTSTRAP PERCENTIL DE 95%*
            %*********************************
            P975=prctile(LmediaB,97.5);
            P25=prctile(LmediaB,2.5);
            LIp95=P25;
            LSp95=P975;

            if LIp95<0
                  LIp95=0;
            end
            ICp95=[LIp95 LSp95];

            %*********************************
            %*I.C. BOOTSTRAP PERCENTIL DE 99%*
            %'*********************************
            P995=prctile(LmediaB,99.5);
            P005=prctile(LmediaB,0.5);
            LIp99=P005;
            LSp99=P995;

            if LIp99<0
                  LIp99=0;
            end
            ICp99=[LIp99 LSp99];
      end


function [LIBC95, LSBC95, LIBC99, LSBC99]=ICBCLoss(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global Lmedia LmediaB B
            %********************
```

```
            %***I.C. BC DE 95%***
            %********************

            LmediaBOrd=sort(LmediaB);
            Cont=0;
            for i=1:B
                  if LmediaBOrd(i)<=Lmedia
                        Cont=Cont+1;
                  end
            end
            P0=Cont/B;
            z0=norminv(P0);
            z1=norminv(0.025);
            z2=norminv(0.975);
            PI=normcdf(2*z0+z1);
            PS=normcdf(2*z0+z2);
            LIBC95=prctile(LmediaB,PI*100);
            LSBC95=prctile(LmediaB,PS*100);

            if LIBC95<0
                  LIBC95=0;
            end
            ICBC95=[LIBC95 LSBC95]

            %********************
            %***I.C. BC DE 99%***
            %********************

            LmediaBOrd=sort(LmediaB);
            Cont=0;
            for i=1:B
                  if LmediaBOrd(i)<=Lmedia
                        Cont=Cont+1;
                  end
            end
            P0=Cont/B;
            z0=norminv(P0);
            z1=norminv(0.005);
            z2=norminv(0.995);
            PI=normcdf(2*z0+z1);
            PS=normcdf(2*z0+z2);
            LIBC99=prctile(LmediaB,PI*100);
            LSBC99=prctile(LmediaB,PS*100);

            if LIBC99<0
                  LIBC99=0;
            end
            ICBC99=[LIBC99 LSBC99];
      end


function [LIBCa95, LSBCa95, LIBCa99, LSBCa99]=ICBCaLoss(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global Lmedia LmediaB B a
            %**********************
            %***I.C. BCa DE 95%***
            %**********************
```

```
        LmediaBOrd=sort(LmediaB);
        Cont=0;
        for i=1:B
              if LmediaBOrd(i)<=Lmedia
                    Cont=Cont+1;
              end
        end
        P0=Cont/B;
        z0=norminv(P0);
        z1=norminv(0.025);
        z2=norminv(0.975);
        PI=normcdf(z0+((z0+z1/(1-a*(z0+z1)))));
        PS=normcdf(z0+((z0+z2/(1-a*(z0+z2)))));
        LIBCa95=prctile(LmediaB,PI*100);
        LSBCa95=prctile(LmediaB,PS*100);

        if LIBCa95<0
              LIBCa95=0;
        end
        ICBCa95=[LIBCa95 LSBCa95];

        %**********************
        %***I.C. BCa DE 99%***
        %**********************

        LmediaBOrd=sort(LmediaB);
        Cont=0;
        for i=1:B
              if LmediaBOrd(i)<=Lmedia
                    Cont=Cont+1;
              end
        end
        P0=Cont/B;
        z0=norminv(P0);
        z1=norminv(0.005);
        z2=norminv(0.995);
        PI=normcdf(z0+((z0+z1/(1-a*(z0+z1)))));
        PS=normcdf(z0+((z0+z2/(1-a*(z0+z2)))));
        LIBCa99=prctile(LmediaB,PI*100);
        LSBCa99=prctile(LmediaB,PS*100);

        if LIBCa99<0
              LIBCa99=0;
        end
        ICBCa99=[LIBCa99 LSBCa99];
  end


function [LIZ95SN, LSZ95SN, LIZ99SN, LSZ99SN]=ICZSN(varargin)
  %Essa função foi adaptada da função
  %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

  global SNNTBI SEBOOTSR
  %*****************************************************
  %*I.C. BOOSTRAT PADRAO DE 95% PARA A RAZAO SINAL-RUIDO*
  %*****************************************************
  z1=norminv(0.025);
  z2=norminv(0.975);
  LIZ95SN=(SNNTBI+z1*SEBOOTSR);
  LSZ95SN=(SNNTBI+z2*SEBOOTSR);
```

```
      ICZ95SN=[LIZ95SN LSZ95SN];

      %*******************************************************
      %*I.C. BOOSTRAT PADRAO DE 99% PARA A RAZAO SINAL-RUIDO*
      %*******************************************************
      z1=norminv(0.005);
      z2=norminv(0.995);
      LIZ99SN=(SNNTBI+z1*SEBOOTSR);
      LSZ99SN=(SNNTBI+z2*SEBOOTSR);
      ICZ99SN=[LIZ99SN LSZ99SN];
end


function [LIt95SN, LSt95SN, LIt99SN, LSt99SN]=ICtSN(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global SNNTBI SEBOOTSR n
      %*************************************************
      %*I.C. BOOTSTRAP t DE 95% PARA A RAZAO SINAL-RUIDO*
      %*************************************************
      t1=tinv(0.025,n-1);
      t2=tinv(0.975,n-1);
      LIt95SN=(SNNTBI+t1*SEBOOTSR);
      LSt95SN=(SNNTBI+t2*SEBOOTSR);
      ICt95SN=[LIt95SN LSt95SN];

      %*************************************************
      %*I.C. BOOTSTRAP t DE 99% PARA A RAZAO SINAL-RUIDO*
      %*************************************************
      t1=tinv(0.005,n-1);
      t2=tinv(0.995,n-1);
      LIt99SN=(SNNTBI+t1*SEBOOTSR);
      LSt99SN=(SNNTBI+t2*SEBOOTSR);
      ICt99SN=[LIt99SN LSt99SN];
end


function [LIp95SN, LSp95SN, LIp99SN, LSp99SN]=ICpSN(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global SNNTBIB
      %*********************************************************
      %*I.C. BOOTSTRAP PERCENTIL DE 95% PARA A RAZAO SINAL-RUIDO*
      %*********************************************************
      P975=prctile(SNNTBIB,97.5);
      P25=prctile(SNNTBIB,2.5);
      LIp95SN=P25;
      LSp95SN=P975;
      ICp95SN=[LIp95SN LSp95SN];

      %*********************************************************
      %*I.C. BOOTSTRAP PERCENTIL DE 99% PARA A RAZAO SINAL-RUIDO*
      %*********************************************************
      P995=prctile(SNNTBIB,99.5);
      P005=prctile(SNNTBIB,0.5);
      LIp99SN=P005;
      LSp99SN=P995;
      ICp99SN=[LIp99SN LSp99SN];
```

```
end


function [LIBC95SN, LSBC95SN, LIBC99SN, LSBC99SN]=ICBCSN(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global SNNTBI SNNTBIB B
      %*******************************************
      %***I.C. BC DE 95% PARA A RAZAO SINAL-RUIDO***
      %*******************************************

      SNNTBIBBOrd=sort(SNNTBIB);
      Cont=0;
      for i=1:B
            if SNNTBIBBOrd(i)<=SNNTBI
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.025);
      z2=norminv(0.975);
      PI=normcdf(2*z0+z1);
      PS=normcdf(2*z0+z2);
      LIBC95SN=prctile(SNNTBIB,PI*100);
      LSBC95SN=prctile(SNNTBIB,PS*100);
      ICBC95SN=[LIBC95SN LSBC95SN];

      %*******************************************
      %***I.C. BC DE 99% PARA A RAZAO SINAL-RUIDO***
      %*******************************************

      SNNTBIBBOrd=sort(SNNTBIB);
      Cont=0;
      for i=1:B
            if SNNTBIBBOrd(i)<=SNNTBI
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.005);
      z2=norminv(0.995);
      PI=normcdf(2*z0+z1);
      PS=normcdf(2*z0+z2);
      LIBC99SN=prctile(SNNTBIB,PI*100);
      LSBC99SN=prctile(SNNTBIB,PS*100);
      ICBC99SN=[LIBC99SN LSBC99SN];
end


function [LIBCa95SN, LSBCa95SN,LIBCa99SN,LSBCa99SN]=ICBCaSN(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global SNNTBI SNNTBIB B aSN
      %**********************
      %***I.C. BCa DE 95%***
      %**********************
```

```
      SNNTBIBBOrd=sort(SNNTBIB);
      Cont=0;
      for i=1:B
            if SNNTBIBBOrd(i)<=SNNTBI
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.025);
      z2=norminv(0.975);
      PI=normcdf(z0+((z0+z1/(1-aSN*(z0+z1)))));
      PS=normcdf(z0+((z0+z2/(1-aSN*(z0+z2)))));
      LIBCa95SN=prctile(SNNTBIB,PI*100);
      LSBCa95SN=prctile(SNNTBIB,PS*100);
      ICBCa95SN=[LIBCa95SN LSBCa95SN];

      %*********************
      %***I.C. BCa DE 99%***
      %*********************

      SNNTBIBBOrd=sort(SNNTBIB);
      Cont=0;
      for i=1:B
            if SNNTBIBBOrd(i)<=SNNTBI
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.005);
      z2=norminv(0.995);
      PI=normcdf(z0+((z0+z1/(1-aSN*(z0+z1)))));
      PS=normcdf(z0+((z0+z2/(1-aSN*(z0+z2)))));
      LIBCa99SN=prctile(SNNTBIB,PI*100);
      LSBCa99SN=prctile(SNNTBIB,PS*100);
      ICBCa99SN=[LIBCa99SN LSBCa99SN];
end


function [LIZ95CP, LSZ95CP, LIZ99CP, LSZ99CP]=ICZCP(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global SEBOOTCp Cp

      %I.C. BOOSTRAT PADRAO DE 95% PARA O INDICE Cp*
      z1=norminv(0.025);
      z2=norminv(0.975);
      LIZ95CP=(Cp+z1*SEBOOTCp);
      LSZ95CP=(Cp+z2*SEBOOTCp);
      if LIZ95CP<0
            LIZ95CP=0;
      end
      ICZ95CP=[LIZ95CP LSZ95CP];

      %*I.C. BOOSTRAT PADRAO DE 99% PARA O INDICE Cp*
      z1=norminv(0.005);
      z2=norminv(0.995);
```

```
      LIZ99CP=(Cp+z1*SEBOOTCp);
      LSZ99CP=(Cp+z2*SEBOOTCp);
      if LIZ99CP<0
            LIZ99CP=0;
      end
      ICZ99CP=[LIZ99CP LSZ99CP];
end


function [LIt95CP, LSt95CP, LIt99CP, LSt99CP]=ICtCP(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global Cp SEBOOTCp n

      %I.C. BOOTSTRAP t DE 95% PARA O INDICE Cp*
      t1=tinv(0.025,n-1);
      t2=tinv(0.975,n-1);
      LIt95CP=(Cp+t1*SEBOOTCp);
      LSt95CP=(Cp+t2*SEBOOTCp);

      if LIt95CP<0
            LIt95CP=0;
      end
      ICt95CP=[LIt95CP LSt95CP];

      %*I.C. BOOTSTRAP t DE 99% PARA O INDICE Cp*
      t1=tinv(0.005,n-1);
      t2=tinv(0.995,n-1);
      LIt99CP=(Cp+t1*SEBOOTCp);
      LSt99CP=(Cp+t2*SEBOOTCp);

      if LIt99CP<0
            LIt99CP=0;
      end
      ICt99CP=[LIt99CP LSt99CP];
end


function [LIp95CP, LSp95CP, LIp99CP, LSp99CP]=ICpCP(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global CpBOOT

      %*I.C. BOOTSTRAP PERCENTIL DE 95% PARA PARA O INDICE Cp*
      P975=prctile(CpBOOT,97.5);
      P25=prctile(CpBOOT,2.5);
      LIp95CP=P25;
      LSp95CP=P975;

      if LIp95CP<0
            LIp95CP=0;
      end
      ICp95CP=[LIp95CP LSp95CP];

      %*I.C. BOOTSTRAP PERCENTIL DE 99% PARA O INDICE Cp*
      P995=prctile(CpBOOT,99.5);
      P005=prctile(CpBOOT,0.5);
```

```
      LIp99CP=P005;
      LSp99CP=P995;

      if LIp99CP<0
            LIp99CP=0;
      end
      ICp99CP=[LIp99CP LSp99CP];
end


function [LIBC95CP, LSBC95CP, LIBC99CP, LSBC99CP]=ICBCCP(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global CpBOOT B Cp

      %***I.C. BC DE 95% PARA PARA O INDICE Cp*
      CpBOOTBOrd=sort(CpBOOT);
      Cont=0;
      for i=1:B
            if CpBOOTBOrd(i)<=Cp
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.025);
      z2=norminv(0.975);
      PI=normcdf(2*z0+z1);
      PS=normcdf(2*z0+z2);
      LIBC95CP=prctile(CpBOOT,PI*100);
      LSBC95CP=prctile(CpBOOT,PS*100);

      if LIBC95CP<0
            LIBC95CP=0;
      end
      ICBC95CP=[LIBC95CP LSBC95CP];

      %***I.C. BC DE 99% PARA O INDICE Cp*
      CpBOOTBOrd=sort(CpBOOT);
      Cont=0;
      for i=1:B
            if CpBOOTBOrd(i)<=Cp
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.005);
      z2=norminv(0.995);
      PI=normcdf(2*z0+z1);
      PS=normcdf(2*z0+z2);
      LIBC99CP=prctile(CpBOOT,PI*100);
      LSBC99CP=prctile(CpBOOT,PS*100);

      if LIBC99CP<0
            LIBC99CP=0;
      end
      ICBC99CP=[LIBC99CP LSBC99CP];
end
```

```
function [LIBCa95CP, LSBCa95CP,LIBCa99CP,LSBCa99CP]=ICBCaCP(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global CpBOOT B Cp acp

      %***I.C. BCa DE 95% PARA O INDICE Cp****
      CpBOOTBOrd=sort(CpBOOT);
      Cont=0;
      for i=1:B
            if CpBOOTBOrd(i)<=Cp
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.025);
      z2=norminv(0.975);
      PI=normcdf(z0+((z0+z1/(1-acp*(z0+z1)))));
      PS=normcdf(z0+((z0+z2/(1-acp*(z0+z2)))));
      LIBCa95CP=prctile(CpBOOT,PI*100);
      LSBCa95CP=prctile(CpBOOT,PS*100);
      if LIBCa95CP<0
            LIBCa95CP=0;
      end
      ICBCa95CP=[LIBCa95CP LSBCa95CP];

      %***I.C. BCa DE 99% PARA O INDICE Cp****
      CpBOOTBOrd=sort(CpBOOT);
      Cont=0;
      for i=1:B
            if CpBOOTBOrd(i)<=Cp
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.005);
      z2=norminv(0.995);
      PI=normcdf(z0+((z0+z1/(1-acp*(z0+z1)))));
      PS=normcdf(z0+((z0+z2/(1-acp*(z0+z2)))));
      LIBCa99CP=prctile(CpBOOT,PI*100);
      LSBCa99CP=prctile(CpBOOT,PS*100);
      if LIBCa99CP<0
            LIBCa99CP=0;
      end
      ICBCa99CP=[LIBCa99CP LSBCa99CP];
end


function [LIZ95CPk, LSZ95CPk, LIZ99CPk, LSZ99CPk]=ICZCPk(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global  Cpk SEBOOTCpk

      %*I.C. BOOSTRAT PADRAO DE 95% PARA O INDICE Cpk*
      z1=norminv(0.025);
```

```
      z2=norminv(0.975);
      LIZ95CPk=(Cpk+z1*SEBOOTCpk);
      LSZ95CPk=(Cpk+z2*SEBOOTCpk);
      if LIZ95CPk<0
            LIZ95CPk=0;
      end
      ICZ95CPk=[LIZ95CPk LSZ95CPk];

      %*I.C. BOOSTRAT PADRAO DE 99% PARA O INDICE Cpk*
      z1=norminv(0.005);
      z2=norminv(0.995);
      LIZ99CPk=(Cpk+z1*SEBOOTCpk);
      LSZ99CPk=(Cpk+z2*SEBOOTCpk);
      if LIZ99CPk<0
            LIZ99CPk=0;
      end
      ICZ99CPk=[LIZ99CPk LSZ99CPk];
end


function [LIt95CPk, LSt95CPk, LIt99CPk, LSt99CPk]=ICtCPk(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global n SEBOOTCpk Cpk

      %*I.C. BOOTSTRAP t DE 95% PARA O INDICE Cpk*
      t1=tinv(0.025,n-1);
      t2=tinv(0.975,n-1);
      LIt95CPk=(Cpk+t1*SEBOOTCpk);
      LSt95CPk=(Cpk+t2*SEBOOTCpk);

      if LIt95CPk<0
            LIt95CPk=0;
      end
      ICt95CPk=[LIt95CPk LSt95CPk];

      %*I.C. BOOTSTRAP t DE 99% PARA O INDICE Cpk*
      t1=tinv(0.005,n-1);
      t2=tinv(0.995,n-1);
      LIt99CPk=(Cpk+t1*SEBOOTCpk);
      LSt99CPk=(Cpk+t2*SEBOOTCpk);

      if LIt99CPk<0
            LIt99CPk=0;
      end
      ICt99CPk=[LIt99CPk LSt99CPk];
end


function [LIp95CPk, LSp95CPk, LIp99CPk, LSp99CPk]=ICpCPk(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global CpkBOOT

      %*I.C. BOOTSTRAP PERCENTIL DE 95% PARA PARA O INDICE Cpk*
      P975=prctile(CpkBOOT,97.5);
      P25=prctile(CpkBOOT,2.5);
      LIp95CPk=P25;
```

```
      LSp95CPk=P975;

      if LIp95CPk<0
            LIp95CPk=0;
      end
      ICp95CPk=[LIp95CPk LSp95CPk];

      %'*I.C. BOOTSTRAP PERCENTIL DE 99% PARA O INDICE Cpk*
      P995=prctile(CpkBOOT,99.5);
      P005=prctile(CpkBOOT,0.5);
      LIp99CPk=P005;
      LSp99CPk=P995;

      if LIp99CPk<0
            LIp99CPk=0;
      end
      ICp99CPk=[LIp99CPk LSp99CPk];
end


function [LIBC95CPk, LSBC95CPk,LIBC99CPk,LSBC99CPk]=ICBCCPk(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global CpkBOOT B Cpk

      %***I.C. BC DE 95% PARA PARA O INDICE Cpk*
      CpkBOOTBOrd=sort(CpkBOOT);
      Cont=0;
      for i=1:B
            if CpkBOOTBOrd(i)<=Cpk
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.025);
      z2=norminv(0.975);
      PI=normcdf(2*z0+z1);
      PS=normcdf(2*z0+z2);
      LIBC95CPk=prctile(CpkBOOT,PI*100);
      LSBC95CPk=prctile(CpkBOOT,PS*100);

      if LIBC95CPk<0
            LIBC95CPk=0;
      end
      ICBC95CPk=[LIBC95CPk LSBC95CPk];

      %***I.C. BC DE 99% PARA O INDICE Cpk*
      CpkBOOTBOrd=sort(CpkBOOT);
      Cont=0;
      for i=1:B
            if CpkBOOTBOrd(i)<=Cpk
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.005);
      z2=norminv(0.995);
```

```
      PI=normcdf(2*z0+z1);
      PS=normcdf(2*z0+z2);
      LIBC99CPk=prctile(CpkBOOT,PI*100);
      LSBC99CPk=prctile(CpkBOOT,PS*100);

      if LIBC99CPk<0
            LIBC99CPk=0;
      end
      ICBC99CPk=[LIBC99CPk LSBC99CPk];
end


function [LIBCa95CPk,LSBCa95CPk,LIBCa99CPk,LSBCa99CPk]= ICBCaCPk(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global CpkBOOT Cpk B acpk

      %***I.C. BCa DE 95% PARA O INDICE Cpk****
      CpkBOOTBOrd=sort(CpkBOOT);
      Cont=0;
      for i=1:B
            if CpkBOOTBOrd(i)<=Cpk
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.025);
      z2=norminv(0.975);
      PI=normcdf(z0+((z0+z1/(1-acpk*(z0+z1)))));
      PS=normcdf(z0+((z0+z2/(1-acpk*(z0+z2)))));
      LIBCa95CPk=prctile(CpkBOOT,PI*100);
      LSBCa95CPk=prctile(CpkBOOT,PS*100);
      if LIBCa95CPk<0
            LIBCa95CPk=0;
      end
      ICBCa95CPk=[LIBCa95CPk LSBCa95CPk];

      %***I.C. BCa DE 99% PARA O INDICE Cpk****
      CpkBOOTBOrd=sort(CpkBOOT);
      Cont=0;
      for i=1:B
            if CpkBOOTBOrd(i)<=Cpk
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.005);
      z2=norminv(0.995);
      PI=normcdf(z0+((z0+z1/(1-acpk*(z0+z1)))));
      PS=normcdf(z0+((z0+z2/(1-acpk*(z0+z2)))));;
      LIBCa99CPk=prctile(CpkBOOT,PI*100);
      LSBCa99CPk=prctile(CpkBOOT,PS*100);
      if LIBCa99CPk<0
            LIBCa99CPk=0;
      end
      ICBCa99CPk=[LIBCa99CPk LSBCa99CPk];
end
```

```
function [LIZ95CPkm, LSZ95CPkm, LIZ99CPkm,LSZ99CPkm]= ICZCPkm(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global Cpkm SEBOOTCpkm

      %*I.C. BOOSTRAT PADRAO DE 95% PARA O INDICE Cpkm*
      z1=norminv(0.025);
      z2=norminv(0.975);
      LIZ95CPkm=(Cpkm+z1*SEBOOTCpkm);
      LSZ95CPkm=(Cpkm+z2*SEBOOTCpkm);
      if LIZ95CPkm<0
            LIZ95CPkm=0;
      end
      ICZ95CPkm=[LIZ95CPkm LSZ95CPkm];

