

UNIVERSIDADE ESTADUAL DE CAMPINAS-UNICAMP  
FACULDADE DE ENGENHARIA QUÍMICA-FEQ

ÁREA DE CONCENTRAÇÃO:  
SISTEMAS DE PROCESSOS QUÍMICOS E INFORMÁTICA

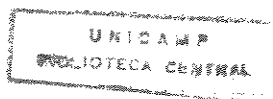
RECONCILIAÇÃO E RETIFICAÇÃO DE DADOS  
E  
CLASSIFICAÇÃO DE VARIÁVEIS  
DE PROCESSO

Autora:  
TALITA FURLANETTO MENDES  
Engenheiro Químico-M.E.

Orientador:  
Dr. JOÃO ALEXANDRE FERREIRA DA ROCHA PEREIRA

Tese apresentada no Curso de Pós-Graduação da FEQ-UNICAMP como parte dos requisitos para obter o título de Doutor em Engenharia Química.

Campinas-SP  
Brasil  
Março, 1995



9513617

UNIDADE	BC
INSTITUIÇÃO	UNICAMP
NUM. DE IDENTIFICAÇÃO	M522r
DATA DE AQUISIÇÃO	25/9/95
PREÇO	433,95
DATA DE AQUISIÇÃO	0 [ ] 0 [x]
PREÇO	78\$11,00
DATA	18/10/95
N.º CPD	

CM-00078452-2

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA CENTRAL - UNICAMP

Mendes, Talita Furlanetto

M522r

Reconciliação e retificação de dados e classificação de variáveis de processo / Talita Furlanetto Mendes.-- Campinas. SP : [s.n.], 1995.

Orientador: João Alexandre Ferreira da Rocha Pereira.  
Tese ( doutorado ) - Universidade Estadual de Campinas, Faculdade de Engenharia Química.

1. Processos Químicos - Análise. 2.Redundância (Engenharia ). 3.\* Reconciliação de dados. 4.\*Retificação de dados. I. Pereira, João Alexandre Ferreira da Rocha. II. Universidade Estadual de Campinas. Faculdade de Engenharia Química. III. Título.

Para

Geraldo  
Carolina e Henrique  
Angelo e Carolina

*O que é difícil se torna fácil,  
quando passa de desconhecido a conhecido  
e, nesta passagem, parece encolher,  
porque deixa um vazio.*



## AGRADECIMENTOS

À Universidade Federal do Rio Grande do Sul-UFRGS, pela concessão de bolsa de estudos através do Programa Institucional de Capacitação Docente-PICD/CAPES.

Ao Prof. João Alexandre F. da Rocha Pereira por me conduzir ao estudo deste assunto e pela orientação.

Aos professores da FEQ que contribuíram para minha capacitação.

Aos professores da Escola de Engenharia da UFRGS e, em especial, aos do Departamento de Engenharia Química, que aprovaram meu afastamento para a minha capacitação.

Ao Prof. Argimiro Secchi por me indicar caminhos em alguns problemas referentes à Linguagem C.

A Paulo Cesar Furlanetto Marques por me ajudar a resolver problemas de máquina.

Aos funcionários das bibliotecas da Escola de Engenharia da UFRGS e da Fundação de Ciência e Tecnologia do Estado do Rio Grande do Sul, que me auxiliaram.

A meu esposo Geraldo S. Mendes e a meus filhos, Carolina e Henrique, porque conviveram comigo durante a realização deste trabalho.

A todos que, de alguma forma, contribuíram para a concretização deste trabalho.

## RESUMO

Os dados de processo, obtidos em unidades industriais, são a base para decisões que vão influir na qualidade dos produtos que gera e de ações que podem, em certos casos, afetar a segurança industrial. Quando obtidos em equipamentos em escala de laboratório, são o alicerce onde se fundamentam o projeto de novos processos. Em vista disto, quanto mais próximo os valores das quantidades medidas estiverem dos valores verdadeiros correspondentes, mais acertadas e confiáveis as ações fundamentadas na sua análise.

Neste trabalho, foram desenvolvidos procedimentos computacionais para reconciliação e retificação de dados de processo, que buscam obter as melhores estimativas, de acordo com algum critério, para as quantidades medidas, seguidas da obtenção de estimativas para as quantidades não-medidas observáveis. Eles são aplicáveis a balanços de massa multicomponentes, sujeitos a restrições lineares, podendo ou não ocorrer reações químicas.

Estes procedimentos utilizam projeção matricial para simplificação do problema de obtenção dos ajustes das quantidades medidas, o que é feito através do método dos mínimos quadrados ponderados e dos multiplicadores de Lagrange. Do conjunto de restrições, fazem parte, as equações de balanços de massa, podendo ser incluídas restrições adicionais, como a razão de partição conhecida de um divisor de fluxo. Considera-se que, quando a composição de um componente de uma corrente total é medida, a vazão total daquela corrente também é medida. As vazões medidas das correntes totais são sempre incluídas, o que pode levar à redundância, além de assegurar que a soma das vazões ajustadas dos componentes de uma corrente seja igual à vazão ajustada da corrente total.

Na retificação de dados, são utilizados três testes estatísticos para detectar e identificar a presença de fontes de erros grosseiros, sendo um deles um teste global, aplicado ao processo como um todo, que é testado contra uma distribuição qui-quadrado. Os outros dois, são um teste de medida de potência máxima, aplicado a cada medida individual, e um teste de desbalanço nodal, aplicado a cada restrição remanescente, após a remoção das quantidades não-medidas, que são testados contra uma distribuição gaussiana.

Relatam-se os resultados da aplicação do programa de reconciliação de dados (Programa RECON) a cinco processos distintos: a um processo de síntese de amônia que contém quatro nós, a um reator químico, onde ocorrem duas reações químicas, a uma rede de vapor com nove nós e vinte correntes, a um processo de síntese de ácido nítrico com oito nós e sete espécies químicas e a um circuito de moagem.

Relatam-se os resultados da aplicação do programa de retificação de dados (Programa RETIF) a cinco casos gerados para o processo de síntese de amônia, a quatro casos do processo de síntese de ácido nítrico e a três casos do circuito de moagem.

Paralelamente, foi elaborado um procedimento computacional para classificação de variáveis de processo, designado TCLASS, aplicável a variáveis de balanços de massa multicomponentes, em que as restrições se referem essencialmente a quantidades extensivas, a menos das equações de normalização. As vazões e concentrações de componentes medidas são classificadas em redundantes e em não-redundantes e as vazões não-medidas e a conversão de cada reação química são classificadas em observáveis e não-observáveis. Este programa pode

ser utilizado para definição da colocação de pontos de medição de sistemas de monitoramento de processos.

Relatam-se, a aplicação do programa TCLASS a oito casos do processo de síntese de amônia e a dois casos de um processo de evaporação-cristalização de nitrato de potássio.

## SUMÁRIO

1. Introdução	1
2. Revisão da Literatura	4
3. Reconciliação de Dados de Processo	20
3.1 Introdução	21
3.2 Obtenção de Informações Básicas	22
3.2.1 Valores das Variáveis Medidas	22
3.2.2 A Matriz Variância-Covariância	22
3.2.3 A Estrutura do Processo	24
3.3 Procedimentos para Reconciliação de Dados de Processo	26
3.4 Decomposição do Problema de Reconciliação de Dados de Processo	27
3.5 Reconciliação de Dados de Processo Através de Projeção Matricial	29
3.6 Bases do Procedimento Computacional de Reconciliação de Dados e Cooptação	32
3.7 O Programa de Reconciliação de Dados de Processo	35
3.8 Aplicações do Programa RECON	38
4. Retificação de Dados de Processo	51
4.1 Introdução	52
4.2 Os Dados de Processo e os Erros	52
4.3 Tratamento de Dados de Processo Contendo Erros Grosseiros	53
4.4 Retificação de Dados de Processo Através de Projeção Matricial	58
4.5 Bases do Procedimento Computacional para Retificação de Dados e Cooptação	64
4.6 O Programa de Retificação de Dados de Processo	66
4.7 Aplicações do Programa RETIF	67
5. Classificação de Variáveis de Processo	80
5.1 Introdução	81
5.2 Procedimentos Para Classificação de Variáveis de Processo	81
5.3 Classificação de Variáveis de Processo Através de Projeção Matricial	83
5.4 Bases do Procedimento Computacional de Classificação de Variáveis de Processo	89
5.5 O Programa de Classificação de Variáveis de Processo	91
5.6 Aplicações do Programa TCLASS	92
6. Conclusões e Sugestões	100
6.1 Conclusões	101
6.2 Sugestões	103
Apêndice A. Métodos Matemáticos	104
Apêndice B. Conceitos e Testes Estatísticos	112
Apêndice C. Listagem do Programa RECON	124
Apêndice D. Listagem do Programa RETIF	144
Apêndice E. Listagem do Programa TCLASS	185
Lista de Símbolos	213
Referências Bibliográficas	216
Abstract	222

---

## CAPÍTULO 1. INTRODUÇÃO

Os valores medidos de variáveis de processo podem conter erros de medida, pois são obtidos através de instrumentos que, além de serem inerentemente imprecisos, são utilizados para a medição de variáveis aleatórias. O caráter aleatório das medidas experimentais pode ser decorrente de vários fenômenos, como ruídos de processo, ruídos de sinal e perturbações ambientais. Portanto, um dado conjunto de valores medidos pode levar ao não-obedecimento das leis fundamentais de conservação de massa e de energia, referentes ao processo em questão. Estes erros podem ser de dois tipos: erros aleatórios, considerados independentes e como tendo uma distribuição normal com média zero e erros grosseiros, que são causados por eventos não-aleatórios, como instrumentos descalibrados, mau funcionamento de instrumentos de medição ou devido ao emprego de modelo do processo incompleto ou impreciso.

Por outro lado, eles são a base de atividades importantes como projeto, planejamento da produção, análise, controle e otimização de processo e análise de falha de uma unidade. Em vista disso, são utilizados para obter estimativas de parâmetros de processo, como eficiência de estágio, coeficientes de transferência de calor e de massa e velocidades de reações químicas, por exemplo.

O uso de dados contendo erros pode conduzir ao mau desempenho de uma unidade e a operações perigosas. É válido notar que, em alguns casos, as atividades de controle e otimização de um processo estão baseadas em pequenas melhorias em seu desempenho, de modo que erros nos dados ou métodos arbitrários para analisá-los, podem mascarar ou ampliar mudanças reais no desempenho de uma unidade. Por outro lado, os dados de laboratório ou de plantas-piloto, que contém erros e são usados para desenvolver modelos para o projeto de unidades, podem levar ao mau dimensionamento das unidades do processo.

Quando os dados medidos são sujeitos somente a erros aleatórios, eles podem ser aprimorados através da reconciliação de dados de processo, que trata da obtenção de estimativas (valores corrigidos) para os valores medidos através de um critério previamente estabelecido, sendo o método dos mínimos quadrados ponderados, o comumente empregado. Estes valores corrigidos são utilizados, muitas vezes, para a obtenção de estimativas para valores que não foram medidos, em decorrência daquelas situações onde as correntes são sujeitas a condições operacionais extremas que impedem a amostragem ou medição, ou contém instrumento danificado ou não são medidos por razões econômicas. A obtenção de estimativas para valores não-medidos se chama coaptação.

Quando há suspeita de que os dados medidos contém valores com erros grosseiros ou grosseiros, é necessário identificar e remover as quantidades a eles correspondentes, antes da reconciliação final dos dados. Caso contrário, ela pode levar a uma série de pequenas incorreções nas demais quantidades medidas, originando dados ruins, isto sem considerar o fato de que a presença de erros grosseiros invalida a base estatística dos procedimentos de reconciliação de dados. A obtenção de estimativas para os valores medidos, que considera como possível a presença de erros grosseiros, recebe a designação de retificação de dados.

É importante notar que nem sempre é possível ajustar todos os dados medidos. Para que haja ajustes é necessário que as medidas sejam redundantes ou tenham covariância com outras quantidades medidas. Da mesma forma, nem sempre é possível estimar os valores de quantidades não-medidas, que podem ou não ser observáveis. Esta questão é tratada na classificação de variáveis de processo, que permite determinar quais dos valores não-medidos que podem ser unicamente determinados a partir dos valores medidos. Por outro lado, no caso

de quantidades medidas pode ser fundamental conhecer seus valores mesmo se houver falha nos instrumentos responsáveis por sua medição. A classificação de variáveis de processo contempla a classificação de quantidades medidas em redundantes e não-redundantes.

Além disso, a classificação de variáveis de processo pode ser usada para analisar o desempenho de sistemas de monitoramento e definir a colocação de instrumentos de medição de unidades de processo.

Estes problemas de haver erros nos dados sempre existiram. Entretanto, quando as medições e a aquisição de dados eram feitas, habitualmente, por operadores de planta e engenheiros, a precisão e consistência de dados era verificada com base em seu conhecimento e experiência do processo. Agora, quando estas atividades são realizadas, automaticamente, por computadores, um volume maior de dados deve ser verificado sem o benefício da vigilância e intervenção humana. Então, surge a necessidade de se contar com uma base racional e sistemática para análise e tratamento de dados de processo, que deve ser implementada em computador.

Em vista disto, buscou-se o desenvolvimento de metodologias e procedimentos computacionais, que tratam da reconciliação e da retificação de dados de processo de sistemas multicomponentes, no estado estacionário, sujeitos a restrições lineares, e da classificação de variáveis de processo, cujos desempenhos são verificados pela aplicação a diversos casos. As metodologias e procedimentos estudados levaram ao desenvolvimento de três programas computacionais que focalizam especificamente cada uma dessas situações, que são tratadas em detalhe nos capítulos 3, 4 e 5.

## CAPÍTULO 2. REVISÃO DA LITERATURA



Os dados de processo são o alicerce em que se fundamentam atividades de controle de processo, como planejamento e controle da produção e programação da operação, bem como da análise da performance do processo. Portanto, quanto mais próximos esses dados estiverem dos valores reais, mais confiáveis serão os resultados das análises deles derivadas. Por outro lado, se os valores medidos estiverem mascarados por erros de medição consideráveis, eles poderão conduzir a ações de controle inadequadas e indesejáveis.

Inicialmente, a aquisição de dados de processo era uma atividade humana e a precisão e consistência dos valores medidos eram verificadas pelo analista, com base em seu conhecimento do processo e em sua experiência. Entretanto, quando ela passou a ser feita por computadores, tornou-se necessária a existência de algoritmos para testar os dados medidos. A instalação de computadores digitais no meio-ambiente operacional de refinarias e complexos químicos, bem como os avanços na instrumentação de processos, possibilitam que a aquisição de dados seja feita de forma ampla, freqüente e regular. Todavia, para que sejam realmente úteis, é necessário dispor de metodologia que permita a obtenção de estimativas confiáveis e consistentes das variáveis de processo, a partir dos valores medidos.

Foi na década de 60, que surgiram os primeiros estudos sobre procedimentos para ajuste de dados de processo e testes para verificação de dados. Desde então, este assunto tem sido objeto de vários trabalhos, com enfoques diversos, entre os quais se incluem a questão da detecção e identificação de erros grosseiros e a classificação de variáveis medidas e não-medidas.

Apresenta-se, a seguir, uma análise de trabalhos sobre ajuste de dados de processo, que tem caráter amplo, quanto ao tipo de restrições (lineares e não-lineares) e quando às condições de operação (estado estacionário ou não-estacionário), por exemplo. Incluem-se, ainda, assuntos relacionados como classificação de variáveis de processo. Para dar uma idéia da forma como foi obtida, informa-se que aborda trabalhos publicados na literatura, a partir de meados de 1960, que foram identificados, inicialmente, a partir de referências cruzadas do artigo de Crowe (1986) e de pesquisa no Engineering Index, no período de 1985 a 1992 e no Current Contents, entre 1991 e agosto de 1994. Além dos trabalhos apresentados a seguir, outros podem ser encontrados no item 4.3, que inclui artigos sobre o problema da detecção e identificação de erros grosseiros.

Ripps (1965), para resolver o problema de ajuste de dados experimentais para o fechamento de balanços de massa e de energia de sistemas no estado estacionário, empregou o método dos mínimos quadrados ponderados, que tinha as equações dos balanços de massa nodais como condições, e utilizou os multiplicadores de Lagrange para resolver o sistema de equações. Observa que se deve fazer distinção entre dois tipos de erros nos dados: pequenos erros aleatórios e erros grosseiros, embora não defina um critério para sua identificação. Acrescenta que, quando há erros grosseiros, as medidas a eles correspondentes devem ser eliminadas, após o que se recalculam os ajustes. Apresenta um algoritmo que facilita a repetição do procedimento de ajuste, no caso em que se deseja estudar o efeito da remoção de várias medidas, uma de cada vez. Relata que, freqüentemente, ao ajuste total mínimo corresponde o conjunto de dados mais correto.

Václavek (1969a) estudou o ajuste de dados de processo de balanços de engenharia de sistemas complexos, utilizando o método dos mínimos quadrados, cuja solução foi obtida através dos multiplicadores de Lagrange. Discute o problema de como escolher um esquema de balanço de correntes medidas, para que a combinação de dados medidos seja consistente

com as equações de balanço. Simplifica o sistema de equações, a ser resolvido, analisando a topologia do sistema e unindo nós ligados por correntes não-medidas. Considera somente o balanço de uma única quantidade, que pode ser, por exemplo, um balanço total de massa ou o de um dos componentes.

Václavěk (1969b), estudando o aprimoramento de dados medidos de balanços, em sistemas complexos de engenharia química, derivou um algoritmo para classificar as correntes medidas e as correntes não-medidas, de acordo com suas propriedades topológicas no esquema de balanço. As medidas são dispostas em dois grupos e em um deles se incluem aquelas que podem ter seu valor obtido por medida direta e também através dos valores de outras correntes medidas. No outro, são colocadas as que somente podem ser determinadas pela medida direta. No caso de correntes não-medidas, também são possíveis duas alternativas: num dos grupos são colocadas aquelas que podem ser determinadas, através das relações de balanço, a partir das estimativas dos valores medidos, e no outro, aquelas que não podem ser determinadas. Define o que chama de esquema de balanço reduzido (RBS), obtido pela simplificação do balanço original do sistema, do qual foram retiradas todas as correntes não-medidas e as correntes medidas que não podem ser corrigidas. Observa que o valor mínimo da função objetivo do RBS é o mesmo que o da função correspondente ao esquema de balanço inicial.

Umeda et alii. (1971) tratam do problema de ajuste de dados e de estimar parâmetros de plantas químicas, como um problema de otimização sob várias restrições, que são as expressões matemáticas das relações entre as variáveis de processo, em condições estacionárias. Utilizaram o Método Complexo de Box para resolver o problema de otimização, que é aplicado a um processo constituído por dois reatores de tanque agitado contínuos (CSTR) e duas colunas de destilação com malha de reciclo.

Nogita (1972) utilizou o método dos mínimos quadrados ponderados para o ajuste de dados de processo, no estado estacionário, e um critério de teste estatístico, para verificar a presença de erros grosseiros. Apresenta um algoritmo para a identificação de erros grosseiros pela eliminação seqüencial de valores medidos, onde compara o efeito da remoção de cada medida no valor numérico da função-teste proposta, para chegar ao conjunto consistente mais provável, que corresponde ao menor valor da função. Considera o caso em que todas variáveis são medidas.

Murthy (1973) apresenta um procedimento simples para ajustar vazões molares de várias espécies, que entram e saem de um reator químico, de forma que as equações de conservação dos elementos sejam obedecidas, sendo estas as únicas restrições impostas. Utilizou o método dos mínimos quadrados ponderados, sujeito às equações de conservação. O problema é resolvido utilizando os multiplicadores de Lagrange. Este método não exige qualquer conhecimento das reações químicas que ocorrem no reator.

Murthy (1974) estudou o ajuste de dados de vazões, através de um reator químico, de modo a satisfazer as equações de balanços de massa, quando as reações são conhecidas e existem restrições adicionais, tais como a irreversibilidade de certas reações e o fato de certas espécies serem inertes, sob as condições de reação. Observa que, quando as reações são conhecidas, é melhor formular o problema de ajuste de dados, utilizando um conjunto de reações independentes. Porém, se o número de reações químicas independentes no sistema for igual ao máximo possível (que é igual ao número de espécies químicas menos a característica da matriz dos elementos) e se as reações independentes forem todas reversíveis, as correções obtidas usando este procedimento são idênticas às que seriam obtidas através de método

descrito no estudo citado no parágrafo anterior. Apresenta um conjunto de fórmulas simples para resolver o problema proposto, utilizando o método dos mínimos quadrados ponderados e a substituição algébrica direta para resolver o sistema de equações obtido.

Almásy e Sztanó (1975) estudaram a verificação da existência de erros e a correção de medidas de variáveis de processos, descritos por modelos lineares. Observam que, à estimativa de medidas por mínimos quadrados, correspondem, também, a estimativa de probabilidade máxima e a estimativa não-tendenciosa de variância mínima, quando os erros de medida seguem uma distribuição Gaussiana com média zero. Isto ocorre quando as medidas são sujeitas somente a pequenos erros aleatórios. A estatística-teste proposta para verificar a presença de erros grosseiros se baseia na distribuição qui-quadrado, sendo aplicada ao conjunto de dados como um todo, cuja matriz dispersão dos erros é conhecida. Apresentam um algoritmo para procurar a fonte de erro grosseiro, considerando que todas as suposições sobre a distribuição dos erros se mantêm, exceto o valor esperado zero do erro de uma única medida. Quando a matriz dispersão dos erros não é conhecida, sugerem o uso de outro algoritmo para a verificação dos dados, que se baseia na matriz dispersão dos defeitos do modelo. Observam que, na prática, não se tem uma matriz exata de dispersão dos erros, mas uma aproximação. Todavia, provam que os erros das estimativas não são sensíveis a pequenos erros na estimativa dessa matriz. O método de pesquisa de erros grosseiros tem como limitação o fato de admitir a possibilidade da presença de um único erro sistemático, sendo adequado ao caso em que o quociente entre o erro grosseiro e a dispersão dos erros regulares não é muito pequeno.

Václavek et alii.(1976a) descrevem um método para ajustar parâmetros de correntes medidas, no cálculo de balanços de massa multicomponentes e citam as relações básicas para resolver os ajustes. Os parâmetros selecionados são as vazões totais e as concentrações dos componentes das correntes. Observam que nem sempre todos os parâmetros das correntes são medidos e que, então, pode-se reduzir o fluxograma de massa inicial. Utilizam os conceitos de fluxograma reduzido de primeiro e de segundo grau e classificam as correntes em quatro categorias, de acordo com sua distribuição de parâmetros medidos e não-medidos. Utilizaram o método dos mínimos quadrados ponderados para obtenção das estimativas, o qual foi resolvido utilizando o método dos multiplicadores de Lagrange. Em outro artigo (1976b), descrevem os algoritmos para a obtenção dos fluxogramas reduzidos de primeiro e de segundo grau e para classificação dos parâmetros das correntes, que tornam possível simplificar a solução do sistema de equações. Uma restrição desse estudo é que prevê, somente, a possibilidade de classificação das correntes em quatro categorias, contemplando unicamente os casos em que todos os componentes de uma corrente têm sua concentração medida ou que a concentração de nenhum deles é medida, quando o parâmetro considerado é a concentração.

Václavek e Louchka (1976) apresentam dois algoritmos relacionados ao ajuste de dados de processo. Um deles é usado para classificar os parâmetros das correntes, que são a vazão total e a concentração dos componentes, a fim de encontrar os parâmetros medidos redundantes e os parâmetros não-medidos determináveis, para simplificar os cálculos de ajuste de dados de processo. Observam que o uso sucessivo desse algoritmo, aplicado a variações individuais de um processo, é um método indireto para projetar pontos de medição. O outro destina-se à seleção direta das medidas necessárias para obter os parâmetros desejados de uma unidade. Ele também é baseado na idéia de classificação: todos os parâmetros são classificados em mensuráveis (conjunto A) e não-mensuráveis (conjunto B), de acordo com a viabilidade técnica, e em requeridos (conjunto C) e não requeridos (conjunto D), de acordo com os objetivos dos cálculos de balanço. As correntes são agrupadas em somente quatro categorias e

os autores consideram que, quando a composição de uma corrente é medida, as concentrações de todos os seus componentes são medidas.

Mah et alii. (1976) mostram como informações inerentes às restrições do processo e estatísticas medidas podem ser utilizadas para aprimorar dados de fluxo de processo e de inventário. Provam que é possível, através de análise teórico-gráfica, simplificar a reconciliação de dados conflitantes, a qual é feita através do método dos mínimos quadrados, cuja função-objetivo é sujeita a um conjunto de restrições modificado. Este é equivalente ao conjunto inicial e é obtido agregando, dois a dois, os nós ligados por arcos (correntes) não-medidas. Os valores estimados de variáveis não-medidas são obtidos a partir das equações restritivas modificadas. Propõem o uso de uma estatística-teste aplicada, inicialmente, aos balanços nodais individuais e, posteriormente, a agregados de nós, para verificar a presença de erros grosseiros e localizá-los antes de efetuar a reconciliação de dados e a cooptação. Usando esse critério, elaboraram um esquema para a identificação das fontes de erro. Os dados considerados são as vazões totais medidas, num dado instante.

Hlavacék (1977), num artigo de revisão sobre análise de plantas complexas, aborda a questão da estimativa e ajuste de dados de processo. Observa que este problema não tem merecido a mesma ênfase dada ao problema de estimativa de parâmetros de um modelo particular ou à discriminação entre modelos rivais, os quais têm sido objeto de grande parte dos procedimentos estatísticos desenvolvidos até então. Relata que para o ajuste de balanços de um componente vem sendo empregada, desde 1964, a estimativa pelos mínimos quadrados e que o ajuste de balanços multicomponentes é mais complicado, pois as restrições são não-lineares. Registra, como possíveis, cinco modos de solução para este caso: 1) substituição algébrica direta de variáveis das restrições na função-objetivo, 2) minimização via multiplicadores de Lagrange, 3) solução direta por programação não-linear, 4) minimização por programação linear e 5) critério minimax de Chebyshev.