      %*I.C. BOOSTRAT PADRAO DE 99% PARA O INDICE Cpkm*')
      z1=norminv(0.005);
      z2=norminv(0.995);
      LIZ99CPkm=(Cpkm+z1*SEBOOTCpkm);
      LSZ99CPkm=(Cpkm+z2*SEBOOTCpkm);
      if LIZ99CPkm<0
            LIZ99CPkm=0;
      end
      ICZ99CPkm=[LIZ99CPkm LSZ99CPkm];
end


function [LIt95CPkm, LSt95CPkm, LIt99CPkm, LSt99CPkm]= ICtCPkm(varargin)
      global n SEBOOTCpkm Cpkm

      %*I.C. BOOTSTRAP t DE 95% PARA O INDICE Cpkm*
      t1=tinv(0.025,n-1);
      t2=tinv(0.975,n-1);
      LIt95CPkm=(Cpkm+t1*SEBOOTCpkm);
      LSt95CPkm=(Cpkm+t2*SEBOOTCpkm);

      if LIt95CPkm<0
            LIt95CPkm=0;
      end
      ICt95CPkm=[LIt95CPkm LSt95CPkm];

      %*I.C. BOOTSTRAP t DE 99% PARA O INDICE Cpkm*
      t1=tinv(0.005,n-1);
      t2=tinv(0.995,n-1);
      LIt99CPkm=(Cpkm+t1*SEBOOTCpkm);
      LSt99CPkm=(Cpkm+t2*SEBOOTCpkm);

      if LIt99CPkm<0
            LIt99CPkm=0;
      end
      ICt99CPkm=[LIt99CPkm LSt99CPkm];
end


function [LIp95CPkm, LSp95CPkm, LIp99CPkm, LSp99CPkm]= ICpCPkm(varargin)
```

```
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global CpkmBOOT

      %'*I.C. BOOTSTRAP PERCENTIL DE 95% PARA PARA O INDICE Cpkm*
      P975=prctile(CpkmBOOT,97.5);
      P25=prctile(CpkmBOOT,2.5);
      LIp95CPkm=P25;
      LSp95CPkm=P975;

      if LIp95CPkm<0
            LIp95CPkm=0;
      end
      ICp95CPkm=[LIp95CPkm LSp95CPkm];

      %*I.C. BOOTSTRAP PERCENTIL DE 99% PARA O INDICE Cpkm*
      P995=prctile(CpkmBOOT,99.5);
      P005=prctile(CpkmBOOT,0.5);
      LIp99CPkm=P005;
      LSp99CPkm=P995;

      if LIp99CPkm<0
            LIp99CPkm=0;
      end
      ICp99CPkm=[LIp99CPkm LSp99CPkm];
      end


function[LIBC95CPkm, LSBC95CPkm, LIBC99CPkm, LSBC99CPkm]=ICBCCPkm(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global CpkmBOOT B Cpkm

      %***I.C. BC DE 95% PARA PARA O INDICE Cpkm*
      CpkmBOOTBOrd=sort(CpkmBOOT);
      Cont=0;
      for i=1:B
            if CpkmBOOTBOrd(i)<=Cpkm
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.025);
      z2=norminv(0.975);
      PI=normcdf(2*z0+z1);
      PS=normcdf(2*z0+z2);
      LIBC95CPkm=prctile(CpkmBOOT,PI*100);
      LSBC95CPkm=prctile(CpkmBOOT,PS*100);

      if LIBC95CPkm<0
            LIBC95CPkm=0;
      end
      ICBC95CPkm=[LIBC95CPkm LSBC95CPkm];

      %***I.C. BC DE 99% PARA O INDICE Cpkm*
      CpkmBOOTBOrd=sort(CpkmBOOT);
      Cont=0;
```

```
      for i=1:B
            if CpkmBOOTBOrd(i)<=Cpkm
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.005);
      z2=norminv(0.995);
      PI=normcdf(2*z0+z1);
      PS=normcdf(2*z0+z2);
      LIBC99CPkm=prctile(CpkmBOOT,PI*100);
      LSBC99CPkm=prctile(CpkmBOOT,PS*100);

      if LIBC99CPkm<0
            LIBC99CPkm=0;
      end
      ICBC99CPkm=[LIBC99CPkm LSBC99CPkm];
end


function [LIBCa95CPkm, LSBCa95CPkm, LIBCa99CPkm,
LSBCa99CPkm]=ICBCaCPkm(varargin)
      %Essa função foi adaptada da função
      %"perdasruido(Amostra,B,mu,delta,cf,P)" da Tese de MARQUES (2015).

      global CpkmBOOT B Cpkm acpkm

      %***I.C. BCa DE 95% PARA O INDICE Cpkm****
      CpkmBOOTBOrd=sort(CpkmBOOT);
      Cont=0;
      for i=1:B
            if CpkmBOOTBOrd(i)<=Cpkm
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.025);
      z2=norminv(0.975);
      PI=normcdf(z0+((z0+z1/(1-acpkm*(z0+z1)))));
      PS=normcdf(z0+((z0+z2/(1-acpkm*(z0+z2)))));
      LIBCa95CPkm=prctile(CpkmBOOT,PI*100);
      LSBCa95CPkm=prctile(CpkmBOOT,PS*100);
      if LIBCa95CPkm<0
            LIBCa95CPkm=0;
      end
      ICBCa95CPkm=[LIBCa95CPkm LSBCa95CPkm];

      %'***I.C. BCa DE 99% PARA O INDICE Cpkm****
      CpkmBOOTBOrd=sort(CpkmBOOT);
      Cont=0;
      for i=1:B
            if CpkmBOOTBOrd(i)<=Cpkm
                  Cont=Cont+1;
            end
      end
      P0=Cont/B;
      z0=norminv(P0);
      z1=norminv(0.005);
```

```
z2=norminv(0.995);
PI=normcdf(z0+((z0+z1/(1-acpkm*(z0+z1)))));
PS=normcdf(z0+((z0+z2/(1-acpkm*(z0+z2)))));;
LIBCa99CPkm=prctile(CpkmBOOT,PI*100);
LSBCa99CPkm=prctile(CpkmBOOT,PS*100);
if LIBCa99CPkm<0
      LIBCa99CPkm=0;
end
ICBCa99CPkm=[LIBCa99CPkm LSBCa99CPkm];
end


%-----------------------------
%Função para nomear as colunas
%-----------------------------
function []=name(varargin)
global design3 namefat design nameExp escolha

%VARIÁVEIS
%escolha -  matriz com as colunas escolinhas para o design
%namefat - nome dos fatores

[li,co]=size(design3);
[li2,co2]=size(design);
tam=length(escolha);

% ------------ vetor de nomes para as interações -------------------
for a=1:co
  nunu=num2str(a);
 namefat{a}=['Fator' ' ' nunu] ;
end

%-------- vetor de nome para matrix experiento --------------------
for a=1:co2
  nameExp{a}=[];
end
for a=1:co
  aux=escolha(a);
  nunu=num2str(a);
  nameExp{aux}=['Fator' ' ' nunu];
end
for a=1:co2
  aux=isempty(nameExp{a});
  if aux==1
    nameExp{a}='e';
  end
end

end


%-----------------------------------
%nomear interação
%-----------------------------------
function []=nameMODIFICADO(varargin)
%VARIAVEIS
%escolha  -  matriz com as colunas escolinhas para o design
%namefat  -  nome dos fatores
%conflito -  parametro para aviso quando fatores principais se confundem
com
%    interações
```

```
%aux90  -  indice da interação

global design3 design4 namefat design nameExp escolha interação...
lista_Int IntPro alternativa listv ...
nameANOM popup2 L_name aux90 conflito

%CONTROLE
CONFLITONAME=conflito
if conflito == 1
  return
end

[li,co]=size(design3);
[li2,co2]=size(design);
[li4,co4]=size(design4);
tam=length(escolha);
tam2=length(listv);

% ------------ vetor de nomes para as interações -------------------
for a=1:co
  nunu=num2str(a);
  namefat{a}=['Fator' ' ' nunu] ;
end

%-------- vetor de nome para matrix experiento --------------------
for a=1:co2
  nameExp{a}=[];
  nameANOM{a}=[];
end
%fatores
for a=1:co
  aux=escolha(a);
  nunu=num2str(a);
  nameExp{aux}=['Fator' ' ' nunu];
  nameANOM{aux}=['Fator' ' ' nunu];
end
%interacao
for a=1:tam2
  if conflito == 1
  else
    if get(popup2,'value') == 2
          aux=lista_Int(a);
          aux2=find(aux==interacao);
          aux3=IntPro{alternativa}(aux2);
          nameExp{aux}=char(aux3);
          nameANOM{aux}=char(aux3);
    end

    if get(popup2,'value') == 3

      switch L_name
        case 'L9 (3^4)'
        aux=lista_Int(a);
        aux2=find(aux==interacao);
        aux3=IntPro{alternativa}(aux2);
        nameExp{aux}=char(aux3);
        nameANOM{aux}=char(aux3);
        aux2=interacao(aux2+1) ;%aqui começa
        nameExp{aux2}=char(aux3);
        nameANOM{aux2}=char(aux3);
```

```
        case 'L27 (3^13)'
        aux56=aux90{a};
        aux3=IntPro{alternativa}(aux56) ;
        aumento= aux90{a}-1;
        aux78=aux90{a}+aumento;
        aux79=aux90{a}+aumento+1;
        aux80=interacao(aux78);
        aux81=interacao(aux79);
        nameExp{aux80}=char(aux3);
        nameANOM{aux80}=char(aux3);
        nameExp{aux81}=char(aux3);
        nameANOM{aux81}=char(aux3);

      end
    end

    if get(popup2,'value') == 4
          aux=lista_Int(a);
          aux2=find(aux==interacao);
          aux3=IntPro{alternativa}(aux2);
          nameExp{aux}=char(aux3);
          nameANOM{aux}=char(aux3);
          aux66=interacao(aux2+1) ;%aqui começa
          nameExp{aux66}=char(aux3);
          nameANOM{aux66}=char(aux3);
          aux88=interacao(aux2+2) ;%aqui começa
          nameExp{aux88}=char(aux3);
          nameANOM{aux88}=char(aux3);
    end

    if get(popup2,'value') == 5
          aux=lista_Int(a);
          aux2=find(aux==interacao);
          aux3=IntPro{alternativa}(aux2);
          nameExp{aux}=char(aux3);
          nameANOM{aux}=char(aux3);
          aux66=interacao(aux2+1) ;%aqui começa
          nameExp{aux66}=char(aux3);
          nameANOM{aux66}=char(aux3);
          aux88=interacao(aux2+2) ;%aqui começa
          nameExp{aux88}=char(aux3);
          nameANOM{aux88}=char(aux3);
          aux21=interacao(aux2+3) ;%aqui começa
          nameExp{aux21}=char(aux3);
          nameANOM{aux21}=char(aux3);
    end

    if get(popup2,'value') == 6
    switch L_name
      case 'L54 (2 x 3^25)'
        if alternativa == 2
        %não acontece nada
        else
              if              strcmp(listv(a),'1x2')==1              ||
strcmp(listv(a),'1x9')==1 || strcmp(listv(a),'2x9')==1
                %nada acontece
              else
                aux56=aux90{a};
                aux3=IntPro{alternativa}(aux56) ;
```

```
                aumento= aux90{a}-1;
                aux78=aux90{a}+aumento;
                aux79=aux90{a}+aumento+1;
                aux80=interacao(aux78);
                aux81=interacao(aux79);
                nameExp{aux80}=char(aux3);
                nameANOM{aux80}=char(aux3);
                nameExp{aux81}=char(aux3);
                nameANOM{aux81}=char(aux3);
              end
        end
    end
    end
  end
end

if get(popup2,'value') == 6
  nameANOM=nameANOM(~cellfun('isempty',nameANOM));
else
  nameANOM=nameANOM(1:co4);
  nameANOM=nameANOM(~cellfun('isempty',nameANOM));
end

%colunas livres
for a=1:co2
  aux=isempty(nameExp{a});
  if aux==1
    nameExp{a}='e';
  end
end

nameExp;

%--------------- limpar variaveis---------------------
clearvars aux aux2 aux3
end



%-----------------------------------------------------------------
%Gera VETOR INTERAÇÃO e GRUPO de acordo com o numero de fatores
%-----------------------------------------------------------------
function []=vetor_interacao(varargin)
global popup1 vetint design3 escolha g namefatshow2 namefat

%vetint - vetor de todas as interações possiveis

bb= get(popup1,'value');
bb2=get(popup1,'value');

mat{1,1}=0;
hh=1;
for cc=1:(bb2-1) %coluna
  aa=cc+1;
  for li=1:(bb-1) %linha
    dd=num2str(escolha(aa));
    tt=num2str(escolha(cc));
    text=strcat(tt,'x',dd);
    mat{li,cc}=text;
    g{hh}=horzcat(design3(:,cc),design3(:,aa));
```

```
    namefatshow2{hh}=[namefat(cc) namefat(aa)];
    aa=aa+1;
    hh=hh+1;
  end
  bb=bb-1;
end

mat2=find(~cellfun(@isempty,mat));
vetint=mat(mat2); %char()

%------------------ Limpar variaveis ----------------------------
clearvars mat g bb bb2 hh
end


function []=vetor_interacaoMODIFICADO(varargin)
global popup1 vetint
%vetint - vetor de todas as interações possiveis

bb= get(popup1,'value');
bb2=get(popup1,'value');

mat{1,1}=0;
hh=1;
for cc=1:(bb2-1) %coluna
  aa=cc+1;
  for li=1:(bb-1) %linha
    dd=num2str(aa);
    tt=num2str(cc);
    text=strcat(tt,'x',dd);
    mat{li,cc}=text;
    aa=aa+1;
    hh=hh+1;
  end
  bb=bb-1;
end

mat2=find(~cellfun(@isempty,mat));
vetint=mat(mat2); %char()

%------------------ Limpar variaveis ----------------------------
clearvars mat g bb bb2 hh

end %Gera VETOR INTERAÇÃO e GRUPO de acordo com o numero de fatores



%-------------------------
%função de agrupamento
%-------------------------
function GRUPING(varargin)
global popup1 design3 g namefatshow2 namefat

%vetint - vetor de todas as interações possiveis

bb= get(popup1,'value');
bb2=get(popup1,'value');

hh=1;
```

```
for cc=1:(bb2-1) %coluna
  aa=cc+1;
  for li=1:(bb-1) %linha
    g{hh}=horzcat(design3(:,cc),design3(:,aa));
    namefatshow2{hh}=[namefat(cc) namefat(aa)];
    aa=aa+1;
    hh=hh+1;
  end
  bb=bb-1;
end

%------------------ Limpar variaveis ---------------------------
clearvars g bb bb2 hh
end

%----------------------------------
%Escolha de colunas do Design
%----------------------------------
function []=escolha_vetor(varargin)
%VARIAVEIS
%escolha - colunas da matriz design de acordo com numero de fatores

global escolha design3 popup1 popup2 des L_name

valpopup1=get(popup1,'value'); %fatores
valpopup2=get(popup2,'value'); %niveis

switch L_name

  case 'L4 (2^3)'
    escolha=[1 2 3]

  case 'L8 (2^7)'
    if valpopup1 <= 4
      escolha=[1 2 4 7]
    else
       escolha=[1 2 3 4 5 6 7]
    end

  case 'L12 (2^11)'
    escolha=[1 2 3 4 5 6 7 8 9 10 11]

  case 'L16 (2^15)'
    if valpopup1 <= 4
       escolha=[1 2 4 8]
    elseif valpopup1 == 5
      escolha=[1 2 4 8 15]
    elseif (valpopup1 > 5) & (valpopup1 <= 8)
       escolha=[1 2 4 8 7 11 13 14]
    elseif valpopup1 >8
       escolha=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]
    end

  case 'L32 (2^31)'
    if valpopup1 <= 4
      escolha=[1 2 4 8]
    elseif valpopup1 == 5
      escolha=[1 2 4 8 16]
    elseif valpopup1 == 6
```

```
      escolha=[1 2 4 8 16 31]
    elseif (valpopup1 > 6) & (valpopup1 <= 16)
      escolha=[1 2 4 8 16 7 11 13 14 19 21 22 25 26 28 31]
    elseif valpopup1 >16
      escolha=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30 31]
    end

  case 'L9 (3^4)'
    escolha=[1 2 3 4]

  case 'L27 (3^13)'
    if valpopup1 <=7
      escolha=[1 2 5 9 10 12 13]
    else
      escolha =[1 2 3 4 5 6 7 8 9 10 11 12 13]
    end

  case  'L16 (4^5)'
    escolha=[1 2 3 4 5]

  case 'L25 (5^6)'
    escolha=[1 2 3 4 5 6]

  case 'L18 (2 x 3^7)'
    escolha=[1 2 3 4 5 6 7 8]

  case 'L32 (2 x 4^9)'
    escolha=[1 2 3 4 5 6 7 8 9 10]

  case 'L36 (2^11 x 3^12)'
      escolha=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
      22 23 24 25 26]

  case 'L50 (2 x 5^11)'
    escolha=[1 2 3 4 5 6 7 8 9 10 11 12]

  case 'L54 (2 x 3^25)'
    if valpopup1 <=3
      escolha=[1 2 9]
    else
      escolha=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
      22 23 24 25 26]
    end
end

k=escolha(1);
design3=des(:,k);
Sescolha=valpopup1;

for h=2:Sescolha
  k=escolha(h);
  design3=horzcat(design3,des(:,k));
end

end

%cria variavel design3 quando não tem interação
function DESIGNN_3(varargin)
```

```
global escolha design3 des popup1

valpopup1=get(popup1,'value'); %fatores

k=escolha(1);
design3=des(:,k);
Sescolha=valpopup1;

for h=2:Sescolha
  k=escolha(h);
  design3=horzcat(design3,des(:,k));
end
design3;
end



%------------------------------
%cria variavel design4 quando tem interação
%-----------------------------------------
function DESIGNN_4(varargin)
global escolha design3 des popup1 design4 interacao listv IntPro
alternativa lista_Int conflito popup2

%VARIAVEIS

%CONTROLE
CONFLITODESIGN4=conflito
if conflito == 1
  return
end

listv
n_int=length(listv);

for par1=1:n_int
  listv(par1)
  IntPro{alternativa}
  aux9=strmatch(listv(par1),IntPro{alternativa})

  if get(popup2,'value') == 2
    if par1 == 1
      lista_Int=interacao(aux9)
    else
      lista_Int=[lista_Int interacao(aux9)]
    end
  end

  if get(popup2,'value') == 3
    aumento=aux9-1;
    aux10=aux9+aumento;
    if par1 == 1
       lista_Int=[interacao(aux10) interacao(aux10+1)]
    else
      lista_Int=[lista_Int interacao(aux10) interacao(aux10+1)]
    end
  end

  if get(popup2,'value') == 4
```

```
          if aux9 == 1
          aux10=1;
          if par1 == 1
            lista_Int=[interacao(aux10) interacao(aux10+1)
interacao(aux10+2)];
          else
            lista_Int=[lista_Int interacao(aux10) interacao(aux10+1)
interacao(aux10+2)];
          end
          elseif aux9 == 2
          aux10=4;
          if par1 == 1
            lista_Int=[interacao(aux10) interacao(aux10+1)
interacao(aux10+2)];
          else
            lista_Int=[lista_Int interacao(aux10) interacao(aux10+1)
interacao(aux10+2)];
          end
          else
           aux10= (aux9*2)+(aux9-2);
           if par1 == 1
            lista_Int=[interacao(aux10) interacao(aux10+1)
interacao(aux10+2)];
          else
            lista_Int=[lista_Int interacao(aux10) interacao(aux10+1)
interacao(aux10+2)];
          end
          end

        end

        if get(popup2,'value') == 5
          lista_Int=[interacao(aux9) interacao(aux9+1) interacao(aux9+2)
interacao(aux9+3)];
        end

        if get(popup2,'value') == 6
          if alternativa == 2
            design4=design3;
          return
          else
          aumento=aux9-1
          aux10=aux9+aumento
          aux69 = 1;
          if strcmp(listv(par1),'1x2')==1 || strcmp(listv(par1),'1x9')==1 ||
strcmp(listv(par1),'2x9')==1
            if aux69 == 1
              lista_Int=[];
            else
              lista_Int=lista_Int;
            end
            aux69 = aux69+1;
          else

            if aux69 == 1
              lista_Int=[interacao(aux10) interacao(aux10+1)];
            else
              lista_Int=[lista_Int interacao(aux10) interacao(aux10+1)];
            end
            aux69 = aux69+1;
          end
```

```
    end
  end

end

vet_fatint=[escolha lista_Int];
vet_fatintOrder=sort(vet_fatint,'ascend');

%CONTROLE
LISTAFATORESINTERACAO=vet_fatint

n_vet=length(vet_fatintOrder) ;

k=vet_fatintOrder(1);
design4=des(:,k);

for h=2:n_vet
  k=vet_fatintOrder(h);
  design4=horzcat(design4,des(:,k));
end

design4;
end


%-------------------------------------
%Matriz ortogonal
%-------------------------------------
function [design]=callback_design(varargin)
s=varargin{3};
global tbl3 linha coluna popup1 popup2 des bg4 intOK L_name nameExp
check5 MIX design

MIX=0;

valpopup1=get(popup1,'value'); %fatores
valpopup2=get(popup2,'value'); %niveis

if (valpopup1==1) || (valpopup2==1)

  msgbox('Escolha o nível e a quantidade de fatores.');
  return

elseif (valpopup1<=31) && (valpopup2==2) %2 niveis

  s.fig(4)=figure('position',[200 200 200 300],...
  'menubar','none',...
  'name','Matriz Ortogonal',...
  'numbertitle','off',...
  'resize','off');

  bg4=uibuttongroup(s.fig(4),...
    'title','Escolha a Matriz');

  s.r1=uicontrol(bg4,...
    'style','radiobutton',...
    'string','L4 (2^3)',...
    'visible','off',...
```

```
  'position',[10 240 200 40]);

s.r2=uicontrol(bg4,...
  'style','radiobutton',...
  'string','L8 (2^7)',...
  'visible','off',...
  'position',[10 200 200 40]);

s.r3=uicontrol(bg4,...
  'style','radiobutton',...
  'string','L12 (2^11)',...
  'visible','off',...
  'position',[10 160 200 40]);

s.r4=uicontrol(bg4,...
  'style','radiobutton',...
  'string','L16 (2^15)',...
  'visible','off',...
  'position',[10 120 200 40]);

s.r5=uicontrol(bg4,...
  'style','radiobutton',...
  'string','L32 (2^31)',...
  'visible','off',...
  'position',[10 80 200 40]);

set(bg4,'SelectedObject',[]);

if valpopup1 <=3
  set(s.r1,'visible','on');
  set(s.r2,'visible','on');
  set(s.r3,'visible','on');
  set(s.r4,'visible','on');
  set(s.r5,'visible','on');
elseif valpopup1 <=7
  set(s.r2,'visible','on');
  set(s.r3,'visible','on');
  set(s.r4,'visible','on');
  set(s.r5,'visible','on');
elseif valpopup1 <=11
  set(s.r3,'visible','on');
  set(s.r4,'visible','on');
  set(s.r5,'visible','on');
 elseif valpopup1 <=15
  set(s.r4,'visible','on');
  set(s.r5,'visible','on');
elseif valpopup1 <= 31
  set(s.r5,'visible','on');
end

waitfor(bg4,'SelectedObject');
opdesign1=get(bg4,'SelectedObject');
L_name=get(opdesign1,'string');

switch L_name

  case 'L4 (2^3)'
  design = [
    1 1 1;
    1 2 2;
```

```
  2 1 2;
  2 2 1;
  ];

linha=4;
coluna=3;
des=design;
escolha_vetor
name
vetor_interacao

case 'L8 (2^7)'
design = [
  1 1 1 1 1 1 1;
  1 1 1 2 2 2 2;
  1 2 2 1 1 2 2;
  1 2 2 2 2 1 1;
  2 1 2 1 2 1 2;
  2 1 2 2 1 2 1;
  2 2 1 1 2 2 1;
  2 2 1 2 1 1 2;
  ];

linha=8;
coluna=7;
des=design;
escolha_vetor
name
vetor_interacao

case 'L12 (2^11)'
design = [
  1 1 1 1 1 1 1 1 1 1 1;
  1 1 1 1 1 2 2 2 2 2 2;
  1 1 2 2 2 1 1 1 2 2 2;
  1 2 1 2 2 1 2 2 1 1 2;
  1 2 2 1 2 2 1 2 1 2 1;
  1 2 2 2 1 2 2 1 2 1 1;
  2 1 2 2 1 1 2 2 1 2 1;
  2 1 2 1 2 2 2 1 1 1 2;
  2 1 1 2 2 2 1 2 2 1 1;
  2 2 2 1 1 1 1 2 2 1 2;
  2 2 1 2 1 2 1 1 1 2 2;
  2 2 1 1 2 1 2 1 2 2 1;
  ];

linha=12;
coluna=11;
des=design;
escolha_vetor
name
vetor_interacao

case 'L16 (2^15)'
design = [
  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
  1 1 1 1 1 1 1 2 2 2 2 2 2 2 2;
  1 1 1 2 2 2 2 1 1 1 1 2 2 2 2;
  1 1 1 2 2 2 2 2 2 2 2 1 1 1 1;
  1 2 2 1 1 2 2 1 1 2 2 1 1 2 2;
```

```
    1 2 2 1 1 2 2 2 2 1 1 2 2 1 1;
    1 2 2 2 2 1 1 1 1 2 2 2 2 1 1;
    1 2 2 2 2 1 1 2 2 1 1 1 1 2 2;
    2 1 2 1 2 1 2 1 2 1 2 1 2 1 2;
    2 1 2 1 2 1 2 2 1 2 1 2 1 2 1;
    2 1 2 2 1 2 1 1 2 1 2 2 1 2 1;
    2 1 2 2 1 2 1 2 1 2 1 1 2 1 2;
    2 2 1 1 2 2 1 1 2 2 1 1 2 2 1;
    2 2 1 1 2 2 1 2 1 1 2 2 1 1 2;
    2 2 1 2 1 1 2 1 2 2 1 2 1 1 2;
    2 2 1 2 1 1 2 2 1 1 2 1 2 2 1;
    ];

  linha=16;
  coluna=15;
  des=design;
  escolha_vetor
  name

  case 'L32 (2^31)'
  design = [
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2;
1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2;
1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1;
1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1 2 2 2 2;
1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1;
1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 1 1 1 1;
1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2;
1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2;
1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1;
1 2 2 1 1 2 2 2 2 1 1 2 2 1 1 1 1 2 2 1 1 2 2 2 2 1 1 2 2 1 1;
1 2 2 1 1 2 2 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 1 1 2 2 1 1 2 2;
1 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 1 1;
1 2 2 2 2 1 1 1 1 2 2 2 2 1 1 2 2 1 1 1 1 2 2 2 2 1 1 1 1 2 2;
1 2 2 2 2 1 1 2 2 1 1 1 1 2 2 1 1 2 2 2 2 1 1 2 2 1 1 1 1 2 2;
1 2 2 2 2 1 1 2 2 1 1 1 1 2 2 2 2 1 1 1 1 2 2 1 1 2 2 2 2 1 1;
2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2;
2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1;
2 1 2 1 2 1 2 2 1 2 1 2 1 2 1 1 2 1 2 1 2 1 2 2 1 2 1 2 1 2 1;
2 1 2 1 2 1 2 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 1 2 1 2 1 2 1 2;
2 1 2 2 1 2 1 1 2 1 2 2 1 2 1 1 2 1 2 2 1 2 1 1 2 1 2 2 1 2 1;
2 1 2 2 1 2 1 1 2 1 2 2 1 2 1 2 1 2 1 1 2 1 2 2 1 2 1 1 2 1 2;
2 1 2 2 1 2 1 2 1 2 1 1 2 1 2 1 2 1 2 2 1 2 1 2 1 2 1 1 2 1 2;
2 1 2 2 1 2 1 2 1 2 1 1 2 1 2 2 1 2 1 1 2 1 2 1 2 1 2 2 1 2 1;
2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1;
2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2;
2 2 1 1 2 2 1 2 1 1 2 2 1 1 2 1 2 2 1 1 2 2 1 2 1 1 2 2 1 1 2;
2 2 1 1 2 2 1 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 1 2 2 1 1 2 2 1;
2 2 1 2 1 1 2 1 2 2 1 2 1 1 2 1 2 2 1 2 1 1 2 1 2 2 1 2 1 1 2;
2 2 1 2 1 1 2 1 2 2 1 2 1 1 2 2 1 1 2 1 2 2 1 2 1 1 2 1 2 2 1;
2 2 1 2 1 1 2 2 1 1 2 1 2 2 1 1 2 2 1 2 1 1 2 2 1 1 2 1 2 2 1;
2 2 1 2 1 1 2 2 1 1 2 1 2 2 1 2 1 1 2 1 2 2 1 1 2 2 1 2 1 1 2;
];

linha=32;
coluna=31;
des=design;
escolha_vetor
name
vetor_interacao
```

```
  end
  close(s.fig(4));
elseif (valpopup1<=31) && (valpopup2==3) %3 niveis

  s.fig(4)=figure('position',[200 200 200 300],...
  'menubar','none',...
  'name','Matriz Ortogonal',...
  'numbertitle','off',...
  'resize','off');

  bg4=uibuttongroup(s.fig(4),...
  'title','Escolha a Matriz');

  s.r1=uicontrol(bg4,...
  'style','radiobutton',...
  'string','L9 (3^4)',...
  'visible','off',...
  'position',[10 240 200 40]);

  s.r2=uicontrol(bg4,...
  'style','radiobutton',...
  'string','L27 (3^13)',...
  'visible','off',...
  'position',[10 200 200 40]);

  set(bg4,'SelectedObject',[]);

  if valpopup1 <=4
  set(s.r1,'visible','on');
  set(s.r2,'visible','on');
  elseif valpopup1 <= 13
  set(s.r2,'visible','on');
  end

  waitfor(bg4,'SelectedObject');
  opdesign1=get(bg4,'SelectedObject');
  L_name=get(opdesign1,'string');
  close(s.fig(4));

  switch L_name

  case 'L9 (3^4)'
  design = [
  1 1 1 1;
  1 2 2 2;
  1 3 3 3;
  2 1 2 3;
  2 2 3 1;
  2 3 1 2;
  3 1 3 2;
  3 2 1 3;
  3 3 2 1;
  ];

  linha=9;
  coluna=4;
  des=design;
  escolha_vetor
```

```
  name
  vetor_interacao


  case 'L27 (3^13)'
  design = [
  1 1 1 1 1 1 1 1 1 1 1 1 1;
  1 1 1 1 2 2 2 2 2 2 2 2 2;
  1 1 1 1 3 3 3 3 3 3 3 3 3;
  1 2 2 2 1 1 1 2 2 2 3 3 3;
  1 2 2 2 2 2 2 3 3 3 1 1 1;
  1 2 2 2 3 3 3 1 1 1 2 2 2;
  1 3 3 3 1 1 1 3 3 3 2 2 2;
  1 3 3 3 2 2 2 1 1 1 3 3 3;
  1 3 3 3 3 3 3 2 2 2 1 1 1;
  2 1 2 3 1 2 3 1 2 3 1 2 3;
  2 1 2 3 2 3 1 2 3 1 2 3 1;
  2 1 2 3 3 1 2 3 1 2 3 1 2;
  2 2 3 1 1 2 3 2 3 1 3 1 2;
  2 2 3 1 2 3 1 3 1 2 1 2 3;
  2 2 3 1 3 1 2 1 2 3 2 3 1;
  2 3 1 2 1 2 3 3 1 2 2 3 1;
  2 3 1 2 2 3 1 1 2 3 3 1 2;
  2 3 1 2 3 1 2 2 3 1 1 2 3;
  3 1 3 2 1 3 2 1 3 2 1 3 2;
  3 1 3 2 2 1 3 2 1 3 2 1 3;
  3 1 3 2 3 2 1 3 2 1 3 2 1;
  3 2 1 3 1 3 2 2 1 3 3 2 1;
  3 2 1 3 2 1 3 3 2 1 1 3 2;
  3 2 1 3 3 2 1 1 3 2 2 1 3;
  3 3 2 1 1 3 2 3 2 1 2 1 3;
  3 3 2 1 2 1 3 1 3 2 3 2 1;
  3 3 2 1 3 2 1 2 1 3 1 3 2;
  ];

  linha=27;
  coluna=13;
  des=design;
  escolha_vetor
  name
  vetor_interacao

  end

elseif (valpopup1<=31) && (valpopup2==4) %4niveis

  s.fig(4)=figure('position',[200 200 200 300],...
  'menubar','none',...
  'name','Matriz Ortogonal',...
  'numbertitle','off',...
  'resize','off');

  bg4=uibuttongroup(s.fig(4),...
  'title','Escolha a Matriz');

  s.r1=uicontrol(bg4,...
  'style','radiobutton',...
  'string','L16 (4^5)',...
  'position',[10 240 200 40]);
```

```
  set(bg4,'SelectedObject',[]);
  waitfor(bg4,'SelectedObject');
  opdesign1=get(bg4,'SelectedObject');
  L_name=get(opdesign1,'string');
  close(s.fig(4));


  switch L_name

  case 'L16 (4^5)'
  design = [
  1 1 1 1 1;
  1 2 2 2 2;
  1 3 3 3 3;
  1 4 4 4 4;
  2 1 2 3 4;
  2 2 1 4 3;
  2 3 4 1 2;
  2 4 3 2 1;
  3 1 3 4 2;
  3 2 4 3 1;
  3 3 1 2 4;
  3 4 2 1 3;
  4 1 4 2 3;
  4 2 3 1 4;
  4 3 2 4 1;
  4 4 1 3 2;

  ];
  linha=16;
  coluna=5;
  des=design;
  escolha_vetor
  name
  vetor_interacao

  end

elseif (valpopup1<=31) && (valpopup2==5) %5niveis

  s.fig(4)=figure('position',[200 200 200 300],...
  'menubar','none',...
  'name','Matriz Ortogonal',...
  'numbertitle','off',...
  'resize','off');

  bg4=uibuttongroup(s.fig(4),...
  'title','Escolha a Matriz');

  s.r1=uicontrol(bg4,...
  'style','radiobutton',...
  'string','L25 (5^6)',...
  'position',[10 240 200 40]);

  set(bg4,'SelectedObject',[]);
  waitfor(bg4,'SelectedObject');
  opdesign1=get(bg4,'SelectedObject');
  L_name=get(opdesign1,'string');
  close(s.fig(4));
```

```
  switch L_name

  case 'L25 (5^6)'
  design = [
  1 1 1 1 1 1;
  1 2 2 2 2 2;
  1 3 3 3 3 3;
  1 4 4 4 4 4;
  1 5 5 5 5 5;
  2 1 2 3 4 5;
  2 2 3 4 5 1;
  2 3 4 5 1 2;
  2 4 5 1 2 3;
  2 5 1 2 3 4;
  3 1 3 5 2 4;
  3 2 4 1 3 5;
  3 3 5 2 4 1;
  3 4 1 3 5 2;
  3 5 2 4 1 3;
  4 1 4 2 5 3;
  4 2 5 3 1 4;
  4 3 1 4 2 5;
  4 4 2 5 3 1;
  4 5 3 1 4 2;
  5 1 5 4 3 2;
  5 2 1 5 4 3;
  5 3 2 1 5 4;
  5 4 3 2 1 5;
  5 5 4 3 2 1;
  ];
  linha=25;
  coluna=6;
  des=design;
  escolha_vetor
  name
  vetor_interacao

  end

elseif (valpopup1<=31) && (valpopup2==6) %mix

  s.fig(4)=figure('position',[200 200 200 300],...
  'menubar','none',...
  'name','Matriz Ortogonal',...
  'numbertitle','off',...
  'resize','off');

  bg4=uibuttongroup(s.fig(4),...
  'title','Escolha a Matriz');

  s.r1=uicontrol(bg4,...
  'style','radiobutton',...
  'string','L18 (2 x 3^7)',...
  'visible','off',...
  'position',[10 240 200 40]);

  s.r2=uicontrol(bg4,...
  'style','radiobutton',...
  'string','L32 (2 x 4^9)',...
```

```
'visible','off',...
'position',[10 200 200 40]);

s.r3=uicontrol(bg4,...
'style','radiobutton',...
'string','L36 (2^11 x 3^12)',...
'visible','off',...
'position',[10 160 200 40]);

s.r4=uicontrol(bg4,...
'style','radiobutton',...
'string','L50 (2 x 5^11)',...
'visible','off',...
'position',[10 120 200 40]);

s.r5=uicontrol(bg4,...
'style','radiobutton',...
'string','L54 (2 x 3^25)',...
'visible','off',...
'position',[10 80 200 40]);

set(bg4,'SelectedObject',[]);

if valpopup1 <=8
      set(s.r1,'visible','on');
      set(s.r2,'visible','on');
      set(s.r3,'visible','on');
      set(s.r4,'visible','on');
      set(s.r5,'visible','on');
elseif valpopup1 <=10
      set(s.r2,'visible','on');
      set(s.r3,'visible','on');
      set(s.r4,'visible','on');
      set(s.r5,'visible','on');
elseif valpopup1 <=12
      set(s.r3,'visible','on');
      set(s.r4,'visible','on');
      set(s.r5,'visible','on');
elseif valpopup1 <= 23
      set(s.r3,'visible','on');
      set(s.r5,'visible','on');
elseif valpopup1 <= 26
      set(s.r5,'visible','on');
end

waitfor(bg4,'SelectedObject');
opdesign1=get(bg4,'SelectedObject');
L_name=get(opdesign1,'string');
close(s.fig(4));

switch L_name

case 'L18 (2 x 3^7)'
design = [
1 1 1 1 1 1 1 1;
1 1 2 2 2 2 2 2;
1 1 3 3 3 3 3 3;
1 2 1 1 2 2 3 3;
1 2 2 2 3 3 1 1;
1 2 3 3 1 1 2 2;
```

```
1 3 1 2 1 3 2 3;
1 3 2 3 2 1 3 1;
1 3 3 1 3 2 1 2;
2 1 1 3 3 2 2 1;
2 1 2 1 1 3 3 2;
2 1 3 2 2 1 1 3;
2 2 1 2 3 1 3 2;
2 2 2 3 1 2 1 3;
2 2 3 1 2 3 2 1;
2 3 1 3 2 3 1 2;
2 3 2 1 3 1 2 3;
2 3 3 2 1 2 3 1;
];

linha=18;
coluna=8;
des=design;
escolha_vetor
name
vetor_interacao
MIX=1;

case 'L32 (2 x 4^9)'
design = [
1 1 1 1 1 1 1 1 1;
1 1 2 2 2 2 2 2 2;
1 1 3 3 3 3 3 3 3;
1 1 4 4 4 4 4 4 4;
1 2 1 1 2 2 3 3 4 4;
1 2 2 2 1 1 4 4 3 3;
1 2 3 3 4 4 1 1 2 2;
1 2 4 4 3 3 2 2 1 1;
1 3 1 2 3 4 1 2 3 4;
1 3 2 1 4 3 2 1 4 3;
1 3 3 4 1 2 3 4 1 2;
1 3 4 3 2 1 4 3 2 1;
1 4 1 2 4 3 3 4 2 1;
1 4 2 1 3 4 4 3 1 2;
1 4 3 4 2 1 1 2 4 3;
1 4 4 3 1 2 2 1 3 4;
2 1 1 4 1 4 2 3 2 3;
2 1 2 3 2 3 1 4 1 4;
2 1 3 2 3 2 4 1 4 1;
2 1 4 1 4 1 3 2 3 2;
2 2 1 4 2 3 4 1 3 2;
2 2 2 3 1 4 3 2 4 1;
2 2 3 2 4 1 2 3 1 4;
2 2 4 1 3 2 1 4 2 3;
2 3 1 3 3 1 2 4 4 2;
2 3 2 4 4 2 1 3 3 1;
2 3 3 1 1 3 4 2 2 4;
2 3 4 2 2 4 3 1 1 3;
2 4 1 3 4 2 4 2 1 3;
2 4 2 4 3 1 3 1 2 4;
2 4 3 1 2 4 2 4 3 1;
2 4 4 2 1 3 1 3 4 2;
];
linha=32;
coluna=10;
des=design;
escolha_vetor
```

```
name
vetor_interacao
MIX=1;

case 'L36 (2^11 x 3^12)'
design = [
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2;
1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3;
1 1 1 1 1 2 2 2 2 2 2 1 1 1 1 2 2 2 2 3 3 3 3;
1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 1 1 1 1;
1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 1 1 1 1 2 2 2 2;
1 1 2 2 2 1 1 1 2 2 2 1 1 2 3 1 2 3 3 1 2 2 3;
1 1 2 2 2 1 1 1 2 2 2 2 2 3 1 2 3 1 1 2 3 3 1;
1 1 2 2 2 1 1 1 2 2 2 3 3 1 2 3 1 2 2 3 1 1 2;
1 2 1 2 2 1 2 2 1 1 2 1 1 3 2 1 3 2 3 2 1 3 2;
1 2 1 2 2 1 2 2 1 1 2 2 2 1 3 2 1 3 1 3 2 1 3;
1 2 1 2 2 1 2 2 1 1 2 3 3 2 1 3 2 1 2 1 3 2 1;
1 2 2 1 2 2 1 2 1 2 1 1 2 3 1 3 2 1 3 3 2 1 2;
1 2 2 1 2 2 1 2 1 2 1 2 3 1 2 1 3 2 1 1 3 2 3;
1 2 2 1 2 2 1 2 1 2 1 3 1 2 3 2 1 3 2 2 1 3 1;
1 2 2 2 1 2 2 1 2 1 1 1 2 3 2 1 1 3 2 3 3 2 1;
1 2 2 2 1 2 2 1 2 1 1 2 3 1 3 2 2 1 3 1 1 3 2;
1 2 2 2 1 2 2 1 2 1 1 3 1 2 1 3 3 2 1 2 2 1 3;
2 1 2 2 1 1 2 2 1 2 1 1 2 1 3 3 3 1 2 2 1 2 3;
2 1 2 2 1 1 2 2 1 2 1 2 3 2 1 1 1 2 3 3 2 3 1;
2 1 2 2 1 1 2 2 1 2 1 3 1 3 2 2 2 3 1 1 3 1 2;
2 1 2 1 2 2 2 1 1 1 2 1 2 2 3 3 1 2 1 1 3 3 2;
2 1 2 1 2 2 2 1 1 1 2 2 3 3 1 1 2 3 2 2 1 1 3;
2 1 2 1 2 2 2 1 1 1 2 3 1 1 2 2 3 1 3 3 2 2 1;
2 1 1 2 2 2 1 2 2 1 1 1 3 2 1 2 3 3 1 3 1 2 2;
2 1 1 2 2 2 1 2 2 1 1 2 1 3 2 3 1 1 2 1 2 3 3;
2 1 1 2 2 2 1 2 2 1 1 3 2 1 3 1 2 2 3 2 3 1 1;
2 2 2 1 1 1 1 2 2 1 2 1 3 2 2 2 1 1 3 2 3 1 3;
2 2 2 1 1 1 1 2 2 1 2 2 1 3 3 3 2 2 1 3 1 2 1;
2 2 2 1 1 1 1 2 2 1 2 3 2 1 1 1 3 3 2 1 2 3 2;
2 2 1 2 1 2 1 1 1 2 2 1 3 3 3 2 3 2 2 1 2 1 1;
2 2 1 2 1 2 1 1 1 2 2 2 1 1 1 3 1 3 3 2 3 2 2;
2 2 1 2 1 2 1 1 1 2 2 3 2 2 1 2 1 1 3 1 1 3 3;
2 2 1 1 2 1 2 1 2 2 1 1 3 1 2 3 2 3 1 2 2 3 1;
2 2 1 1 2 1 2 1 2 2 1 2 1 2 3 1 3 1 2 3 3 1 2;
2 2 1 1 2 1 2 1 2 2 1 3 2 3 1 2 1 2 3 1 1 2 3;
];
linha=36;
coluna=23;
des=design;
escolha_vetor
name
vetor_interacao

case  'L50 (2 x 5^11)'
design = [
1 1 1 1 1 1 1 1 1 1 1 1;
1 1 2 2 2 2 2 2 2 2 2 2;
1 1 3 3 3 3 3 3 3 3 3 3;
1 1 4 4 4 4 4 4 4 4 4 4;
1 1 5 5 5 5 5 5 5 5 5 5;
1 2 1 2 3 4 5 1 2 3 4 5;
1 2 2 3 4 5 1 2 3 4 5 1;
1 2 3 4 5 1 2 3 4 5 1 2;
1 2 4 5 1 2 3 4 5 1 2 3;
```