Knepper e Gorman (1980) apresentam as equações para ajustamento de dados e estimativa de parâmetros, desenvolvidas a partir da teoria das inversas generalizadas. Elas se destinam à análise de conjuntos de dados, sujeitos a restrições, e ao planejamento de medições experimentais, aplicando-se a casos lineares e não-lineares. Examinaram a convergência de problemas não-lineares, para o que sugerem o uso de um algoritmo computacional. Desenvolvem a matriz variância-covariância dos parâmetros calculados, a dos dados ajustados e a covariância entre elas, que são usadas na decisão da validade de medidas experimentais adicionais, para aumentar a precisão. Descrevem um teste qui-quadrado multivariado, que permite verificar as discrepâncias nas restrições, que são inconsistentes com a matriz variância-covariância dos dados medidos. Descrevem, ainda, outros testes estatísticos para que as estimativas sejam livres de erros grosseiros.

Romagnoli e Stephanopoulos (1980) estudaram a retificação de dados medidos em plantas químicas complexas, que operam no estado estacionário e a obtenção de estimativas de variáveis não-medidas. Propõem o uso de um algoritmo para classificar as correntes medidas e as não-medidas, baseado na matriz ocorrência, a fim de reduzir a tamanho do problema. Apresentam, também, diferentes algoritmos referentes à solução de problemas práticos especiais, como seleção de parâmetros corrigíveis, seleção de medidas necessárias para que um processo seja determinável e seleção de medidas necessárias para que certos parâmetros sejam determináveis. Propõem o uso de testes estocásticos para verificar a consistência de dados do processo e detectar erros grosseiros. Este estudo se aplica tanto ao caso de restrições lineares como não-lineares, considerando a situação em que quando a composição de certas correntes são medidas, elas são medidas para todos os componentes.

Romagnoli e Stephanopoulos (1981) estudaram a retificação de dados medidos de processo, na presença de erros grosseiros. Apresentam um método para identificar a fonte de erros, através da eliminação seriada de uma ou mais medidas de cada vez. Utilizaram o método dos mínimos quadrados para a obtenção dos ajustes, tendo como restrições os balanços de massa nodais. No método desenvolvido, faz-se o processamento seqüencial das equações de balanço. Argumentam que ele simplifica a identificação de erros grosseiros, durante o processo de reconciliação, e facilita o tratamento de dados de processo de grande dimensão, pois fornece um critério simples individualizado para cada balanço, ao contrário da função-teste global. O critério é baseado na distribuição qui-quadrado dos resíduos de balanço. Utilizam análise estrutural para identificar um conjunto reduzido de medidas suspeitas de erro. Este estudo é restrito ao caso de balanços lineares, que, na prática, são encontrados somente em balanços globais; todavia, pode ser aplicado ao caso de balanços de componentes e de energia linearizáveis.

Mah (1981) apresenta uma revisão sobre estudos em áreas relacionadas ao ajuste de dados de processo, envolvendo a reconciliação de dados de processo, a classificação de variáveis e medidas, a detecção e identificação de erros grosseiros e a definição de pontos de tomada de medidas, que formam uma base para o projeto e análise de sistemas de monitoramento do desempenho de processos. Observa que, inicialmente, dava-se ênfase aos procedimentos de cálculo para a reconciliação, que deu lugar à questão de classificação de variáveis e medidas e à detecção e identificação de erros grosseiros. Argumenta que trabalho posterior é necessário para avaliar e generalizar os algoritmos desenvolvidos e para integrar os ganhos específicos em diferentes aspectos de sistemas de monitoramento do desempenho, para que tenham uso industrial direto.

Stanley e Mah (1981a) apresentam dois algoritmos para determinação da observabilidade local e global e redundância de variáveis individuais e medições de processos, que se baseiam em técnicas de análise teórico-gráfica. O tratamento considera somente restrições de conservação de massa e de energia, porém observam que pode ser estendido a processos com outras restrições adicionais, como frações de partição especificadas.

Stanley e Mah (1981b) desenvolveram uma teoria sobre observabilidade e redundância aplicável a sistemas no estado estacionário, através de analogias com sistemas dinâmicos. Demonstram a importância desses conceitos na predição do desempenho qualitativo de estimadores, como os mínimos quadrados condicionados e outros. Mostram como a observabilidade local está diretamente ligada à unicidade local da solução da equação de medida e como a não-observabilidade local provoca a falha do estimador tanto no caso de estimador no estado estacionário, como no estado quase-estacionário. Para restrições e medições lineares, as condições para observabilidade local também valem para observabilidade global e o sistema é decomponível em subsistemas (redundante, estritamente observável e não-observável).

Crowe et alii.(1983) estudaram o ajuste de dados de vazões de sistemas multicomponentes, no estado estacionário, utilizando os mínimos quadrados ponderados, de forma que as leis de conservação de massa fossem obedecidas. O caso básico contém restrições que são lineares nas vazões dos componentes e na vazão total. Consideram que nem todos os parâmetros são medidos e utilizam projeção matricial para decompor o problema, objetivando, inicialmente, ajustar os dados medidos corrigíveis e, posteriormente, calcular os valores estimados de parâmetros não-medidos observáveis. O método é estendido para casos contendo restrições bilineares, que envolvem parâmetros desconhecidos, como a fração de divisão de um divisor de fluxo. Utilizam três estatísticas-teste para verificar a presença de erros

grosseiros nas medidas, aplicadas ao conjunto de dados como um todo, a cada nó e a cada medida, baseadas na distribuição normal ou na distribuição qui-quadrado.

Almásy e Mah (1984) observam que, quase sem exceção, todos os desenvolvimentos de técnicas sistemáticas para reconciliação de dados de processo se baseiam na suposição de que os erros de medida são variáveis aleatórias, que obedecem uma distribuição estatística conhecida, cuja matriz covariância é também conhecida, mas muito pouco tem sido dito sobre como este conhecimento pode ser obtido. Neste trabalho, delineiam e avaliam um método para estimar esta matriz, fazendo uso dos resíduos das restrições, que são calculados a partir dos dados disponíveis.

Tamhane e Mah (1985), em artigo de revisão da literatura, abordam a questão da reconciliação de dados e detecção de erros grosseiros em redes de processos químicos. Consideram processos que se encontram no estado estacionário e cujos valores medidos representam um instantâneo do processo, num dado momento. Assumem que, na ausência de erros grosseiros, o valor esperado do erro é zero e que a matriz covariância é conhecida e positiva por definição. Observam que o método dos mínimos quadrados ponderados é comumente usado para obter estimativas de variáveis medidas e não-medidas, devido às suas propriedades de obter a melhor estimativa no caso não-condicionado. Apresentam o método de solução devido a Crowe et alii.(1983), que é adequado ao caso em que, também, estão presentes restrições não provenientes do arranjo de fluxo das correntes. No que se refere à detecção de erros grosseiros, observam que, até pouco tempo, trabalhos de engenharia química não eram descritos no corpo da pesquisa estatística. Descrevem três tipos de testes propostos para detectar erros grosseiros. O primeiro é uma estatística-teste global, proposta por vários autores, baseada numa distribuição qui-quadrado, cuja desvantagem apontada é a necessidade de um esquema de teste para detectar as fontes de erros grosseiros, pois analisa o processo como um todo; observam que para calcular esta estatística não é necessária a reconciliação prévia dos dados. O segundo é uma estatística-teste nodal, destinada a verificar em separado cada desbalanço nodal, que é testada frente a uma distribuição normal padronizada, proposta por Reilly e Carpani, em 1963, e por Mah et alii (1976). Comentam que a aplicabilidade do algoritmo básico, desenvolvido pelos últimos, para usar esta estatística, é limitada, pois considera que os erros grosseiros em diferentes correntes, conectadas num mesmo nó, não se cancelam. Em vista disto, recomendam seu uso como teste suplementar. O terceiro é um teste de medida, proposto pelos próprios autores deste artigo de revisão, que é aplicado aos ajustes (resíduos) individuais de cada medida, que também são testados contra uma distribuição normal. Relatam uma dificuldade associada ao uso deste teste e uma forma de evitá-la: trata-se da não-identificabilidade de erros grosseiros ocorrentes sempre que as mesmas duas correntes conectam os mesmos dois nós, nos casos em que as restrições são somente devidas à rede do processo. Este inconveniente pode ser evitado incorporando outras restrições envolvendo, usualmente, informações estequiométricas adicionais. Para resolver o caso de restrições não-lineares, reportam-se ao algoritmo proposto por Knepper e Gorman (1980), do tipo Gauss-Newton, e ao caso especial descrito por Crowe et alii.(1983). Finalmente fazem comentários quanto ao problema de obtenção da matriz covariância, quando há correlação entre observações consecutivas, ao problema de observações dependentes do tempo e à necessidade de muito trabalho teórico para tratar a questão de restrições não-lineares. Além disso, observam que os esquemas de detecção de erros grosseiros devem ser aperfeiçoados para incluir restrições relacionadas com a não-negatividade natural dos parâmetros e que o não-fechamento das equações de balanço pode ser devido à mal especificação das restrições e sugerem o uso dos testes nodal e de medida combinados.

Romagnoli (1983), ao tratar do problema de reconciliação de dados em processos químicos, discute a relevância da redundância e a possibilidade de obter estimativas no processamento de dados. Define as condições gerais para a completa determinação do sistema, usando as propriedades de solução das equações de balanço. Desenvolve uma técnica recorrente para ajustamento de dados para processar as restrições do sistema, que considera adequada para um esquema de detecção e identificação de falhas. Trata ainda do problema de estimar parâmetros do sistema na presença de um vetor desvio desconhecido constante (dos erros sistemáticos), acrescentando um termo à função que modela as equações das medidas. As correções devidas correspondentes são aplicadas aos resultados do ajuste original que não considera a presença desses desvios. O caso estudado se refere a um processo no estado estacionário, descrito por um sistema linear ou que foi linearizado, cujas medidas são funcionalmente independentes.

Narasimhan et alii.(1986) descrevem um teste estatístico composto, para determinar se as variáveis de um processo sofreram mudanças de estados estacionários. Esta determinação é importante em aplicações como na reconciliação de dados de processo, cujo procedimento depende do comportamento da unidade. É chamado composto pois se realiza em duas etapas: na primeira, determina-se se as variâncias de um conjunto de variáveis (testadas simultaneamente) obtidas em dois períodos de tempo consecutivos são iguais. Na segunda etapa, de acordo com a resposta da primeira, aplica-se o teste adequado, em que se verifica se os valores verdadeiros correspondentes às variáveis, cujas variâncias foram testadas na primeira etapa, são iguais. Se a resposta do teste do segundo estágio for negativa, diz-se que o sistema está no estado não-estacionário. Observam que o desempenho do teste composto é caracterizado por duas quantidades: a probabilidade de ocorrer um erro do Tipo I e sua potência. Estudam o efeito dos seguintes fatores para avaliar o desempenho do teste composto: níveis de significância dos testes, a dimensão do conjunto de variáveis testadas simultaneamente, o número de medições que compõe cada média no tempo e a formulação das hipóteses.

Kretsovalis e Mah (1987a) estudaram o efeito da adição ou remoção de uma única medida na precisão das estimativas, no caso de processos no estado estacionário. A precisão dos dados reconciliados é medida pelo traço da sua matriz covariância dos erros estimados. Provam que a redundância jamais afetará, adversamente, a precisão da estimativa e que ela sempre a melhorará, quando as medições se relacionarem com as variáveis de processo de modo diferente ao das restrições. Este tratamento considera que todas as variáveis de processo são observáveis ou, equivalentemente, que podem ser determinadas de forma única e que os erros grosseiros foram eliminados, previamente.

Kretsovalis e Mah (1987b) desenvolveram algoritmos para classificar as variáveis e as medidas em redes de fluxo multicomponentes. Notam que esta classificação é importante para a operação de um processo, pois permite avaliar o impacto da falha de um instrumento, por exemplo. Além disso, também fornece uma base racional para o projeto de sistemas de monitoramento de processos, no que se refere ao posicionamento e definição de instrumentos de diferentes tipos, custos e precisão. Observam que, matematicamente falando, se for possível mudar o valor de uma variável sem violar as condições de conservação, então esta mudança é dita possível. Todavia, isto implica que os valores de certas variáveis relacionadas deverão variar de forma conveniente, para que as restrições permaneçam satisfeitas. Num dado processo, com um certo conjunto de pontos de medição, se for feita uma mudança possível numa variável, sem que ela seja detectada (observada) pelos instrumentos de medição, então a variável é dita não-observável. Por esta definição, uma variável medida é certamente observável, mas a não-medida pode ser observável ou não. Entretanto, se a medição de uma



variável medida for suprimida e, assim mesmo, ela permanecer observável, então ela é chamada redundante. Neste trabalho, o tratamento para redes de fluxo multicomponentes exclui balanços de energia e reações químicas, mas não impõe quaisquer restrições quanto ao número de medidas das frações molares das correntes. Para o desenvolvimento dos algoritmos, fizeram uso de conceitos teórico-gráficos, tratando em etapas, os vários gráficos derivados do gráfico do processo.

Kretsovalis e Mah (1988a) notam que a observabilidade de uma variável ou a redundância de uma medida são informações básicas para o projeto de sistemas de monitoramento do desempenho de processos. Neste trabalho, descrevem os princípios de classificação de um tratamento destinado a redes de processo generalizadas, incluindo balanços de massa totais e multicomponentes, balanços de energia, reações, trocas de calor e divisores de correntes. Os algoritmos elaborados exploram as propriedades teórico-gráficas da rede e a resolubilidade das equações restritivas para determinar a observabilidade. Em outro artigo (1988b), descrevem os dois algoritmos desenvolvidos para classificação, fazendo uma análise detalhada de cada um, bem como ilustram sua aplicação, através de exemplos.

Narasimhan e Mah (1988) descrevem a aplicação do método da razão de probabilidade condicional generalizada (GLR) para identificar erros grosseiros causados por desvios em instrumentos de medição e controladores, vazamentos no processo e falha dos controladores, em processos dinâmicos de malha fechada. Este método dá um esquema geral para identificar diferentes tipos de erros grosseiros, cujo efeito no processo pode ser modelado.

Pai e Fisher (1988) tratam do problema de reconciliação de dados sujeitos a restrições não-lineares. Observam que a solução de problemas gerais deste tipo exige, invariavelmente, algum tipo de método iterativo. Desenvolvem um procedimento iterativo, que utiliza a projeção matricial de Crowe et alii.(1983) e combina um esquema quase-Newton com um esquema Gauss-Newton para resolver problemas de reconciliação de dados não-lineares. Neste estudo, propõem uma atualização tipo Broyden para substituir os valores anteriores das derivadas, de modo que a convergência possa ser melhorada, sem calculá-las repetidamente. Esta atualização é feita, usando os resíduos da função no novo ponto, para gerar, a partir da matriz anterior, uma aproximação do Jacobiano.

Prasad e Kuester (1988) investigaram o uso de dois procedimentos, um seqüencial e o outro não-seqüencial, para ajuste de dados, visando sua implementação em microcomputador, aplicado a um processo de liquefação, no estado estacionário ou quase-estacionário. Como estimador seqüencial, utilizaram um filtro de Kalman no estado estacionário, que, conforme observam, representa, essencialmente, uma solução recorrente de problemas de mínimos quadrados lineares, quando os ruídos do processo e das medidas são considerados variáveis Gaussianas normais. Este estimador se mostrou adequado para a estimativa de vazões em linha, servindo também como um integrador de dados de processo. Os dados integrados, livres de erros sistemáticos, podem ser usados na técnica não-seqüencial. O problema de coaptação de dados foi resolvido, de forma não-seqüencial, utilizando a abordagem de balanços reduzidos e uma rotina de otimização condicionada não-linear para minimizar a função-objetivo dos mínimos quadrados ponderados e obter os dados ajustados, seguida de uma rotina de otimização geral, não-condicionada, não-linear. A abordagem não-seqüencial se mostrou mais adequada para implementação fora de linha e seus principais benefícios são o ajustamento das composições e o cálculo de parâmetros do modelo do processo, que satisfazem as restrições dos balanços materiais. Investigaram, ainda, a adequabilidade de diferentes esquemas de detecção de erros grosseiros, descritos na literatura, aplicados às duas metodologias.



Crowe (1989) tratou da questão da redundância e observabilidade de dados de processo na reconciliação, no estado estacionário. Tinha em vista que, antes ou durante este procedimento, pode-se querer saber quais as quantidades não-medidas que podem ser determinadas de forma única, a partir de valores, correspondentes aos medidos, consistentes com as restrições do processo e quais não podem. Quanto à redundância, observa que ela, em geral, leva a conflitos entre os dados e os balanços e, portanto, à necessidade de reconciliá-los, possibilitando verificar a confiabilidade de valores medidos. Apresenta a teoria e métodos diretos para classificar quantidades não-medidas, em observáveis e não-observáveis, e para identificar medidas redundantes. Estes métodos se baseiam nas matrizes que descrevem a estrutura da planta e em qualquer conjunto de valores consistente com as restrições. A teoria permite a colocação arbitrária de medições, reações químicas, divisores de fluxo e fluxos de energia puros. Nenhuma consideração sobre a estrutura das restrições é feita, além da que diz que elas são lineares nas quantidades extensivas não-medidas.

Holly e Crowe (1989) aplicaram o procedimento de reconciliação de dados a medidas de vazões mássicas de uma planta de extração, baseado na minimização da soma ponderada dos quadrados dos ajustes, descrito por Crowe et alii.(1983). Tratam de aspectos importantes como de testes para verificar a uniformidade dos dados e da questão de estimar variâncias e covariâncias. Utilizaram três estatísticas-teste para detectar erros grosseiros: um teste global, em que os desbalanços são testados coletivamente contra uma distribuição qui-quadrado, um teste de medida aplicado a cada ajuste individual e um teste de desbalanço nodal, sendo os dois últimos testados contra uma distribuição normal. Verificaram que a eliminação de qualquer uma de duas determinadas vazões remove o erro grosseiro aparente nos dados.

Serth et alii.(1993) relatam estudos para obter estimativas para a eficiência de estágio em um processo de separação por destilação "flash", num único estágio, envolvendo a detecção de erros grosseiros e a reconciliação de dados. Observam que, são evitados problemas relacionados com a falta de independência entre as eficiências dos componentes e dificuldades computacionais, associadas à introdução de quantidades no equilíbrio nas equações restritivas na reconciliação de dados, quando é usada a definição de eficiência dada pela eficiência de vaporização ou pela eficiência de Murphree modificada, podendo a última ser baseada nas frações molares ou nas vazões dos componentes. Como o caso de reconciliação é não-linear, citam um procedimento de regressão linear iterativo para resolvê-lo, ao qual é incorporado um algoritmo para eliminar erros sistemáticos. Nesse método, o problema de reconciliação é resolvido de forma recorrente. Em cada etapa, os resíduos são testados para detectar erros grosseiros e a medida correspondente ao resíduo mais significativo é removida do conjunto de valores medidos, até que nenhum resíduo seja estatisticamente significativo. Os desvios-padrão relativos usados para erros aleatórios, cuja covariância foi tomada como igual a zero, eram equivalentes a 0,4% para temperaturas e 2,5% para as vazões, frações molares e fluxo de calor. Concluem que, utilizando as definições de eficiência sugeridas, é possível obter estimativas efetivas para a eficiência e também a efetiva detecção de erros grosseiros. Neste estudo, a melhor detecção de erro foi obtida usando a eficiência de Murphree baseada nas frações molares.

Leibovici et alii.(1993) relatam duas aplicações da reconciliação de dados a processos de separação através de destilação "flash", sendo um deles em dois estágios. Ao tratar seus dados consideraram que todas as composições têm desvios-padrão iguais. Observaram que os valores reconciliados da separação em dois estágios, que tiveram mudanças significativas, estão mais próximos dos valores previstos pela equação de estado de Soave-Redlich-Kwong do que os valores medidos correspondentes.

Phillips e Harrison (1993) aplicaram a reconciliação de dados e um método chamado Teste de Medidas Iterativo Modificado, para identificação de erros grosseiros, a dados de simulação e a dados experimentais não-lineares obtidos em reator em escala piloto, onde ocorriam duas reações simultâneas. Os valores medidos e reconciliados foram usados para determinar as velocidades das reações usando regressão linear. Nos dados da simulação, foram introduzidos erros grosseiros para testar o desempenho do método de detecção de erros grosseiros citado, que foi aplicado aos dados experimentais. Observaram que as melhores velocidades para ambas as reações foram obtidas quando o conjunto de dados identificado como contendo erros grosseiros foi eliminado e quando a regressão foi feita usando dados reconciliados a partir do conjunto de dados remanescentes, que não continha erros grosseiros.

Serth e Heenan (1986) apresentam algoritmos para detectar erros grosseiros em dados de processo, sujeitos a restrições lineares, aplicando-os a um sistema de medição de vapor, por meio de simulação. Foram usados o teste de medida (MT), o teste dos pseudo-nós (MP), anteriormente desenvolvidos, e várias modificações deles, além de um método novo (SC). Concluíram que os algoritmos SC e MIMT (teste de medidas iterativo modificado) se constituem em métodos efetivos e confiáveis para detectar erros e reconciliar dados de sistemas de medição de vapor. Ambos possibilitaram a detecção de aproximadamente 80% dos erros grosseiros nos dados e propiciaram reduções médias nos erros totais superiores a 60%. Observaram, entretanto, que o MIMT é computacionalmente mais eficiente, exigindo menos tempo de computação, em média, do que o SC. Verificaram que com o algoritmo MT foi obtido um desempenho fraco, por razões parcialmente relacionadas com a estrutura do sistema e com a faixa dos valores empregados. Observam que o sucesso do método SC sugere que uma abordagem útil para detectar erros grosseiros pode ser através de uma combinação de algoritmos diferentes de modo a explorar a força de cada um.

Madron (1985b) comparou métodos para identificar fontes de erros grosseiros aplicáveis a balanços de massa de um único componente, para verificar sua eficácia e o volume de trabalho envolvido, considerando a ocorrência de um único erro grosseiro. Utilizou os três métodos mais freqüentemente empregados para isto: análise dos ajustes, análise dos resíduos das equações do modelo matemático, que se baseiam na distribuição normal padronizada, e a eliminação sucessiva de quantidades medidas, que se baseia na distribuição qui-quadrado. Concluiu, de acordo com o caso analisado e com base em sua experiência ao solucionar problemas reais, que os métodos propostos são úteis na detecção de medidas sujeitas a erros grosseiros. Todavia, observa que devemos ter em mente que, só raramente, estes métodos são capazes de identificar de forma definitiva a fonte de erro grosseiro. Na maioria dos casos, eles permitem, somente, que seja delineado um conjunto de prováveis fontes de erro grosseiro e a determinação da fonte de erro se torna um problema de análise do próprio processo de medição. Entre os métodos disponíveis, recomenda a análise dos ajustes, cujos resultados podem ser refinados pelo método da eliminação sucessiva das quantidades medidas.

Serth et alii.(1987) estenderam o uso do teste de medida iterativo modificado (MIMT) desenvolvido para sistemas sujeitos a restrições lineares ao caso de sistemas não-lineares. Testaram o algoritmo desenvolvido através de simulação, usando os dados de um circuito de moagem industrial. Foram detectados, corretamente, aproximadamente 80% dos erros sistemáticos dos dados, obtendo-se uma redução média no erro total de mais do que 60%. Foram estudados oito casos, contemplando-se o efeito da correlação entre os erros em dois deles. Os resultados indicaram que o algoritmo é robusto com respeito às correlações entre os erros e que o seu desempenho global foi comparável ao obtido previamente com um sistema linear quase do mesmo tamanho.

Mathiesen (1974) estudou o ajustamento de um conjunto de  $m$  medidas redundantes de modo a se adequarem a um conjunto de  $p$  equações e  $q$  inequações como um problema de programação linear. Usou a soma dos valores absolutos dos ajustes ponderados como a função-objetivo a ser minimizada. O método de ajustamento linear se baseia na idéia de que os erros grandes e pequenos, dentro de uma certa faixa, têm a mesma probabilidade, isto é, são erros distribuídos retangularmente. O resultado é que  $m$  menos  $p$  medidas são aceitas sem qualquer ajuste, enquanto as  $p$  medidas remanescentes são ajustadas para satisfazer o sistema de equações, ao contrário do método quadrático, onde são feitos ajustes em todas as medidas. Observa que o método linear se constitui de um método mais grosseiro porém mais rápido e, com respeito a erros grandes, mais estável do que o quadrático, permitindo também a inclusão de inequações de modo fácil. Na sua opinião, o método linear será mais útil na prática do que o quadrático, embora o último seja o mais mencionado na literatura.

Liebman et alii.(1992) propuseram o uso de um novo método de reconciliação de dados dinâmicos não-lineares, usando programação não-linear. Argumentam que ele se constitui de um meio mais robusto para tratar sistemas de engenharia química, que operam dinamicamente, em regiões altamente não-lineares, onde o filtro de Kalman ou o filtro de Kalman estendido, tipicamente usados na reconciliação de dados de processo governados por equações dinâmicas, podem ser imprecisos. Além disto, o método proposto pode ser aplicado a problemas que envolvem restrições dadas por inequações e para aqueles cuja função-objetivo não é dada pelos mínimos quadrados ponderados. Observam que, através do uso de técnicas de solução e otimização simultâneas aprimoradas, seu algoritmo fornece um esquema geral para obtenção de estimativas de parâmetros e de estado, e que considera, ainda, a questão de medidas tendenciosas. Enfim, demonstram o uso desse método e suas extensões ao caso de um reator.

Tjoa e Biegler (1991) observam que, na reconciliação de dados de processo, os dados são ajustados para satisfazer as restrições do processo, minimizando o erro no sentido dos mínimos quadrados. Neste estudo, em lugar disto, minimizam uma função-objetivo que é construída usando o princípio da probabilidade máxima, para construir uma nova distribuição, que considera as contribuições devidas a erros aleatórios e grosseiros, também chamada de distribuição Gaussiana contaminada. Observam que a vantagem da minimização desta função é que ela dá estimativas não-tendenciosas na presença de erros grosseiros e que, simultaneamente, pode ser construído um teste para detectar estes erros, baseado na função distribuição, sem ser necessário supor que as restrições sejam lineares. Desenvolveram um método de programação quadrática sucessiva (SQP) híbrido e demonstram a praticidade da estratégia empregada através de dois exemplos. Observam que os resultados mostraram que o método dá estimativas não-tendenciosas e é efetivo na identificação de erros grosseiros.