```
1 2 5 1 2 3 4 5 1 2 3 4;
1 3 1 3 5 2 4 4 1 3 5 2;
1 3 2 4 1 3 5 5 2 4 1 3;
1 3 3 5 2 4 1 1 3 5 2 4;
1 3 4 1 3 5 2 2 4 1 3 5;
1 3 5 2 4 1 3 3 5 2 4 1;
1 4 1 4 2 5 3 5 3 1 4 2;
1 4 2 5 3 1 4 1 4 2 5 3;
1 4 3 1 4 2 5 2 5 3 1 4;
1 4 4 2 5 3 1 3 1 4 2 5;
1 4 5 3 1 4 2 4 2 5 3 1;
1 5 1 5 4 3 2 4 3 2 1 5;
1 5 2 1 5 4 3 5 4 3 2 1;
1 5 3 2 1 5 4 1 5 4 3 2;
1 5 4 3 2 1 5 2 1 5 4 3;
1 5 5 4 3 2 1 3 2 1 5 4;
2 1 1 1 4 5 4 3 2 5 2 3;
2 1 2 2 5 1 5 4 3 1 3 4;
2 1 3 3 1 2 1 5 4 2 4 5;
2 1 4 4 2 3 2 1 5 3 5 1;
2 1 5 5 3 4 3 2 1 4 1 2;
2 2 1 2 1 3 3 2 4 5 5 4;
2 2 2 3 2 4 4 3 5 1 1 5;
2 2 3 4 3 5 5 4 1 2 2 1;
2 2 4 5 4 1 1 5 2 3 3 2;
2 2 5 1 5 2 2 1 3 4 4 3;
2 3 1 3 3 1 2 5 5 4 2 4;
2 3 2 4 4 2 3 1 1 5 3 5;
2 3 3 5 5 3 4 2 2 1 4 1;
2 3 4 1 1 4 5 3 3 2 5 2;
2 3 5 2 2 5 1 4 4 3 1 3;
2 4 1 4 5 4 1 2 5 2 3 3;
2 4 2 5 1 5 2 3 1 3 4 4;
2 4 3 1 2 1 3 4 2 4 5 5;
2 4 4 2 3 2 4 5 3 5 1 1;
2 4 5 3 4 3 5 1 4 1 2 2;
2 5 1 5 2 2 5 3 4 4 3 1;
2 5 2 1 3 3 1 4 5 5 4 2;
2 5 3 2 4 4 2 5 1 1 5 3;
2 5 4 3 5 5 3 1 2 2 1 4;
2 5 5 4 1 1 4 2 3 3 2 5;
];
linha=50;
coluna=12;
des=design;
escolha_vetor
name
vetor_interacao
MIX=1;

case 'L54 (2 x 3^25)'
design = [
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;
1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2;
1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3;
1 1 2 2 2 2 2 2 1 1 1 1 1 1 2 3 2 3 2 3 2 3 2 3 2;
1 1 2 2 2 2 2 2 2 2 2 2 2 2 3 1 3 1 3 1 3 1 3 1 3;
1 1 2 2 2 2 2 2 3 3 3 3 3 3 1 2 1 2 1 2 1 2 1 2 1;
1 1 3 3 3 3 3 3 1 1 1 1 1 1 3 2 3 2 3 2 3 2 3 2 3;
1 1 3 3 3 3 3 3 2 2 2 2 2 2 1 3 1 3 1 3 1 3 1 3 1;
1 1 3 3 3 3 3 3 3 3 3 3 3 3 2 1 2 1 2 1 2 1 2 1 2;
```

```
  1 2 1 1 2 2 3 3 1 1 2 2 3 3 1 1 1 1 2 3 2 3 3 2 3 2;
  1 2 1 1 2 2 3 3 2 2 3 3 1 1 2 2 2 2 3 1 3 1 1 3 1 3;
  1 2 1 1 2 2 3 3 3 3 1 1 2 2 3 3 3 3 1 2 1 2 2 1 2 1;
  1 2 2 2 3 3 1 1 1 1 2 2 3 3 2 3 2 3 3 2 3 2 1 1 1 1;
  1 2 2 2 3 3 1 1 2 2 3 3 1 1 3 1 3 1 1 3 1 3 2 2 2 2;
  1 2 2 2 3 3 1 1 3 3 1 1 2 2 1 2 1 2 2 1 2 1 3 3 3 3;
  1 2 3 3 1 1 2 2 1 1 2 2 3 3 3 2 3 2 1 1 1 1 2 3 2 3;
  1 2 3 3 1 1 2 2 2 2 3 3 1 1 1 3 1 3 2 2 2 2 3 1 3 1;
  1 2 3 3 1 1 2 2 3 3 1 1 2 2 2 1 2 1 3 3 3 3 1 2 1 2;
  1 3 1 2 1 3 2 3 1 2 1 3 2 3 1 1 2 3 1 1 3 2 2 3 3 2;
  1 3 1 2 1 3 2 3 2 3 2 1 3 1 2 2 3 1 2 2 1 3 3 1 1 3;
  1 3 1 2 1 3 2 3 3 1 3 2 1 2 3 3 1 2 3 3 2 1 1 2 2 1;
  1 3 2 3 2 1 3 1 1 2 1 3 2 3 2 3 3 2 2 3 1 1 3 2 1 1;
  1 3 2 3 2 1 3 1 2 3 2 1 3 1 3 1 1 3 3 1 2 2 1 3 2 2;
  1 3 2 3 2 1 3 1 3 1 3 2 1 2 1 2 2 1 1 2 3 3 2 1 3 3;
  1 3 3 1 3 2 1 2 1 2 1 3 2 3 3 2 1 1 3 2 2 3 1 1 2 3;
  1 3 3 1 3 2 1 2 2 3 2 1 3 1 1 3 2 2 1 3 3 1 2 2 3 1;
  1 3 3 1 3 2 1 2 3 1 3 2 1 2 2 1 3 3 2 1 1 2 3 3 1 2;
  2 1 1 3 3 2 2 1 1 3 3 2 2 1 1 1 3 2 3 2 2 3 2 3 1 1;
  2 1 1 3 3 2 2 1 2 1 1 3 3 2 2 2 1 3 1 3 3 1 3 1 2 2;
  2 1 1 3 3 2 2 1 3 2 2 1 1 3 3 3 2 1 2 1 1 2 1 2 3 3;
  2 1 2 1 1 3 3 2 1 3 3 2 2 1 2 3 1 1 1 1 3 2 3 2 2 3;
  2 1 2 1 1 3 3 2 2 1 1 3 3 2 3 1 2 2 2 2 1 3 1 3 3 1;
  2 1 2 1 1 3 3 2 3 2 2 1 1 3 1 2 3 3 3 3 2 1 2 1 1 2;
  2 1 3 2 2 1 1 3 1 3 3 2 2 1 3 2 2 3 2 3 1 1 1 1 3 2;
  2 1 3 2 2 1 1 3 2 1 1 3 3 2 1 3 3 1 3 1 2 2 2 2 1 3;
  2 1 3 2 2 1 1 3 3 2 2 1 1 3 2 1 1 2 1 2 3 3 3 3 2 1;
  2 2 1 2 3 1 3 2 1 2 3 1 3 2 1 1 2 3 3 2 1 1 3 2 2 3;
  2 2 1 2 3 1 3 2 2 3 1 2 1 3 2 2 3 1 1 3 2 2 1 3 3 1;
  2 2 1 2 3 1 3 2 3 1 2 3 2 1 3 3 1 2 2 1 3 3 2 1 1 2;
  2 2 2 3 1 2 1 3 1 2 3 1 3 2 2 3 3 2 1 1 2 3 1 1 3 2;
  2 2 2 3 1 2 1 3 2 3 1 2 1 3 3 1 1 3 2 2 3 1 2 2 1 3;
  2 2 2 3 1 2 1 3 3 1 2 3 2 1 1 2 2 1 3 3 1 2 3 3 2 1;
  2 2 3 1 2 3 2 1 1 2 3 1 3 2 3 2 1 1 2 3 3 2 2 3 1 1;
  2 2 3 1 2 3 2 1 2 3 1 2 1 3 1 3 2 2 3 1 1 3 3 1 2 2;
  2 2 3 1 2 3 2 1 3 1 2 3 2 1 2 1 3 3 1 2 2 1 1 2 3 3;
  2 3 1 3 2 3 1 2 1 3 2 3 1 2 1 1 3 2 2 3 3 2 1 1 2 3;
  2 3 1 3 2 3 1 2 2 1 3 1 2 3 2 2 1 3 3 1 1 3 2 2 3 1;
  2 3 1 3 2 3 1 2 3 2 1 2 3 1 3 3 2 1 1 2 2 1 3 3 1 2;
  2 3 2 1 3 1 2 3 1 3 2 3 1 2 2 3 1 1 3 2 1 1 2 3 3 2;
  2 3 2 1 3 1 2 3 2 1 3 1 2 3 3 1 2 2 1 3 2 2 3 1 1 3;
  2 3 2 1 3 1 2 3 3 2 1 2 3 1 1 2 3 3 2 1 3 3 1 2 2 1;
  2 3 3 2 1 2 3 1 1 3 2 3 1 2 3 2 2 3 1 1 2 3 3 2 1 1;
  2 3 3 2 1 2 3 1 2 1 3 1 2 3 1 3 3 1 2 2 3 1 1 3 2 2;
  2 3 3 2 1 2 3 1 3 2 1 2 3 1 2 1 1 2 3 3 1 2 2 1 3 3;
  ];

  linha=54;
  coluna=26;
  des=design;
  escolha_vetor
  name
  vetor_interacao

 end

else

end
```

```
set(tbl3,'data',design);
[r1,c1]=size(design);
set(tbl3,'columnname',nameExp(1:c1));
intOK=1;
set(check5,'value',0);

end



%-----------------------------
%menu ajuda
%-----------------------------
function ajuda(varargin)
global color

HelpFig=figure('position',[100 100 150 500],...
  'menubar','none',...
  'name','Ajuda',...
  'numbertitle','off',...
  'resize','off');

txt0=uicontrol('Style','text',...
 'String','Taguchi Design',...
 'ButtonDownFcn',@click2,...
 'Enable', 'Inactive',...
 'position',[0 450 130 40],...
 'backgroundcolor', color,...
 'visible','off');

txt1=uicontrol('Style','text',...
 'String','Amostra',...
 'ButtonDownFcn',@click2,...
 'Enable', 'Inactive',...
 'position',[0 420 130 40],...
 'backgroundcolor', color);

txt2=uicontrol('Style','text',...
 'String','Níveis/Fatores',...
 'ButtonDownFcn',@click2,...
 'Enable', 'Inactive',...
 'position',[0 380 130 40],...
 'backgroundcolor', color);

txt3=uicontrol('Style','text',...
 'String','Matriz ortogonal',...
 'ButtonDownFcn',@click2,...
 'Enable', 'Inactive',...
 'position',[0 340 130 40],...
 'backgroundcolor', color);

txt4=uicontrol('Style','text',...
 'String','Interaçao',...
 'ButtonDownFcn',@click2,...
 'Enable', 'Inactive',...
 'position',[0 300 130 40],...
 'backgroundcolor', color);

txt5=uicontrol('Style','text',...
 'String','Razão sinal/ruido',...
```

```
 'ButtonDownFcn',@click2,...
 'Enable', 'Inactive',...
 'position',[0 260 130 40],...
 'backgroundcolor', color);

txt6=uicontrol('Style','text',...
 'String','Função perda',...
 'ButtonDownFcn',@click2,...
 'Enable', 'Inactive',...
 'position',[0 220 130 40],...
 'backgroundcolor', color);

txt7=uicontrol('Style','text',...
 'String','Capacidade de processo',...
 'ButtonDownFcn',@click2,...
 'Enable', 'Inactive',...
 'position',[0 180 130 40],...
 'backgroundcolor', color);

txt8=uicontrol('Style','text',...
 'String','Intervalo de confiança',...
 'ButtonDownFcn',@click2,...
 'Enable', 'Inactive',...
 'position',[0 140 130 40],...
 'backgroundcolor', color);

txt9=uicontrol('Style','text',...
 'String','ANOVA',...
 'ButtonDownFcn',@click2,...
 'Enable', 'Inactive',...
 'position',[0 100 130 40],...
 'backgroundcolor', color);

txt10=uicontrol('Style','text',...
 'String','Gráficos',...
 'ButtonDownFcn',@click2,...
 'Enable', 'Inactive',...
 'position',[0 65 130 40],...
 'backgroundcolor', color);

txt11=uicontrol('Style','text',...
 'String','Exportação',...
 'ButtonDownFcn',@click2,...
 'Enable', 'Inactive',...
 'position',[0 30 130 40],...
 'backgroundcolor', color);

txt12=uicontrol('Style','text',...
 'String','NaN/Inf',...
 'ButtonDownFcn',@click2,...
 'Enable', 'Inactive',...
 'position',[0 0 130 40],...
 'backgroundcolor', color);

end


%--------------------------------
%Texto menu  ajuda
%--------------------------------
```

```
function click2(varargin)
h=get(gco,'string');

switch h
  case 'Taguchi Design'
  text={};
  helpfig2(text,'Ajuda: Sobre Taguchi');

  case 'Amostra'
  text={'Janela principal > Digite no campo "Amostra Mestre" OU aperte
"importar dados"','','Digitar dados','Exemplo: 34 24; 23 11; 12 20; 12 24'...
  'Obs: Para separar os dados horizontalmente deve se deixar um espaço
entre as observações, para separar os dados verticalmente coloque " ; ".',...
  '','Importar dados','O programa aceita a importação de dados através
de arquivos .XLS, basta clicar no botão "Importar Dados" e selecionar o
arquivo que deseja',...
  'Obs: Para leitura do arquivo corretamente no arquivo .XLS os número
devem ser escritos com " , " em vez de " . " ',...
  '','Entrada de dados DEVE SER BALANCEADA','Exemplo dados não
balanceados: 2 1 3; 2 1; 2 3 4 1; 3 4 1 2',....
  'Para deixar eles balanceados utilize "NaN" nos dados que faltam',
'Exemplo dados balanceados: 2 1 3 NaN; 2 1 NaN NaN; 2 3 4 1; 3 4 1 2'};
  helpfig2(text,'Ajuda: Amostra');

  case 'Níveis/Fatores'
  text={'Janela principal > Níveis > Fatores', '', 'A tabela abaixo
mostra os designs disponíveis de acordo com o nível e número de fatores',...
  '','---------------- número de níveis -------------------',...
  '','Design 2  3 4 5',...
  '','L4  2-3  ',...
  '','L8  2-7  ',...
  '','L12 2-11   ',...
  '','L16 2-15   ',...
  '','L32 2-31   ',...
  '','L9 2-4 ',...
  '','L27  2-13  ',...
  '','L16  2-5 ',...
  '','L25  2-6',...
  '','','A tabela a baixo mostra os designs disponíveis para nível
único de acordo com o nível e numero de fatores',...
  '','---------------- número de níveis -------------------',...
  '','Design 2  3 4 5',...
  '','L18  1 1-7 ',...
  '','L32  1 1-9 ',...
  '','L36 1-11 1-12  ',...
  '','L50  1   1-11',...
  '','L54  1 1-25'};
  helpfig2(text,'Ajuda: Níveis/Fatores');

  case 'Matriz ortogonal'
  text={' Pagina Inicial > Níveis > Fatores > Design',...
'','','L4 (2^3)',...
'','1 1 1',...
'','1 2 2',...
'','2 1 2',...
'','2 2 1',...
'','','L8 (2^7)',...
'','1 1 1 1 1 1 1',...
'','1 1 1 2 2 2 2',...
'','1 2 2 1 1 2 2',...
'','1 2 2 2 2 1 1',...
```

```
        '','2 1 2 1 2 1 2'...
        '','2 1 2 2 1 2 1',...
        '','2 2 1 1 2 2 1',...
        '','2 2 1 2 1 1 2',...
        '','','L12 (2^11)',...
        '','1 1 1 1 1 1 1 1 1 1 1',...
        '','1 1 1 1 1 2 2 2 2 2 2',...
        '','1 1 2 2 2 1 1 1 2 2 2',...
        '','1 2 1 2 2 1 2 2 1 1 2',...
        '','1 2 2 1 2 2 1 2 1 2 1',...
        '','1 2 2 2 1 2 2 1 2 1 1',...
        '','2 1 2 2 1 1 2 2 1 2 1',...
        '','2 1 2 1 2 2 2 1 1 1 2',...
        '','2 1 1 2 2 2 1 2 2 1 1',...
        '','2 2 2 1 1 1 1 2 2 1 2',...
        '','2 2 1 2 1 2 1 1 1 2 2',...
        '','2 2 1 1 2 1 2 1 2 2 1',...
        '','',' L16 (2^15)',...
        '','1 1 1 1 1 1 1 1 1 1 1 1 1 1 1',...
        '','1 1 1 1 1 1 1 1 2 2 2 2 2 2 2',...
        '','1 1 1 2 2 2 2 1 1 1 1 2 2 2 2',...
        '','1 1 1 2 2 2 2 2 2 2 2 1 1 1 1',...
        '','1 2 2 1 1 2 2 1 1 2 2 1 1 2 2',...
        '','1 2 2 1 1 2 2 2 2 1 1 2 2 1 1',....
        '','1 2 2 2 2 1 1 1 1 2 2 2 2 1 1',...
        '','1 2 2 2 2 1 1 2 2 1 1 1 1 2 2',...
        '','2 1 2 1 2 1 2 1 2 1 2 1 2 1 2',...
        '','2 1 2 1 2 1 2 2 1 2 1 2 1 2 1',...
        '','2 1 2 2 1 2 1 1 2 1 2 2 1 2 1',...
        '','2 1 2 2 1 2 1 2 1 2 1 1 2 1 2',...
        '','2 2 1 1 2 2 1 1 2 2 1 1 2 2 1',...
        '','2 2 1 1 2 2 1 2 1 1 2 2 1 1 2',...
        '','2 2 1 2 1 1 2 1 2 2 1 2 1 1 2',...
        '','2 2 1 2 1 1 2 2 1 1 2 1 2 2 1',...
        '','','L32(2^31)',...
'','1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1',...
'','1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2',...
'','1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2',...
'','1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1',...
'','1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 2',...
'','1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1 1',...
'','1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 1',...
'','1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 2',...
'','1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 2',...
'','1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 2 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1',...
'','1 2 2 1 1 2 2 2 2 2 1 1 2 2 1 1 1 1 2 2 1 1 2 2 2 2 1 1 2 2 1 1',...
'','1 2 2 1 1 2 2 2 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 1 1 1 2 2 1 1 2 2',...
'','1 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1 1 2 2 2 2 1 1 1',...
'','1 2 2 2 2 1 1 1 1 2 2 2 2 1 1 2 2 1 1 1 1 2 2 2 2 1 1 1 1 2 2 2',...
'','1 2 2 2 2 1 1 2 2 1 1 1 1 2 2 1 1 2 2 2 2 1 1 2 2 1 1 1 1 2 2 2',...
'','1 2 2 2 2 1 1 2 2 1 1 1 1 2 2 2 2 1 1 1 1 2 2 1 1 2 2 2 2 1 1 1',...
'','2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1',...
'','2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 1',...
'','2 1 2 1 2 1 2 2 1 2 1 2 1 2 1 1 2 1 2 1 2 1 2 2 1 2 1 2 1 2 1 1',...
'','2 1 2 1 2 1 2 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 1 2 1 2 1 2 1 2 1',...
'','2 1 2 2 1 2 1 1 2 1 2 2 1 2 1 1 2 1 2 2 1 2 1 1 2 1 2 2 1 2 1 1',...
'','2 1 2 2 1 2 1 1 2 1 2 2 1 2 1 2 1 2 1 1 2 1 2 2 1 2 1 1 2 1 2 1',...
'','2 1 2 2 1 2 1 2 1 2 1 1 2 1 2 1 2 1 2 2 1 2 1 2 1 2 1 1 2 1 2 1',...
'','2 1 2 2 1 2 1 2 1 2 1 1 2 1 2 2 1 2 1 1 2 1 2 1 2 1 2 2 1 2 1 1',...
'','2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1',...
'','2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 1',...
```

```
'','2 2 1 1 2 2 1 2 1 1 2 2 1 1 2 1 2 2 1 1 2 2 1 2 1 1 2 2 1 1 2',...
'','2 2 1 1 2 2 1 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 1 2 2 1 1 2 2 1',...
'','2 2 1 2 1 1 2 1 2 2 1 2 1 1 2 1 2 2 1 2 1 1 2 1 2 2 1 2 1 1 2',...
'','2 2 1 2 1 1 2 1 2 2 1 2 1 1 2 2 1 1 2 1 2 2 1 2 1 1 2 1 2 2 1',...
'','2 2 1 2 1 1 2 2 1 1 2 1 2 2 1 1 2 2 1 2 1 1 2 2 1 1 2 1 2 2 1',...
'','2 2 1 2 1 1 2 2 1 1 2 1 2 2 1 2 1 1 2 1 2 2 1 1 2 2 1 2 1 1 2',...
    '','','L9 (3^4)',...
    '','1 1 1 1',...
    '','1 2 2 2',...
    '','1 3 3 3',...
    '','2 1 2 3',...
    '','2 2 3 1',...
    '','2 3 1 2',...
    '','3 1 3 2',...
    '','3 2 1 3',...
    '','3 3 2 1',...
    '','','L27 (3^13)',...
    '','1 1 1 1 1 1 1 1 1 1 1 1 1',...
    '','1 1 1 1 2 2 2 2 2 2 2 2 2',...
    '','1 1 1 1 3 3 3 3 3 3 3 3 3',...
    '','1 2 2 2 1 1 1 2 2 2 3 3 3',...
    '','1 2 2 2 2 2 2 3 3 3 1 1 1',...
    '','1 2 2 2 3 3 3 1 1 1 2 2 2',...
    '','1 3 3 3 1 1 1 3 3 3 2 2 2',...
    '','1 3 3 3 2 2 2 1 1 1 3 3 3',...
    '','1 3 3 3 3 3 3 2 2 2 1 1 1',...
    '','2 1 2 3 1 2 3 1 2 3 1 2 3',...
    '','2 1 2 3 2 3 1 2 3 1 2 3 1',...
    '','2 1 2 3 3 1 2 3 1 2 3 1 2',...
    '','2 2 3 1 1 2 3 2 3 1 3 1 2',...
    '','2 2 3 1 2 3 1 3 1 2 1 2 3',...
    '','2 2 3 1 3 1 2 1 2 3 2 3 1',...
    '','2 3 1 2 1 2 3 3 1 2 2 3 1',...
    '','2 3 1 2 2 3 1 1 2 3 3 1 2',...
    '','2 3 1 2 3 1 2 2 3 1 1 2 3',...
    '','3 1 3 2 1 3 2 1 3 2 1 3 2',...
    '','3 1 3 2 2 1 3 2 1 3 2 1 3',...
    '','3 1 3 2 3 2 1 3 2 1 3 2 1',...
    '','3 2 1 3 1 3 2 2 1 3 3 2 1',...
    '','3 2 1 3 2 1 3 3 2 1 1 3 2',...
    '','3 2 1 3 3 2 1 1 3 2 2 1 3',...
    '','3 3 2 1 1 3 2 3 2 1 2 1 3',...
    '','3 3 2 1 2 1 3 1 3 2 3 2 1',...
    '','3 3 2 1 3 2 1 2 1 3 1 3 2',...
    '','','L16 (4^5)',...
    '','1 1 1 1 1',...
    '','1 2 2 2 2',...
    '','1 3 3 3 3',...
    '','1 4 4 4 4',...
    '','2 1 2 3 4',...
    '','2 2 1 4 3',...
    '','2 3 4 1 2',...
    '','2 4 3 2 1',...
    '','3 1 3 4 2',...
    '','3 2 4 3 1',...
    '','3 3 1 2 4',...
    '','3 4 2 1 3',...
    '','4 1 4 2 3',...
    '','4 2 3 1 4',...
    '','4 3 2 4 1',...
    '','4 4 1 3 2',...
```