Verneuil et alii.(1992) enfocam a questão da eliminação de erros grosseiros, dividindo-a em três etapas: 1)a detecção da presença de erros grosseiros, 2)a identificação dos erros grosseiros, buscando as prováveis fontes de erros e 3)a sua eliminação pela remoção apropriada de dados corrompidos por este tipo de erro. Consideram a análise dos resíduos e a dos ajustes individuais, ferramentas poderosas para detectar erros grosseiros, porém cuja eficiência não é ilimitada. Observam que a eficiência da detecção de erros grosseiros pode ser expressa pela potência do teste (que é a probabilidade de detectar um erro grosseiro), mas sugerem o uso dos assim chamados valores limiares. Valor limiar é um valor tal que há 90% de probabilidade de que o erro grosseiro seja detectado. Para a identificação dos erros grosseiros sugerem que algumas quantidades sejam excluídas, a priori, da pesquisa de erros, como aquelas cujos valores limiares são extremamente altos e aquelas que não são ajustáveis. Observam que, do ponto de vista de identificação de erros grosseiros, somente variáveis

medidas diretamente devem ocorrer no modelo matemático. Recomendam o uso dos seguintes métodos para eliminação de erros grosseiros: análise dos resíduos das equações, análise dos ajustes padronizados e eliminação de quantidades medidas. Sugerem o uso de informações preliminares, como intervalos dentro dos quais todos os valores podem cair, para discriminar entre dois ou mais valores suspeitos de conter erros grosseiros e descrevem o método Credibilidade de Identificação. Argumentam que estes métodos podem ser usados quando houver mais de um erro grosseiro e que o método de eliminação sucessiva pode ser empregado quando os ajustes padronizados diferirem significativamente, em valores absolutos. Elimina-se o maior valor e se for detectada a presença de erro grosseiro, o procedimento é repetido, terminando quando não for detectado erro ou se não houver quantidades redundantes remanescentes. Todavia, observam que os métodos de identificação de erros grosseiros raramente fornecem resultados não-duvidosos, especialmente quando há mais de um deles. Observam, ainda, que a inconsistência dos dados pode ser causada por modelos matemáticos inadequados e que os modelos de balanço podem ser invalidados pela ocorrência de vazamentos de material, de desvios do estado estacionário, pela inadequação da estequiometria e perdas de calor.

Stanley e Mah (1977) mostram como podem ser estimadas temperaturas e vazões de uma rede de processo, a partir de um modelo quase-estacionário e de um filtro de Kalman discreto. Observam que os dados necessários para aplicações deste tipo se encontram disponíveis em muitas unidades em operação e que as necessidades computacionais estão dentro das capacidades dos computadores de processo. Usando dados coletados durante um período de dois meses, desenvolveram um modelo no estado quase-estacionário para a rede de processo, o qual levou à aplicação de um filtro de Kalman discreto. Verificaram que, ajustando os pesos relativos dos ruídos de processo e covariâncias dos ruídos das medidas, o filtro podia ser ajustado para dar uma redução significativa nos erros (de até 70%) e responder rápida e estavelmente a mudanças bruscas nas condições de processo. Observaram que a redundância das medidas contribui significativamente para a redução global dos erros e que experimentos posteriores mostraram que o filtro funcionou de forma razoavelmente boa com desvios isolados. Entretanto, notam que a inclusão indiscriminada de desvios e de parâmetros, como de variáveis a serem estimadas não é recomendável.

Crowe (1992) derivou um teste de potência máxima (MP), baseado na distribuição normal, para detectar erros grosseiros no balanço das espécies, em torno de unidades de processo individuais, aplicável às restrições originais e ao caso bilinear. Observa que o teste, anteriormente desenvolvido, era limitado pois se aplicava às restrições remanescentes após a eliminação das vazões não-medidas e ao caso de restrições lineares. Mostra que o quadrado da estatística-teste MP é precisamente igual à redução na soma ponderada dos quadrados dos ajustes, que resulta da remoção daquela restrição. Ilustra com dois exemplos.

Ramamurthi et alii.(1993), objetivando mostrar a importância da reconciliação de dados dinâmicos para melhorar o desempenho de uma malha fechada, propõem o uso de uma estratégia de estimativa baseada no horizonte linearizado sucessivamente (SLHE) para estimar parâmetros e estados. Adotaram uma estratégia de dois níveis para a estimativa de entradas e saídas do processo, as quais são corrompidas por erros de medidas. A estimativa das entradas é desacoplada da estimativa das saídas e dos parâmetros, através de uma estratégia de dois níveis, resultando na redução significativa do tempo de computação. Realizaram estudos de simulação em malha aberta e fechada, em um reator contínuo agitado, para demonstrar a efetividade da estratégia proposta. Observam que foram obtidos resultados satisfatórios para estimativa de estado e parâmetros, em pontos de operação da malha aberta estável e instável e numa região que exibia comportamento de ciclo limite. Demonstram, ainda, o desempenho de

um controlador preditivo não-linear, usando as estimativas obtidas pelo procedimento SLHE. Notaram que o desempenho do sistema de controle, usando esta abordagem, é virtualmente idêntico a um método baseado em programação não-linear, que usa o modelo não-linear completo.

Stephenson e Shewchuk (1986) abordam as possibilidades de uso do programa de simulação MASSBALL para reconciliar dados de processo. Este programa está baseado na abordagem orientada para equações, num ambiente modular. É construído um modelo selecionado, interligando blocos-padrão modulares, referentes às operações fundamentais (como mistura e separação de correntes, adição ou consumo de calor e reações), que estabelecem as equações, mas não as resolvem. No caso de reconciliação de dados, o programa permite a entrada de dados medidos redundantes e inconsistentes, diretamente, num modelo sub-restringido. Observam que a única exigência é que o subconjunto de valores medidos deve ser suficiente para satisfazer os graus de liberdade não-restringidos. As medidas são reconciliadas minimizando a soma ponderada dos quadrados das diferenças normalizadas entre valores medidos e calculados, que satisfazem equações restritivas do modelo e outras restrições estabelecidas.

Rollins e Roelfs (1992) relatam seus estudos para estender procedimento de obtenção de estimadores não-tendenciosos para os valores de variáveis de processo, quando há desvios nas medidas e vazamentos no processo, ao caso em que as restrições são bilineares. Descrevem três maneiras de testar hipóteses, quando as restrições são bilineares, sendo que duas consideram a aproximação à distribuição estatística verdadeira e usam a teoria normal, podendo ser usadas de imediato. Na terceira, propõem um teste de hipóteses, usando a distribuição exata, todavia esta distribuição ainda não foi desenvolvida. Verificaram que cada técnica tem méritos e limitações próprias, porém, somente a abordagem em dois estágios e a linearização podem ser usadas de imediato, demandando estudos para melhor compreender as aplicações mais adequadas. Entretanto, a abordagem que emprega a distribuição exata exigirá o desenvolvimento da distribuição e de testes estatísticos.

Darouach e Zasadzinski (1991), estudando uma solução para o problema de balanços materiais lineares dinâmicos, desenvolveram um algoritmo baseado no modelo singular ou modelo dinâmico linear generalizado, para os quais a representação espacial do estado-padrão e o filtro de Kalman não podem ser usados. O algoritmo é baseado no método desenvolvido para o estado estacionário e conduz a um esquema recorrente, que consideram muito útil no processamento em tempo real. Ele reduz problemas computacionais tais como singularidades e erros de arredondamento, que podem ocorrer em sistemas complexos. Observam que a convergência deste método foi provada no caso da reconciliação de dados dinâmicos.

Keller et alii.(1992) propõem um algoritmo analítico para estimar a matriz covariância dos erros de medida, que é empregada na reconciliação de dados de processo. Ele se baseia nos resíduos de restrições lineares, calculados a partir dos dados disponíveis. Observam que uma exigência indispensável para sua aplicação é redundância espacial suficiente e que este algoritmo pode ser aplicado a medidas correlacionadas, se todos os sensores sujeitos a tais medidas forem conhecidos. Apresentam dois exemplos.

Rollins e Davis (1992) relatam uma abordagem para detectar erros grosseiros que fornece estimativas não-tendenciosas para variáveis de processo, quando existe medidas tendenciosas e/ou vazamentos. Ela é aplicável a sistemas descritos pelo seguinte modelo: os erros são normalmente distribuídos, o estado é estacionário e as restrições são lineares. Apresentam as equações para obter as estimativas das variáveis de processo, bem como as

equações para identificar medidas tendenciosas e vazamentos. Incluem-se, entre elas, a função potência para um teste global e dois testes dos componentes de nível- $\alpha$  e suas funções de potência. Comparam esta abordagem com uma estratégia de compensação em série, variando o nível de significância, a matriz variância-covariância, o tamanho dos desvios das medidas, o número de variáveis tendenciosas e o tamanho da amostra. Avaliam e comparam a precisão das estimativas dos desvios e o desempenho em detectar a presença de vazamentos no processo.

MacDonald e Howat (1988) estudaram a extensão de técnicas de reconciliação de dados sujeitos a restrições não-lineares para estimar parâmetros de processo, aplicando-as a uma operação de destilação "flash", em um único estágio. Empregaram dois procedimentos, um deles chamado de sequencial ou desacoplado e o outro chamado acoplado. No primeiro, os dados são reconciliados para satisfazer os balanços de massa e energia e após são obtidas estimativas para os parâmetros, usando a estimativa de probabilidade máxima. No segundo, faz-se a reconciliação de dados para satisfazer as restrições e obter as estimativas para os parâmetros (neste caso, as eficiências), simultaneamente. Observam que o primeiro é mais rápido computacionalmente e mais facilmente adaptável aos algoritmos de reconciliação existentes, mas não é estatisticamente rigoroso, enquanto o segundo é. Concluíram que ambos têm sucesso ao reduzir significativamente os erros de medida das variáveis do processo. Verificaram que os dados ajustados podem ser tidos como mais precisos do que os valores medidos, com exceção das pressões de alimentação e de operação, que não são ajustadas nos dois casos. Observaram que o procedimento acoplado ajusta melhor as variáveis de processo do que o outro e que ambos resultam em estimativas mais exatas e precisas para as eficiências do que se os dados não fossem ajustados. Todavia, na maioria dos casos não há uma diferença significativa nas eficiências calculadas pelos dois procedimentos. Verificaram que o procedimento acoplado exige, substancialmente, maior tempo de computação, embora ele forneça somente melhorias marginais em relação ao procedimento desacoplado, com exceção da predição do desvio-padrão da eficiência.

Sanchez et alii.(1992) relatam possibilidades de uso do pacote computacional PLADAT, que permite a classificação de variáveis operacionais de uma planta química completa. Os resultados da classificação são usados para reduzir o tamanho do problema de reconciliação, que se baseia em uma técnica de programação não-linear (programação quadrática sucessiva). O algoritmo adotado para a classificação é uma variação de um procedimento desenvolvido anteriormente, que se baseia na determinação de um conjunto de saída para as equações de balanço de massa e de energia. As medidas com erros grosseiros são previamente localizadas e eliminadas, utilizando métodos estatísticos. Discutem, ainda a aplicação deste pacote para definir a colocação de novos instrumentos de medição no sistema de instrumentação de uma seção de uma planta de etileno existente, que tem 150 correntes e 45 unidades, de modo que, com o menor número de instrumentos, seja possível o completo conhecimento de todas as correntes do processo.

Aldrich e Devender (1993) aconselham o uso de redes neuronais para detectar erros sistemáticos em sistemas de processo, obtendo-se uma relação interna das relações entre as distribuições dos resíduos das medidas e dos resíduos das restrições do processo, através de redes neuronais de retro-propagação. Observam que a maior vantagem das redes neuronais é que elas podem ser usadas para detectar erros sistemáticos em sistemas sujeitos a restrições não-lineares, ao contrário dos métodos estatísticos convencionais. Demostram o seu uso, através de dois exemplos: a localização de erros sistemáticos múltiplos num circuito de flotação industrial e em dados medidos de um circuito de hidrociclones de três estágios, sujeito a restrições de processo não-lineares. Observam que os exemplos, discutidos neste trabalho, indicam que as redes neuronais de retro-propagação de camada única fornecem uma alternativa

atrativa aos processos estatísticos tradicionais, para a detecção de erros sistemáticos medidos sujeitos a restrições não-lineares, e que elas são um opção atrativa para a detecção, em linha, desses erros.

Terry e Himmelblau (1993) observam que um dos problemas que engenheiros se deparam é a identificação e eliminação de erros grosseiros de dados medidos e sua retificação, para que satisfaçam restrições de processo tais como os balanços de massa e de energia, que os descrevem. Acreditam que a chave para uma retificação de dados válida é o emprego de um bom modelo e que embora os balanços de massa e energia sejam oriundos de leis fundamentais, freqüentemente, são feitas tantas simplificações no seu desenvolvimento, que não há nada melhor do que modelos estritamente empíricos para representar um processo. Sugerem o uso de redes neuronais artificiais, como uma forma de construção de modelos não-paramétricos. Notam que redes neuronais compostas de funções básicas produzem excelentes modelos e podem ser usadas para retificar dados. Demostram a sua aplicação através de dados simulados de um trocador de calor, no estado estacionário, cujos erros aleatórios e grosseiros foram retificados com sucesso. Ao comparar os resultados obtidos com a retificação de dados por mínimos quadrados condicionados via programação não-linear, concluem que o tratamento usando a rede neuronal artificial se mostrou superior.

---

## **CAPÍTULO 3. RECONCILIAÇÃO DE DADOS DE PROCESSO**



### 3.1 Introdução

Como os dados medidos de parâmetros de correntes de processo são sujeitos a erros de medida (excluindo-se, aqui, os erros sistemáticos) é necessário ajustá-los, de forma que as restrições do processo sejam satisfeitas, de acordo com um critério estabelecido. O problema básico de análise de dados é estimar variáveis de processo, a partir de valores medidos, e ao fazer isto minimizar o erro total.

Numa planta química, há equipamentos e correntes de processo e há variáveis de processo associadas (vazão, entalpia, por exemplo) a cada corrente. Ao analisarem-se as vazões totais de cada corrente  $j$  ( $j = 1, 2, \dots, m$ ), como variáveis de processo e representando os valores medidos por  $\tilde{x}_j$ , os valores verdadeiros por  $x_j$  e os valores estimados por  $\hat{x}_j$ , definem-se os erros de medida ( $e_j$ ) e os ajustes ( $a_j$ ) como:

$$e_j = \tilde{x}_j - x_j$$

$$a_j = \hat{x}_j - \tilde{x}_j$$

O valor verdadeiro de uma variável é aquele que seria obtido na medição se não houvesse característica estocástica a ela associada. Consequentemente, o valor real de uma variável de processo é, por um lado, um valor hipotético, postulado como existente. Através do processo de reconciliação, o que se obtém são estimativas dos parâmetros das correntes, obtidas a partir dos valores medidos, que devem obedecer a certas restrições. O modo como as estimativas são obtidas e quais os ajustes feitos dependem de suposições sobre a estatística dos erros de medida.

As atividades de ajuste de dados de processo podem ser divididas em várias etapas. A primeira consiste na obtenção de informações, como a relação dos valores medidos e suas variâncias ou covariâncias e os coeficientes estequiométricos das reações químicas. É também necessário conhecer o fluxograma do processo, no qual devem constar todas as correntes, pois fornece informações sobre a estrutura do processo. Além disso, de acordo com o problema de ajuste, podem ser incluídas outras restrições. A segunda parte diz respeito à simplificação do sistema de equações a minimizar, fazendo-se presente quando houver quantidades não-medidas. A terceira parte é aquela em que se realiza o ajuste de dados, sendo chamada de reconciliação de dados de processo, quando se trata de ajustar valores de medidas redundantes, corrompidas por erros aleatórios. Em geral, como nem todas as variáveis de um processo são medidas, a reconciliação é seguida da assim chamada coaptação, que trata a questão de estimar valores de variáveis não-medidas, a partir dos valores estimados (ou corrigidos) de variáveis medidas.

Quando houver suspeita de que há medidas com erro sistemático, deve ser considerada a sua detecção e remoção, pois sua presença invalida a base estatística do procedimento de ajuste de dados. Este assunto é abordado no Capítulo 4, sobre retificação de dados de processo.

Neste item, trata-se do ajuste de dados de processo, cujos valores medidos contêm somente erros aleatórios e que são sujeitos a restrições lineares, no estado estacionário.

A situação mais simples de ajuste de dados de processo é aquela em que se quer obter a melhor estimativa de variáveis medidas, cujos valores são aparentemente conflitantes, sendo

chamada de reconciliação de dados. Porém, um caso bastante comum é aquele em que se quer obter estimativas de variáveis não-medidas, a partir dos valores corrigidos de variáveis medidas, o que é chamado de coaptação. Neste caso, é feito o ajuste de dados medidos e após são obtidas as estimativas das quantidades não-medidas, desde que elas sejam observáveis.

Este trabalho considera a possibilidade de obter estimativas para quantidades não-medidas, além de estimativas para as quantidades medidas.

### 3.2. Obtenção de Informações Básicas

São aqui chamadas de informações básicas, os valores das variáveis medidas, a matriz variância-covariância e a estrutura do processo correspondentes. Esta última permite obter as restrições do processo, que dizem respeito ao arranjo das correntes. Todavia, em determinados casos podem ser necessárias restrições adicionais, como relações de equilíbrio termodinâmico, reações químicas e seus coeficientes estequiométricos, por exemplo.

#### 3.2.1. Valores das Variáveis Medidas

Como a análise de dados é estatística por natureza e sabendo que a redundância nas medidas melhora a precisão das estimativas, devemos identificar todos os pontos em questão, assinalando quais os parâmetros medidos. No caso de balanços totais de massa, consideram-se as vazões das correntes e no caso de balanços multicomponentes as vazões e a composição das correntes ou as vazões totais e dos componentes.

Os parâmetros podem ser medidos de forma direta ou indireta. De acordo com Mickley et alii.(1957), medidas diretas são aquelas em que a quantidade medida é obtida por observação direta do instrumento de medição. No caso de medida indireta, o valor de uma quantidade é determinado por cálculo, a partir das magnitudes de outras quantidades diretamente medidas, através de um relação funcional existente entre elas. Portanto, no caso de medidas indiretas, é necessário ter os dados (o valor no instante considerado e sua variância) das quantidades primárias, diretamente medidas, bem como conhecer a função que as relaciona, que são utilizadas na obtenção da matriz covariância.

#### 3.2.2. A Matriz Variância-Covariância

Ao reconciliar dados de processo, procura-se tratar os dados de forma que o erro total seja mínimo. Visto que as medidas podem ter graus de precisão diferentes, é conveniente tomá-los em consideração ao minimizar o erro total, pois se isto não for feito as medidas menos precisas terão uma participação maior no erro total. Como a variância mede a dispersão em torno da média, é comum adotar-se o seu inverso como fator ponderante dos quadrados dos ajustes individuais.

Para obter uma estimativa da variância de uma variável diretamente medida é preciso ter um conjunto de medidas dessa quantidade. No caso de uma variável medida indiretamente é necessário conhecer sua relação com as variáveis primárias, além de um conjunto de medidas destas últimas.

Madron et alii.(1977), ao fazerem uma análise estatística de um reator químico, descrevem o modo de cálculo da matriz covariância. Observam que, usualmente, o número de moles de uma dada espécie, gerada por reações químicas, não é medido diretamente, sendo obtido por cálculo, através de variáveis primárias, que são diretamente medidas. Tratam de expressar a matriz covariância, através das variâncias das variáveis primárias medidas, considerando que elas são sujeitas a erros que são variáveis aleatórias, com média zero e com uma distribuição normal, sendo estatisticamente independentes. Consideram duas formas de expressar as variáveis indiretamente medidas, para obter as expressões dos erros destas variáveis, por meio dos erros das variáveis primárias. Citam o método sugerido por Box para o cálculo aproximado da matriz covariância. Finalmente observam que, na maioria dos casos, nem as variâncias das variáveis primárias, nem seus valores verdadeiros são conhecidos. Portanto, eles são substituídos por estimativas desses valores, por exemplo, pela variância da amostra, em lugar da variância, e pelos valores medidos ao invés dos valores verdadeiros das variáveis primárias.

Almásy e Mah (1984) tratam do problema de obter uma estimativa para a matriz covariância dos erros de medida, observando que ela pode ser obtida através de método direto ou de método indireto. Comentam que a matriz obtida pelo primeiro depende de variações nos valores esperados das variáveis medidas e que, em vista disto, as estimativas serão ruins, quando essas variações forem comparáveis ou excederem, em magnitude, os erros de medida. Propõem o uso de um método indireto, que se baseia no conteúdo de informações inerentes às restrições de conservação usadas na reconciliação de dados de processo. A matriz covariância dos erros de medida é obtida a partir da matriz covariância dos resíduos das equações de balanço da amostra, cujo valor esperado é zero, se não houver desvios sistemáticos nas medidas. Os resíduos são calculados diretamente a partir dos valores medidos, sem serem necessárias estimativas para os valores verdadeiros das variáveis medidas. A matriz covariância dos erros ( $Q$ ) é estimada pela minimização condicionada da soma dos quadrados dos elementos de fora da diagonal de  $Q$ , uma vez que consideram que os erros de medida não são correlacionados, ou são pouco correlacionados, de forma que  $Q$  é diagonal, ou diagonalmente dominante. Notam, de acordo com resultados de estudos de simulação, que, no limite, quando o tamanho da amostra tende a infinito, a estimativa da covariância se aproxima de seu valor real. Observam ainda que: 1) o método indireto funciona satisfatoriamente quando as variâncias têm magnitudes semelhantes e que seu desempenho se deteriora à medida que sua faixa de variação se amplia; 2) o método indireto é moderadamente sensível à correlação dos erros de medida, tendo um desempenho aceitável a níveis de coeficiente de correlação de 0,1, não sendo recomendável para níveis superiores e 3) em todos os experimentos, os erros de estimativa relativos são reduzidos, substancialmente, quando o tamanho da amostra aumenta. Concluem que se o processo estiver realmente no estado estacionário, o método direto é adequado, mas no caso de variações consideráveis dos valores verdadeiros das variáveis medidas, durante o período de obtenção de dados, deve ser dada preferência ao método indireto, que funcionará melhor quanto maior for a redundância espacial das medidas. Notam que tanto um método quanto o outro só são válidos quando não há erros grosseiros nos dados. Quando houver este tipo de erro, a calibração direta dos instrumentos contra valores de referência deverá ser preferida.

Para o cálculo da matriz variância-covariância, nos procedimentos computacionais, utilizou-se o método descrito no Apêndice B2, de acordo com Madron et alii.(1977).

### 3.2.3. A Estrutura do Processo

A estrutura do processo estabelece a interdependência entre as vazões e dados do inventário do processo, que é expressa matematicamente através de balanços de massa, no caso de ajuste de vazões mássicas.

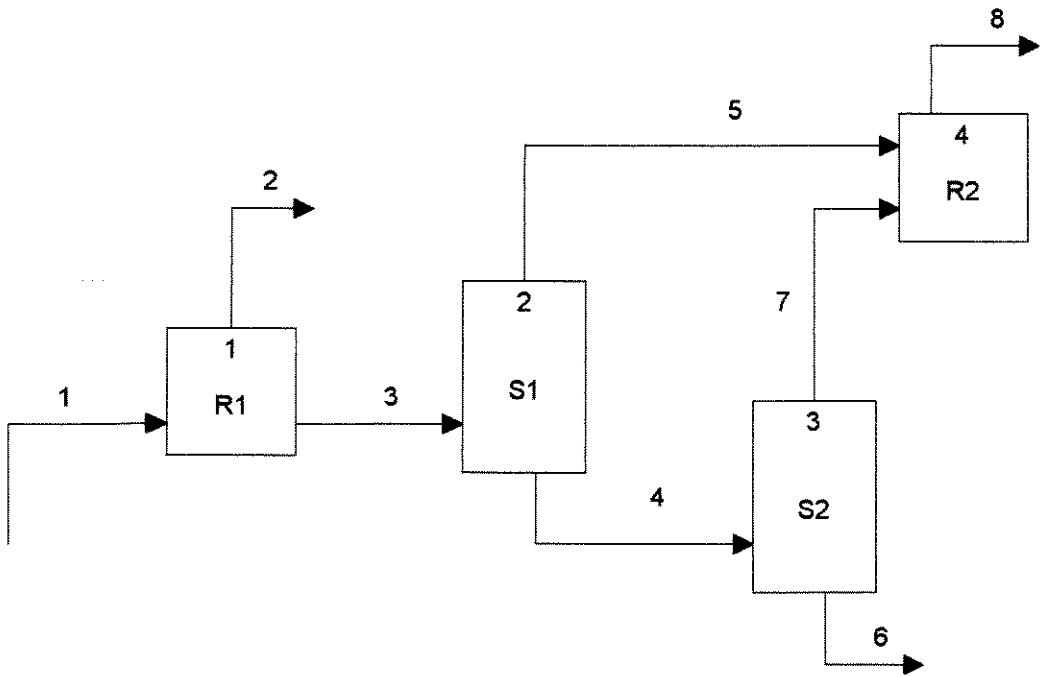
A estrutura de processos químicos pode ser representada por meio de gráfico orientado, chamado de gráfico de processo, formado por arcos dirigidos e por nós. Os arcos correspondem geralmente às correntes e seu sentido é igual ao das correntes que simbolizam. Os nós, em geral, representam as unidades, tanques e junções do fluxograma do processo. O gráfico do processo pode conter somente um subconjunto de nós, quando, num dado instante, somente uma parte do processo estiver sendo considerada. Ele sempre contém um nó meio-ambiente, representado por E, que pode ser visto como um complemento do processo, de onde partem as correntes de alimentação (incluídas as utilidades) e para onde se dirigem as correntes de produtos. Na Figura 3.1.a, mostra-se o fluxograma de um processo simples, constituído de dois reatores (R1 e R2) e de duas colunas de destilação (S1 e S2), e, na Figura 3.1.b, o gráfico de processo correspondente.