```
'','','L25 (5^6)',...
'','1 1 1 1 1 1',...
'','1 2 2 2 2 2',...
'','1 3 3 3 3 3',...
'','1 4 4 4 4 4',...
'','1 5 5 5 5 5',...
'','2 1 2 3 4 5',...
'','2 2 3 4 5 1',...
'','2 3 4 5 1 2',...
'','2 4 5 1 2 3',...
'','2 5 1 2 3 4',...
'','3 1 3 5 2 4',...
'','3 2 4 1 3 5',...
'','3 3 5 2 4 1',...
'','3 4 1 3 5 2',...
'','3 5 2 4 1 3',...
'','4 1 4 2 5 3',...
'','4 2 5 3 1 4',...
'','4 3 1 4 2 5',...
'','4 4 2 5 3 1',...
'','4 5 3 1 4 2',...
'','5 1 5 4 3 2',...
'','5 2 1 5 4 3',...
'','5 3 2 1 5 4',...
'','5 4 3 2 1 5',...
'','5 5 4 3 2 1',...
'','','L18 (2 x 3^7)',...
'','1 1 1 1 1 1 1 1',...
'','1 1 2 2 2 2 2 2',...
'','1 1 3 3 3 3 3 3',...
'','1 2 1 1 2 2 3 3',...
'','1 2 2 2 3 3 1 1',...
'','1 2 3 3 1 1 2 2',...
'','1 3 1 2 1 3 2 3',...
'','1 3 2 3 2 1 3 1',...
'','1 3 3 1 3 2 1 2',...
'','2 1 1 3 3 2 2 1',...
'','2 1 2 1 1 3 3 2',...
'','2 1 3 2 2 1 1 3',...
'','2 2 1 2 3 1 3 2',...
'','2 2 2 3 1 2 1 3',...
'','2 2 3 1 2 3 2 1',...
'','2 3 1 3 2 3 1 2',...
'','2 3 2 1 3 1 2 3',...
'','2 3 3 2 1 2 3 1',...
'','','L32 (2 x 4^9)',...
'','1 1 1 1 1 1 1 1 1 1',...
'','1 1 2 2 2 2 2 2 2 2',...
'','1 1 3 3 3 3 3 3 3 3',...
'','1 1 4 4 4 4 4 4 4 4',...
'','1 2 1 1 2 2 3 3 4 4',...
'','1 2 2 2 1 1 4 4 3 3',...
'','1 2 3 3 4 4 1 1 2 2',...
'','1 2 4 4 3 3 2 2 1 1',...
'','1 3 1 2 3 4 1 2 3 4',...
'','1 3 2 1 4 3 2 1 4 3',...
'','1 3 3 4 1 2 3 4 1 2',...
'','1 3 4 3 2 1 4 3 2 1',...
'','1 4 1 2 4 3 3 4 2 1',...
'','1 4 2 1 3 4 4 3 1 2',...
'','1 4 3 4 2 1 1 2 4 3',...
```

```
'','1 4 4 3 1 2 2 1 3 4',...
'','2 1 1 4 1 4 2 3 2 3',...
'','2 1 2 3 2 3 1 4 1 4',...
'','2 1 3 2 3 2 4 1 4 1',...
'','2 1 4 1 4 1 3 2 3 2',...
'','2 2 1 4 2 3 4 1 3 2',...
'','2 2 2 3 1 4 3 2 4 1',...
'','2 2 3 2 4 1 2 3 1 4',...
'','2 2 4 1 3 2 1 4 2 3',...
'','2 3 1 3 3 1 2 4 4 2',...
'','2 3 2 4 4 2 1 3 3 1',...
'','2 3 3 1 1 3 4 2 2 4',...
'','2 3 4 2 2 4 3 1 1 3',...
'','2 4 1 3 4 2 4 2 1 3',...
'','2 4 2 4 3 1 3 1 2 4',...
'','2 4 3 1 2 4 2 4 3 1',...
'','2 4 4 2 1 3 1 3 4 2',...
'','','L36 (2^11 x 3^12)',...
'','1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1,'...
'','1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2',...
'','1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3',...
'','1 1 1 1 1 1 2 2 2 2 2 2 1 1 1 1 2 2 2 2 3 3 3 3',...
'','1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 1 1 1 1',...
'','1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 1 1 1 1 2 2 2 2',...
'','1 1 2 2 2 1 1 1 2 2 2 1 1 2 3 1 2 3 3 1 2 2 3',...
'','1 1 2 2 2 1 1 1 2 2 2 2 2 3 1 2 3 1 1 2 3 3 1',...
'','1 1 2 2 2 1 1 1 2 2 2 3 3 1 2 3 1 2 2 3 1 1 2',...
'','1 2 1 2 2 1 2 2 1 1 2 1 1 3 2 1 3 2 3 2 1 3 2',...
'','1 2 1 2 2 1 2 2 1 1 2 2 2 1 3 2 1 3 1 3 2 1 3',...
'','1 2 1 2 2 1 2 2 1 1 2 3 3 2 1 3 2 1 2 1 3 2 1',...
'','1 2 2 1 2 2 1 2 1 2 1 1 2 3 1 3 2 1 3 3 2 1 2',...
'','1 2 2 1 2 2 1 2 1 2 1 2 3 1 2 1 3 2 1 1 3 2 3',...
'','1 2 2 1 2 2 1 2 1 2 1 3 1 2 3 2 1 3 2 2 1 3 1',...
'','1 2 2 2 1 2 2 1 2 1 1 1 2 3 2 1 1 3 2 3 3 2 1',...
'','1 2 2 2 1 2 2 1 2 1 1 2 3 1 3 2 2 1 3 1 1 3 2',...
'','1 2 2 2 1 2 2 1 2 1 1 3 1 2 1 3 3 2 1 2 2 1 3',...
'','2 1 2 2 1 1 2 2 1 2 1 1 2 1 3 3 3 1 2 2 1 2 3',...
'','2 1 2 2 1 1 2 2 1 2 1 2 3 2 1 1 1 2 3 3 2 3 1',...
'','2 1 2 2 1 1 2 2 1 2 1 3 1 3 2 2 2 3 1 1 3 1 2',...
'','2 1 2 1 2 2 2 1 1 1 2 1 2 2 3 3 1 2 1 1 3 3 2',...
'','2 1 2 1 2 2 2 1 1 1 2 2 3 3 1 1 2 3 2 2 1 1 3',...
'','2 1 2 1 2 2 2 1 1 1 2 3 1 1 2 2 3 1 3 3 2 2 1',...
'','2 1 1 2 2 2 1 2 2 1 1 1 3 2 1 2 3 3 1 3 1 2 2',...
'','2 1 1 2 2 2 1 2 2 1 1 2 1 3 2 3 1 1 2 1 2 3 3'...
'','2 1 1 2 2 2 1 2 2 1 1 3 2 1 3 1 2 2 3 2 3 1 1',...
'','2 2 2 1 1 1 1 2 2 1 2 1 3 2 2 2 1 1 3 2 3 1 3',...
'','2 2 2 1 1 1 1 2 2 1 2 2 1 3 3 3 2 2 1 3 1 2 1',...
'','2 2 2 1 1 1 1 2 2 1 2 3 2 1 1 1 3 3 2 1 2 3 2',...
'','2 2 1 2 1 2 1 1 1 2 2 1 3 3 3 2 3 2 2 1 2 1 1',...
'','2 2 1 2 1 2 1 1 1 2 2 2 1 1 1 3 1 3 3 2 3 2 2',...
'','2 2 1 2 1 2 1 1 1 2 2 3 2 2 1 2 1 1 3 1 1 3 3',...
'','2 2 1 1 2 1 2 1 2 2 1 1 3 1 2 3 2 3 1 2 2 3 1',...
'','2 2 1 1 2 1 2 1 2 2 1 2 1 2 3 1 3 1 2 3 3 1 2',...
'','2 2 1 1 2 1 2 1 2 2 1 3 2 3 1 2 1 2 3 1 1 2 3',...
'','','L50 (2 x 5^11)',...
'','1 1 1 1 1 1 1 1 1 1 1 1 1',...
'','1 1 2 2 2 2 2 2 2 2 2 2 2',...
'','1 1 3 3 3 3 3 3 3 3 3 3 3',...
'','1 1 4 4 4 4 4 4 4 4 4 4 4',...
'','1 1 5 5 5 5 5 5 5 5 5 5 5',...
'','1 2 1 2 3 4 5 1 2 3 4 5',...
```

```
'','1 2 2 3 4 5 1 2 3 4 5 1',...
'','1 2 3 4 5 1 2 3 4 5 1 2',...
'','1 2 4 5 1 2 3 4 5 1 2 3',...
'','1 2 5 1 2 3 4 5 1 2 3 4',...
'','1 3 1 3 5 2 4 4 1 3 5 2',...
'','1 3 2 4 1 3 5 5 2 4 1 3',...
'','1 3 3 5 2 4 1 1 3 5 2 4',...
'','1 3 4 1 3 5 2 2 4 1 3 5',...
'','1 3 5 2 4 1 3 3 5 2 4 1',...
'','1 4 1 4 2 5 3 5 3 1 4 2',...
'','1 4 2 5 3 1 4 1 4 2 5 3',...
'','1 4 3 1 4 2 5 2 5 3 1 4',...
'','1 4 4 2 5 3 1 3 1 4 2 5',...
'','1 4 5 3 1 4 2 4 2 5 3 1',...
'','1 5 1 5 4 3 2 4 3 2 1 5',...
'','1 5 2 1 5 4 3 5 4 3 2 1',...
'','1 5 3 2 1 5 4 1 5 4 3 2',...
'','1 5 4 3 2 1 5 2 1 5 4 3',...
'','1 5 5 4 3 2 1 3 2 1 5 4',...
'','2 1 1 1 4 5 4 3 2 5 2 3',...
'','2 1 2 2 5 1 5 4 3 1 3 4',...
'','2 1 3 3 1 2 1 5 4 2 4 5',...
'','2 1 4 4 2 3 2 1 5 3 5 1',...
'','2 1 5 5 3 4 3 2 1 4 1 2',...
'','2 2 1 2 1 3 3 2 4 5 5 4'...
'','2 2 2 3 2 4 4 3 5 1 1 5',...
'','2 2 3 4 3 5 5 4 1 2 2 1',...
'','2 2 4 5 4 1 1 5 2 3 3 2',...
'','2 2 5 1 5 2 2 1 3 4 4 3',...
'','2 3 1 3 3 1 2 5 5 4 2 4',...
'','2 3 2 4 4 2 3 1 1 5 3 5',...
'','2 3 3 5 5 3 4 2 2 1 4 1',...
'','2 3 4 1 1 4 5 3 3 2 5 2',...
'','2 3 5 2 2 5 1 4 4 3 1 3',...
'','2 4 1 4 5 4 1 2 5 2 3 3',...
'','2 4 2 5 1 5 2 3 1 3 4 4',...
'','2 4 3 1 2 1 3 4 2 4 5 5',...
'','2 4 4 2 3 2 4 5 3 5 1 1',...
'','2 4 5 3 4 3 5 1 4 1 2 2',...
'','2 5 1 5 2 2 5 3 4 4 3 1',...
'','2 5 2 1 3 3 1 4 5 5 4 2',...
'','2 5 3 2 4 4 2 5 1 1 5 3',...
'','2 5 4 3 5 5 3 1 2 2 1 4',...
'','2 5 5 4 1 1 4 2 3 3 2 5',...
'','','L54 (2 x 3^25)',...
'','1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1',...
'','1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2',...
'','1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3',...
'','1 1 2 2 2 2 2 2 1 1 1 1 1 1 2 3 2 3 2 3 2 3 2 3 2',...
'','1 1 2 2 2 2 2 2 2 2 2 2 2 2 3 1 3 1 3 1 3 1 3 1 3',...
'','1 1 2 2 2 2 2 2 3 3 3 3 3 3 1 2 1 2 1 2 1 2 1 2 1',...
'','1 1 3 3 3 3 3 3 1 1 1 1 1 1 3 2 3 2 3 2 3 2 3 2 3',...
'','1 1 3 3 3 3 3 3 2 2 2 2 2 2 1 3 1 3 1 3 1 3 1 3 1',...
'','1 1 3 3 3 3 3 3 3 3 3 3 3 3 2 1 2 1 2 1 2 1 2 1 2',...
'','1 2 1 1 2 2 3 3 1 1 2 2 3 3 1 1 1 1 2 3 2 3 3 2 3',...
'','1 2 1 1 2 2 3 3 2 2 3 3 1 1 2 2 2 2 3 1 3 1 1 3 1',...
'','1 2 1 1 2 2 3 3 3 3 1 1 2 2 3 3 3 3 1 2 1 2 2 1 2',...
'','1 2 2 2 3 3 1 1 1 1 2 2 3 3 2 3 2 3 3 2 3 2 1 1 1',...
'','1 2 2 2 3 3 1 1 2 2 3 3 1 1 3 1 3 1 1 3 1 3 2 2 2',...
'','1 2 2 2 3 3 1 1 3 3 1 1 2 2 1 2 1 2 2 1 2 1 3 3 3',...
'','1 2 3 3 1 1 2 2 1 1 2 2 3 3 3 2 3 2 1 1 1 1 2 3 2',...
```

```
        '','1 2 3 3 1 1 2 2 2 2 3 3 1 1 1 3 1 3 2 2 2 2 3 1 3 1',...
        '','1 2 3 3 1 1 2 2 3 3 1 1 2 2 2 1 2 1 3 3 3 3 1 2 1 2',...
        '','1 3 1 2 1 3 2 3 1 2 1 3 2 3 1 1 2 3 1 1 3 2 2 3 3 2',...
        '','1 3 1 2 1 3 2 3 2 3 2 1 3 1 2 2 3 1 2 2 1 3 3 1 1 3',...
        '','1 3 1 2 1 3 2 3 3 1 3 2 1 2 3 3 1 2 3 3 2 1 1 2 2 1',...
        '','1 3 2 3 2 1 3 1 1 2 1 3 2 3 2 3 3 2 2 3 1 1 3 2 1 1',...
        '','1 3 2 3 2 1 3 1 2 3 2 1 3 1 3 1 1 3 3 1 2 2 1 3 2 2',...
        '','1 3 2 3 2 1 3 1 3 1 3 2 1 2 1 2 2 1 1 2 3 3 2 1 3 3',...
        '','1 3 3 1 3 2 1 2 1 2 1 3 2 3 3 2 1 1 3 2 2 3 1 1 2 3',...
        '','1 3 3 1 3 2 1 2 2 3 2 1 3 1 1 3 2 2 1 3 3 1 2 2 3 1',...
        '','1 3 3 1 3 2 1 2 3 1 3 2 1 2 2 1 3 3 2 1 1 2 3 3 1 2',...
        '','2 1 1 3 3 2 2 1 1 3 3 2 2 1 1 1 3 2 3 2 2 3 2 3 1 1',...
        '','2 1 1 3 3 2 2 1 2 1 1 3 3 2 2 2 1 3 1 3 3 1 3 1 2 2',...
        '','2 1 1 3 3 2 2 1 3 2 2 1 1 3 3 3 2 1 2 1 1 2 1 2 3 3',...
        '','2 1 2 1 1 3 3 2 1 3 3 2 2 1 2 3 1 1 1 1 3 2 3 2 2 3',...
        '','2 1 2 1 1 3 3 2 2 1 1 3 3 2 3 1 2 2 2 2 1 3 1 3 3 1',...
        '','2 1 2 1 1 3 3 2 3 2 2 1 1 3 1 2 3 3 3 3 2 1 2 1 1 2',...
        '','2 1 3 2 2 1 1 3 1 3 3 2 2 1 3 2 2 3 2 3 1 1 1 1 3 2',...
        '','2 1 3 2 2 1 1 3 2 1 1 3 3 2 1 3 3 1 3 1 2 2 2 2 1 3',...
        '','2 1 3 2 2 1 1 3 3 2 2 1 1 3 2 1 1 2 1 2 3 3 3 3 2 1',...
        '','2 2 1 2 3 1 3 2 1 2 3 1 3 2 1 1 2 3 3 2 1 1 3 2 2 3,'...
        '','2 2 1 2 3 1 3 2 2 3 1 2 1 3 2 2 3 1 1 3 2 2 1 3 3 1',...
        '','2 2 1 2 3 1 3 2 3 1 2 3 2 1 3 3 1 2 2 1 3 3 2 1 1 2',...
        '','2 2 2 3 1 2 1 3 1 2 3 1 3 2 2 3 3 2 1 1 2 3 1 1 3 2',...
        '','2 2 2 3 1 2 1 3 2 3 1 2 1 3 3 1 1 3 2 2 3 1 2 2 1 3',...
        '','2 2 2 3 1 2 1 3 3 1 2 3 2 1 1 2 2 1 3 3 1 2 3 3 2 1',...
        '','2 2 3 1 2 3 2 1 1 2 3 1 3 2 3 2 1 1 2 3 3 2 2 3 1 1',...
        '','2 2 3 1 2 3 2 1 2 3 1 2 1 3 1 3 2 2 3 1 1 3 3 1 2 2',...
        '','2 2 3 1 2 3 2 1 3 1 2 3 2 1 2 1 3 3 1 2 2 1 1 2 3 3',...
        '','2 3 1 3 2 3 1 2 1 3 2 3 1 2 1 1 3 2 2 3 3 2 1 1 2 3',...
        '','2 3 1 3 2 3 1 2 2 1 3 1 2 3 2 2 1 3 3 1 1 3 2 2 3 1',...
        '','2 3 1 3 2 3 1 2 3 2 1 2 3 1 3 3 2 1 1 2 2 1 3 3 1 2',...
        '','2 3 2 1 3 1 2 3 1 3 2 3 1 2 2 3 1 1 3 2 1 1 2 3 3 2',...
        '','2 3 2 1 3 1 2 3 2 1 3 1 2 3 3 1 2 2 1 3 2 2 3 1 1 3',...
        '','2 3 2 1 3 1 2 3 3 2 1 2 3 1 1 2 3 3 2 1 3 3 1 2 2 1',...
        '','2 3 3 2 1 2 3 1 1 3 2 3 1 2 3 2 2 3 1 1 2 3 3 2 1 1',...
        '','2 3 3 2 1 2 3 1 2 1 3 1 2 3 1 3 3 1 2 2 3 1 1 3 2 2',...
        '','2 3 3 2 1 2 3 1 3 2 1 2 3 1 2 1 1 2 3 3 1 2 2 1 3 3'};
        helpfig2(text,'Ajuda: Matriz ortogonal','adjust');

        case 'Interação'
        text={'Pagina Inicial > Níveis > Fatores > Design > Interação',...
        '','','**As interações que são confundidas com os fatores principais
não serão exibidas **',...
        '','','As matrizes de interação para as respectivas matrizes
ortogonais são:',...
        '','L8 (2^7)',...
        '','(1) 3 2 5 4 7 6',...
        '','  (2) 1 6 7 4 5',...
        '',' (3)7 6 5 4',...
        '',' (4)1 2 3',...
        '',' (5)3 2',...
        '',' (6)1',...
        '','','L16 (2^15)',...
        '','(1) 3  2  5  4  7  6  9  8  11 10 13 12 15 14',...
        '',' (2) 1  6  7  4  5  10 11 8  9  14 15 12 13',...
        '','  (3) 7  6  5  4  11 10 9  8  15 14 13 12',...
        '',' (4) 1  2  3  12 13 14 15 8  9  10 11',...
        '','  (5) 3  2  13 12 15 14 9  8  11 10',...
        '',' (6) 1  14 15 12 13 10 11 8  9 ',...
        '','  (7) 15 14 13 12 11 10 9  8 ',...
```