Ao gráfico de processo podem ser aplicadas todas as representações algébricas de um gráfico dirigido. Faz-se uso frequente da matriz incidência A para representar a estrutura de um processo, cujas linhas correspondem aos nós (exclue-se o nó meio-ambiente) e cujas colunas correspondem aos arcos do gráfico de processo. Considere-se uma planta química que tenha  $n$  ( $i=1,2,\dots,n$ ) unidades de processo ou nós e  $m$  ( $j=1,2,\dots,m$ ) correntes. Então, a cada elemento ( $a_{ij}$ ) corresponderá o número 1, se a corrente  $j$  entrar no nó  $i$ ; o número -1, se a corrente  $j$  sair do nó  $i$  e o número 0, se a corrente  $j$  não estiver ligada ao nó  $i$ . A matriz incidência (A) do processo da Figura 3.1 é

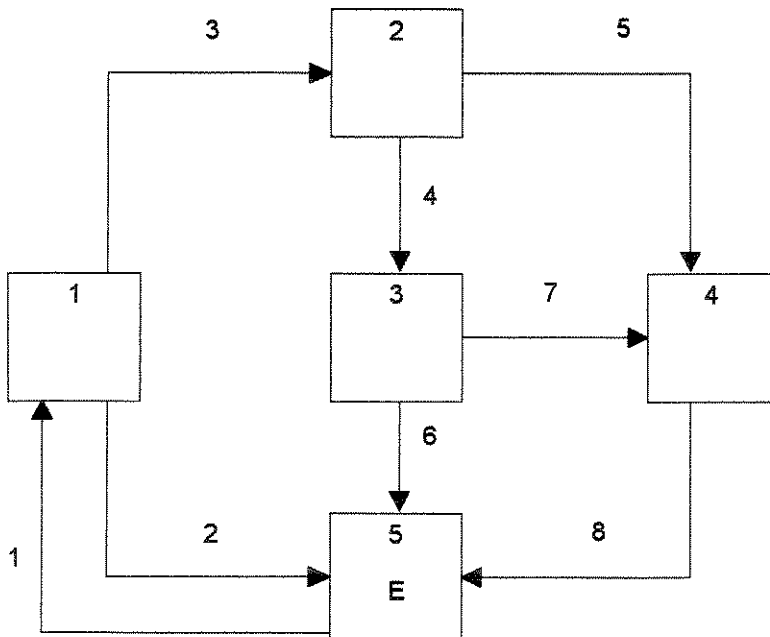
$$A = \begin{array}{c} \text{corrente} \\ \text{nó} \end{array} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{bmatrix} \end{array}$$

No caso de  $n+1$  nós conectados, somente  $n$  linhas da matriz incidência são linearmente independentes. Esta propriedade vem do fato de que há, fisicamente,  $n$  balanços nodais independentes, posto que ao nó  $n+1$  corresponde o balanço total da rede. De fato, quaisquer  $n$  balanços nodais podem ser escolhidos para formar um conjunto de linhas de A independentes e, usualmente, é deixado de fora o nó meio-ambiente. Então, nesta notação, A é uma matriz  $n \times m$ , com característica  $n$ . A matriz incidência permite a representação de arcos múltiplos entre cada par de nós, sendo usada, na sua forma mais simples, para expressar balanços totais de massa, podendo também representar balanços multicomponentes. Neste caso, cada elemento é um bloco, que contém C linhas que correspondem a um equipamento e C colunas que correspondem a uma corrente, sendo C igual ao número de componentes da dita corrente. A

inequação  $m > n$  é sempre válida, pois não tem sentido uma situação em que o número de arcos é menor do que o de nós, pois haveria equipamentos completamente desconectados.



a) fluxograma.



b) gráfico do processo.

Figura 3.1 - Fluxograma e Gráfico de um Processo Envolvendo Dois Reatores e Duas Colunas de Destilação.

### 3.3. Procedimentos Para Reconciliação de Dados de Processo

Na reconciliação de dados, busca-se a obtenção da melhor estimativa para valores de variáveis medidas, de forma que as restrições do processo sejam satisfeitas. O caso mais simples é aquele referente a vazões totais, onde as restrições são dadas pelas equações de conservação de massa relativas a cada equipamento, tanque ou junção do processo.

Para que se possa efetuar a reconciliação de dados é necessário que haja redundância. Diz-se que uma medida é redundante quando for possível obter uma estimativa de seu valor, a partir de dados medidos de outras variáveis do sistema, uma vez que a sua estrutura seja conhecida.

De acordo com Kumar (1981), a redundância das medidas aumenta a exatidão das estimativas, posto que a análise de dados é estatística por natureza. Na reconciliação de dados de vazão mássica, em que todas as vazões são medidas, tem-se a redundância máxima e o sistema é dito super-determinado.

O modo como as estimativas são obtidas e quais os ajustes feitos dependem de suposições sobre a estatística dos erros de medida.

Em geral, parte-se do pressuposto que todas as medidas são sujeitas somente a erros de medida aleatórios, que seguem, portanto, uma distribuição normal com média zero. Considera-se que têm sua matriz covariância conhecida, positiva por definição. Por exemplo, no caso de reconciliação de vazões totais, considera-se que todas as medidas são estatisticamente independentes e normalmente distribuídas. Consequentemente, a cada erro individual está associado um desvio-padrão (ou variância) e a média é zero. Quando cada instrumento funciona independentemente dos outros, o conjunto de medidas geradas corresponde à situação de teste multiamostrais.

As estimativas das variáveis medidas são obtidas, frequentemente, de forma que o valor da função que expressa a soma ponderada dos quadrados dos ajustes individuais seja mínimo, através do método dos mínimos quadrados.

De acordo com Crowe et alii.(1983), o problema de reconciliação foi considerado pela primeira vez por Kuehn e Davidson (1961), que o resolveram buscando os ajustes ótimos através dos mínimos quadrados, usando os multiplicadores de Lagrange.

Almásy e Sztanó (1975) observam que se os erros de medida tiverem uma distribuição Gaussiana, com média zero, então o vetor das estimativas, obtido através do método dos mínimos quadrados dos ajustes, é também a estimativa de probabilidade máxima e ainda a estimativa não-tendenciosa de variância mínima.

Václavek (1969b) descreve o uso do método dos mínimos quadrados para ajustar dados de vazão total de processo. Observa que, como o número de equações ( $n$ ), dado pelos balanços nodais, é menor do que o número de incógnitas ( $m$ ), dado pelo número de correntes, não há solução única no caso geral. Portanto, devem ser acrescentadas condições que as estimativas dos parâmetros ( $\hat{x}_j$ ) devem satisfazer. Se os dados são tratados pelos mínimos quadrados, tais condições são dadas pelo valor mínimo da função-objetivo ( $f$ ), dada por

$$f = \mathbf{a}_t^2 = \sum_{j=1}^m p_j a_j^2 \quad \text{para } j = 1, \dots, m$$

que é a soma ponderada dos quadrados dos ajustes individuais ou o quadrado do ajuste total ( $a_t$ ). Os fatores  $p_j$  são os fatores ponderantes e sua presença faz com que a participação dos ajustes individuais no ajuste total seja diferenciada. Se todos os  $p_j$  fossem iguais a 1, o quadrado do ajuste total seria simplesmente a soma dos quadrados dos ajustes individuais. Neste caso, as medidas com maiores ajustes (mais erradas) dariam maiores contribuições ao ajuste total. É comum considerar os fatores ponderantes como iguais ao inverso das variâncias respectivas. Assim, as medidas com maiores variâncias (e conseqüentemente com maior probabilidade de erro, posto que a variância mede a dispersão em torno da média) têm sua participação atenuada no ajuste total.

Então, a função-objetivo  $f$  é minimizada com a condição de que as vazões estimadas satisfaçam as equações de balanço de massa. A solução é obtida utilizando o método dos multiplicadores de Lagrange.

O caso de balanços multicomponentes também foi estudado por Václavěk et alii. (1976a), onde a função-objetivo é modificada, acrescentando-se um termo referente aos ajustes das medidas de concentração. Todavia, estabelecem a condição de que quando a concentração de um dos componentes de uma corrente é medida, todas o são.

### 3.4. Decomposição do Problema de Reconciliação de Dados de Processo

Na maioria dos casos de reconciliação de dados de processo, nem todos os parâmetros das correntes de um sistema são medidos. Nesta situação, há diversas alternativas para simplificar o procedimento de reconciliação. Deve-se, geralmente, eliminar as variáveis não-medidas, com o objetivo de diminuir a dimensão do problema de minimização, no qual são determinados os ajustes das quantidades medidas.

Isto é possível, pois os conflitos com as leis de conservação só podem aparecer em seções do processo, onde tenham sido feitas medições em todas as correntes de uma equação de balanço (que pode ser a soma de balanços nodais) ou onde haja redundância.

Václavěk (1969a e 1969b) introduziu o conceito de esquema de balanço reduzido, com o intuito de reduzir o esquema de balanço inicial de sistemas, que contêm variáveis não-medidas, de modo a possibilitar a redução do problema de ajuste de dados de plantas químicas complexas. Junto com seus colaboradores (1976a e 1976b), propõe algoritmos para classificar os parâmetros das correntes de um processo, que servem para identificar quais os parâmetros medidos passíveis de correção e quais os parâmetros não-medidos que podem ser determinados a partir dos valores corrigidos. Argumentam que somente os parâmetros medidos redundantes podem ser reconciliados e somente estes precisam ser tratados na etapa de minimização da função-objetivo, reduzindo a dimensão do problema. Os algoritmos propostos permitem classificar tanto as vazões mássicas totais, como as concentrações dos componentes das correntes, através, respectiva e sucessivamente, dos gráficos de correntes materiais reduzidos de primeiro e de segundo grau, aplicando-se a balanços multicomponentes. Todavia, ainda assim, são restritos, pois os parâmetros das correntes podem ser classificados,

somente, em quatro categorias, de forma que uma corrente deva ter todas ou nenhuma concentração de seus componentes medida e pode, ou não, ter sua vazão total medida.

Mah et alii. (1976), estudando o ajuste de vazões totais de processo, descrevem um procedimento, que se baseia em análise teórico-gráfica, para decompor problemas de reconciliação de dados, onde há correntes não-medidas. O procedimento elimina os arcos do gráfico do processo com vazões mássicas não-medidas (arcos não-medidos) e transforma o problema de reconciliação num problema constituído somente por arcos de vazões medidas (arcos medidos). Obtêm isto, agregando quaisquer dois nós, que têm um arco não-medido entre eles, removendo, portanto, todos os arcos internos entre eles. Todos os arcos externos são preservados nesta fusão. Este procedimento é repetido até que todos os arcos não-medidos sejam eliminados. A reconciliação é feita sobre a rede transformada, assim obtida. Em sequência, após a reconciliação, fazem a estimativa das vazões não-medidas contidas nas árvores-arcos do gráfico do processo. Observam que as medidas de vazão podem ser determinadas de forma única se, e somente se, os arcos não-medidos formarem gráficos acíclicos, isto é, árvores.

Romagnoli e Stephanopoulos (1980), tendo em vista que a aplicação do procedimento de reconciliação de dados a plantas químicas complexas, em escala industrial, gera problemas de dimensão muito grande, trataram de reduzir o tamanho do problema de mínimos quadrados, através da classificação das variáveis de processo medidas e não-medidas. Usando as propriedades de resolubilidade estruturais das equações de balanço, definiram as condições gerais para a verificar quais dos parâmetros não-medidos são determináveis. Pela aplicação direta dessas condições, desenvolveram um método que classifica os parâmetros de processo em quatro categorias: 1) determinável, 2) indeterminável, 3) superdeterminado e 4) estritamente determinado. Nas duas primeiras, são enquadradas as variáveis não-medidas e nas duas últimas as medidas. Usando esta estratégia de classificação, pode ser feita a redução dos balanços iniciais a um conjunto de equações de tamanho menor que o original, que contém somente os parâmetros medidos superdeterminados. Argumentam que o método é bastante geral e pode ser usado em conjunto com balanços de massa e calor, lineares e não-lineares. Entretanto, ao desenvolver os algoritmos consideraram que todos os parâmetros mensuráveis eram medidos e que quando a composição de uma dada corrente era medida, isto era feito para todos os seus componentes.

Crowe et alii. (1983) utilizaram projeção matricial para simplificar o problema de reconciliação de dados de processo, contendo parâmetros não-medidos. Definiram uma matriz projeção, que pode ser construída, diretamente, e que elimina, de forma efetiva, as quantidades não-medidas do conjunto de restrições, produzindo um conjunto reduzido de balanços. Uma vantagem desta abordagem com relação às alternativas apresentadas nos parágrafos anteriores é que não assume como premissa que as concentrações de todos os componentes de uma corrente devam ser medidas, quando uma delas for medida, o que lhe confere um caráter mais amplo do que o das demais. Ela permite tratar casos que incluem correntes cuja composição é parcialmente medida e mesmo conjuntos de equações restritivas lineares com matrizes arbitrarias. Observaram que esta metodologia é equivalente aos esquemas de balanços reduzidos de Václavek (1969b), no caso em que uma corrente é completamente medida ou não-medida. Além disso, a metodologia foi estendida para permitir a inclusão de restrições não-lineares, que contém parâmetros desconhecidos, como a fração de divisão de um divisor de fluxo.

Esta última metodologia é empregada nos procedimentos computacionais para reconciliação, retificação e classificação de variáveis de processo.



### 3.5. Reconciliação de Dados de Processo Através de Projeção Matricial

Crowe et alii.(1983) desenvolveram um procedimento para reconciliação de vazões de processo, para sistemas no estado estacionário, que têm correntes medidas e não-medidas, cujas restrições são lineares. Utilizaram projeção matricial, para obter um conjunto reduzido das equações de balanço, referente às correntes cujos dados medidos podem ser ajustados, de forma que a soma ponderada dos quadrados dos ajustes seja mínima e que as leis de conservação sejam obedecidas.

A planta química considerada tinha  $K$  unidades de processo (ou nós),  $J$  correntes e  $C$  componentes numa corrente. A estrutura da planta é expressa através de sua matriz incidência  $A$ , cujas linhas correspondem aos nós e cujas colunas, às correntes. Para obter a matriz dos balanços  $B$ , substituiu-se cada elemento  $\pm 1$  da matriz incidência por  $\pm I$  (a matriz identidade) e cada elemento 0 pela matriz nula. Então, a matriz  $B$  é uma matriz-bloco, em que cada um contém  $C$  linhas e  $C$  colunas, que correspondem a um nó e a uma corrente, respectivamente. Para cada nó  $k$ , em que ocorrem  $R_k$  reações químicas, constrói-se uma matriz estequiometria  $S_k$ . Cada linha de  $S_k$ , corresponde a uma reação química  $r$  ( $r = 1, 2, \dots, R_k$ ) e cada coluna a um componente  $c$  ( $c = 1, 2, \dots, C$ ). O elemento  $S_{krc}$  é o coeficiente estequiométrico do componente  $c$  na reação  $r$ , que ocorre no nó  $k$ , seguindo a convenção de negativo, positivo ou zero para um reagente, produto ou inerte, respectivamente. Para cada nó reacional, define-se um vetor  $\zeta_k$  das extensões das reações, que não é medido ou conhecido, inicialmente. Então, define-se a matriz estequiometria  $S$ , com blocos  $S_k$ , não-zero ( $R_k$  por  $C$ ), somente para os nós reacionais, onde cada linha do bloco corresponde a uma reação.

Se  $x_j$  é o vetor das vazões dos componentes na corrente  $j$  e ainda se  $x^T = [x_1^T \ x_2^T \ \dots \ x_j^T]$ , então os balanços materiais da planta podem ser escritos como

$$Bx + S^T \zeta = 0 \quad (3.1)$$

onde  $\zeta$  é o vetor das extensões de todas as reações de todos os nós reacionais.

Balanços de energia podem ser adicionados à definição do problema, considerando a taxa de fluxo de entalpia como o  $(C+1)$ -ésimo componente. Então,  $B$  terá  $K$  blocos-linhas, um para cada nó, com  $C+1$  linhas cada um. A matriz  $S_k$  é modificada, adicionando uma coluna contendo a mudança de entalpia-padrão para cada reação (baseada num produto com coeficiente estequiométrico unitário). Além disso, o balanço de energia em cada nó pode conter um fluxo de calor  $q_k$  (usualmente não-medido).

Quando todos os componentes de uma corrente fazem parte dos balanços, deve-se garantir que, se a vazão total for medida, a vazão total ajustada seja igual à soma das vazões dos componentes ajustadas. Se  $x_{tj}$  é a vazão total da corrente  $j$ ,

$$1^T x_j = x_{tj} \quad (3.2)$$

onde  $1^T$  é um vetor linha, cujos elementos são o número 1. A matriz  $B$  é, então, aumentada, adicionando uma linha e uma coluna para cada uma dessas correntes, e o vetor  $x_t$  das vazões totais é juntado a  $x$  de modo que a eq.3.1 se torna

$$\begin{bmatrix} B & & & & 0 \\ \hline 1^T & 0 & \dots & \dots & -I \\ \bullet & 1^T & 0 & \dots & \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} x \\ \dots \\ x_t \end{bmatrix} + \begin{bmatrix} S^T \\ \dots \\ 0 \end{bmatrix} \zeta = 0 \quad (3.3)$$

Por simplicidade, é mantido o balanço dado pela eq.3.1.

O vetor  $\tilde{x}$  das vazões medidas dos componentes é composto pelos vetores  $\tilde{x}_j$  de cada corrente, mas contém somente tantos elementos quanto valores medidos. A vazão de qualquer componente, cuja concentração não é medida, é incluída em  $u$ , o vetor das vazões desconhecidas.

As colunas de  $B$  são separadas de forma que

$$B = [B_0 : B_1 : B_3]$$

$B_0, B_1$  e  $B_3$  contêm, respectivamente, as colunas correspondentes às vazões dos componentes exatamente conhecidas ( $b_0$ ), às vazões dos componentes medidas ( $\tilde{x}$ ) e às vazões dos componentes não-medidas ( $u$ ).

As medidas  $\tilde{x}$  devem ser ajustadas por quantidades "a", de forma que os balanços materiais, dados pela eq.3.1, sejam válidos para as vazões ajustadas, isto é,

$$B_0 b_0 + B_1(\tilde{x} + a) + B_3 u + S^T \zeta = 0 \quad (3.4)$$

Supõe-se que os balanços sejam linearmente independentes.

As vazões estimadas são dadas por

$$\hat{x} = \tilde{x} + a \quad (3.5)$$

Para simplificar, definem

$$P \equiv [B_3 : S^T] \quad (3.6)$$

e

$$v^T \equiv [u^T : \zeta^T] \quad (3.7)$$

de modo que a eq.3.4 fica

$$B_0 b_0 + B_1(\tilde{x} + a) + Pv = 0 \quad (3.8)$$

Definem um problema inicial dado por

$$\min_{a, u, \zeta} f(a) = a^T \Sigma^{-1} a \quad (3.9)$$

sujeita à eq.3.8.  $\Sigma$  é, geralmente, uma matriz ponderante, positiva por definição, mas aqui é a matriz variância-covariância das medidas  $\bar{x}$ . Supõe-se, usualmente, que ela é uma matriz constituída de blocos diagonais, em que cada bloco corresponde a uma corrente, embora esta consideração não seja necessária. Todas as vazões exatamente conhecidas devem ser separadas num vetor dos valores constantes para evitar que  $\Sigma$  seja singular.

Este problema inicial pode ser resolvido utilizando os multiplicadores de Lagrange. Todavia, é definido um segundo problema, cuja solução é também solução do problema inicial.

Então, é definida uma matriz  $Y$ , cujas colunas varrem o espaço-nulo de  $P^T$ , isto é,

$$P^T Y = 0 \quad (3.10)$$

Assim, qualquer vetor  $w (\neq 0)$ , tal que  $P^T w = 0$ , é uma combinação linear das colunas de  $Y$ . Portanto, da eq.3.8,

$$Y^T [B_0 b + B_1 (\bar{x} + a)] = 0 \quad (3.11)$$

Isto leva à definição do segundo problema

$$\begin{aligned} &\min f(a) \\ &a \\ &\text{sujeita à eq.3.11} \end{aligned}$$

Se  $a_*$  é solução mínima deste problema, deve-se resolver

$$P v_* = -B_1 (\bar{x} + a_*) - B_0 b_0 \quad (3.12)$$

para obter  $v_*$ .

Existe uma solução mínima para o segundo problema, pois a matriz variância é positiva por definição e, portanto, o valor de  $a_*$  que minimiza  $f(a)$  é único. Observam que o vetor dos ajustes mínimos ( $a_*$ ) pode ser encontrado, mesmo se não houver um único  $v_*$ . O vetor  $v_*$  que satisfaz a eq.3.12, existe e é único se, e somente se, as colunas da matriz  $P$  forem linearmente independentes. Isto significa que

$$P v = 0 \quad \Rightarrow \quad v = 0 \quad (3.13)$$

e leva a um teste computacional direto para verificar a unicidade de  $v_*$ . A não-unicidade de  $v_*$  é equivalente a ser possível satisfazer balanços materiais envolvendo apenas parâmetros não-medidos.

O método para construção da matriz  $Y$ , a qual ao pré-multiplicar a equação de balanços (eq.3.8) retira dela as quantidades não-medidas, é o seguinte:

i) reduzir as colunas de  $P$ , para obter  $P_r$  com colunas linearmente independentes. Isto equivale a pós-multiplicar  $P$  por uma matriz não-singular  $F$ , de forma que

$$P F = [P_r ; 0] \quad (3.14)$$

ii) separar  $P_r$ , através de uma matriz permutação, de forma que

$$P_r = \begin{bmatrix} P_1 \\ \text{---} \\ P_2 \end{bmatrix} \quad (3.15)$$

onde  $P_1$  é quadrada e não-singular. Então,

$$Y^T = [-P_2 P_1^{-1} : I] \quad (3.16)$$

Observam que nos casos em que não há reação química, o uso de  $Y$  é equivalente ao esquema de balanço reduzido de Václavěk (1969a) aplicado a cada componente, em separado. Tamhane e Mah (1985) também relatam que a pré-multiplicação da matriz dos balanços pela matriz  $Y$  é equivalente ao procedimento de decomposição, baseado em análise teórico-gráfica, proposto por Mah et alii. (1986), no caso de sistemas com restrições devidas somente ao arranjo do processo.

A determinação dos ajustes é feita usando os multiplicadores de Lagrange. Resolve-se o segundo problema, encontrando um ponto estacionário  $(a_*, \lambda_*)$  do Lagrangiano

$$L(a, \lambda) \equiv \frac{1}{2} f(a) - \lambda^T Y^T [B_0 b_0 + (\bar{x} + a)] \quad (3.17)$$

Isto dá

$$a_* = \sum B_1^T Y \lambda_* \quad (3.18)$$

e

$$H \lambda_* = -Y^T [B_0 b_0 + B_1 \bar{x}] \quad (3.19)$$

com

$$H = Y^T B_1 \sum B_1^T Y \quad (3.20)$$

A eq.3.19 é resolvida obtendo-se  $\lambda_*$ , que é substituído na eq.3.18 para o cálculo de  $a_*$ , desde que  $Y^T B_1$  tenha característica igual ao número de linhas. Então, a eq.3.12 é resolvida para obter  $v_*$ .

### 3.6. Bases do Procedimento Computacional de Reconciliação de Dados e Coaptação

Desenvolveu-se um procedimento computacional para reconciliar quantidades medidas e para obtenção de estimativas de quantidades não-medidas observáveis de processos no estado estacionário, sujeitos a restrições lineares, envolvendo balanços de massa multicomponentes, ocorrendo ou não reações químicas, usando projeção matricial.

Como se trata do caso linear, são incluídas somente aquelas situações em que quando a concentração de uma espécie química de uma corrente é medida, a vazão total correspondente

também é medida, não importando o número de componentes que tem concentração medida naquela corrente.

A vazão total de cada corrente é sempre considerada, destinando-se a ela a última coluna de cada um dos blocos, correspondente a cada uma das correntes, da matriz das restrições. Em vista disto, à matriz incidência são adicionadas linhas referentes ao que se chamou de balanços das correntes individuais, que são restrições de corrente. Desta forma, garante-se que a soma das vazões corrigidas dos componentes de uma corrente seja igual a sua vazão total corrigida e, além disso, contribui-se para aumentar a redundância. Se for o caso, podem-se adicionar outras restrições de corrente, como a referente à fração de partição conhecida de um divisor de fluxo.

Através do vetor de informações das variáveis (viv) é feita a separação das colunas da matriz B, obtendo-se as matrizes  $B_1$ , que é  $m \times q$ , e P, que é  $m \times n$ . Considerando que um dado sistema tenha c espécies químicas,  $n_i$  correntes totais e  $r_q$  reações químicas, este vetor tem um número de elementos igual a  $n_i \times (c+1) + r_q$ . Então, a cada corrente correspondem  $c+1$  elementos de viv, que correspondem às vazões dos componentes, seguidas da vazão total daquela corrente. Porém, em geral, nem todas as espécies fazem parte de cada corrente, assim coloca-se 0 no elemento de viv correspondente à vazão da espécie que não pertence àquela corrente, 1 quando pertencer e tiver vazão do componente medida e 3 quando pertencer e não for medida. No elemento referente à corrente total, coloca-se 0, quando ela contiver uma única espécie; quando for constituída por mais de um componente, coloca-se 1 quando pelo menos a vazão de um deles for medida e 3 quando nenhuma for medida. Os últimos  $r_q$  elementos de viv correspondem cada um deles à conversão de uma reação química e são sempre iguais a 3, pois elas não são medidas. Considerou-se que não há vazões exatamente conhecidas.

Para incluir uma reação química, basta informar através da matriz estequiometria TE, que é  $r_q$  por  $c+2$ , onde cada linha corresponde a uma reação química, o número do nó, onde ela ocorre e seus coeficientes estequiométricos, colocando-os com sinal negativo se forem reagentes, positivo se forem produtos e 0 se forem inertes ou não participarem da reação, ou ainda se forem referentes às correntes totais.

As colunas da matriz das restrições que corresponderem a elementos de viv iguais a 1 farão parte da matriz  $B_1$ , as que corresponderem a elementos iguais a 3 integrarão a matriz P.

Como a obtenção da matriz das restrições é feita de forma ampla, isto é, como se todas as espécies entrassem ou saíssem de todos os nós, é necessário retirar algumas de suas linhas, que correspondem, por exemplo, a restrições inexistentes. Para isto, foi elaborado o vetor de informações dos equipamentos (vie), que, no caso de um sistema com  $m_i$  nós e  $n_r$  restrições de corrente adicionais, tem  $m_i \times (c+1) + n_r \times (c+1) + n_i$  elementos. Cada grupo de  $c+1$  elementos deste vetor corresponde a um nó (ou restrição adicional), atribuindo-se a eles o valor 0, quando a espécie química correspondente não fizer contato com aquele nó (ou não participar daquela restrição) ou corresponder à corrente total e 1 quando o componente correspondente entrar ou sair daquele nó. Nos últimos  $n_i$  elementos, correspondentes aos balanços individuais, coloca-se 0 no elemento correspondente a uma corrente com um único componente e 1 no elemento correspondente a uma corrente com mais do que um componente. As linhas removidas correspondem aos elementos 0 de vie.

Para obtenção da matriz Y, tal que  $P^T Y = 0$ , utilizou-se o seguinte método:

i) através de um processo de eliminação, permeado por trocas de linhas e colunas, transformou-se uma cópia da matriz  $P$  na matriz  $PL$ , que é uma matriz triangular superior, que tem a forma genérica dada abaixo, onde os elementos da diagonal são os "pivots" (portanto não-iguais a zero), de modo a obter a maior matriz não-singular no canto superior esquerdo; os elementos acima da diagonal podem ou não ser zero. As trocas de linhas e/ou de colunas são registradas através de um ponteiro para linhas ( $ip$ ) e outro para colunas ( $ic$ ), que são usados para efetuar, posteriormente, as mesmas trocas em  $P$ . Através de  $PL$ , determina-se a característica de  $(P) = r$ .

$$PL = \begin{bmatrix} * & * & * & * & * & * & * & \dots & * & * & * \\ 0 & * & * & * & * & * & * & \dots & * & * & * \\ 0 & 0 & * & * & * & * & * & \dots & * & * & * \\ 0 & 0 & 0 & * & * & * & * & \dots & * & * & * \\ 0 & 0 & 0 & 0 & * & * & * & \dots & * & * & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix}$$

ii) trocam-se as linhas e/ou colunas de  $P$ , de acordo com os ponteiros  $ip$  e  $ic$ .

iii) separa-se a matriz  $P$  de forma que:

$$P = \begin{bmatrix} P_1 & \vdots & P_3 \\ \dots & \dots & \dots \\ P_2 & \vdots & P_4 \end{bmatrix}$$

onde  $P_1$  é a maior matriz quadrada não-singular ( $r \times r$ ) contida em  $P$ ,  $P_2$  é  $(m-r) \times r$  e  $P_3$  é  $r \times (n-r)$ .

iv) obtém-se  $Y^T = [-P_2 P_1^{-1} : I]$ , que é  $(m-r) \times m$ , fazendo o produto indicado e completando as colunas de  $Y^T$  com a matriz identidade  $I$ , que é  $(m-r) \times (m-r)$ .

v) desfazem-se as trocas de coluna de  $Y^T$  pós-multiplicando-a pela matriz  $H_1$ , obtida após troca de linhas de uma matriz identidade  $(m \times m)$  de acordo com as informações do ponteiro para linhas  $ip$ .

Quando todas as quantidades são medidas  $Y^T = I$ , que então é  $m \times m$ .

A determinação dos ajustes é feita usando os multiplicadores de Lagrange, conforme descrito no item anterior.

Quando  $r = n$ , todos os valores não-medidos são observáveis. A determinação de quais dos valores não-medidos são observáveis é feita através da forma descrita na classificação de variáveis (Capítulo 4), tomando a solução da equação geral para o caso particular em que  $w=0$ , em que ela depende somente das colunas de  $P$ .

A obtenção de estimativas para os valores observáveis é feita através da equação:

$$v = -P_1^{-1}B_{1m}\hat{x}$$

onde:

$v$  = vetor das estimativas das quantidades observáveis,  $r \times 1$

$P_1^{-1}$  = inversa de  $P_1$ ,  $r \times r$

$B_{1m}$  = matriz  $r \times q$ , obtida após troca de linhas de  $B_1$ , de acordo com  $i_p$ , tomando as suas  $r$  primeiras linhas

$\hat{x}$  = valores corrigidos das quantidades medidas

Deve-se notar que nem sempre todas as quantidades medidas sofrem ajustes e que o vetor das estimativas de quantidades observáveis contém o valor zero na posição correspondente a quantidades não-observáveis.

### 3.7. O Programa de Reconciliação de Dados de Processo

Este programa destina-se à reconciliação de vazões totais e dos componentes das correntes de um processo, seguida da coaptação, podendo tratar de sistemas com um único componente ou multicomponentes, sujeitos a restrições lineares. Prevê a ocorrência de reações químicas, bem como a adição de restrições do tipo razão de divisão de um divisor de fluxo, quando ela for conhecida. Este programa, designado Programa RECON, foi escrito em linguagem Turbo C e sua listagem é dada no Apêndice C.

No Programa RECON, utiliza-se projeção matricial, para simplificar o problema de obtenção de ajustes de quantidades medidas, que permite remover as quantidades não-medidas das restrições. A matriz projeção é obtida através de método desenvolvido neste trabalho, descrito no item 3.6. Os ajustes dos valores medidos são feitos através do método dos mínimos quadrados ponderados e dos multiplicadores de Lagrange. A partir dos valores corrigidos das quantidades medidas, são obtidas estimativas para os valores de quantidades não-medidas observáveis, que podem incluir vazões e a extensão de reações químicas, do modo descrito no item 3.6. As vazões totais das correntes são incluídas nas equações de balanço de forma que a vazão total corrigida de uma corrente é igual à soma das vazões corrigidas de seus componentes.

Dentro do próprio programa é calculada a matriz variância-covariância, que é obtida considerando que as variâncias e covariâncias das vazões dos componentes dependem das frações correspondentes e da variância da corrente total, desprezando a participação das variâncias das frações dos componentes. Isto é válido, considerando aqueles casos onde as últimas são muito pequenas em comparação com as variâncias das correntes globais, como no

caso da determinação da composição de correntes gasosas através de análise cromatográfica, onde o erro pode ser da ordem de 0,05%. Nesta situação, o produto do quadrado da vazão pela variância da concentração é praticamente igual a zero. Além disso, os elementos de fora da diagonal foram divididos por dois.

Considerou-se que:

- o número máximo de nós é 10;
- o número máximo de correntes totais é 20;
- o número máximo de espécies químicas é 10;
- o número máximo de reações químicas é 5 e
- o número máximo de restrições adicionais é 3.

Os dados de entrada podem ser introduzidos através do teclado diretamente ou através da leitura de um arquivo de dados tipo texto.

Os dados de entrada são os seguintes e devem ser gravados, em arquivo de dados, na seqüência dada abaixo:

- número de correntes totais ( $n_i$ );
- número de nós ( $m_i$ );
- número de espécies químicas ( $c$ );
- número de reações químicas ( $r_q$ );
- número de restrições adicionais ( $n_r$ );
- número de vazões medidas ( $q$ ), que é a soma do número de vazões dos componentes medidas, com o número de vazões totais medidas.

-matriz de informações das correntes  $M(n_i \times 2)$ ; esta matriz tem número de linhas igual ao número de correntes totais e duas colunas. A cada linha corresponde uma corrente tomada na ordem numérica, sendo que o primeiro elemento de cada linha deve conter o número do nó no qual a corrente entra e o segundo elemento, o número do nó do qual ela sai. Quando uma corrente entrar ou sair do meio-ambiente, deve ser colocado o número 100.

-vetor variância das correntes totais ( $varct[n_i]$ ); este vetor tem número de elementos igual ao número de correntes totais, cujos valores devem ser colocados na ordem numérica crescente dessas correntes, colocando um 0 no elemento correspondente a uma corrente não-medida.

-matriz das frações parciais  $MFM(n_i \times (c+1))$ ; esta é usada para calcular a matriz variância-covariância, tendo  $n_i$  linhas e  $c+1$  colunas. A cada linha corresponde uma corrente tomada na ordem numérica, cujos elementos devem conter os valores das frações dos componentes, quando medidos; devendo-se colocar o número 2 ou 0, quando não-medidos ou



não integrarem aquela corrente, respectivamente. No elemento correspondente à corrente total, que contiver um único componente, deve-se colocar 0; enquanto no elemento correspondente à corrente total, que tiver mais de um componente deve-se colocar 1 ou 2, quando medida ou não-medida, respectivamente.

-matriz estequiometria  $TE(rq \times (c+2))$ ; que tem número de linhas igual ao número de reações químicas consideradas e número de elementos igual ao número de espécies químicas presentes no sistema acrescido de duas unidades. No primeiro elemento de cada linha, deve constar o número do nó, onde ocorre a reação correspondente, seguida de seus coeficientes estequiométricos, colocando-os com sinal negativo se forem reagentes, positivo se forem produtos e 0 se forem inertes ou não participarem da reação ou ainda se forem referentes às correntes totais (que ocupam o último elemento de cada linha).

-matriz das restrições  $TR(nr \times ni)$ ; que tem número de linhas igual ao número de restrições adicionais e número de colunas igual ao número de correntes totais. Cada linha corresponde a uma restrição, sendo que cada elemento se refere a uma corrente total. Um elemento leva o número 0, quando a corrente correspondente não fizer parte daquela restrição, ou algum coeficiente numérico diferente de zero. Por exemplo, no caso do divisor de fluxo, onde entra a corrente T5 e saem as correntes T6 e T7, pertencentes a um sistema com 7 correntes totais, onde a vazão total da corrente 6 é 2,0% da corrente 5, a restrição pode ser dada por:

$$0,02 T5 - T6 = 0$$

e a linha da matriz TR conteria os seguintes elementos:

$$0 \ 0 \ 0 \ 0 \ 0,02 \ -1 \ 0$$

-vetor das vazões medidas, onde são colocadas as vazões dos componentes de cada corrente, seguidas da vazão total de cada uma, na ordem numérica crescente das correntes.

-vetor informações das variáveis (viv); quando se refere a um dado sistema que tem c espécies químicas, ni correntes totais e rq reações químicas, ele deve conter um número de elementos igual a  $ni \times (c+1) + rq$ . Então, a cada corrente correspondem c+1 elementos de viv, que correspondem às vazões dos componentes, seguidas da vazão total daquela corrente. Porém, em geral, nem todas as espécies fazem parte de cada corrente, assim coloca-se 0 no elemento de viv correspondente à vazão da espécie que não pertence àquela corrente, 1 quando pertencer e tiver vazão do componente medida e 3 quando pertencer e não for medida. No elemento referente à corrente total, coloca-se 0, quando ela contiver uma única espécie; quando for constituída por mais de um componente, coloca-se 1 quando pelo menos a vazão de um deles for medida e 3 quando nenhuma for medida. Os últimos rq elementos de viv correspondem cada um deles à conversão de uma reação química e são sempre iguais a 3, pois elas não são medidas.

-vetor informações dos equipamentos (vie); no caso de um sistema com mi nós e nr restrições de corrente adicionais, ele tem  $mi \times (c+1) + nr \times (c+1) + ni$  elementos. Cada grupo de c+1 elementos deste vetor corresponde a um nó (ou restrição adicional), atribuindo-se a eles o valor 0, quando a espécie química correspondente não fizer contato com aquele nó (ou não participar daquela restrição) ou corresponder à corrente total e 1 quando o componente correspondente entrar ou sair daquele nó (ou participar daquela restrição). Os últimos ni elementos dizem respeito às equações dos balanços individuais, colocando-se 0 no elemento

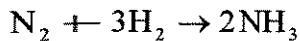
correspondente a uma corrente com um único componente e 1 no elemento correspondente a uma corrente com mais do que um componente.

### 3.8. Aplicações do Programa RECON

A fim de testar e analisar o seu desempenho em situações diversas, o programa RECON foi aplicado a cinco sistemas diferentes, usando um computador pessoal, que são: um processo de síntese de amônia, um reator químico onde ocorrem duas reações químicas, uma rede de vapor, um processo de síntese de ácido nítrico e um circuito de moagem.

#### a) Processo de Síntese de Amônia

Aplicou-se a reconciliação de dados a um processo simples de síntese de amônia, empregado anteriormente por Crowe et alii. (1983). O número de espécies químicas envolvidas é 4, que são nitrogênio, hidrogênio, amônia e argônio, designados, respectivamente, pelas letras A, B, C e D. Os componentes presentes em cada corrente são mostrados no fluxograma da Figura 3.2. A amônia é gerada no nó número 2, através da seguinte reação:



O nó número 4 é um divisor de fluxo, cuja restrição correspondente faz parte do conjunto de restrições, considerando que vazão da corrente 6 é igual a 2% da corrente 5.

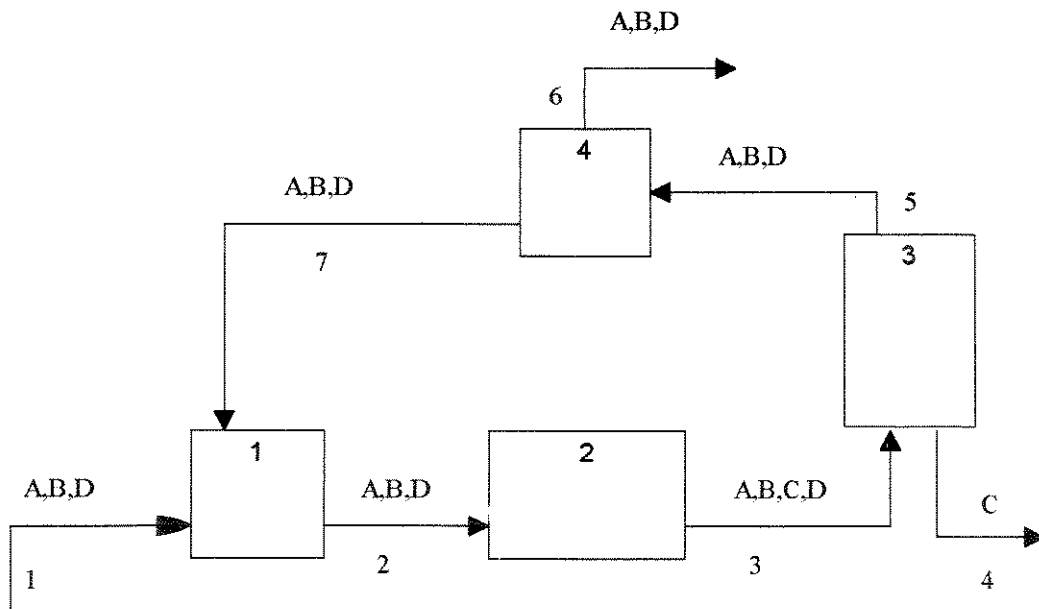


Figura 3.2 - Fluxograma do Processo de Síntese da Amônia

As vazões reais das correntes medidas são apresentadas na Tabela 3.1, onde a letra T, seguida de um número, representa a vazão total daquela corrente. Os valores medidos foram obtidos introduzindo erros aleatórios, que seguem uma distribuição normal, aos valores reais, utilizando tabela encontrada em Freund (1988), que fornece números normais aleatórios. Para

obter os valores de uma variável aleatória, a partir de uma tabela desse tipo, faz-se uso da fórmula:

$$x = \mu + \sigma z$$

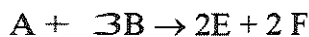
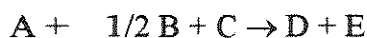
que transforma valores de uma variável aleatória com distribuição normal padronizada em valores de uma variável aleatória com distribuição normal com média  $\mu$  e desvio-padrão  $\sigma$ . Como no uso das tabelas de números aleatórios comuns, a escolha da página e da coluna em que se vai começar é feita ao acaso.

A Tabela 3 - 1 apresenta ainda os resultados da reconciliação de dados, bem como as estimativas para as quantidades não-medidas observáveis. Verifica-se, neste exemplo, que as quantidades medidas foram corrigidas e que todas as quantidades não-medidas são observáveis. A reconciliação de dados levou à redução do erro relativo percentual médio (obtido através da média aritmética dos módulos dos erros relativos percentuais individuais), que é igual a 1,28% para os valores medidos e igual a 1,18% para os valores reconciliados.

As frações medidas dos componentes das correntes totais e os desvios-padrão dessas correntes, que são usados no cálculo da matriz variância-covariância, são dados na Tabela 3.2.

### b) Reator Químico

Aplicou-se a reconciliação de dados ao caso de um reator químico. O número de espécies químicas envolvidas é 7, designadas pelas letras A, B, C, D, E, F e G; a última é um inerte. Os componentes presentes em cada corrente são mostrados no fluxograma da Figura 3.3. Este sistema é constituído por um único nó, onde ocorrem duas reações químicas, cujas estequiometrias são dadas por:



A primeira é a reação primária e a segunda é uma reação secundária. Ambas são irreversíveis; D é o produto desejado, enquanto E e F não têm valor comercial. Há um total de 9 correntes totais neste processo, 6 são correntes de alimentação, cujos produtos são separados em 3 correntes.

As vazões medidas, que correspondem a um conjunto típico de dados sem erros grosseiros, gerado através de computador, conforme trabalho de Phillips e Harrison (1993), são apresentadas na Tabela 3.3, onde a letra T representa a vazão total daquela corrente. Esta tabela apresenta, também, os resultados da reconciliação e da coaptação. Verifica-se que todas as quantidades não-medidas são observáveis, porém nem todas as quantidades medidas sofrem ajustes, como as vazões das correntes A1, B2, F3 e G4.

As frações medidas de cada componente das correntes totais, usadas no cálculo da matriz variância-covariância, são dadas na Tabela 3.4. Considerou-se que o desvio-padrão de cada corrente é igual a 2,81% dos valores reais.

Tabela 3.1. Dados do caso da síntese de amônia (em kgmol/s)

Quantidade	Valor real	Valor medido	Valor estimado
A1	32,70	32,33	32,15
B1	98,04	98,52	96,28
D1	0,40	0,40	0,40
T1	131,14	132,24	128,83
A2	100,56	103,86	101,00
B2	nm	nm	294,89*
D2	20,15	20,11	19,81
T2	419,70	415,53	415,70
A3	69,23	69,10	70,26
B3	nm	nm	202,65
C3	nm	nm	61,49*
D3	nm	nm	19,81*
T3	357,04	351,04	354,21
C4	62,66	59,32	61,49
A5	nm	nm	70,26*
B5	205,00	205,28	202,65
D5	nm	nm	19,81*
T5	294,38	291,02	292,72
A6	nm	nm	1,41*
B6	nm	nm	4,05*
D6	nm	nm	0,40*
T6	nm	nm	5,85*
A7	nm	nm	68,85*
B7	nm	nm	198,60*
D7	nm	nm	19,41*
T7	nm	nm	286,87*
R1	nm	nm	30,74*

Obs.: nm = não-medida

\* =valores observáveis

Tabela 3.2. Dados para o Cálculo da Variância do Caso da Síntese de Amônia

Corrente	Frações		Molares		Variância
	A	B	C	D	
T1	0,2493	0,7476	0	0,0031	13,7700
T2	0,2396	nm	0	0,0480	141,0700
T3	0,1939	nm	nm	nm	102,0899
T4	0	0	1,0	0	3,14
T5	nm	0,6964	0	nm	69,3399

nm= não-medida

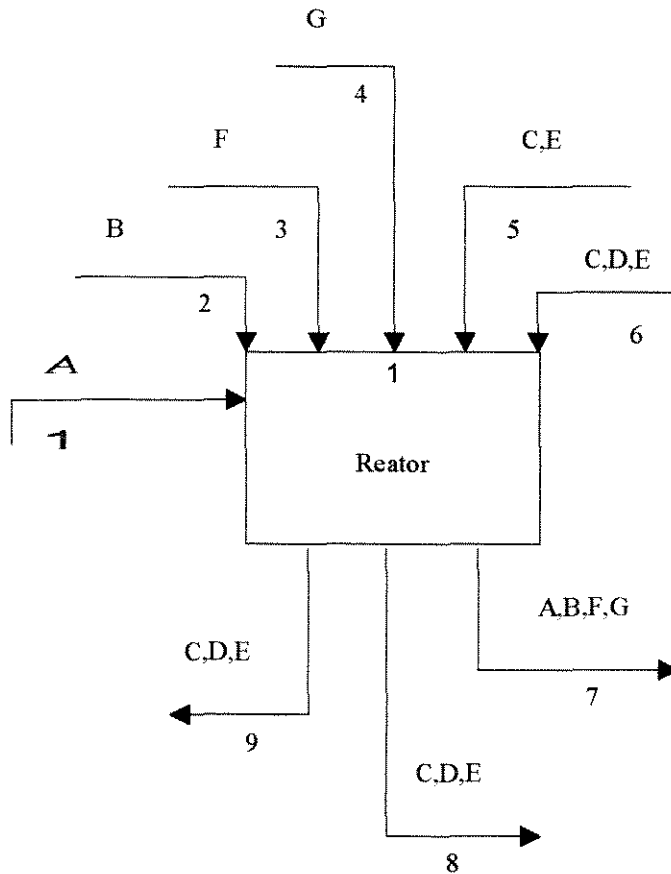


Figura 3.3- Fluxograma do Reator Químico

Tabela 3.3 - Resultados do Caso do Reator Químico (mol/h)

Quantidade	Valor Medido	Val. Reconc.	Quantidade	Valor Medido	Valor Reconc.
A1	3,4620	3,4620	F7	nm	0,5917*
B2	0,7346	0,7346	G7	nm	1,5851*
F3	0,4671	0,4671	T7	nm	5,6194*
G4	1,5851	1,5851	C8	0,0496	0,0497
C5	1,2939	1,2670	D8	0,1546	0,1551
E5	0,1830	0,1807	E8	0,0479	0,0480
T5	1,4769	1,4477	T8	0,2522	0,2528
C6	0,3477	0,3457	C9	1,1983	1,2265
D6	0,0194	0,0193	D9	0,1976	0,2007
E6	0,0473	0,0471	E9	0,6330	0,6410
T6	0,4145	0,4122	T9	2,0289	2,0682
A7	nm	3,0632*	R1	nm	0,3365*
B7	nm	0,3794*	R2	nm	0,0623*

nm = não-medida

\* = quantidade observável

Tabela 3.4 – Dados para o Cálculo da Variância no Caso do Reator Químico

Corrente	Frações Molares dos Componentes das Correntes						
	A	B	C	D	E	F	G
T1	1,0000	0	0	0	0	0	0
T2	0	1,0000	0	0	0	0	0
T3	0	0	0	0	0	1,0000	0
T4	0	0	0	0	0	0	1,0000
T5	0	0	0,8761	0	0,1239	0	0
T6	0	0	0,8389	0,0469	0,1142	0	0
T7	nm	nm	0	0	0	nm	nm
T8	0	0	0,1969	0,6132	0,1899	0	0
T9	0	0	0,5906	0,0974	0,3120	0	0

nm= não-medida

### c) Rede de Vapor

Para aplicar o programa RECON a um sistema com um único componente, foi escolhida uma rede de vapor. Esta, conforme mostra o fluxograma da Figura 3.4, compõe-se de 8 nós e 20 correntes e um único componente, que é a água.

Na Tabela 3.5, constam os valores reais das correntes, obtidos através dos valores empregados por Serth e Heenan (1986); contém, ainda, os valores medidos que foram obtidos mediante a introdução de erros aleatórios, conforme descrito anteriormente e os valores reconciliados.

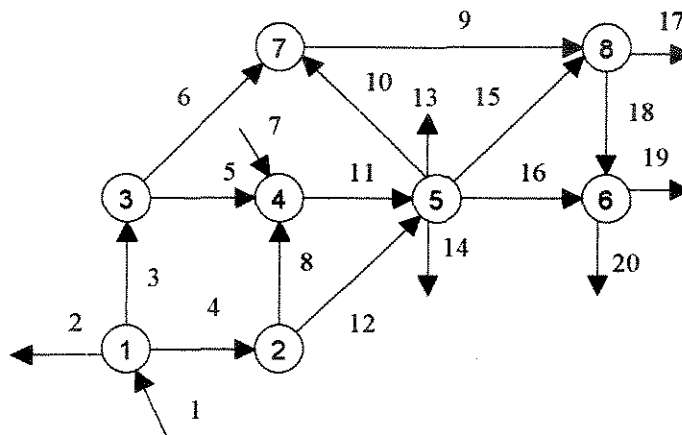


Figura 3.4 -Fluxograma da Rede de Vapor

Para a obtenção da matriz variância, considerou-se que todas as correntes têm desvio-padrão igual a 2,5% de seus valores reais.

A reconciliação de dados levou à redução do erro relativo percentual médio (obtido através da média aritmética do módulo dos erros relativos percentuais individuais), que é igual a 2,03% para os valores medidos e igual a 1,65% para os valores reconciliados.

Tabela 3.5 - Resultados para a Rede de Vapor (em kg/h)

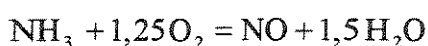
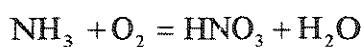
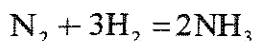
Quantidade	Valor Real	Valor Medido	Valor Reconc.
A1	166,37	167,84	167,08
A2	0,86	0,87	0,87
A3	53,27	53,11	53,45
A4	112,27	112,22	112,76
A5	0,86	0,83	0,83
A6	52,41	52,20	52,62
A7	109,95	110,78	110,71
A8	1,00	0,99	0,99
A9	67,27	68,84	66,76
A10	14,86	14,07	14,14
A11	111,82	113,16	112,53
A12	111,67	112,66	111,77
A13	32,73	32,87	32,93
A14	91,86	93,77	94,24
A15	23,64	24,34	24,25
A16	60,00	56,90	58,74
A17	85,45	84,21	85,77
A18	5,45	5,22	5,24
A19	46,64	47,29	46,30
A20	18,82	17,85	17,69

#### d) Processo de Síntese de Ácido Nítrico

O programa RECON foi aplicado a um processo complexo (com reciclo) de síntese de ácido nítrico, que é constituído de 9 nós e de 16 correntes, contendo 7 espécies químicas, que são:  $N_2$ ,  $H_2$ ,  $NH_3$ ,  $O_2$ ,  $HNO_3$ ,  $H_2O$  e  $NO$ , representadas por A, B, C, D, E, F e G, respectivamente.