```
        '',' (8) 1  2  3  4  5  6  7 ',...
        '','  (9) 3  2  5  4  7  6 ',...
        '','  (10) 1  6  7  4  5 ',...
        '','   (11) 7  6  5  4 ',...
        '','  (12) 1  2  3 ',...
        '','   (13) 3  2 ',...
        '','   (14) 1 ',...
        '','','L32 (2^31)',...
        '','(1) 3 2 5 4 7 6 9 8 11 10 13 12 15 14 17 16 19 18 21 20 23 22 25
24 27 26 29 28 31 30',...
        '',' (2) 1 6 7 4 5 10 11 8 9 14 15 12 13 18 19 16 17 22 23 20 21 26
27 24 25 30 31 28 29',...
        '',' (3) 7 6 5 4 11 10 9 8 15 14 13 12 19 18 17 16 23 22 21 20 27
26 25 24 31 30 29 28',...
        '',' (4) 1 2 3 12 13 14 15 8 9 10 11 20 21 22 23 16 17 18 19 28 29
30 31 24 25 26 27',...
        '',' (5) 3 2 13 12 15 14 9 8 11 10 21 20 23 22 17 16 19 18 29 28 31
30 25 24 27 26',...
        '',' (6) 1 14 15 12 13 10 11 8 9 22 23 20 21 18 19 16 17 30 31 28
29 26 27 24 25',...
        '',' (7) 15 14 13 12 11 10 9 8 23 22 21 20 19 18 17 16 31 30 29 28
27 26 25 24',...
        '',' (8) 1  2  3 4  5  6 7 24 25 26 27 28 29 30 31 16 17 18 19 20
21 22 23',...
        '',' (9) 3  2 5  4  7 6 25 24 27 26 29 28 31 30 17 16 19 18 21 20 23
22',...
        '',' (10) 1 3  7  4 5 26 27 24 25 30 31 28 29 18 19 16 17 22 23 20
21',...
        '',' (11) 7  6  5 4 27 26 25 24 31 30 29 28 19 18 17 16 23 22 21
20',...
        '','  (12) 1  2 3 28 29 30 31 24 25 26 27 20 21 22 23 16 17 18 19',...
        '','   (13) 3 2 29 28 31 30 25 24 27 26 21 20 23 22 17 16 19 18',...
        '','   (14) 1 30 31 28 29 26 27 24 25 22 23 20 21 18 19 16 17',...
        '','   (15) 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16',...
        '','   (16) 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15',...
        '','  (17) 3  2  5  4  7  6  9  8 11 10 13 12 15 14',...
        '','   (18) 1  6  7  4  5  10 11 8  9 14 15 12 13',...
        '','  (19) 7  6  5  4  11 10 9  8 15 14 13 12',...
        '','   (20) 1  2  3  12 13 14 15 8  9 10 11',...
        '','  (21) 3  2  13 12 15 14 9  8 11 10',...
        '','   (22) 1  14 15 12 13 10 11 8  9',...
        '','    (23) 15 14 13 12 11 10 9  8',...
        '','    (24) 1  2  3  4  5 6  7',...
        '','   (25) 3  2  5  4 7  6',...
        '','    (26) 1  6  7 4  5',...
        '','   (27) 7  6 5  4',...
        '','    (28) 1 2  3',...
        '','    (29) 3  2',...
        '','   (30) 1',...
      '','','L27 (3^13)',...
      '','(1) 3  2  2  6  5  5  9  8  8  12 11 11',...
      '','  4  4  3  7  7  6  10 10 9  13 13 12',...
      '',' (2) 1  1  8  9  10 5  6  7  5  6  7 ',...
      '',' 4  3  11 12 13 11 12 13 8  9  10',...
      '','  (3) 1  9  10 8  7  5  6  6  7  5 ',...
      '','  2  13 11 12 12 13 11 10 8  9 ',...
      '',' (4) 10 8  9  6  7  5  7  5  6 ',...
      '',' 12 13 11 13 11 12 9  10 8 ',...
      '','  (5) 1  1  2  3  4  2  4  3 ',...
      '','  7  6  11 13 12 8  10 9 ',...
      '',' (6) 1  4  2  3  3  2  4 ',...
```

```
        '',' 5  13 12 11 10 9  8 ',...
        '','  (7) 3  4  2  4  3  2 ',...
        '','  12 11 13 9  8  10',...
        '',' (8) 1  1  2  3  4 ',...
        '',' 10 9  5  7  6 ',...
        '','  (9) 1  4  2  3 ',...
        '','  8  7  6  5 ',...
        '','  (10) 3  4  2 ',...
        '','   6  5  7 ',...
        '','   (11) 1  1 ',...
        '','  13 12',...
        '','   (12) 1',...
        '','   11',};
        helpfig2(text,'Ajuda: Interaçao','adjust');

        case 'Razão sinal/ruido'
        text={'Página inicial > Opções > Característica do processo',...
        '','Razão sinal/ruído',...
        '','','Nominal é mehor I',...
        'S/N = 10log[ÿ^2/S^2]',...
        'ÿ = média das amostras',...
        'Obs: Número mínimo de amostras exigidas 2.',...
        '','','Nominal é melhor II',...
        'S/N = -10log(S^2)',...
        'Obs: Número mínimo de amostras exigidas 2.',...
        '','','Menor é melhor',...
        'S/N = -10log[(1/n)*sum(y^2)]',...
        '','','Maior é melhor',...
        'S/N = -10log[(1/n)*sum(1/y^2)]',...
        '','** A exibição dos intervalos de confiança é mediante a seleção
da opção na janela de opções**'};
        helpfig2(text,'Ajuda: Razão Sinal-Ruído');

        case 'Função perda'
        text={'Página inicial > Opções > Característica do processo > Função
perda',...
        '','O programa calcula a função perda média de Taguchi',...
        '','','Função Perda média de Taguchi',...
        'A característica funcional "Nominal-é-melhor Tipo I" e "Nominal-é-
melhor Tipo II", são equivalentes a apenas "Nominal é melhor" para o caso da
função perda media.',...
        '','Nominal-é-menor',...
        'L(y)=(P/d^2)*(S^2+(ÿ-m)^2)',...
        '','','Menor-é-melhor',...
        'L(y)=(P/d^2)*(S^2+ÿ^2)',...
        '','','Maior-é-melhor',...
        'L(y)=(P*d^2)*(1/n)*sum((1/y)^2)',...
        '','','ÿ = média da amostra',...
        'm = valor nominal',...
        'k=(P/d^2)',...
        'd = Delta',...
        'P = valor de retrabalho',...
        'S^2=variância',...
        '','** A exibição dos intervalos de confiança é mediante a seleção
da opção na janela de opções**'};
        helpfig2(text,'Ajuda: Função perda','adjust');

        case 'Capacidade de processo'
        text={'Página inicial > Opções > Característica do processo >
Capacidade de processo',...
        '','Os índices só serão calculados se:',...
```

```
        '1 - Amostra vier de uma distribuição normal',...
        '','Cp',...
        'Cp = (LSE-LIE)/6S',...
        '','','Cpk',...
        'Cpk = Min[(mu-LIE)/3S,(LSE-mu)/3S]',...
        '','','Cpkm',...
        'Cpkm = (LSE-LIE)/6*sqrt(S^2+(mu-T)^2)',...
        '','','LSE = Limite superior de especificação',...
        'LIE = Limite inferior de especificação',...
        'mu = média amostral',...
        'T = valor nominal (valor alvo)',...
        'S = desvio padrão',...
        'S^2= variância',...
        '','** A exibição dos intervalos de confiança é mediante a seleção
da opção na janela de opções**'};
        helpfig2(text,'Ajuda: Capacidade de processo','adjust');

        case 'Intervalo de confiança'
        text={'Página inicial > Opções > Característica do processo >
Intervalo de confiança',...
        '','','Z',...
        'Tem a suposição de normalidade da distribuição do parâmetro e a
medida de variabilidade é o desvio padrão bootstrap.',...
        '','','t',...
        'Funciona bem quando a distribuição da estatística é aproximadamente
normal e a presença de viés pequeno.',...
        '','','percentil',...
        'Funciona bem quando a distribuição da estatística é aproximadamente
normal e a presença de viés pequeno.',...
        '','','Bc',...
        'Utilizado para corrigir o vício e assimetria da distribuição.',...
        '','','Bca',...
        'Utiliza-se quando a assimetria é muito forte.',...
        '','','** Ao selecionar a opção do intervalo de confiança, todas as
outras opções que estão selecionadas terão os intervalos de confiança
calculados**'};
        helpfig2(text,'Ajuda: Intervalo de confiança','adjust');

        case 'ANOVA'
        text={'Página inicial > Opções > Calcular > ANOVA'...
        '','A analise da variancia (ANOVA) permite identificar quais fatores
são mais relevantes.'...
        'Ela pode ser usada nos dados da amostra, ou na razão
sinal/ruído.',...
        '','','Obs: Para que a ANOVA seja usada corretamente, apenas na
primeira vez, ela deve ser usada após efetuar os cálculos.',...
        'Após esse primeiro uso, pode-se usar ela livremente.',...
        'Repetir esse processo sempre que mudar a matriz experimento.'};
        helpfig2(text,'Ajuda: ANOVA');

        case 'Gráficos'
        text={'Página inicial > Opções > Calcular > Gráficos',...
        '','Gráfico efeitos principais',...
        'O gráfico de efeitos principais permite analisar o comportamento dos
fatores quando se muda de nível.',...
        '','Gráfico de interação',...
        'Esse gráfico permite visualizar se existe interação entre os fatores
escolhidos.',...
        '','','**O gráfico de interações só será exibido se as interações
forem selecionadas na janela de interações**'};
        helpfig2(text,'Ajuda: Gráficos');
```

```
        case 'Exportação'
        text={'Página inicial > Opções > Calcular > Exportar',...
        '','As opções de exportação selecionadas serão exportadas na pasta
selecionada como "Result.xls"',...
        '','','**Só é possível ter um arquivo "Result.xls" em cada pasta,
caso a pasta já possua o arquivo o programa ira subscrever o arquivo
existente.**' };
        helpfig2(text,'Ajuda: Exportação');

        case 'NaN/Inf'
        text={'A aparição de "NaN" ou "Inf" nos resultados significa que
algum pré-requisito não foi comprido, consulte o respectivo tópico na aba
"Ajuda".'};
        helpfig2(text,'Ajuda: NaN/Inf');
      end
      end


      %-------------------------------------
      % confuguração menu ajuda
      %-------------------------------------
      function [varargout]=helpfig2(varargin)
      %Adaptado de
      % (c) Matthias Schwaiger, 2011
      % http://www.matzix.de
      % $Revision: 29 $
      % $Date: 2011-11-14 22:11:51 +0100 (Mo, 14. Nov 2011) $
      % $Author: Matthias
      title=mfilename;
      maxlength=75; % chars per line
      data.maxlines=25; % lines
      data.charheight=16;
      data.charwidth=8;

      autoadjust=0;
      aboutfig=0;

      if nargin == 0 || (~iscellstr(varargin{1}) && ~ischar(varargin{1}))
        info=regexp('$Id: helpfig.m 29 2011-11-14 21:11:51Z Matthias $','^.*
(?<revision>[0-9]+) (?<date>[0-9\-]+) .*','names');
        helptxt={upper(mfilename),'',['Revision: ' info.revision],['Date: '
info.date],'',[char(169)  '  Matthias  Schwaiger,  2011-'  info.date(1:4)],'
http://www.matzix.de'};
        title='About HELPFIG';
        autoadjust=1;
        aboutfig=1;
      elseif nargin == 1 && ischar(varargin{1}) && ...
        any(size(varargin{1})==1) && ...
        ~isempty(regexp(varargin{1},'^[a-zA-z0-9_]+$','once')) && ...
        exist([varargin{1} '.m'],'file')
        helptxt=help(varargin{1});
        title=['Help on ' upper(varargin{1})];
        autoadjust=1;
        if   strcmp(varargin{1},mfilename), aboutfig=1; end
        else
        helptxt=varargin{1};
      end
      if ischar(helptxt) && any(size(helptxt)==1)
        helptxt=regexp(helptxt,'[\n\r]','split');
      end
```

```
      if ~iscellstr(helptxt)
        helptxt=cellstr(helptxt);
      end
      helptxt=regexprep(helptxt,'[\f\t\v]','');
      helptxt=regexprep(helptxt,'[\n\r]+','');
      helptxt=helptxt(:);

      for k=2:nargin
        switch k
        case 2,
      if ischar(varargin{k}), title=varargin{k};
      end
        case 3,
      if   isnumeric(varargin{k}),   maxlength=round(max(1,varargin{k}(1)));
      elseif
           ischar(varargin{k})       &&       strcmpi(varargin{k},'adjust'),
      autoadjust=1;
      end
        case 4,
      if isnumeric(varargin{k}),data.maxlines=round(max(1,varargin{k}(1)));
      end
        end
      end

      if autoadjust
        maxlength=max(cellfun(@length,helptxt));
        data.maxlines=length(helptxt);
      end

      scr=get(0,'ScreenSize');
      maxlength=max(1,min(floor((scr(3)-70)/data.charwidth),maxlength));
      data.maxlines=max(1,min(floor((scr(4)-
100)/data.charheight),data.maxlines));
      usescrollbar=1;
      if     sum(cellfun(@(x)     ceil(length(x)/maxlength),helptxt))     <=
data.maxlines
        usescrollbar=0;
      end
      width=38+maxlength*data.charwidth+usescrollbar*22;
      height=30+data.maxlines*data.charheight;
      data.curline=1;
      data.mousepos=[0 0];

      fig=figure;
      set(fig,'tag','helpfig','name',title,'NumberTitle','off', ...
        'menubar','none','resize','off', ...
        'position',round([(scr(3)-width)/2      (scr(4)-height)/2      width
height]));
      try set(fig,'DockControl','off'); catch; end %#ok<CTCH>
      drawnow;

      data.frame=uicontrol('style','frame', 'units','pixels', ...
        'position',[8 8 width-16 height-16], ...
        'enable','inactive');
      data.ui(1)=uicontrol('style','text', 'units','pixels', ...
        'position',[16 16 width-32-usescrollbar*22 height-32], ...
        'backgroundcolor',get(data.frame,'backgroundcolor'), ...
        'horizontalalignment','left', ...
        'fontsize',10,'FontName', 'FixedWidth', ...
        'enable','inactive');
```

```
      if usescrollbar
        data.ui(2)=uicontrol('style','slider', 'units','pixels', ...
        'position',[width-29 9 20 height-18]);
      end

      % contents and callback setup
      % reformat helptxt if necessary
      k=1;
      while k<=length(helptxt)
        if length(helptxt{k})>maxlength
        idx=strfind(helptxt{k}(1:maxlength),' ');
        if ~isempty(idx)
        idx=idx(end);
        else
        idx=maxlength;
        end
        helptxt=[helptxt(1:k-1);                          {helptxt{k}(1:idx)};
{helptxt{k}(idx+1:end)}; helptxt(k+1:end)];
        end
        k=k+1;
      end
      data.txt=helptxt;
      data.maxline=max(1,length(helptxt)-data.maxlines+1);
      % cellfun(@disp,helptxt);

      % set up GUI contents and callback functions
      set(data.ui(1),'string',helptxt(data.curline:min(data.curline+data.ma
xlines-1,length(helptxt))));
      set(gcf,'keypressfcn',@helpfigtxtscroll);
      set(data.frame,'keypressfcn',@helpfigtxtscroll);
      set(data.ui(1),'keypressfcn',@helpfigtxtscroll);
      if usescrollbar
        if length(helptxt) <= data.maxlines
            set(data.ui(2),'min',1,'max',2,'value',1,'enable','off');
        else
        set(data.ui(2),'min',1,'max',data.maxline, ...
        'sliderstep', min(1,[1 data.maxlines]/(data.maxline-1)), ...
        'value',data.maxline-data.curline+1, ...
        'callback',@helpfigtxtscroll);
        end
        set(data.ui(2),'keypressfcn',@helpfigtxtscroll);
      end

      try
        set(fig,'WindowButtonDownFcn',{@helpfigtxtscroll,'down'});
        set(fig,'WindowButtonUpFcn',{@helpfigtxtscroll,'up'});
        set(fig,'WindowScrollWheelFcn',@helpfigtxtscroll);
      catch %#ok
      end

      if aboutfig && nargin == 0
        set(gcf,'tag','about_helpfig');
      elseif aboutfig
        set(gcf,'tag','help_helpfig');
      end
      set(fig,'userdata',data,'HandleVisibility','off');


      % finish
      if nargout ~=0
```

```
        varargout{1}=fig;
      end
      end



      %----------------------------------------
      %Função scroll menu Help
      %----------------------------------------
      function helpfigtxtscroll(src,evnt,varargin)
      %adaptado de
      %   (c) Matthias Schwaiger, 2011
      %       http://www.matzix.de
      % $Revision: 29 $
      % $Date: 2011-11-14 22:11:51 +0100 (Mo, 14. Nov 2011) $
      % $Author: Matthias $

      data=get(gcbf,'userdata');
      if nargin == 3
        pos=get(data.ui(1),'position');
        pos=[pos(1) pos(2)+pos(4)];
        pos=ceil((get(gcbf,'CurrentPoint') - pos)./[data.charwidth -
data.charheight]);
        pos(1)=max(0,min(max(cellfun(@length,data.txt)),pos(1)));
        pos(2)=max(1,min(min(data.maxlines,pos(2))+data.curline-
1,data.maxline+data.maxlines-1));
        % get zero position
        if strcmp(varargin{1},'down')
             data.mousepos=pos;
        if pos(1)~=0
             set(data.frame,'backgroundcolor',[0.7 0.7 1]);
        else
            set(data.frame,'backgroundcolor',[0.3 0.3 1]);
        end
        elseif strcmp(varargin{1},'up')
        if all([pos(1) data.mousepos(1)]==0) % select complete lines
             pos(1)=max(cellfun(@length,data.txt));
        end
        if all(pos==data.mousepos)
            txt=[' ' data.txt{pos(2)} ' '];
            idx=strfind(txt,' ');
             txt=txt(max(idx(idx<=pos(1)+1))+1:min(idx(idx>=pos(1)+1))-1);
        if ~isempty(txt)
            set(data.frame,'backgroundcolor',[0.3 1 0.3]);
             drawnow
            pause(0.2);
        end
        else
            txt=char(data.txt);

txt=txt(min(pos(2),data.mousepos(2)):max(pos(2),data.mousepos(2)),max(1,min
(pos(1),data.mousepos(1))):max(pos(1),data.mousepos(1)));
        txt2='';
        for k=1:size(txt,1)
             txt2=[txt2 deblank(txt(k,:)) 10]; %#ok<AGROW>
        end
        txt=txt2(1:end-1);
        end
        try
        if ~isempty(txt)
             clipboard('copy',txt);
```

```
        else
             clipboard('copy',' ');
        end
        catch %#ok
        disp('clipboard is not available!');
        end
        data.mousepos=[0 0];

set(data.frame,'backgroundcolor',get(data.ui(1),'backgroundcolor'));
        % disp(clipboard('paste'));
        end
        set(gcbf,'userdata',data);
        drawnow;
        return
      elseif length(data.ui) >= 2 && src==data.ui(2)
        data.curline=round(data.maxline-get(data.ui(2),'value')+1);
      elseif isfield(evnt,'Key')
        % disp(lower(evnt.Key)); % development
        switch lower(evnt.Key)
        case 'escape'
        delete(gcbf);
        return
        case 'a'
        if ~any(strcmp(get(gcbf,'tag'),{'about_helpfig','help_helpfig'}))
            helpfig;
        end
        case 'f1'
        if ~strcmp(get(gcbf,'tag'),'help_helpfig')
             helpfig('helpfig');
        end
        case 'c'
        if any(strcmp(evnt.Modifier,'control'))
             txt='';
        for k=1:length(data.txt)
             txt=[txt deblank(data.txt{k}) 10]; %#ok<AGROW>
        end
        try
        clipboard('copy',txt(1:end-1));
        catch %#ok
        disp('clipboard is not available!');
        end
        end
        case 'home'
            data.curline=1;
        case 'end'
            data.curline=data.maxline;
        case {'leftarrow','pageup'}
             data.curline=max(1,data.curline-data.maxlines);
        case {'rightarrow','pagedown'}
             data.curline=min(data.maxline,data.curline+data.maxlines);
        case 'uparrow'
             data.curline=max(1,data.curline-1);
        case 'downarrow'
             data.curline=min(data.maxline,data.curline+1);
        end
      elseif isfield(evnt,'VerticalScrollCount')