No fluxograma do sistema mostrado na Figura 3.5, são indicados os componentes de cada corrente.

Neste sistema, ocorrem três reações químicas, cada uma em um reator, cujas estequiometrias são dadas por:



Estas reações têm lugar nos nós número 2, 5 e 6, respectivamente.

O nó número 9 é um divisor de fluxo e a restrição correspondente faz parte do conjunto de restrições impostas, estabelecendo que a vazão da corrente 16 é igual a 5,0 % da corrente 6.

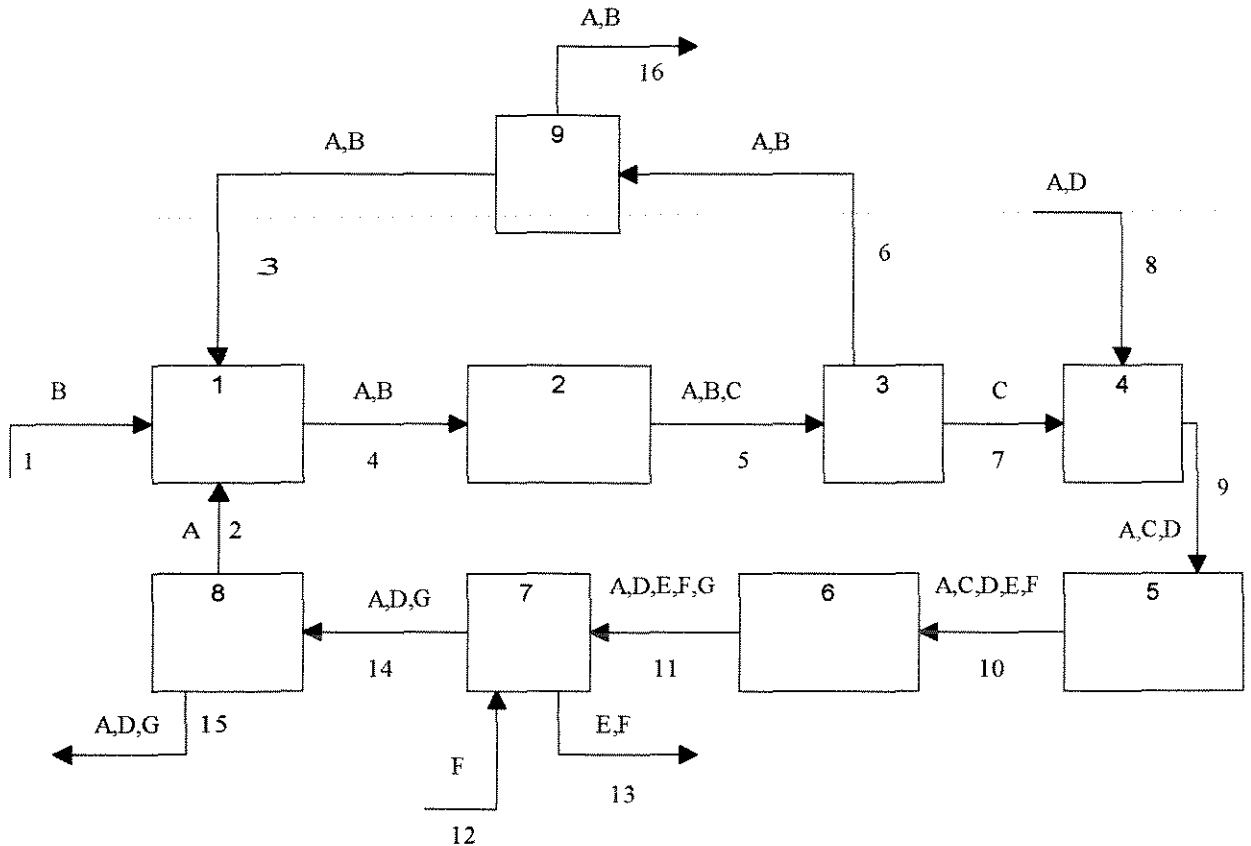


Figura 3.5- Fluxograma do Processo de Síntese de Ácido Nítrico

As vazões medidas, que foram obtidas mediante a introdução de erros aleatórios, conforme modo descrito anteriormente, a partir de dados ajustados de estudo de simulação de Cardoso (1991), são apresentadas na Tabela 3.6; a letra T, seguida de um número, representa a vazão total daquela corrente. Esta tabela apresenta, também, os resultados da reconciliação, cujos valores são dados em kgmol/h.

Verifica-se que todas as quantidades medidas foram corrigidas e que todas as quantidades não-medidas são observáveis e que a reconciliação de dados levou à redução do erro relativo percentual médio (obtido através da média aritmética do módulo dos erros relativos percentuais individuais), que é igual a 0,81% para os valores medidos e igual a 0,55% para os valores reconciliados.

Considerou-se que as correntes totais têm desvio-padrão igual a 2,5% dos valores reais e foram usadas as frações medidas dadas na Tabela 3.7 para o cálculo da matriz variância-covariância.



Tabela 3.6 - Resultados para o Caso da Síntese de Ácido Nítrico (kgmol/h)

Quantidade	Valor Real	Valor Medido	Valor Reconc.
B1	796,52	793,39	797,95
A2	320,62	313,18	321,15
A3	1704,38	1700,36	1669,05
B3	1971,43	1981,62	1952,73
T3	3675,81	3687,17	3651,79
A4	2025,01	2028,43	2020,20
B4	2767,95	2783,78	2750,68
T4	4792,96	4759,55	4770,88
A5	1794,09	1793,21	1788,48
B5	2075,19	2045,24	2055,51
C5	461,84	465,39	463,45
T5	4331,12	4286,16	4307,44
A6	1794,09	1793,43	1788,48
B6	2075,19	2049,93	2055,51
T6	3869,28	3774,79	3843,99
C7	461,84	463,06	463,45
A8	4851,63	4869,14	4813,33
D8	1271,45	1275,52	1272,30
T8	6123,08	6005,82	6085,63
A9	4851,63	4772,60	4813,33
C9	461,84	466,06	463,45
D9	1271,45	1268,68	1272,30
T9	6584,92	6553,64	6549,08
A10	4851,63	4868,90	4813,33
C10	59,93	61,17	60,22
D10	467,62	465,88	465,84
E10	401,91	403,60	403,23
F10	401,91	406,50	403,23
T10	6183,01	6080,25	6145,85
A11	4851,63	4795,69	4813,33
D11	392,71	389,22	390,57
E11	401,91	399,00	403,23
F11	491,81	496,92	493,56
G11	59,93	59,72	60,22
T11	6197,99	6105,95	6160,90
F12	394,82	394,53	391,78
E13	401,91	403,86	403,23
F13	886,63	878,39	885,34
T13	1288,54	1275,87	1288,57
A14	4851,63	4831,74	4813,33
D14	392,71	395,16	390,57
G14	59,93	60,85	60,22
T14	5304,27	5313,50	5264,11

nm = não-medida

continuação da Tabela 3.6 - Resultados para o Caso da Síntese de Ácido Nítrico (kgmol/h)

Quantidade	Valor Real	Valor Medido	Valor Reconc.
A15	4531,01	4469,34	4492,18
D15	392,71	389,36	390,57
G15	59,93	59,40	60,22
T15	4983,65	5012,31	4942,96
A16	89,70	90,32	89,42
B16	103,76	102,90	102,78
T16	193,46	193,58	192,20
R1		nm	231,72
R2		nm	403,23
R3		nm	60,22

nm = não-medida

Tabela 3.7 - Dados para o Cálculo da Variância no Caso da Síntese de Ácido Nítrico

Corrente	Frações Molares dos Componentes das Correntes						
	A	B	C	D	E	F	G
T1	0	1,0000	0	0	0	0	0
T2	1,0000	0	0	0	0	0	0
T3	0,4641	0,5359	0	0	0	0	0
T4	0,4277	0,5723	0	0	0	0	0
T5	0,4160	0,4770	0,1070	0	0	0	0
T6	0,4659	0,5341	0	0	0	0	0
T7	0	0	1,0000	0	0	0	0
T8	0,7026	0	0	0,2074	0	0	0
T9	0,7372	0	0,0700	0,1927	0	0	0
T10	0,7852	0	0,0096	0,0756	0,0649	0,0647	0
T11	0,7832	0	0	0,0632	0,0648	0,0791	0,0097
T12	0	0	0	0	0	1,0000	0
T13	0	0	0	0	0,3123	0,6877	0
T14	0,9146	0	0	0,0741	0	0	0,0113
T15	0,9088	0	0	0,0791	0	0	0,0121
T16	0,4522	0,5478	0	0	0	0	0

#### e) *Circuito de Moagem*

O programa de reconciliação foi aplicado ao circuito de moagem, mostrado no fluxograma da Figura 3.6. Ele é constituído de quatro nós; dois deles são moinhos (nós 1 e 3), um é uma bomba (nó 2) e o outro é uma bateria de ciclones (nó 4). O material sólido é classificado em três faixas granulométricas, cujas frações são tratadas de forma análoga às espécies químicas de um sistema multicomponentes; os ciclones desempenham papel

semelhante ao de colunas de destilação simples e os moinhos, papel similar ao de um reator químico ao efetuar uma mudança nas quantidades das diferentes faixas granulométricas. Todavia, neste tipo de nó, como não é conhecida a "estequiometria" do processo, somente são consideradas a conservação dos sólidos totais e da água. Na bomba e na bateria de ciclones, além disto, levam-se em conta as restrições relativas à conservação de massa das faixas granulométricas distintas dos sólidos. A seleção das restrições que devem ser incluídas é feita através do vetor  $v_{ie}$ , conforme descrito no item 3.7.

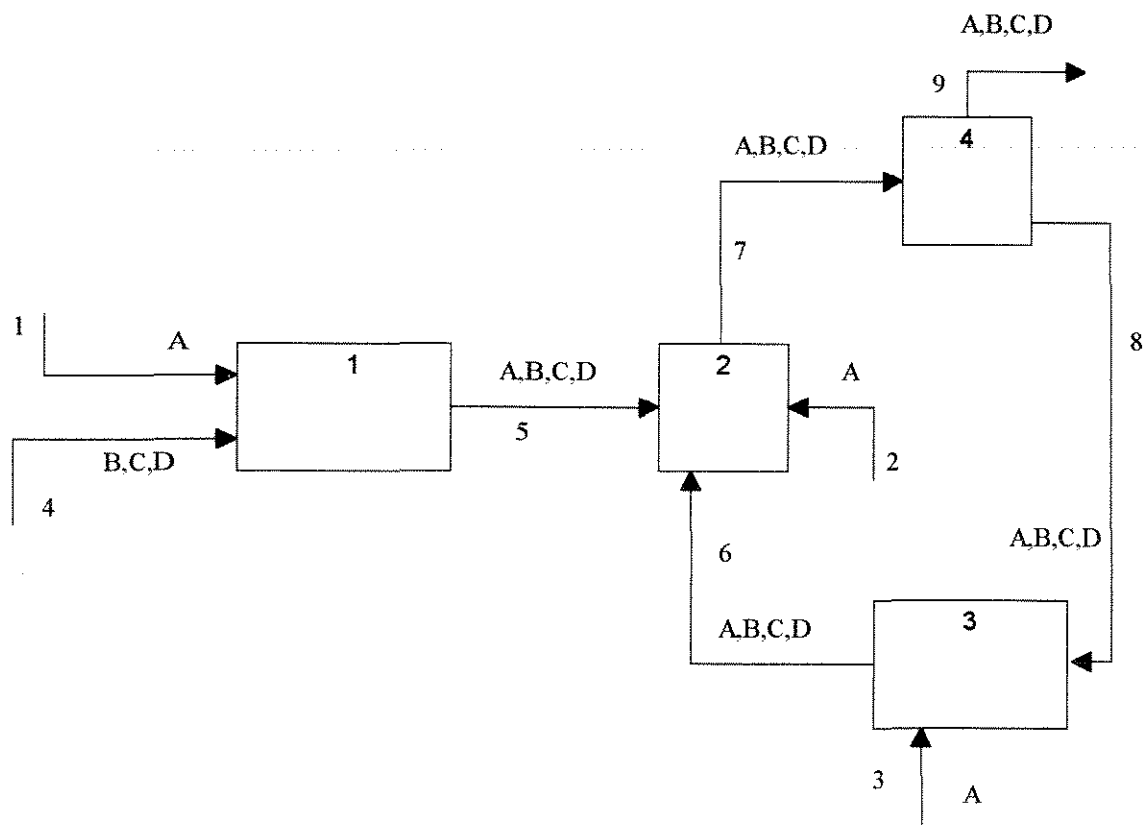


Figura 3.6 - Fluxograma do Circuito de Moagem

Este sistema é tratado como se tivesse 4 componentes, representando-se a água por A e os sólidos das diferentes faixas granulométricas por B, C e D; a vazão total é representada por T. As correntes 1, 2 e 3 contêm somente água; a corrente 4 contém somente sólidos, cujas frações não são medidas. As correntes 6, 7, 8 e 9 contêm água e sólidos, mas somente são medidas as frações dos últimos. Todas as vazões totais são medidas.

Utilizando valores reais das quantidades medidas deste circuito de moagem, dados por Serth et alii.(1987), transformando-os para adaptá-los ao caso linear e após a introdução de erros aleatórios, aplicou-se a reconciliação de dados.

A Tabela 3.8 apresenta os dados iniciais e os resultados da reconciliação, expressos em toneladas métricas. Verifica-se que há valores medidos que não sofreram ajustes e que há quantidades não-medidas que são não-observáveis. A reconciliação de dados levou à redução do erro relativo percentual médio (obtido através da média aritmética do módulo dos erros

relativos percentuais individuais), que é igual a 1,17% para os valores medidos, passando a igual a 0,80% para os valores reconciliados.

Considerou-se que o desvio-padrão das correntes é igual a 1% dos valores reais. As frações mássicas medidas de cada faixa granulométrica das correntes de sólidos, usadas no cálculo da matriz variância-covariância, são dadas na Tabela 3.9.

Tabela 3.8 - Resultados para o Circuito de Moagem (toneladas métricas/h)

Quantidade	Valor Real	Valor Medido	Valor Reconc.
A1	99,70	99,90	99,35
A2	320,00	314,12	314,64
A3	25,00	24,81	24,82
B4	nm	nm	no
C4	nm	nm	no
D4	nm	nm	no
T4	257,90	259,43	259,43
A5	nm	nm	99,35*
B5	24,28	24,20	24,29
C5	141,75	144,31	143,99
D5	91,90	93,06	92,87
T5	357,60	355,74	360,51
A6	nm	nm	272,02*
B6	11,43	11,49	11,40
C6	512,76	517,42	513,15
D6	313,52	315,38	311,12
T6	1109,40	1122,69	1107,70
A7	nm	nm	686,01*
B7	35,74	35,62	35,70
C7	654,58	647,59	657,14
D7	405,29	401,16	403,99
T7	1787,00	1795,92	1782,84
A8	nm	nm	247,20*
B8	35,57	35,63	35,56
C8	603,52	613,59	607,02
D8	183,53	196,59	196,83
T8	1084,30	1074,58	1086,61
A9	nm	nm	438,81*
B9	0,14	0,1386	0,14
C9	50,66	49,91	50,12
D9	207,09	205,62	207,17
T9	702,70	691,29	696,23

nm = não-medida

no = não-observável

\* = quantidades observáveis

Tabela 3.9 – Dados para o Cálculo da Variância no Caso do Circuito de Moagem

Corrente	Frações Mássicas das Faixas Granulométricas		
	B	C	D
T5	0,0679	0,3964	0,2570
T6	0,0103	0,4622	0,2826
T8	0,0200	0,3663	0,2268
T7	0,0328	0,5566	0,1831
T9	0,0002	0,0721	0,2947

Conclusões resultantes da aplicação deste programa:

O programa de reconciliação desenvolvido foi aplicado a cinco processos diferentes, procurando-se através deles estudar diferentes situações, cujos resultados acima descritos originaram as seguintes observações:

-em todos os casos estudados, onde foi possível calculá-lo, houve uma redução no erro relativo percentual médio após a reconciliação, ou, em outras palavras, o erro relativo percentual médio dos valores reconciliados é menor do que o dos valores medidos.

-os processos estudados incluem diferentes situações, tanto quanto ao tipo de restrições, como a ocorrência ou não de reações químicas, como quanto à obtenção ou não de ajustes para as quantidades medidas e de estimativas para as quantidades não-medidas, que são destacadas a seguir para cada um deles:

a) trata de sistema multicomponente, com quatro nós, quatro espécies químicas e sete correntes totais, onde ocorre uma reação química, contendo um divisor de fluxo cuja razão de divisão é conhecida. As quantidades medidas foram corrigidas e foram obtidas estimativas para todas as quantidades não-medidas;

b) trata de sistema multicomponente com um único nó, com sete componentes, onde ocorrem duas reações químicas, em que todas as quantidades não medidas foram observáveis, porém nem todas as quantidades medidas foram ajustadas;

c) trata de sistema com um único componente e com oito nós, contendo vinte correntes totais, onde todas são medidas e todas foram corrigidas;

d) trata de sistema multicomponente, com nove nós, sendo um deles um divisor de fluxo com razão de divisão conhecida, sete espécies químicas e dezesseis correntes totais, onde ocorrem três reações químicas, cada uma em um reator. Todas os valores medidos foram corrigidos e todas as quantidades não-medidas, que são as extensões das reações químicas, foram estimadas;

e) trata de sistema contendo quatro nós, onde as três faixas granulométricas de sólidos são consideradas como se fossem componentes, onde a "estequiometria" das mudanças granulométricas não é conhecida. Neste caso, nem todos os valores medidos foram ajustados e há valores não-medidos que não são observáveis.

Em vista disso, acredita-se que o programa de reconciliação seja bastante flexível quanto aos processos cujos dados poderá reconciliar, constituindo-se de ferramenta importante para análise de dados de processo.

---

## **CAPÍTULO 4. RETIFICAÇÃO DE DADOS DE PROCESSO**

## 4.1 Introdução

A retificação de dados pode ser vista como um conceito mais amplo do que a reconciliação de dados. Usualmente, a última se refere ao ajustamento de valores medidos de processo para compensar somente os erros aleatórios das medidas, pois se assume que os valores verdadeiros, envolvidos em balanços de massa e energia, são afetados somente por erros aleatórios. A retificação de dados, por outro lado, envolve não somente o ajuste de medidas com este tipo de erro, mas também com erros grosseiros, causados por eventos não-aleatórios como vazamentos, desvios do estado estacionário, mau funcionamento de instrumentos, por exemplo. Deve, normalmente, haver um número pequeno de erros grosseiros presentes num conjunto de dados, em comparação com erros aleatórios. Apesar disso, sua presença invalida a base estatística dos procedimentos de reconciliação, portanto é necessário identificá-los e eliminá-los do conjunto de medidas. Para isto, é feita uma análise baseada em medidas redundantes para verificar o fechamento das relações de balanço.

## 4.2 Os Dados de Processo e os Erros

Usualmente, o valor medido ( $\tilde{x}_j$ ) e o valor verdadeiro ( $x_j$ ) não são exatamente iguais e sua diferença é chamada de erro ( $e_j$ )

$$e_j = \tilde{x}_j - x_j$$

Os erros podem ser classificados em duas categorias distintas. Na primeira, estão os erros muito pequenos decorrentes de flutuações normais do processo, de desvios pequenos na amostragem e medição e variação aleatória na operação de um instrumento. Na maioria dos processos químicos, estas flutuações são uma pequena fração do total medido, então o resultado da medição é aproximadamente igual à quantidade física em questão. Estes erros são chamados de erros aleatórios.

Em contraste com esses pequenos erros, podem ocorrer erros grandes, também chamados erros grosseiros, que são devidos a eventos não-aleatórios, como desvios instrumentais não-compensados e mau funcionamento de aparelhos de medição, por exemplo. Além disso, também podem conduzir a erros de estimativa grosseiros, a representação inadequada ou imprecisa de um processo, envolvendo vazamentos e depósitos, e a inadequada consideração de desvios do estado estacionário. No caso de erros grosseiros, os valores medidos podem ter pouca relação com a quantidade física.

A presença de erros grosseiros invalida a base estatística do procedimento de reconciliação de dados, que considera que os erros são variáveis aleatórias, normalmente distribuídas, com média zero e matriz variância-covariância conhecida, positiva por definição. Além disso, um erro grosseiro presente em uma das medidas do conjunto que está sendo reconciliado causará uma série de pequenas correções em outras medidas. Esta distribuição de erros grosseiros tem duas desvantagens consideráveis: ela torna os dados imprecisos como um todo e mascara a fonte de erros, não fornecendo informações para conduzir ao reparo de instrumentos ou outras ações corretivas. Em vista disso, é aconselhável identificar e retirar as medidas que contêm erros grosseiros, para não prejudicar a confiabilidade das demais medidas e para tornar o processo de ajustamento de dados mais eficaz.



### 4.3 Tratamento de Dados de Processo Contendo Erros Grosseiros

Diversos autores relatam formas de tratar erros grosseiros, que envolvem a sua detecção e identificação. A maior parte dos critérios propostos se baseia em testes estatísticos, que se aplicam ao conjunto de dados como um todo ou a cada nó (testando o desbalanço nodal correspondente) ou ao ajuste de cada medida. Às vezes, são empregados em conjunto, utilizando-se mais de uma estatística, num mesmo problema de retificação de dados. Alguns testes são aplicados após a reconciliação de dados, que é seguida da remoção da medida suspeita, refazendo-se o procedimento de ajustamento de dados. Outros são efetuados antes de realizar a reconciliação de dados, o que é possível de acordo com a estatística empregada.

Ripps (1965) faz a remoção de medidas redundantes suspeitas, através de uma variável auxiliar, que é incluída na equação básica de ajustamento de dados. Ele não fornece um critério para selecionar essas medidas e aconselha que seja eliminada uma medida suspeita de cada vez, após o que é realizado o procedimento de reconciliação. Sugere que os valores da função-objetivo, assim obtidos, sejam usados para análise dos resultados. Relata que, frequentemente, o valor mínimo da função-objetivo coincide com o ajustamento mais correto. Todavia, acrescenta que a exatidão é melhor determinada por testes subsequentes, através da razoabilidade de outros parâmetros calculados a partir dos dados ajustados, pela consistência dos parâmetros de uma corrida para outra e recalibração de instrumentos suspeitos, por exemplo. A função-objetivo se refere ao processo como um todo e, também, portanto, a análise em questão.

Nogita (1972) introduz um critério para testar a inconsistência de dados e para detectar a existência de medidas errôneas e perturbações no processo, baseado em considerações estatísticas. Define uma função-teste, que é dada pelo somatório das diferenças entre o valor estimado e o valor medido correspondente, ponderado pelo inverso do desvio-padrão respectivo. Prova que esta função tem uma distribuição normal, com média zero e variância conhecida e testa a função normalizada (pela divisão por seu desvio-padrão), contra uma distribuição normal padronizada, com média zero e variância unitária. Por experiência própria, sugere que uma probabilidade de erro Tipo I (de rejeitar uma hipótese zero correta) de 10% é aceitável, em muitos casos. Então, de uma tabela de distribuição normal padronizada cumulativa, tem-se que a probabilidade da função normalizada estar compreendida no intervalo de  $-1,645$  a  $1,645$  é de 90%. Consequentemente, quando o módulo dessa função for  $> 1,645$ , pode-se dizer que a inconsistência é significativa com uma probabilidade de erro Tipo I de 0,10, prevendo um desbalanço sistemático, como o sistema em estado não-estacionário, com efluxo ou influxo inesperados ou erro grosseiro nas medidas. Para identificação de erros grosseiros, adota a eliminação seqüencial de valores medidos e compara o efeito de cada remoção no valor numérico da função normalizada. Define o conjunto consistente mais provável como aquele que confere o valor mínimo à dita função. De acordo com Mah et alii. (1986), como o critério é aplicado ao conjunto de dados, ou seja, uma única estatística é aplicada a todo o processo, ela pode ser ineficaz para detectar erros grosseiros em redes grandes. Além disso, pelo fato da reconciliação ser realizada antes da pesquisa de erro grosseiro, é dificultada sua identificação, devido ao espalhamento do erro grosseiro.

Almásy e Sztanó (1975) utilizam um teste estatístico para verificar a presença de erros grosseiros, através de uma função obtida pela soma dos quadrados dos defeitos do modelo, ponderada pelo inverso da matriz dispersão. Esta função tem uma distribuição qui-quadrado com número de graus de liberdade igual ao número de equações do sistema. Se for considerado um nível de significância suficientemente pequeno, então a probabilidade do valor

da função ultrapassar o valor crítico da distribuição qui-quadrado, sob condições normais é tão pequena, que pode-se inferir que alguma das hipóteses iniciais (média igual a zero ou a característica Gaussiana, por exemplo) não se mantém. Para identificar a fonte de erro grosseiro, supõem que todas as considerações sobre a distribuição dos erros são válidas, exceto o valor esperado zero para o erro de um único ponto de medida.

Mah et alii. (1976) observam que, antes de efetuar os cálculos de ajustamento de dados, é necessário testar a validade da hipótese, em que se supõe que todos os erros presentes nos dados são erros de medida aleatórios, com média zero e variância conhecida, pois a presença de erros sistemáticos invalida a base estatística do procedimento de reconciliação de dados. Com esse objetivo, propõem a construção de funções-teste, uma para cada nó, definidas pelo produto do desbalanço associado a cada nó pelo inverso da raiz quadrada da soma das variâncias das correntes associadas aquele nó. Sob a hipótese nula de que não há erro grosseiro, o valor esperado da função é zero. A função, assim definida, tem uma distribuição normal e sua variância é 1. Com base nessa função, podem ser aplicados testes a cada nó do gráfico de reconciliação, o que, afirmam, é uma vantagem importante quando comparada com funções que avaliam o conjunto de dados como um todo, sendo mais sensível a erros grosseiros do que essas. Relatam, ainda, que é importante considerar a intensidade do erro grosseiro, acusado no esquema de detecção. Através de testes de simulação, verificaram que grande parte das correntes falha no teste de detecção a níveis relativamente baixos de erros grosseiros. Por exemplo, erros grosseiros de até 20 % das vazões foram detectados em 65 % dos casos; quando elevados a 50 % das vazões, foram detectados em 85 % dos casos. Portanto, consideram que o teste parece ser efetivo para os casos estudados com uma probabilidade de 0,10, para erros do Tipo I. Descrevem um algoritmo para isolar e identificar erros grosseiros e vazamentos, que se baseia na aplicação da função-teste a cada nó isolado, considerando que todos os arcos são medidos e depois agrupando, progressivamente, pares de nós, trios de nós, etc, até identificar os arcos ruins (com este tipo de erro).