data.curline=round(max(1,min(data.maxline,data.curline+evnt.VerticalScrollA
mount*evnt.VerticalScrollCount)));
      end
      if length(data.ui) >= 2
```

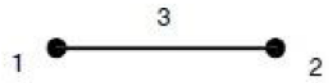
```
        set(data.ui(2),'value',data.maxline-data.curline+1);
      end
      set(gcbf,'userdata',data);
      set(data.ui(1),'string',data.txt(data.curline:min(data.curline+data.m
axlines-1,length(data.txt))));
      drawnow
      end
```

ANEXO A – MATRIZES ORTOGONAIS, MATRIZES TRIANGULARES E GRÁFICOS LINEARES

As tabelas e imagens foram retiradas e adaptadas de Taguchi *et al.* (2005) e Krishnaiah e Shahabudden (2012).

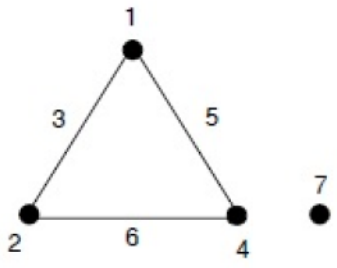
$L_4(2^3)$

Nº	1	2	3
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1

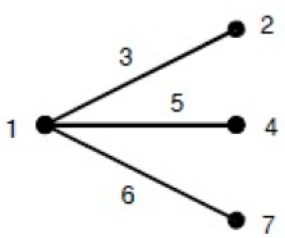


$L_8(2^7)$

Nº	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2



(a)



(b)

$L_8(2^7)$ Interação entre duas colunas

1	2	3	4	5	6	7
(1)	3	2	5	4	7	6
	(2)	1	6	7	4	5
		(3)	7	6	5	4
			(4)	1	2	3
				(5)	3	2
					(6)	1
						(7)

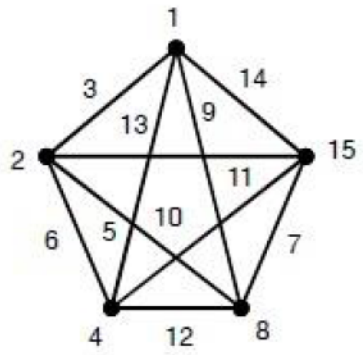
$L_{12}(2^{11})$ *

Nº	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	2	2	2	2	2	2
3	1	1	2	2	2	1	1	1	2	2	2
4	1	2	1	2	2	1	2	2	1	1	2
5	1	2	2	1	2	2	1	2	1	2	1
6	1	2	2	2	1	2	2	1	2	1	1
7	2	1	2	2	1	1	2	2	1	2	1
8	2	1	2	1	2	2	2	1	1	1	2
9	2	1	1	2	2	2	1	2	2	1	1
10	2	2	2	1	1	1	1	2	2	1	2
11	2	2	1	2	1	2	1	1	1	2	2
12	2	2	1	1	2	1	2	1	2	2	1

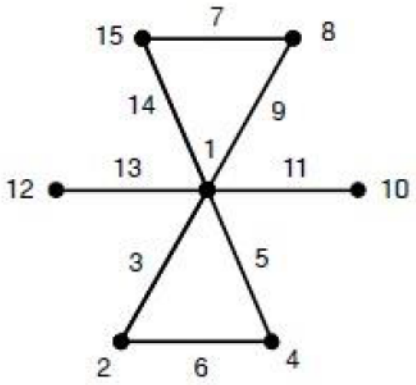
*Essa matriz foi projetada onde as interações são distribuídas mais uniformemente em todas as colunas.

$L_{16}(2^{15})$

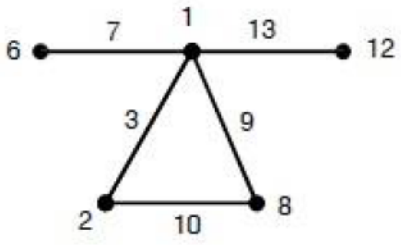
Nº	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
3	1	1	1	2	2	2	2	1	1	1	1	2	2	2	2
4	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
5	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2
6	1	2	2	1	1	2	2	2	2	1	1	2	2	1	1
7	1	2	2	2	2	1	1	1	1	2	2	2	2	1	1
8	1	2	2	2	2	1	1	2	2	1	1	1	1	2	2
9	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
10	2	1	2	1	2	1	2	2	1	2	1	2	1	2	1
11	2	1	2	2	1	2	1	1	2	1	2	2	1	2	1
12	2	1	2	2	1	2	1	2	1	2	1	1	2	1	2
13	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1
14	2	2	1	1	2	2	1	2	1	1	2	2	1	1	2
15	2	2	1	2	1	1	2	1	2	2	1	2	1	1	2
16	2	2	1	2	1	1	2	2	1	1	2	1	2	2	1

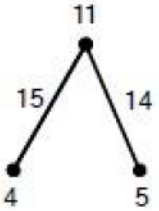


(a)

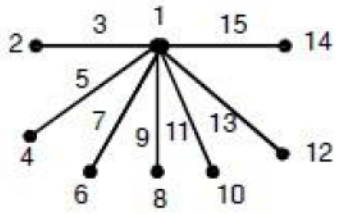


(b)

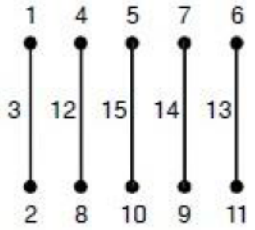




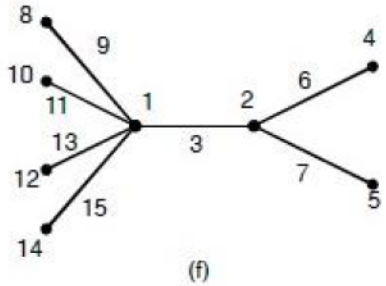
(c)



(d)



(e)



(f)

$L_{16}(2^{15})$ Interação entre duas colunas

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
(1)	3	2	5	4	7	6	9	8	11	10	13	12	15	14
	(2)	1	6	7	4	5	10	11	8	9	14	15	12	13
		(3)	7	6	5	4	11	10	9	8	15	14	13	12
			(4)	1	2	3	12	13	14	15	8	9	10	11
				(5)	3	2	13	12	15	14	9	8	11	10
					(6)	1	14	15	12	13	10	11	8	9
						(7)	15	14	13	12	11	10	9	8
							(8)	1	2	3	4	5	6	7
								(9)	3	2	5	4	7	6
									(10)	1	6	7	4	5
										(11)	7	6	5	4
											(12)	1	2	3
												(13)	3	2
													(14)	1

$L_{18}(2x3^7)$

Nº	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2
3	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	1	1	1
4	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2
5	1	1	1	2	2	2	2	1	1	1	1	2	2	2	2	1	1	1
6	1	1	1	2	2	2	2	1	1	1	1	2	2	2	2	2	2	2
7	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1
8	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1	2	2	2
9	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2
10	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2	2	2	1
11	1	2	2	1	1	2	2	2	2	1	1	2	2	1	1	1	1	2
12	1	2	2	1	1	2	2	2	2	1	1	2	2	1	1	2	2	1
13	1	2	2	2	2	1	1	1	1	2	2	2	2	1	1	1	1	2
14	1	2	2	2	2	1	1	1	1	2	2	2	2	1	1	2	2	1
15	1	2	2	2	2	1	1	2	2	1	1	1	1	2	2	1	1	2
16	1	2	2	2	2	1	1	2	2	1	1	1	1	2	2	2	2	1
17	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1
18	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	2	1	2
19	2	1	2	1	2	1	2	2	1	2	1	2	1	2	1	1	2	1
20	2	1	2	1	2	1	2	2	1	2	1	2	1	2	1	2	1	2
21	2	1	2	2	1	2	1	1	2	1	2	2	1	2	1	1	2	1
22	2	1	2	2	1	2	1	1	2	1	2	2	1	2	1	2	1	2
23	2	1	2	2	1	2	1	2	1	2	1	1	2	1	2	1	2	1
24	2	1	2	2	1	2	1	2	1	2	1	1	2	1	2	2	1	2
25	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2
26	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1	2	1	1
27	2	2	1	1	2	2	1	2	1	1	2	2	1	1	2	1	2	2
28	2	2	1	1	2	2	1	2	1	1	2	2	1	1	2	2	1	1
29	2	2	1	2	1	1	2	1	2	2	1	2	1	1	2	1	2	2
30	2	2	1	2	1	1	2	1	2	2	1	2	1	1	2	2	1	1
31	2	2	1	2	1	1	2	2	1	1	2	1	2	2	1	1	2	2
32	2	2	1	2	1	1	2	2	1	1	2	1	2	2	1	2	1	1

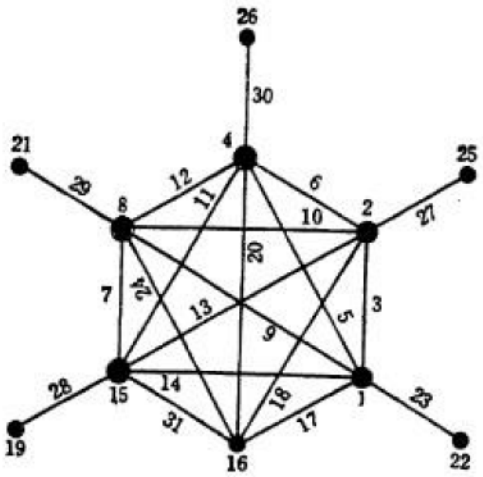
$L_{32}(2^{31})$

(Continua)

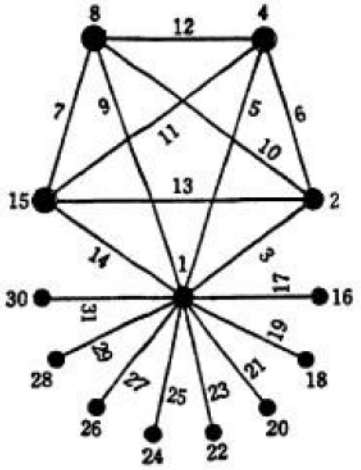
Nº	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		3	2	5	4	7	6	9	8	11	10	13	12	15	14	17	16	19	18	21
2			1	6	7	4	5	10	11	8	9	14	15	12	13	18	19	16	17	22
3				7	6	5	4	11	10	9	8	15	14	13	12	19	18	17	16	23
4					1	2	3	12	13	14	15	8	9	10	1 1	20	21	22	23	16
5						3	2	13	12	15	14	9	8	11	1 0	21	20	23	22	17
6							1	14	15	12	13	10	11	8	9	22	23	20	21	18
7								15	14	13	12	11	10	9	8	23	22	21	20	19
8									1	2	3	4	5	6	7	24	25	26	27	28
9										3	2	5	4	7	6	25	24	27	26	29
10											1	3	7	4	5	26	27	24	25	30
11												7	6	5	4	27	26	25	24	31
12													1	2	3	28	29	30	31	24
13														3	2	29	28	31	30	25
14															1	30	31	28	29	26
15																31	30	29	28	27
16																	1	2	3	4
17																		3	2	5
18																			1	6
19																				7
20																				
21																				
22																				
23																				
24																				
25																				
26																				
27																				
28																				
29																				
30																				

(Conclusão)

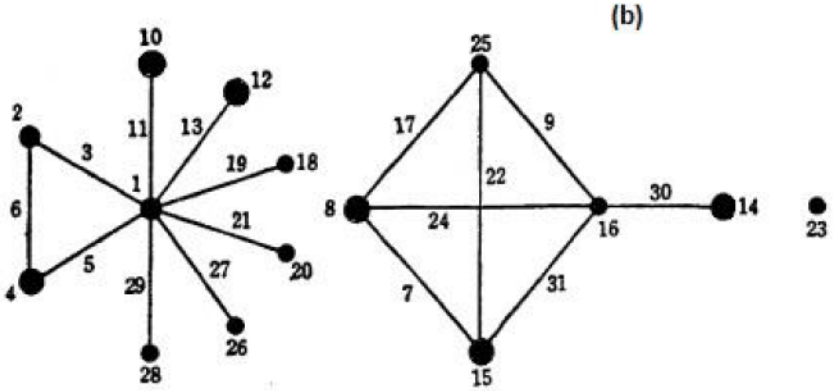
Nº	21	22	23	24	25	26	27	28	29	30	31
1	20	23	22	25	24	27	26	29	28	31	30
2	23	20	21	26	27	24	25	30	31	28	29
3	22	21	20	27	26	25	24	31	30	29	28
4	17	18	19	28	29	30	31	24	25	26	27
5	16	19	18	29	28	31	30	25	24	27	26
6	19	16	17	30	31	28	29	26	27	24	25
7	18	17	16	31	30	29	28	27	26	25	24
8	29	30	31	16	17	18	19	20	21	22	23
9	28	31	30	17	16	19	18	21	20	23	22
10	31	28	29	18	19	16	17	22	23	20	21
11	30	29	28	19	18	17	16	23	22	21	20
12	25	26	27	20	21	22	23	16	17	18	19
13	24	27	26	21	20	23	22	17	16	19	18
14	27	24	25	22	23	20	21	18	19	16	17
15	26	25	24	23	22	21	20	19	18	17	16
16	5	6	7	8	9	10	11	12	13	14	15
17	4	7	6	9	8	11	10	13	12	15	14
18	7	4	5	10	11	8	9	14	15	12	13
19	6	5	4	11	10	9	8	15	14	13	12
20	1	2	3	12	13	14	15	8	9	10	11
21		3	2	13	12	15	14	9	8	11	10
22			1	14	15	12	13	10	11	8	9
23				15	14	13	12	11	10	9	8
24					1	2	3	4	5	6	7
25						3	2	5	4	7	6
26							1	6	7	4	5
27								7	6	5	4
28									1	2	3
29										3	2
30											1


26
30
21
4
25
29
12
11
6
8
10
2
27
20
7
24
13
3
5
9
28
15
14
18
1
23
31
17
19
16
22

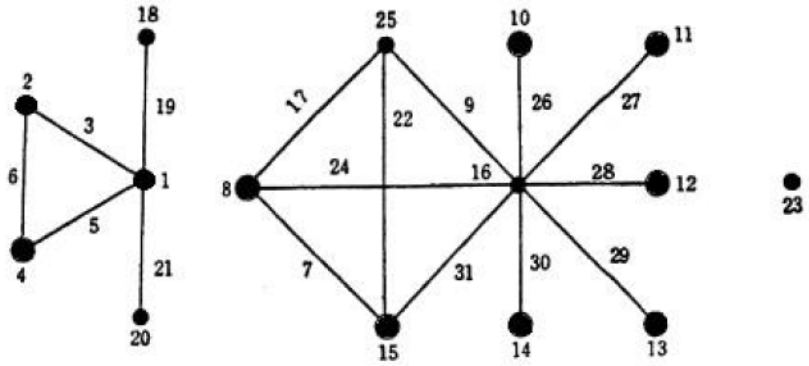
(a)


8
12
4
7
9
11
5
6
10
15
13
2
14
1
3
30
31
17
16
19
28
24
27
25
23
21
18
26
24
22
20

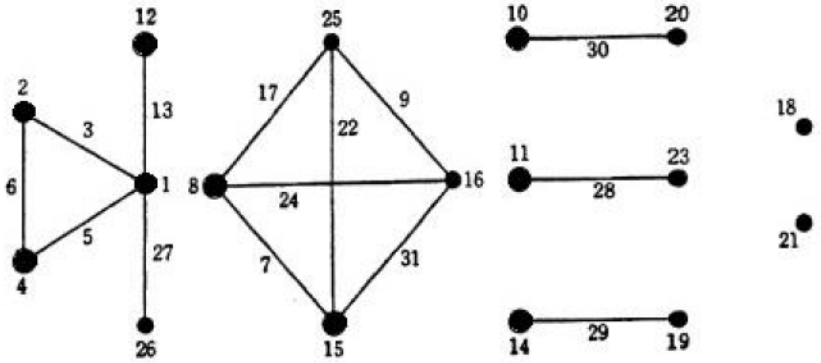
(b)


10
12
2
11
13
3
19
18
1
6
21
20
5
4
27
29
26
28
25
17
9
22
30
8
14
24
16
23
7
31
15

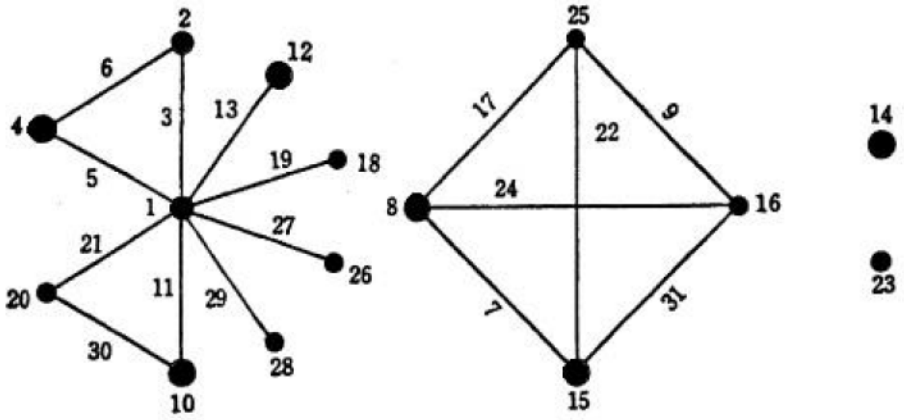
(c)


18
25
10
11
2
19
3
17
22
9
26
27
6
1
24
16
28
12
8
23
5
4
21
7
31
30
29
20
15
14
13

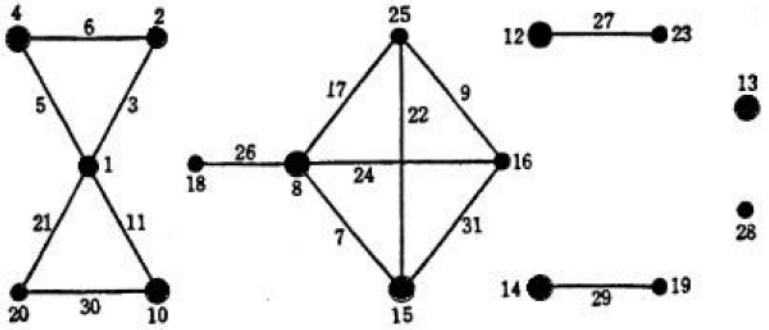
(d)


12
25
10
20
30
2
13
3
17
9
22
18
6
1
8
16
24
11
23
28
5
27
7
31
21
4
26
15
14
29
19

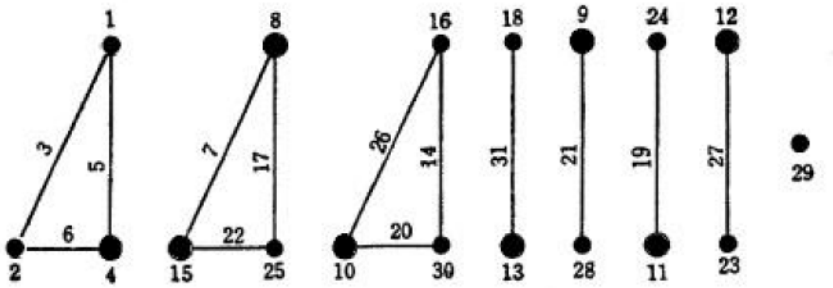
(e)


2
25
6
12
4
3
13
17
9
22
14
19
18
5
1
8
16
24
27
21
26
23
20
11
29
7
31
30
28
10
15

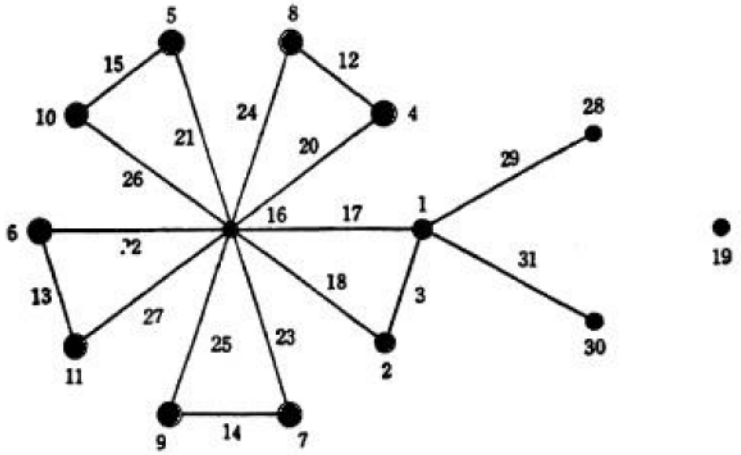
(f)


4
6
2
25
12
27
23
5
3
17
9
22
13
26
18
1
8
16
24
21
11
7
31
28
20
30
10
15
14
29
19

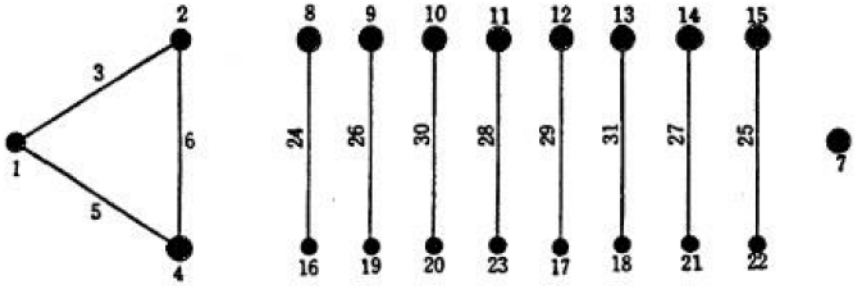
(g)


1
8
16
18
9
24
12
3
5
7
17
26
14
31
21
19
27
29
6
22
20
2
4
15
25
10
30
13
28
11
23

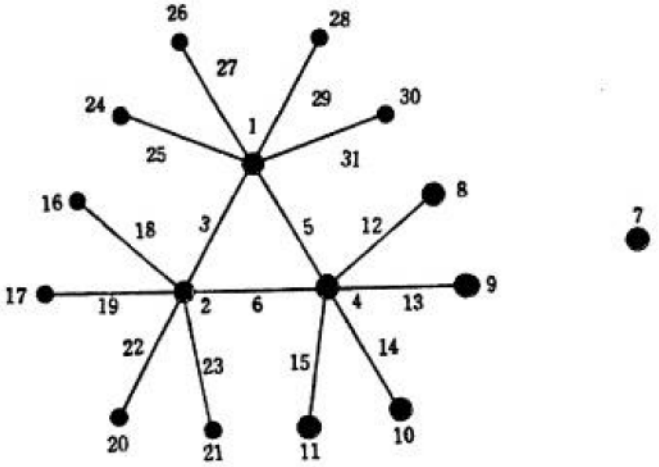
(h)


5
8
15
12
10
4
24
28
21
20
29
26
16
17
1
6
32
31
19
18
13
3
27
25
23
30
11
2
9
14
7

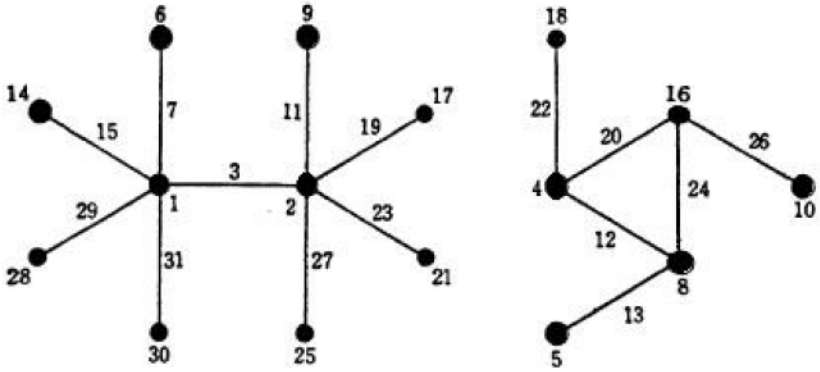
(i)


2
8
9
10
11
12
13
14
15
3
24
26
30
28
29
31
27
25
7
6
1
5
16
19
20
23
17
18
21
22
4

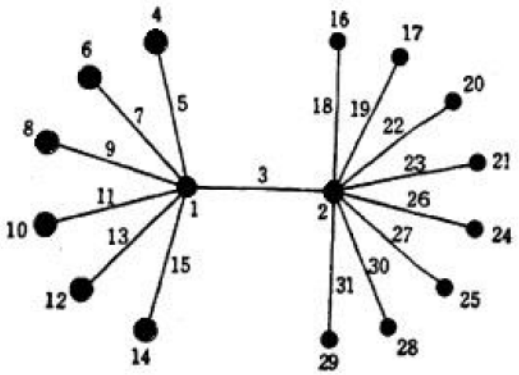
(j)


26
28
27
24
29
30
1
25
31
16
8
18
3
5
12
7
17
19
2
6
4
13
9
22
23
15
14
20
21
11
10

(k)


6
9
18
14
7
11
17
22
16
15
19
20
26
3
4
24
29
1
2
23
10
12
28
31
27
21
8
13
30
25
5

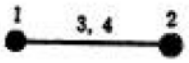
(l)


4
16
17
6
18
19
20
7
5
8
22
9
23
21
3
11
26
1
2
10
24
13
27
15
30
12
31
25
14
29
28

(m)

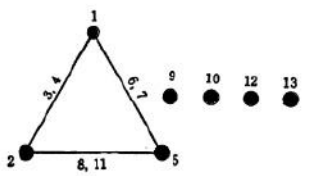
$L_9(3^4)$

Nº	1	2	3	4
1	1	1	1	1
2	1	2	2	2
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

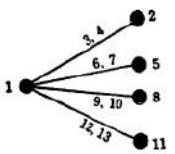


$L_{27}(3^{13})$

No.	1	2	3	4	5	6	7	8	9	10	11	12	13
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2	2	2	2	2
3	1	1	1	1	3	3	3	3	3	3	3	3	3
4	1	2	2	2	1	1	1	2	2	2	3	3	3
5	1	2	2	2	2	2	2	3	3	3	1	1	1
6	1	2	2	2	3	3	3	1	1	1	2	2	2
7	1	3	3	3	1	1	1	3	3	3	2	2	2
8	1	3	3	3	2	2	2	1	1	1	3	3	3
9	1	3	3	3	3	3	3	2	2	2	1	1	1
10	2	1	2	3	1	2	3	1	2	3	1	2	3
11	2	1	2	3	2	3	1	2	3	1	2	3	1
12	2	1	2	3	3	1	2	3	1	2	3	1	2
13	2	2	3	1	1	2	3	2	3	1	3	1	2
14	2	2	3	1	2	3	1	3	1	2	1	2	3
15	2	2	3	1	3	1	2	1	2	3	2	3	1
16	2	3	1	2	1	2	3	3	1	2	2	3	1
17	2	3	1	2	2	3	1	1	2	3	3	1	2
18	2	3	1	2	3	1	2	2	3	1	1	2	3
19	3	1	3	2	1	3	2	1	3	2	1	3	2
20	3	1	3	2	2	1	3	2	1	3	2	1	3
21	3	1	3	2	3	2	1	3	2	1	3	2	1
22	3	2	1	3	1	3	2	2	1	3	3	2	1
23	3	2	1	3	2	1	3	3	2	1	1	3	2
24	3	2	1	3	3	2	1	1	3	2	2	1	3
25	3	3	2	1	1	3	2	3	2	1	2	1	3
26	3	3	2	1	2	1	3	1	3	2	3	2	1