Hlaváček (1977), em artigo de revisão da literatura, relata que Václavek e Vosolsobé propõem um critério para tratar erros grosseiros, baseado em método heurístico. O critério, aplicado a cada nó individualmente, é definido por uma equação que representa o quociente entre o resíduo de balanço no nó e a vazão medida média correspondente. Quando o módulo do valor obtido for menor ou igual ao valor crítico (compreendido, frequentemente, entre 0,05 e 0,2), consideram que não há erro grosseiro e efetuam o ajustamento de dados pelos mínimos quadrados. Observam que a conclusão de ausência de erro grosseiro pode estar errada, pois pode haver cancelamento de erros, o que, todavia, não é comum. Relata, ainda, que Vosolsobé propõe, em outro trabalho, testar todas as duplas de nós conectadas por correntes internas para localizar a posição da fonte de erro.

Romagnoli e Stephanopoulos (1981) utilizaram uma função-teste dada pelo quadrado do vetor dos resíduos de balanço ponderado pela variância desses resíduos, para verificar a presença de erros grosseiros. Partem do pressuposto de que os erros de medida são normalmente distribuídos, com valor esperado igual a zero. Então, os resíduos das equações de balanço também são distribuídos da mesma forma e têm valor esperado igual a zero. Acrescentam que os erros sistemáticos causam o deslocamento da medida numa direção e as observações são distribuídas ao redor de um certo valor, diferente do valor verdadeiro do parâmetro. A diferença entre esses valores é o erro sistemático. A quantidade medida, neste caso, é o vetor dos resíduos de balanço, cujo valor teórico é conhecido e igual a zero. Considera-se que a incerteza do erro de medida é conhecida, então verifica-se se a média dos resíduos calculada difere significativamente de zero, de modo que a hipótese de média zero não se verifique. Como o vetor dos resíduos tem  $n$  elementos ( $n$  equações), a função-teste terá

uma distribuição qui-quadrado com  $n$  graus de liberdade, que deverá ser testada contra o valor tabelado correspondente, uma vez que seja pré-estabelecida a probabilidade de erro admissível, sendo aceitável em muitos casos o valor 0,10. Sugerem que a identificação de erros grosseiros seja feita através de análise estrutural, cujo procedimento é descrito. Propõem que, alternativamente, seja adotado o processamento sequencial das equações de balanço. Ele inicia com a aplicação do teste a uma equação de balanço, seguida da remoção (e teste) de uma medida de cada vez, quando o teste estatístico revelar a presença de erros grosseiros, com exceção daquelas medidas que foram consideradas sem erros grosseiros nas etapas anteriores. A seguir, vão-se adicionando as equações de balanço, uma a uma, à função-teste, aumentando da mesma forma o número de graus de liberdade da distribuição qui-quadrado tabelada.

Tamhane (1982), estudando o uso de resíduos (ajustes individuais das medidas) para detectar valores extremos em regressões lineares, mostra que um teste baseado no vetor dos resíduos transformado tem, na classe das transformações lineares do vetor dos resíduos, certas propriedades ótimas para revelar a presença de um único valor extremo, quando o observador não está ciente de que há exatamente um desses valores. Observa que é mais poderoso do que outros testes baseados no uso de resíduos, neste caso específico.

Mah e Tamhane (1982) tratam o problema de detecção e identificação da presença de um ou mais erros grosseiros em dados de processo, aplicado ao caso em que a reconciliação de dados é sujeita a restrições lineares. Consideram a situação em que todas as variáveis são medidas e que os erros grosseiros estão antes associados às medições do que a erros no modelo do processo. Observam que um teste simples, baseado nos ajustes ou resíduos (diferença entre o valor medido e o valor estimado) de cada variável, pode ser usado para detectar e identificar diretamente a fonte de erros grosseiros. Descrevem, ainda, um outro teste de medida baseado no vetor dos resíduos transformado (obtido pelo produto do inverso da matriz variância-covariância pelo vetor resíduo), que conforme Tamhane (1982) possui a máxima potência para detectar a presença de uma única medida com erro grosseiro. Este teste pode ser estendido ao caso de erros múltiplos, sem contudo ter assegurada suas propriedades de máxima potência.

Crowe et alii.(1983) tratam da verificação e identificação de erros grosseiros na reconciliação de vazões, em balanços multicomponentes, no estado estacionário. Consideram que toda medida de concentração e vazão é independente e normalmente distribuída, com média desconhecida e variância conhecida. Observam que embora o produto de duas variáveis normais independentes não seja normalmente distribuído, tal produto é aproximadamente normalmente distribuído, se as duas variáveis tiverem somente valores positivos e se seus coeficientes de variação (desvios-padrão relativos) forem suficientemente pequenos, isto é, menores do que 5%. Então, consideram que o vetor das medidas é uma amostra de uma distribuição normal multivariada com média desconhecida e matriz variância-covariância conhecida. Observam que medidas de diferentes correntes são, usualmente, tomadas como estatisticamente independentes, assim como medidas de concentração e de vazão total numa mesma corrente. Entretanto, as vazões dos componentes de uma mesma corrente não são independentes, pois a vazão total é comum. Utilizam três estatísticas-teste, sendo uma delas baseada na distribuição qui-quadrado, a qual é usada para detectar erros grosseiros, coletivamente, em um conjunto de combinações lineares dos erros nas equações do balanço reduzido. Observam que qualquer uma combinação linear desses erros pode ser testada contra uma distribuição normal e utilizam uma que testa o desbalanço particular em cada nó e outra que testa o ajuste particular de cada medida. Aconselham que se empreguem em conjunto essas três estatísticas, para obter tantas evidências quanto possível e observam que, em um

dos casos estudados, um dos testes não revelou a presença de erros grosseiros, enquanto os outros dois revelaram.

Madron (1985), estudando o problema de identificação de erros grosseiros em medidas na engenharia química, propõe uma abordagem, que, ao complementar outras existentes, permite reduzir o número de elementos do conjunto de medidas suspeitas, que deve ser testado na fase final da eliminação desse tipo de erro. Para isto, faz uso de informações disponíveis a priori, sobre a credibilidade da medida, que representa o valor máximo possível do erro grosseiro de cada medida. Sugere o uso do teste global qui-quadrado para detectar erros grosseiros e de três métodos de identificação da fonte de erros descritos na literatura, os quais, acredita, não diferem consideravelmente, quanto à eficiência de identificação. O estudo é restrito ao caso de desvios de um pressuposto modelo de erros, não contemplando erros devidos à representação inadequada do modelo matemático (como vazamentos, por exemplo) e considera a presença de um único erro grosseiro.

Iordache et alii.(1985) tratam da avaliação do desempenho do teste de medida para detecção de erros grosseiros em dados de processo, proposto por Mah e Tamhane (1982), através da avaliação de sua potência, que é a probabilidade de detectar e identificar, corretamente, a presença de erros grosseiros. Consideram que somente um erro grosseiro está presente nas medidas e exploram a influência de diferentes parâmetros na potência do teste. Observam que o efeito das restrições, da configuração da rede e da posição da medida com erro grosseiro podem ser adequadamente considerados através da matriz restrição transformada e que aqueles fatores que tornam as colunas (correntes) dessa matriz mais proporcionais tendem a reduzir as potências associadas a estas colunas e a torná-las umas iguais as outras, enquanto os fatores que as tornam menos proporcionais tendem a aumentar suas potências. Utilizam duas definições para a potência e observam que ela é profundamente afetada pela razão entre a magnitude do erro grosseiro e o desvio-padrão. Além disso, as potências também dependem da faixa e da distribuição dos desvios-padrão, quando eles são diferentes. A razão entre os desvios máximo e mínimo é uma medida do espalhamento das potências e quanto maior esta razão, maior a faixa das potências. Para uma mesma relação magnitude do erro grosseiro/desvio-padrão, quanto menor a magnitude do erro, menor a potência associada àquela medida.

Narasimhan e Mah (1987) descrevem um método para detectar e identificar erros grosseiros devidos a desvios instrumentais e vazamentos em processos químicos, no estado estacionário. Ele pode ser estendido a qualquer tipo de erro que possa ser modelado matematicamente. Como sub-produto ele permite a obtenção de estimativa da magnitude do erro grosseiro. Este método é chamado de método da razão de probabilidade condicional generalizada (GLR) e se baseia no teste estatístico da razão de probabilidade condicional. Permite a identificação de vários erros grosseiros, para o que utiliza uma estratégia baseada na compensação seriada de erros grosseiros. Esta forma de compensação necessita menor tempo de cálculo do que outras existentes, baseadas na eliminação seriada. Observam que o método GLR apresenta vantagens sobre outros métodos para identificação de erros grosseiros em processos químicos, no estado estacionário, já que somente ele possibilita diferenciar os vários tipos de erros. Comentam que isto é importante, pois nem sempre se pode estar certo de que o processo opera, estritamente, no estado estacionário. Para avaliar a performance do teste relatam experiências de simulação, utilizando duas medidas de desempenho. Observam que quando somente erros grosseiros devidos a desvios instrumentais estão presentes, então o teste GLR se reduz ao teste de medida e que ambos têm o mesmo desempenho, em condições idênticas.

Crowe (1988) apresenta um método recorrente para a identificação de erros grosseiros, baseado no prognóstico do efeito da remoção de qualquer conjunto determinado de medidas sobre testes estatísticos, sem efetuar reconciliações completas a cada vez. O objetivo é encontrar um ou mais conjuntos de medidas, cuja retirada fará com que sejam satisfeitos todos os testes estatísticos. A dedução apresentada leva diretamente ao cálculo da estatística de medida e à reconciliação. Mostra que a diminuição do valor da função-objetivo causada pela retirada de uma única medida é igual ao quadrado da estatística de medida de potência máxima correspondente, calculada antes da eliminação. Propõe um algoritmo para detectar os conjuntos suspeitos de erros grosseiros, cuja remoção conduz a valores aceitáveis de todos os testes estatísticos e vazões de processo, que é ilustrado com uma aplicação. Observa que o desempenho desse algoritmo precisa ser testado e comparado com outros publicados, a fim de testar sua habilidade para identificar corretamente erros grosseiros, evitando a ocorrência de erros do Tipo I e do Tipo II. A teoria se restringe a problemas lineares, no estado estacionário.

Narasimhan e Mah (1989) observam que a maioria dos testes estatísticos para identificar erros grosseiros propostos, até então, trata de modelos simples de processos, no estado estacionário. Neles, as variáveis são consideradas diretamente medidas ou não-medidas. Propõem técnicas para transformar, adequadamente, modelos gerais, no estado estacionário, em modelos simples na mesmas condições de operação. Consideram os casos em que variáveis de processo são medidas indiretamente e onde variáveis não-medidas estão presentes nas restrições. Numa situação em que ambos os casos estejam presentes, o modelo geral do processo é transformado no modelo simples em duas etapas. Inicialmente, eliminam-se as variáveis não-medidas e após são tratadas as variáveis indiretamente medidas. Então é possível aplicar qualquer um dos quatro testes estatísticos seguintes, previamente desenvolvidos: o teste global, o teste dos resíduos das restrições, o teste de medida e o teste da razão de probabilidade condicional generalizada.

Kao et alii.(1990) observam que uma das maiores desvantagens da maioria dos métodos propostos, na literatura, para detecção de erros grosseiros é que eles partem do pressuposto de que as observações são não-correlacionadas seriadamente (já que a independência implica em ausência de correlação), o que muitas vezes não é confirmado pelas evidências experimentais. Relatam que observações correlacionadas podem ser causadas por um número de fatores físicos associados com o tempo-morto do processo, com sua dinâmica e controle, bem como devido a fatores relacionados a instrumentos de medição. Nesta investigação, tratam de erros de medida correlacionados seriadamente, em processos químicos no estado estacionário. Entretanto, consideram que variáveis diferentes medidas num mesmo tempo ou em diferentes tempos não são correlacionadas, isto é, que não há correlação cruzada. Usam, como referência, o teste de medida e mostram que é bastante significativo o efeito do fato de negligenciar a existência de correlação. Sugerem duas soluções para este problema, que são avaliadas analiticamente e através de simulações em computador. Uma delas se baseia na obtenção da fórmula correta para a variância, que toma em consideração correlações seriadas e o outro envolve a filtragem para remoção de correlações seriadas dos dados de processo, obtendo resíduos não-correlacionados, aos quais podem ser aplicadas as técnicas desenvolvidas para a detecção de erros grosseiros em dados de processo independentes. Observam que os dois métodos são robustos, efetivos e simples de usar e que o último é mais poderoso no caso de vetores de dados individuais, mas exige mais tempo de computação.

Crowe (1989) estende o conceito de potência máxima, derivado por Tamhane (1982) para o teste de medida, ao teste dos resíduos da restrições de balanços lineares, no estado estacionário. O teste de potência máxima tem a maior probabilidade de encontrar corretamente um erro grosseiro em uma medida particular. Observa que a estatística de medida de máxima

potência de uma corrente de alimentação ou de produto, que corresponde a um vetor coluna unitário positivo ou negativo da matriz incidência dos valores medidos, é igual à estatística dos resíduos de potência máxima, para aquele balanço em que aquela corrente aparece. Apresenta dois exemplos, em que compara o teste dos resíduos convencional com o teste de potência máxima. Observa que o teste de potência máxima dos resíduos deve ser usado em conjunto com o teste de medida de potência máxima e com o teste qui-quadrado global. Conforme os exemplos dados, os testes por si só podem não possibilitar a identificação precisa da fonte de erros grosseiros.

#### 4.4 Retificação de Dados de Processo Através de Projeção Matricial

Crowe (1988) apresenta um método para prever o efeito da remoção de um conjunto de medições nos resultados dos testes estatísticos, sem efetuar a reconciliação completa, aplicável a um sistema linear. O objetivo é encontrar um ou mais conjuntos de medidas removidas de modo que as quantidades remanescentes satisfaçam os três testes estatísticos escolhidos. Os testes estatísticos são os mais frequentemente citados na literatura: o teste qui-quadrado global, o teste de desbalanço nodal e o teste de medidas.

A teoria desenvolvida se limita ao caso linear e considera, para simplificar a notação, que os balanços contêm somente quantidades medidas; as quantidades não-medidas poderiam ter sido previamente eliminadas se existissem.

O problema de reconciliação pode, então, ser definido como:

$$\min_a f = a^T \Sigma^{-1} a \quad (4.1)$$

sujeita à condição

$$B(\bar{x} + a) = 0$$

onde  $\bar{x}$  é o vetor das quantidades medidas ( $q \times 1$ ), "a" é o vetor dos ajustes correspondentes e B é uma matriz ( $p \times q$ ), de característica p, que contém a matriz incidência dos balanços das espécies, acrescida, possivelmente, de linhas referentes a restrições de igualdade adicionais. No caso de sistemas com quantidades não-medidas corresponde ao produto  $Y^T B_1$ .

Considera-se que as quantidades medidas têm uma matriz variância-covariância conhecida ou estimada  $\Sigma$ , positiva por definição.

A solução pode ser escrita como:

$$a = -\Sigma B^T H^{-1} e \quad (4.2)$$

onde,  $e = B\bar{x}$  é o vetor dos desbalanços e  $H = B \Sigma B^T$  é uma matriz positiva por definição.

A partir da fórmula para a variância de uma combinação linear de variáveis aleatórias conjuntamente distribuídas, se o valor esperado de "e" é zero e a variância das medidas é  $\Sigma$ ,

então, a variância de "e" é H. Substituindo o valor de "a" dado pela eq.4.2, o valor da função-objetivo, na solução é

$$f = e^{-T} H^{-1} e \quad (4.3)$$

Além disso e de acordo com Scheffe, ela tem uma distribuição qui-quadrado. Esta é a base para o teste qui-quadrado global e também para a premissa que diz que se todas as medidas contendo erros grosseiros tiverem sido eliminadas, o valor da função-objetivo, após a remoção, estará abaixo do valor qui-quadrado tabelado com nível de confiança e graus de liberdade apropriados. Entretanto, se o teste qui-quadrado for satisfeito, não significa que não haja erro grosseiro, pois pode haver um erro grosseiro entre medidas aleatórias de um grande conjunto. Por isto, empregam-se outros testes para diagnosticar, individualmente, as medidas e os desbalanços das espécies.

Para verificar o efeito da remoção de um conjunto particular de l medidas na função-objetivo, separa-se a matriz B em

$$B = [B' : B''] \quad (4.4)$$

em que as l colunas de B'' correspondem aos valores removidos. Considera-se que a característica de B'' é l (< p). Se esses valores removidos fossem efetivamente não-medidos, de acordo com Crowe (1983), definiria-se a matriz R<sup>T</sup>, (p-l) x p por

$$R^T B'' = 0 \quad (4.5)$$

Então, a solução para o caso com remoções é obtida substituindo B por R<sup>T</sup>B e "e" e H por

$$e_r = R^T e \quad (4.6)$$

e

$$H_r = R^T H R \quad (4.7)$$

respectivamente. Então, a variação na função-objetivo f é

$$\begin{aligned} \Delta f &= e_r^T H_r^{-1} e_r - e^T H^{-1} e \\ &= -e^T M e \end{aligned} \quad (4.8)$$

com

$$M = [H^{-1} - R H_r^{-1} R^T] \quad (4.9)$$

Esta matriz, expressa em termos de quantidades disponíveis, pode ser escrita:

$$M = H^{-1} B'' G_1^{-1} B''^T H^{-1} \quad (4.10)$$

onde a matriz G<sub>1</sub>

$$G_1 = B''^T H^{-1} B'' \quad (4.11)$$

A redução na função-objetivo, usando as eq.4.8 e 4.10, pode ser escrita como

$$\Delta f = -(e^T H^{-1} B'') G_1^{-1} (B''^T H^{-1} e) \quad (4.12)$$

Se for adicionada a  $B''$  uma outra medida removida, cuja coluna  $b$  é dependente das colunas de  $B''$ , a matriz  $R$  não mudará, pois ela já satisfaria  $R^T B'' = 0$ . Então, nem  $e$ , nem  $H$ , mudariam, pois a adição de uma coluna dependente a  $B''$  deixa  $M$  e, conseqüentemente, a função-objetivo inalterada.

No caso em que uma única medida é removida, a eq.4.12 fica

$$\delta f = -(b^T H^{-1} e)^2 / (b^T H^{-1} b) \quad (4.13)$$

onde  $b$  é uma única coluna de  $B$ , correspondente à medida removida.

A eq.4.12 permite prever a redução na função-objetivo resultante de cada seqüência de remoções: uma única de cada vez, duas de uma vez e assim por diante. Qualquer conjunto de valores, cuja remoção não reduz suficientemente a função-objetivo, de forma que ela satisfaça o teste qui-quadrado, não pode conter todos os erros grosseiros. Entre os conjuntos de  $l$  remoções, aqueles que satisfizeram o teste qui-quadrado serão examinados posteriormente para verificar qual deles satisfaz, também, o teste do desbalanço e o de medida.

Como verificação final, calculam-se as vazões, incluindo aquelas não-medidas, de cada conjunto que satisfaz os três testes estatísticos, para descartar aqueles que contêm vazões negativas. Qualquer conjunto que, ao ser removido, leva a vazões negativas indica que há uma imperfeição nas medidas remanescentes e que é improvável que todos os erros grosseiros corretos tenham sido removidos, considerando variâncias razoáveis. Vazões negativas podem surgir se as variâncias atribuídas àquelas vazões, especialmente às menores, forem muito grandes.

Se nenhum dos conjuntos, de tamanho  $l$ , satisfizer todos os testes estatísticos e não conduzir a vazões negativas, então devem ser removidas medidas adicionais, se possível.

Um teste geral do desbalanço, já que a variância do vetor desbalanço " $e$ " é  $H$  e a variância de qualquer combinação linear  $w^T e$  é  $w^T H w$ , é dado por

$$z_e(w) = w^T e / (w^T H w)^{1/2} \quad (4.14)$$

Se se considerar que " $e$ " é normalmente distribuído, com valor esperado zero, então  $z_e(w)$  tem média zero e variância unitária. Usualmente,  $w$  deveria ser um vetor unitário, mas isto não é necessário.

Após a remoção de  $l$  medidas, o teste do desbalanço pode ser escrito para qualquer combinação linear  $v$  como

$$\begin{aligned} z_{e_r}(v) &= v^T e_r / (v^T H_r v)^{1/2} \\ &= v^T R^T e / (v^T R^T H R v)^{1/2} \end{aligned} \quad (4.15)$$



Esta equação é muito parecida com a eq.4.14, exceto que  $w$  é substituído por  $Rv$ , onde  $v$  é um vetor unitário. Então, a aplicação do teste de desbalanço para o caso de remoção é direta, sem a necessidade de fazer a reconciliação, embora seja preciso obter a matriz  $R$ .

A matriz  $R_2$  para uma remoção adicional de uma medida correspondente à coluna  $b$  de  $B'$ , pode ser obtida a partir de uma  $R_1$ , anterior àquela remoção (I no caso de não ter havido nenhuma), através de

$$R^T = [c_1 : -\gamma I] R_1^T \quad (4.16)$$

onde

$$c = [-\gamma : c_1^T]^T = R_1^T b \quad (4.17)$$

com  $\gamma \neq 0$ . Se o primeiro elemento de "c" for zero, as colunas do termo entre colchetes da eq.4.16 devem ser permutadas de modo que  $c_1$  seja colocado na posição ocupada por uma escolha arbitrária de  $\gamma \neq 0$  em  $c$ .

De modo similar, e considerando que o desbalanço é normalmente distribuído com média zero e variância unitária  $H$ , o vetor dos ajustes "a", como mostrado anteriormente por Mah e Tamhane (1982) e por Crowe (1983), tem média zero e matriz variância dada por

$$Q = \Sigma B^T H^{-1} B \Sigma \quad (4.18)$$

de modo que "a" pode ser testado contra uma normal univariada através de

$$z_{a,j} = u_j^T a / (u_j^T Q u_j)^{1/2} \quad (4.19)$$

onde  $u_j$  é vetor unitário  $j$ -ésimo, isto é, a  $j$ -ésima coluna da matriz identidade.

Mah e Tamhane (1982) mostraram que um teste de potência máxima para as medidas, antes de qualquer remoção, pode ser definido por

$$\begin{aligned} z_{a,j}^* &= u_j^T \Sigma^{-1} a / u_j^T \Sigma^{-1} Q \Sigma^{-1} u_j)^{1/2} \\ &= -b_j^T H^{-1} e / (b_j^T H^{-1} b_j)^{1/2} \end{aligned} \quad (4.20)$$

onde  $b_j = B u_j$ , uma coluna particular de  $B$ .

Almásy e Sztanó (1975) definiram esta mesma estatística e observaram esta propriedade para o valor que contém erro grosseiro.

Após a remoção, a estatística equivalente para uma medida remanescente pode ser dada por

$$z_{ar,j}^* = -b_j^T R H_r^{-1} R^T e / (b_j^T R H_r^{-1} R^T b_j)^{1/2} \quad (4.21)$$

A diferença entre os numeradores das eq. 4.21 e 4.20 é

$$\begin{aligned}\Delta(\text{num}) &= b_j^T [H^{-1} - R H_r^{-1} R^T] e \\ &= b_j^T M e\end{aligned}\quad (4.22)$$

e, similarmente, a variação no quadrado do denominador é

$$\Delta(\text{denom}^2) = -b_j^T M b_j \quad (4.23)$$

Então, o teste de medida para qualquer dos valores não-removidos pode ser calculado de forma rápida através da eq.4.10 para obter M. Observa-se que a estatística dada pela eq.4.21 poderia ser indefinida para quaisquer dos valores removidos devido às eq. 4.4 e 4.5.

Crowe mostra que, dada uma reconciliação com um conjunto particular de medidas, a redução na função-objetivo, após a remoção de qualquer uma delas, é exatamente igual ao quadrado da estatística de medida de potência máxima correspondente antes de sua remoção. Então, para qualquer medida numa dada reconciliação, comparando a estatística de medida de potência máxima da eq.4.20 com a variação na função-objetivo resultante da remoção daquela medida da eq.4.13, com  $b = b_j$ , encontra-se

$$\delta f = -(z_{s,j}^*)^2 \quad (4.24)$$

Então, dada a reconciliação, pode-se saber a partir das estatísticas de medida quais remoções individuais, se houver, levarão a uma redução suficientemente grande na função-objetivo para torná-la menor do que o valor qui-quadrado tabelado pertinente.

Assim, após a predição do efeito da remoção de qualquer conjunto particular de medidas sobre as estatísticas de medida das remanescentes, a redução adicional na função-objetivo causada por mais uma remoção é similarmente o quadrado da estatística de medida correspondente, antes de sua remoção.

Para calcular as estatísticas de medida após uma remoção adicional, a partir de um conjunto que já teve medidas removidas, considera-se que a função-objetivo e as estatísticas de medida tenham sido calculadas para a remoção de  $l$  medidas, correspondentes às colunas de  $B'$ . A estatística de medida de cada quantidade remanescente é facilmente determinável, após a remoção adicional do valor correspondente à coluna  $b_q$  de  $B'$ .

Para expressar a mudança no numerador e no quadrado do denominador da estatística de medida para a remoção adicional de  $b_q$ , a forma das equações é preservada, porém se substitui "e" por  $e_q$  (da eq.4.6) e H por  $H_r$  (da eq.4.7). A inversa de  $H_r$ , nas equações para estatísticas de medida e ajustes, é substituída por

$$N \equiv R H_r^{-1} R^T \quad (4.25)$$

conforme a eq. 4.21 comparada com a eq.4.20. Então, a derivação para obter a equação para M pode ser feita de forma análoga para dar, com as medidas de  $B'$  previamente removidas,

$$M_n = N b_q b_q^T N / (b_q^T N b_q) \quad (4.26)$$

para a remoção adicional do valor correspondente à  $b_q$ .