27	3	3	2	1	3	2	1	2	1	3	1	3	2



(a)



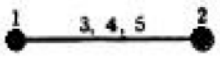
(b)

$L_{27}(3^{13})$

Nº	1	2	3	4	5	6	7	8	9	10	11	12	13
1		3	2	2	6	5	5	9	8	8	12	11	11
		4	4	3	7	7	6	10	10	9	13	13	12
2			1	1	8	9	10	5	6	7	5	6	7
			4	3	11	12	13	11	12	13	8	9	10
3				1	9	10	8	7	5	6	6	7	5
				2	13	11	12	12	13	11	10	8	9
4					10	8	9	6	7	5	7	5	6
					12	13	11	13	11	12	9	10	8
5						1	1	2	3	4	2	4	3
						7	6	11	13	12	8	10	9
6							1	4	2	3	3	2	4
							5	13	12	11	10	9	8
7								3	4	2	4	3	2
								12	11	13	9	8	10
8									1	1	2	3	4
									10	9	5	7	6
9										1	4	2	3
										8	7	6	5
10											3	4	2
											6	5	7
11												1	1
												13	12
12													1
													11

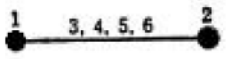
$L_{16}(4^5)$

Nº	1	2	3	4	5
1	1	1	1	1	1
2	1	2	2	2	2
3	1	3	3	3	3
4	1	4	4	4	4
5	2	1	2	3	4
6	2	2	1	4	3
7	2	3	4	1	2
8	2	4	3	2	1
9	3	1	3	4	2
10	3	2	4	3	1
11	3	3	1	2	4
12	3	4	2	1	3
13	4	1	4	2	3
14	4	2	3	1	4
15	4	3	2	4	1
16	4	4	1	3	2



$L_{25}(5^6)$

Nº	1	2	3	4	5	6
1	1	1	1	1	1	1
2	1	2	2	2	2	2
3	1	3	3	3	3	3
4	1	4	4	4	4	4
5	1	5	5	5	5	5
6	2	1	2	3	4	5
7	2	2	3	4	5	1
8	2	3	4	5	1	2
9	2	4	5	1	2	3
10	2	5	1	2	3	4
11	3	1	3	5	2	4
12	3	2	4	1	3	5
13	3	3	5	2	4	1
14	3	4	1	3	5	2
15	3	5	2	4	1	3
16	4	1	4	2	5	3
17	4	2	5	3	1	4
18	4	3	1	4	2	5
19	4	4	2	5	3	1
20	4	5	3	1	4	2
21	5	1	5	4	3	2
22	5	2	1	5	4	3
23	5	3	2	1	5	4
24	5	4	3	2	1	5
25	5	5	4	3	2	1



$L_{18}(2x3^7)$ *

No.	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	2	2
3	1	1	3	3	3	3	3	3
4	1	2	1	1	2	2	3	3
5	1	2	2	2	3	3	1	1
6	1	2	3	3	1	1	2	2
7	1	3	1	2	1	3	2	3
8	1	3	2	3	2	1	3	1
9	1	3	3	1	3	2	1	2
10	2	1	1	3	3	2	2	1
11	2	1	2	1	1	3	3	2
12	2	1	3	2	2	1	1	3
13	2	2	1	2	3	1	3	2
14	2	2	2	3	1	2	1	3
15	2	2	3	1	2	3	2	1
16	2	3	1	3	2	3	1	2
17	2	3	2	1	3	1	2	3
18	2	3	3	2	1	2	3	1

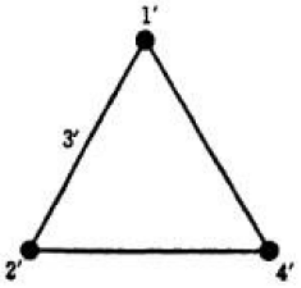
*Essa matriz foi projetada onde as interações são distribuídas mais uniformemente em todas as colunas.

$L_{32}(2x4^9)$

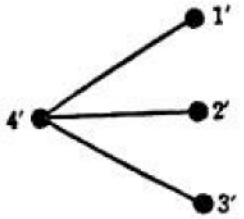
Nº	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	2	2	2	2
3	1	1	3	3	3	3	3	3	3	3
4	1	1	4	4	4	4	4	4	4	4
5	1	2	1	1	2	2	3	3	4	4
6	1	2	2	2	1	1	4	4	3	3
7	1	2	3	3	4	4	1	1	2	2
8	1	2	4	4	3	3	2	2	1	1
9	1	3	1	2	3	4	1	2	3	4
10	1	3	2	1	4	3	2	1	4	3
11	1	3	3	4	1	2	3	4	1	2
12	1	3	4	3	2	1	4	3	2	1
13	1	4	1	2	4	3	3	4	2	1
14	1	4	2	1	3	4	4	3	1	2
15	1	4	3	4	2	1	1	2	4	3
16	1	4	4	3	1	2	2	1	3	4
17	2	1	1	4	1	4	2	3	2	3
18	2	1	2	3	2	3	1	4	1	4
19	2	1	3	2	3	2	4	1	4	1
20	2	1	4	1	4	1	3	2	3	2
21	2	2	1	4	2	3	4	1	3	2
22	2	2	2	3	1	4	3	2	4	1
23	2	2	3	2	4	1	2	3	1	4
24	2	2	4	1	3	2	1	4	2	3
25	2	3	1	3	3	1	2	4	4	2
26	2	3	2	4	4	2	1	3	3	1
27	2	3	3	1	1	3	4	2	2	4
28	2	3	4	2	2	4	3	1	1	3
29	2	4	1	3	4	2	4	2	1	3
30	2	4	2	4	3	1	3	1	2	4
31	2	4	3	1	2	4	2	4	3	1
32	2	4	4	2	1	3	1	3	4	2

$L_{36}(2^{11}x3^{12})$

Nº	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	1	2	3	4
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3	1	1	1	1
4	1	1	1	1	1	2	2	2	2	2	2	1	1	1	1	2	2	2	2	3	3	3	3	1	2	2	1
5	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3	3	3	3	1	1	1	1	1	2	2	1
6	1	1	1	1	1	2	2	2	2	2	2	3	3	3	3	1	1	1	1	2	2	2	2	1	2	2	1
7	1	1	2	2	2	1	1	1	2	2	2	1	1	2	3	1	2	3	3	1	2	2	3	2	1	2	1
8	1	1	2	2	2	1	1	1	2	2	2	2	2	3	1	2	3	1	1	2	3	3	1	2	1	2	1
9	1	1	2	2	2	1	1	1	2	2	2	3	3	1	2	3	1	2	2	3	1	1	2	2	1	2	1
10	1	2	1	2	2	1	2	2	1	1	2	1	1	3	2	1	3	2	3	2	1	3	2	2	2	1	1
11	1	2	1	2	2	1	2	2	1	1	2	2	2	1	3	2	1	3	1	3	2	1	3	2	2	1	1
12	1	2	1	2	2	1	2	2	1	1	2	3	3	2	1	3	2	1	2	1	3	2	1	2	2	1	1
13	1	2	2	1	2	2	1	2	1	2	1	1	2	3	1	3	2	1	3	3	2	1	2	1	1	1	2
14	1	2	2	1	2	2	1	2	1	2	1	2	3	1	2	1	3	2	1	1	3	2	3	1	1	1	2
15	1	2	2	1	2	2	1	2	1	2	1	3	1	2	3	2	1	3	2	2	1	3	1	1	1	1	2
16	1	2	2	2	1	2	2	1	2	1	1	1	2	3	2	1	1	3	2	3	3	2	1	1	2	2	2
17	1	2	2	2	1	2	2	1	2	1	1	2	3	1	3	2	2	1	3	1	1	3	2	1	2	2	2
18	1	2	2	2	1	2	2	1	2	1	1	3	1	2	1	3	3	2	1	2	2	1	3	1	2	2	2
19	2	1	2	2	1	1	2	2	1	2	1	1	2	1	3	3	3	1	2	2	1	2	3	2	1	2	2
20	2	1	2	2	1	1	2	2	1	2	1	2	3	2	1	1	1	2	3	3	2	3	1	2	1	2	2
21	2	1	2	2	1	1	2	2	1	2	1	3	1	3	2	2	2	3	1	1	3	1	2	2	1	2	2
22	2	1	2	1	2	2	2	1	1	1	2	1	2	2	3	3	1	2	1	1	3	3	2	2	2	1	2
23	2	1	2	1	2	2	2	1	1	1	2	2	3	3	1	1	2	3	2	2	1	1	3	2	2	1	2
24	2	1	2	1	2	2	2	1	1	1	2	3	1	1	2	2	3	1	3	3	2	2	1	2	2	1	2
25	2	1	1	2	2	2	1	2	2	1	1	1	3	2	1	2	3	3	1	3	1	2	2	1	1	1	3
26	2	1	1	2	2	2	1	2	2	1	1	2	1	3	2	3	1	1	2	1	2	3	3	1	1	1	3
27	2	1	1	2	2	2	1	2	2	1	1	3	2	1	3	1	2	2	3	2	3	1	1	1	1	1	3
28	2	2	2	1	1	1	1	2	2	1	2	1	3	2	2	2	1	1	3	2	3	1	3	1	2	2	3
29	2	2	2	1	1	1	1	2	2	1	2	2	1	3	3	3	2	2	1	3	1	2	1	1	2	2	3
30	2	2	2	1	1	1	1	2	2	1	2	3	2	1	1	1	3	3	2	1	2	3	2	1	2	2	3
31	2	2	1	2	1	2	1	1	1	2	2	1	3	3	3	2	3	2	2	1	2	1	1	2	1	2	3
32	2	2	1	2	1	2	1	1	1	2	2	2	1	1	1	3	1	3	3	2	3	2	2	2	1	2	3
33	2	2	1	2	1	2	1	1	1	2	2	3	2	2	2	1	2	1	1	3	1	3	3	2	1	2	3
34	2	2	1	1	2	1	2	1	2	2	1	1	3	1	2	3	2	3	1	2	2	3	1	2	2	1	3
35	2	2	1	1	2	1	2	1	2	2	1	2	1	2	3	1	3	1	2	3	3	1	2	2	2	1	3
36	2	2	1	1	2	1	2	1	2	2	1	3	2	3	1	2	1	2	3	1	1	2	3	2	2	1	3



(a)



(b)

$L_{50}(2x5^{11})$

(Continua)

Nº	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	2	2	2	2	2	2	2	2	2	2
3	1	1	3	3	3	3	3	3	3	3	3	3
4	1	1	4	4	4	4	4	4	4	4	4	4
5	1	1	5	5	5	5	5	5	5	5	5	5
6	1	2	1	2	3	4	5	1	2	3	4	5
7	1	2	2	3	4	5	1	2	3	4	5	1
8	1	2	3	4	5	1	2	3	4	5	1	2
9	1	2	4	5	1	2	3	4	5	1	2	3
10	1	2	5	1	2	3	4	5	1	2	3	4
11	1	3	1	3	5	2	4	4	1	3	5	2
12	1	3	2	4	1	3	5	5	2	4	1	3
13	1	3	3	5	2	4	1	1	3	5	2	4
14	1	3	4	1	3	5	2	2	4	1	3	5
15	1	3	5	2	4	1	3	3	5	2	4	1
16	1	4	1	4	2	5	3	5	3	1	4	2
17	1	4	2	5	3	1	4	1	4	2	5	3
18	1	4	3	1	4	2	5	2	5	3	1	4
19	1	4	4	2	5	3	1	3	1	4	2	5
20	1	4	5	3	1	4	2	4	2	5	3	1
21	1	5	1	5	4	3	2	4	3	2	1	5
22	1	5	2	1	5	4	3	5	4	3	2	1
23	1	5	3	2	1	5	4	1	5	4	3	2
24	1	5	4	3	2	1	5	2	1	5	4	3
25	1	5	5	4	3	2	1	3	2	1	5	4
26	2	1	1	1	4	5	4	3	2	5	2	3
27	2	1	2	2	5	1	5	4	3	1	3	4
28	2	1	3	3	1	2	1	5	4	2	4	5
29	2	1	4	4	2	3	2	1	5	3	5	1
30	2	1	5	5	3	4	3	2	1	4	1	2
1	2	2	1	2	1	3	3	2	4	5	5	4
32	2	2	2	3	2	4	4	3	5	1	1	5
33	2	2	3	4	3	5	5	4	1	2	2	1
34	2	2	4	5	4	1	1	5	2	3	3	2
35	2	2	5	1	5	2	2	1	3	4	4	3
36	2	3	1	3	3	1	2	5	5	4	2	4
37	2	3	2	4	4	2	3	1	1	5	3	5

(Conclusão)

38	2	3	3	5	5	3	4	2	2	1	4	1
39	2	3	4	1	1	4	5	3	3	2	5	2
40	2	3	5	2	2	5	1	4	4	3	1	3
41	2	4	1	4	5	4	1	2	5	2	3	3
42	2	4	2	5	1	5	2	3	1	3	4	4
43	2	4	3	1	2	1	3	4	2	4	5	5
44	2	4	4	2	3	2	4	5	3	5	1	1
45	2	4	5	3	4	3	5	1	4	1	2	2
46	2	5	1	5	2	2	5	3	4	4	3	1
47	2	5	2	1	3	3	1	4	5	5	4	2
48	2	5	3	2	4	4	2	5	1	1	5	3
49	2	5	4	3	5	5	3	1	2	2	1	4
50	2	5	5	4	1	1	4	2	3	3	2	5

$L_{54}(2^1x3^{25})$

(Continua)

Nº	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	1	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	1	1	2	2	2	2	2	2	1	1	1	1	1	1	2	3	2	3	2	3	2	3	2	3	2	3
5	1	1	2	2	2	2	2	2	2	2	2	2	2	2	3	1	3	1	3	1	3	1	3	1	3	1
6	1	1	2	2	2	2	2	2	3	3	3	3	3	3	1	2	1	2	1	2	1	2	1	2	1	2
7	1	1	3	3	3	3	3	3	1	1	1	1	1	1	3	2	3	2	3	2	3	2	3	2	3	2
8	1	1	3	3	3	3	3	3	2	2	2	2	2	2	1	3	1	3	1	3	1	3	1	3	1	3
9	1	1	3	3	3	3	3	3	3	3	3	3	3	3	2	1	2	1	2	1	2	1	2	1	2	1
10	1	2	1	1	2	2	3	3	1	1	2	2	3	3	1	1	1	1	2	3	2	3	3	2	3	2
11	1	2	1	1	2	2	3	3	2	2	3	3	1	1	2	2	2	2	3	1	3	1	1	3	1	3
12	1	2	1	1	2	2	3	3	3	3	1	1	2	2	3	3	3	3	1	2	1	2	2	1	2	1
13	1	2	2	2	3	3	1	1	1	1	2	2	3	3	2	3	2	3	3	2	3	2	1	1	1	1
14	1	2	2	2	3	3	1	1	2	2	3	3	1	1	3	1	3	1	1	3	1	3	2	2	2	2
15	1	2	2	2	3	3	1	1	3	3	1	1	2	2	1	2	1	2	2	1	2	1	3	3	3	3
16	1	2	3	3	1	1	2	2	1	1	2	2	3	3	3	2	3	2	1	1	1	1	2	3	2	3
17	1	2	3	3	1	1	2	2	2	2	3	3	1	1	1	3	1	3	2	2	2	2	3	1	3	1
18	1	2	3	3	1	1	2	2	3	3	1	1	2	2	2	1	2	1	3	3	3	3	1	2	1	2
19	1	3	1	2	1	3	2	3	1	2	1	3	2	3	1	1	2	3	1	1	3	2	2	3	3	2
20	1	3	1	2	1	3	2	3	2	3	2	1	3	1	2	2	3	1	2	2	1	3	3	1	1	3
21	1	3	1	2	1	3	2	3	3	1	3	2	1	2	3	3	1	2	3	3	2	1	1	2	2	1
22	1	3	2	3	2	1	3	1	1	2	1	3	2	3	2	3	3	2	2	3	1	1	3	2	1	1
23	1	3	2	3	2	1	3	1	2	3	2	1	3	1	3	1	1	3	3	1	2	2	1	3	2	2
24	1	3	2	3	2	1	3	1	3	1	3	2	1	2	1	2	2	1	1	2	3	3	2	1	3	3
25	1	3	3	1	3	2	1	2	1	2	1	3	2	3	3	2	1	1	3	2	2	3	1	1	2	3
26	1	3	3	1	3	2	1	2	2	3	2	1	3	1	1	3	2	2	1	3	3	1	2	2	3	1
27	1	3	3	1	3	2	1	2	3	1	3	2	1	2	2	1	3	3	2	1	1	2	3	3	1	2
28	2	1	1	3	3	2	2	1	1	3	3	2	2	1	1	1	3	2	3	2	2	3	2	3	1	1
29	2	1	1	3	3	2	2	1	2	1	1	3	3	2	2	2	1	3	1	3	3	1	3	1	2	2
30	2	1	1	3	3	2	2	1	3	2	2	1	1	3	3	3	2	1	2	1	1	2	1	2	3	3
31	2	1	2	1	1	3	3	2	1	3	3	2	2	1	2	3	1	1	1	1	3	2	3	2	2	3
32	2	1	2	1	1	3	3	2	2	1	1	3	3	2	3	1	2	2	2	2	1	3	1	3	3	1
33	2	1	2	1	1	3	3	2	3	2	2	1	1	3	1	2	3	3	3	3	2	1	2	1	1	2
34	2	1	3	2	2	1	1	3	1	3	3	2	2	1	3	2	2	3	2	3	1	1	1	1	3	2
35	2	1	3	2	2	1	1	3	2	1	1	3	3	2	1	3	3	1	3	1	2	2	2	2	1	3
36	2	1	3	2	2	1	1	3	3	2	2	1	1	3	2	1	1	2	1	2	3	3	3	3	2	1
37	2	2	1	2	3	1	3	2	1	2	3	1	3	2	1	1	2	3	3	2	1	1	3	2	2	3

(Conclusão)

38	2	2	1	2	3	1	3	2	2	3	1	2	1	3	2	2	3	1	1	3	2	2	1	3	3	1
39	2	2	1	2	3	1	3	2	3	1	2	3	2	1	3	3	1	2	2	1	3	3	2	1	1	2
40	2	2	2	3	1	2	1	3	1	2	3	1	3	2	2	3	3	2	1	1	2	3	1	1	3	2
41	2	2	2	3	1	2	1	3	2	3	1	2	1	3	3	1	1	3	2	2	3	1	2	2	1	3
42	2	2	2	3	1	2	1	3	3	1	2	3	2	1	1	2	2	1	3	3	1	2	3	3	2	1
43	2	2	3	1	2	3	2	1	1	2	3	1	3	2	3	2	1	1	2	3	3	2	2	3	1	1
44	2	2	3	1	2	3	2	1	2	3	1	2	1	3	1	3	2	2	3	1	1	3	3	1	2	2
45	2	2	3	1	2	3	2	1	3	1	2	3	2	1	2	1	3	3	1	2	2	1	1	2	3	3
46	2	3	1	3	2	3	1	2	1	3	2	3	1	2	1	1	3	2	2	3	3	2	1	1	2	3
47	2	3	1	3	2	3	1	2	2	1	3	1	2	3	2	2	1	3	3	1	1	3	2	2	3	1
48	2	3	1	3	2	3	1	2	3	2	1	2	3	1	3	3	2	1	1	2	2	1	3	3	1	2
49	2	3	2	1	3	1	2	3	1	3	2	3	1	2	2	3	1	1	3	2	1	1	2	3	3	2
50	2	3	2	1	3	1	2	3	2	1	3	1	2	3	3	1	2	2	1	3	2	2	3	1	1	3
51	2	3	2	1	3	1	2	3	3	2	1	2	3	1	1	2	3	3	2	1	3	3	1	2	2	1
52	2	3	3	2	1	2	3	1	1	3	2	3	1	2	3	2	2	3	1	1	2	3	3	2	1	1
53	2	3	3	2	1	2	3	1	2	1	3	1	2	3	1	3	3	1	2	2	3	1	1	3	2	2
54	2	3	3	2	1	2	3	1	3	2	1	2	3	1	2	1	1	2	3	3	1	2	2	1	3	3