A mudança no numerador de  $z_{ar,j}^*$  é, então, por analogia à eq.4.22,

$$\delta(\text{num}_j) = b_j^T M_n e \quad (4.27)$$

que, com a eq.4.26, torna-se

$$\delta(\text{num}_j) = (b_j^T N b_q) (b_q^T N e) / (b_q^T N b_q) \quad (4.28)$$

Similarmente, a partir das eq.4.23 e 4.26,

$$\delta(\text{denom}_j^2) = -(b_j^T N b_q)^2 / (b_q^T N b_q) \quad (4.29)$$

Então, a partir das eq.4.21 e 4.25,

$$z_{ar,j}^* = -(b_j^T N e) / (b_j^T N b_j)^{1/2} \quad (4.30)$$

antes da remoção de  $b_q$ , de modo que o novo valor da estatística de medida é

$$z_{an,j}^* = (z_{ar,j}^* - \Gamma_{n,jq} z_{dr,q}^*) / (1 - \Gamma_{n,jq}^2)^{1/2} \quad (4.31)$$

com

$$\Gamma_{n,jq} = (b_j^T N b_q) / [(b_j^T N b_j) (b_q^T N b_q)]^{1/2} \quad (4.32)$$

Tendo  $M$  e  $H^{-1}$  já calculados, a matriz  $N$  também é disponível a partir das eq.4.9 e 4.25 como

$$N = H^{-1} - M \quad (4.33)$$

de modo que a nova estatística de medida pode ser obtida facilmente.

Usando as eq. 4.2 e 4.9, a mudança no ajuste da  $j$ -ésima medida remanescente, após a remoção das colunas de  $B^n$  é

$$\Delta a_j = u_j^T \Sigma B^T M e \quad (4.34)$$

de modo que a variação adicional em  $a_j$ , causada por uma remoção posterior do valor representado pela coluna  $b_q$  é

$$\delta a_j = u_j^T \Sigma B^T M_n e \quad (4.35)$$

$$\begin{aligned} &= u_j^T \Sigma B^T N b_q (b_q^T N e) / (b_q^T N b_q) \\ &= -u_j^T \Sigma B^T N b_q z_{ar,q}^* / (b_q^T N b_q)^{1/2} \end{aligned} \quad (4.36)$$

A mudança na função-objetivo devida à remoção de uma única medida, como já havia sido mostrado na eq.4.24, é igual ao quadrado da estatística de medida correspondente

anterior àquela remoção. Então, a mudança adicional na função-objetivo, de acordo com a eq.4.30, causada pela remoção posterior da medida referente à coluna  $b_q$  é

$$\begin{aligned}\delta f &= -e^T M_n e \\ &= -(b_q^T N e)^2 / (b_q^T N b_q) \\ &= -(z_{ar,q}^*)^2\end{aligned}\tag{4.37}$$

#### 4.5 Bases do Procedimento Computacional Para Retificação de Dados e Coaptação

Desenvolveu-se um procedimento computacional, que utiliza projeção matricial, para retificação de dados de processo, que envolve a obtenção de estimativas para as quantidades medidas e de estimativas para as quantidades não-medidas observáveis, cuidando de detectar e identificar a presença de erros grosseiros. É aplicável a sistemas no estado estacionário, que são sujeitos a restrições lineares, incluindo balanços de massa multicomponentes, havendo ou não reações químicas. Porém, considera-se que, quando a concentração de um componente de uma corrente é medida, a vazão total daquela corrente também é medida.

As considerações quanto à inclusão das vazões totais são as mesmas descritas no item 3.6, bem como as referentes aos dados de entrada, incluindo os vetores  $v_{iv}$  e  $v_{ie}$ , os dados sobre reações químicas, restrições de corrente adicionais, etc.

Inicialmente, é feita a reconciliação de dados para obter as estimativas dos valores medidos, conforme descrito no item 3.6. A seguir, aplicam-se três testes estatísticos para detectar e identificar possíveis fontes de erros grosseiros, que são o teste global, o teste do desbalanço nodal e o teste de medida. Utilizam-se três testes, pois de acordo com vários pesquisadores é aconselhável o uso de mais de um teste estatístico para melhorar a possibilidade de detectar e de identificar erros grosseiros, corretamente.

Conforme Mah (1990), para aplicação dos dois primeiros não é necessário efetuar a reconciliação, porém eles somente detectam a presença de erros grosseiros, exigindo um procedimento adicional para a identificação da(s) fonte(s) de erro(s) grosseiro(s); enquanto o último exige a reconciliação prévia das quantidades medidas, porém possibilita a identificação direta da(s) fonte(s) deste tipo de erros.

Etapas do procedimento de retificação de dados:

- i) inicialmente, faz-se a reconciliação de dados de processo, reduzindo o problema de reconciliação, através de projeção matricial, quando houver quantidades não-medidas, seguida da determinação de estimativas para os valores não-medidos observáveis;
- ii) calculam-se a estatística global, através da eq.4.3 e as estatísticas de medida das quantidades medidas, através da eq.4.20;
- iii) compara-se o valor da estatística global, com o valor tabelado da função qui-quadrado ( $\chi_{0,05}^2$ ) com  $p$  graus de liberdade (ou a característica de  $Y$ ) e nível de confiança igual

a 0,95; comparam—se as estatísticas de medida, usando a fórmula de Mah e Tamhane (1982) para relacionar o nível de confiança de uma estatística individual com o global, através de

$$\alpha^* = 1 - (1 - \alpha)^{1/s'} \quad (4.38)$$

onde,  $\alpha$  é a probabilidade de rejeitar erradamente a hipótese nula (de que não há erro grosseiro), escolhido como 0,05, enquanto  $\alpha^*$  é o nível de probabilidade correspondente para cada estatística. O inteiro  $s'$  ( $\leq$  número de vazões medidas) representa o número de estatísticas de medida com valores absolutos distintos. Então, uma estatística de medida será considerada muito grande para ser aleatória se exceder o valor crítico ( $z_c$ ) dada pela distribuição normal padronizada. Por exemplo, a um nível de confiança global de 95% e quando há duas estatísticas de medida com valores absolutos distintos,  $\alpha=0,05$  e  $z_c$  é igual a 2,235, sendo tomada no ponto  $\alpha^*/2$  superior da Distribuição Normal Padronizada. Aqui, a estatística de medida, cujo valor absoluto for  $\leq 0,01$  do que uma estatística calculada anteriormente, será marcada através do vetor  $pvza[j]$ , onde  $j=1, \dots, q$ , para evitar que se avalie, sem necessidade (Crowe 1989a), os efeitos de sua remoção, tanto na função-objetivo, como nas estatísticas de medida e do desbalanço;

iv) se o valor da  $f < \chi_{0,05}^2$  e se  $z_{a,j}^* < z_c$  para  $j=1, \dots, q$ , calcula-se a estatística do desbalanço ( $z_c$ ), através da eq.4.14; se  $z_c < z_c$ , então não há indícios de erros grosseiros, devendo-se verificar se algum dos valores não-medidos observáveis é inaceitável (negativo). Aqui,  $z_c$  é o valor da distribuição normal padronizada, a nível de confiança global de 95%, obtido conforme descrito no item iii, considerando que  $s'$  é o número de estatísticas de desbalanço distintas (sendo  $s' \leq p$ );

v) se o valor da  $f > \chi_{0,05}^2$  e/ou se alguma das estatísticas de medida for maior do que o valor crítico e/ou, ainda, se o valor de alguma das estatísticas de desbalanço ( $z_c$ ) for maior do que o valor crítico pertinente, verifica-se se há uma única medida com erro grosseiro;

vi) calculam-se novos valores para a função-objetivo para verificar o efeito da remoção de medidas não-marcadas, através da eq.4.24 e de seu valor inicial, calculado anteriormente. Os valores obtidos são comparados com  $\chi_{0,05}^2$  tabelado, com  $p-1$  graus de liberdade, e os resultados são indicados através do vetor  $J1(j,0)$ , onde  $j=1, \dots, q$ , sendo  $q$  o número de quantidades medidas. A uma dada medida eliminada corresponde o valor zero, quando sua remoção acarretar um valor para a função-objetivo menor que o tabelado e o valor 1 no caso contrário;

vii) calculam-se, através das eq.4.22 e 4.23, as estatísticas de medida das quantidades remanescentes, após a eliminação individual de cada medida não-marcada, que são comparadas com os valores críticos pertinentes; coloca-se o número de estatísticas cujo valor absoluto é maior do que o valor crítico, no vetor  $ZAM(j,0)$ , onde  $j=1, \dots, q$ ; portanto, cada elemento corresponde a uma medida removida;

viii) verifica-se, se a remoção individual de cada medida conduziu simultaneamente a valores satisfatórios para as estatísticas global e de medida através dos vetores  $J1(j,0)$  e  $ZAM(j,0)$ ; os resultados são colocados no vetor  $ZJ1(j,0)$ , onde  $j=1, \dots, q$ , que conterá zero no elemento correspondente à medida removida que conduz a resultados satisfatórios e 1, no caso contrário. Aqui, também registram-se através da matriz  $VZDA(i,j)$ ,  $q \times q$ , quais as medidas

remanescentes que têm valor absoluto  $\leq 0,01$  do que uma estatística calculada anteriormente, para evitar a avaliação desnecessária dos efeitos de sua remoção;

ix) calcula-se a estatística de desbalanço após a remoção de uma medida, através da eq.4.15, para os casos em que ela corresponde a um elemento 0 do vetor  $ZJ1(j,0)$ ; comparam-se os resultados das estatísticas de cada caso com o valor crítico e conta-se o número de estatísticas de desbalanço maiores do que o crítico, colocando o resultado no vetor  $ZEM1(0,j)$ , onde  $j=1, \dots, q$ ;

x) verifica-se quais as remoções individuais, se houver, que conduzem todas as três estatísticas a valores satisfatórios e, para estes casos (e para as medidas marcadas correspondentes, se for o caso), calculam-se os valores corrigidos das quantidades medidas remanescentes após uma remoção, através da eq.4.34, bem como os valores das quantidades não-medidas observáveis; os resultados devem ser analisados para verificar se são aceitáveis (se não há valores negativos);

xi) quando nenhum elemento de  $ZEM1(0,j)$  for igual a zero, significa que as três estatísticas não foram simultaneamente aceitas em nenhum dos casos, então calculam-se novos valores da função-objetivo após a remoção de pares de quantidades medidas, através da eq.4.37 e dos seus valores após remoções individuais; comparam-se os valores obtidos com  $\chi^2_{0,05}$  tabelado, agora com  $p-2$  graus de liberdade, e conta-se o número de pares ( $n_{par}$ ) que fornecem valores satisfatórios, registrando-se quais são as medidas removidas correspondentes através da matriz  $PAR$  ( $n_{par} \times 2$ ); se  $n_{par}=0$ , então há indícios de que existe mais de dois erros grosseiros;

xii) se  $n_{par} > 0$ , calculam-se, através da eq.4.31, as estatísticas de medidas das quantidades remanescentes, após a remoção de cada par de medidas indicado pela matriz  $PAR$ ; comparam-se os valores obtidos com o valor crítico pertinente, colocando-se no vetor  $ZAM2(0,j)$ , onde  $j=1, \dots, n_{par}$ , o número de estatísticas maiores que o valor crítico;

xiii) se  $ZAM2(0, j)=0$ , calculam-se as estatísticas de desbalanço correspondentes, através da eq.4.15, e verifica-se se os valores são satisfatórios, contando o número de estatísticas maiores que o valor crítico, colocando os resultados no vetor  $ZEM2(0, j)$ , onde  $j=1, \dots, n_{par}$ ; quando  $ZEM2(0, j)=0$  calculam-se os valores corrigidos das quantidades medidas remanescentes após duas remoções, através da eq.4.36, bem como os valores das quantidades não-medidas observáveis, incluindo, aqui, os pares que contêm medidas marcadas. Os resultados devem ser analisados para verificar se são aceitáveis (se não há valores negativos);

xiv) se nenhum  $ZAM2(0,j)=0$  e/ou se nenhum  $ZEM2(0,j)=0$ , há indícios de ocorrência de mais de dois erros grosseiros;

#### 4.6 O Programa de Retificação de Dados de Processo

Este programa, designado Programa RETIF, foi desenvolvido de acordo com o descrito nos 3.5 e 3.6, que tratam da parte de reconciliação de dados e nos itens 4.4 e 4.5, que tratam da parte de retificação de dados. Ele pode ser usado para a obtenção de valores corrigidos para as quantidades medidas, que obedeçam às restrições do processo, e de estimativas para as quantidades não-medidas observáveis, investigando a presença e eliminando erros grosseiros.

Este programa se destina à obtenção de estimativas para quantidades medidas e para quantidades não-medidas de sistemas no estado estacionário, sujeitos a restrições lineares, na presença de até dois erros grosseiros. É aplicável a sistemas com um único ou vários componentes, prevendo a ocorrência de reações químicas, com coeficientes estequiométricos definidos, e a inclusão de restrições de corrente adicionais, como a razão de divisão conhecida de um divisor de fluxo.

Faz uso simultâneo de três estatísticas, que são uma estatística global, baseada na distribuição qui-quadrado, e de estatísticas de desbalanço e de medida, testadas contra uma distribuição normal padronizada, para detectar e identificar a ocorrência de possíveis erros grosseiros.

Este programa foi escrito em linguagem Turbo C e sua listagem é dada no Anexo D.

As observações quanto aos dados de entrada, bem com obtenção da matriz variância-covariância, são idênticas às do Programa RECON, dadas no item 3.6, portanto não são aqui repetidas.

O programa inclui valores tabelados para a distribuição qui-quadrado, para o nível de confiança de 95%, com até 50 graus de liberdade, extraídos de Anderson (1978) e Miller et alii.(1990) ou obtidos por interpolação linear (valores referentes a 41 até 50 graus de liberdade). As estatísticas de medida e de desbalanço foram testadas contra um nível de confiança global de 95% (cujo valor crítico é  $z_c = 1,96$  para  $\alpha = 0,05$ , que corresponde à área sob a curva igual a  $1-\alpha/2$ , conforme tabela da Função Distribuição Normal Padronizada, contida em Anderson (1978). A partir do que foram calculados os níveis de confiança individuais, conforme descrito no item 4.5, retirando-se da mesma tabela os valores correspondentes para até 50 estatísticas distintas.

#### 4.7 Aplicações do Programa RETIF

Para testar a operacionalidade deste programa foram estudados os resultados de vários casos relatados a seguir, incluindo um sistema de síntese de amônia, um de ácido nítrico e um circuito de moagem.

##### a) Processo de Síntese de Amônia

O processo de síntese de amônia (SA) é o mesmo descrito no item 3.8. Partindo-se dos valores medidos dados na Tabela 3.1, foram gerados casos, colocando erros em algumas quantidades, cujos valores obtidos foram tomados com duas casas decimais:

- Caso SA1: adicionou-se 10% na vazão de B1;
- Caso SA2: adicionou-se 20% na vazão de A2;
- Caso SA3: adicionou-se 20% na vazão de B5 e 20% em T5;
- Caso SA4: adicionou-se 10% na vazão de A1 e 20% em C4;
- Caso SA5: adicionou-se 20% na vazão de A2, B2 e T2;

A Tabela 4. 1 apresenta os valores resultantes da aplicação do RETIF aos casos SA1, SA2 e SA3 do sistema de síntese de amônia. A seguir são feitas algumas observações sobre eles, onde a letra **J** se refere à função-objetivo (que equivale à estatística do teste global), **JT** ao valor tabelado pertinente da distribuição qui-quadrado e **v** indica o número de graus de liberdade.

Caso SA1:

-Função-objetivo:  $J(\text{todos valores medidos}) = 27,89 > JT(v=8) = 15,51$

-Estatística de medida: Há uma única estatística de medida cujo valor absoluto é maior do que o crítico  $z_c$  (11 estatísticas diferentes) = 2,835:

$$z_a(B1) = -4,82$$

-A única medida, cuja remoção causa uma redução suficiente na função-objetivo (**J1**), é **B1**:

$$J1(B1 \text{ removida}) = 4,62 < JT(v=7) = 14,07$$

-A remoção de **B1** faz com que todos os valores absolutos das estatísticas de medida das quantidades remanescentes sejam inferiores ao valor crítico.

-Todos os valores absolutos das estatísticas de desbalanço são inferiores ao crítico, após a remoção de **B1**.

-Conclusão: foi detectado e identificado corretamente o erro grosseiro da quantidade **B1**, através dos três testes estatísticos.

Caso SA2:

-Função-objetivo:  $J(\text{todos valores medidos}) = 74,67 > JT(v=8) = 15,51$

-Estatística de medida: Há três estatísticas de medida cujo valor absoluto é maior do que o crítico  $z_c$  (12 estatísticas diferentes) = 2,865:

$$Za(A2) = -8,45$$

$$Za(T2) = 3,67$$

$$Za(A3) = 7,20$$

-Porém, a única medida cuja remoção causa uma redução suficiente na função-objetivo (**J1**) é **A2**:

$$J1(A2 \text{ removida}) = 3,36 < JT(v=7) = 14,07$$



Tabela 4.1 – Resultados da Aplicação do RETIF aos Casos SA1, SA2 e SA3 da Síntese de Amônia (quantidades em kgmol/s)

Quantidade	SA1	SA2	SA3 (T2 e T3)**	SA3 (B5 e T5)**
A1	32,06	32,09	32,08	32,19
B1	96,01**	96,19	96,97	96,35
D1	0,40	0,40	0,40	0,40
T1	128,46	128,68	129,45	128,94
A2	100,96	99,75**	101,62	101,04
B2	294,66*	295,44*	340,65*	292,45*
D2	19,82	19,87	19,85	19,81
T2	415,44	415,07	462,11**	413,30
A3	70,31	69,04	70,95	70,26
B3	202,71*	203,32*	248,65*	200,09*
C3	61,30*	61,42*	61,33*	61,57*
D3	19,82*	19,87*	19,85*	19,81*
T3	354,14	353,65	400,79**	351,73
C4	61,30	61,42	61,33	61,57
A5	70,31*	69,04*	70,95*	70,26*
B5	202,71	203,32	248,65	200,09**
D5	19,82*	19,87*	19,85*	19,81*
T5	292,84	292,23	339,46	290,17**
A6	1,41*	1,38*	1,42*	1,41*
B6	4,05*	4,07*	4,97*	4,00*
D6	0,40*	0,40*	0,40*	0,40*
T6	5,86*	5,84*	6,79*	5,80*
A7	68,90*	67,66*	69,53*	68,85*
B7	198,65*	199,25*	243,68*	196,09*
D7	19,42*	19,48*	19,45*	19,42*
T7	286,98*	286,39*	332,67*	284,36*
R1	30,65*	30,71*	30,66*	30,78*

\* indica quantidades não-medidas

\*\* indica quantidades removidas

-A remoção de A2 faz com que os módulos das estatísticas de medida das quantidades remanescentes sejam inferiores ao valor crítico.

-Todos os valores absolutos das estatísticas de desbalanço são inferiores ao crítico, após a remoção de A2.

-Conclusão: foi detectado e identificado corretamente o erro grosseiro da quantidade A2, através dos três testes estatísticos. A quantidade que continha o erro grosseiro, também correspondia o maior valor absoluto da estatística de medida.

#### Caso SA3:

-Função-objetivo:  $J(\text{todos valores medidos}) = 35,46 > J_T(v=8) = 15,51$

-Estatística de medida: Há duas estatísticas de medida cujo valor é maior do que o crítico  $z_c(12 \text{ estatísticas diferentes}) = 2,865$  :

$$Z_a(T2) = 3,46 \text{ e } Z_a(T3) = 3,02$$

-Não há uma única medida, cuja remoção cause uma redução suficiente na função-objetivo (J1). Entretanto, a remoção de dois pares de medidas faz com que isto aconteça:

$$J2(\text{removendo T2 e T3}) = 6,16 < JT(v=6) = 12,59 \text{ e}$$

$$J2(\text{removendo B5 e T5}) = 4,37 < JT(v=6) = 12,59$$

-A remoção de qualquer um dos pares faz com que os valores absolutos de todas as estatísticas de medida das quantidades remanescentes sejam inferiores ao valor crítico.

-Todos os valores absolutos das estatísticas de desbalanço são inferiores ao crítico, após a remoção de qualquer um dos dois pares.

-Conclusão: embora o teste global inicial tenha detectado a presença de erro grosseiro, o teste de medida não indicou corretamente as medidas com erros grosseiros; dois pares de medidas (T2 e T3, B5 e T5) foram considerados como pares contendo erros grosseiros, através dos três testes estatísticos. O par que contém erros grosseiros (B5 e T5) está entre eles e é o que dá o menor valor para a função-objetivo após duas remoções.

A Tabela 4.2 apresenta os valores resultantes da aplicação do RETIF ao caso SA4, do processo de síntese de amônia. A seguir são feitas algumas observações sobre ele, onde a letra J se refere à função-objetivo (que equivale a estatística do teste global), JT ao valor tabelado pertinente da distribuição qui-quadrado e v indica o número de graus de liberdade.

#### Caso SA4:

-Função-objetivo:  $J(\text{todos valores medidos}) = 20,87 > JT(v=8) = 15,51$

-Estatística de medida: Nenhuma estatística de medida tem valor absoluto maior do que o crítico  $z_c(12 \text{ estatísticas diferentes}) = 2,865$ .

-Não há uma única medida, cuja remoção cause uma redução suficiente na função-objetivo (J1) ou seja, torne-a menor do que  $JT(v=7) = 14,07$ . Entretanto, a remoção de 13 pares de medidas faz com que a função-objetivo (J2) tenha valor menor do que o tabelado  $JT(v=6) = 12,59$ :

Par	J2	Par	J2
A1 e B1	12,306	D2 e C4	12,464
A1 e C4	1,961	T2 e C4	11,799
B1 e T1	4,393	A3 e C4	12,190
B1 e C4	9,634	T3 e C4	12,449
D1 e C4	12,503	C4 e B5	11,269
T1 e C4	10,272	C4 e T5	12,315
A2 e C4	11,803		

Tabela 4.2 - Resultados da Aplicação do RETIF ao Caso SA4 da Síntese de Amônia (quantidades em  $\text{kg mol/s}$ )

Quantidade	SA4 (A1 e C4)**	SA4 (B1 e T1)**	SA4 (B1 e C4)**
A1	32,99**	36,14	34,26
B1	98,78	108,23**	102,59**
D1	0,40	0,40	0,40
T1	132,17	144,77**	137,25
A2	101,77	104,23	102,82
B2	296,52*	302,86*	298,80*
D2	20,00	20,23	20,04
T2	418,29	427,32	421,67
A3	70,19	69,48	69,96
B3	201,77*	198,61*	200,22*
C3	63,17*	69,50*	65,72*
D3	20,00*	20,23*	20,04*
T3	355,12	357,81	355,94
C4	63,17**	69,50	65,72**
A5	70,19*	69,48*	69,96*
B5	201,77	198,61	200,22
D5	20,00*	20,23*	20,04*
T5	291,96	288,31	290,22
A6	1,40*	1,39*	1,40*
B6	4,04*	3,97*	4,00*
D6	0,40*	0,40*	0,40*
T6	5,84*	5,77*	5,80*
A7	68,79*	68,09*	68,56*
B7	197,73*	194,64*	196,21*
D7	19,60*	19,82*	19,64*
T7	286,12*	282,54*	284,42*
R1	31,58*	34,75*	32,86*

\* indica quantidades não-medidas

\*\* indica quantidades removidas

-A remoção de três dos treze pares faz com que os valores absolutos de todas as estatísticas de medida das quantidades remanescentes sejam inferiores ao valor crítico. Eles são: A1 e C4, B1 e T1, B1 e C4.

-Todos os valores absolutos das estatísticas de desbalanço são inferiores ao crítico, após a remoção de qualquer um desses três pares.

-Conclusão: embora o teste global inicial tenha detectado a presença de erro grosseiro, o teste de medida não indicou nenhuma medida como contendo este tipo de erro; três pares de medidas foram designados como possíveis suspeitos, através dos três testes estatísticos; o par que contém erros grosseiros (A1 e C4) está entre eles e é o que dá o menor valor para a função-objetivo após duas remoções.

Caso SA5:

-Neste caso, foram introduzidos erros em três quantidades medidas.

-Função-objetivo:  $J(\text{todos valores medidos}) = 72,82 > JT(v=8) = 15,51$

-Estatística de medida: Há três estatísticas de medida cujo valor absoluto é maior do que o crítico  $z_c(11 \text{ estatísticas diferentes}) = 2,835$ :

$$Za(A2) = -4,07$$

$$Za(A3) = 3,77$$

$$Za(C4) = 3,15$$

-Não há uma única medida, cuja remoção cause uma redução suficiente na função-objetivo (J1). Nem a remoção de dois pares de medidas faz com que isto aconteça.

-Conclusão: embora o teste global inicial tenha detectado a presença de erro grosseiro, o teste de medida não indicou corretamente as medidas responsáveis; entretanto, não foram indicados pares de medidas cuja remoção tornasse aceitáveis todas as estatísticas, de modo que há indícios de existir mais de duas medidas com erros grosseiros, o que está de acordo com o estabelecido neste caso.

### b) Processo de Síntese de Ácido Nítrico

O processo de síntese de ácido nítrico (SN) é o mesmo descrito no item 3.8. Partindo-se dos valores medidos dados na Tabela 3.6, foram gerados casos colocando erros em algumas quantidades (cujos valores foram tomados com duas casas decimais) e/ou designando como não-medidas outras:

Caso SN1: mantêm-se todos os valores medidos, acrescentando-se 20% ao valor de B1.

Caso SN2: considera-se, como medido, um grupo de 18 quantidades, mantendo-se os mesmos valores da Tabela 3.6.

Caso SN3: considera-se, como medido, o mesmo grupo de quantidades do caso SN2, aumentando-se em 20% a referente à B1.

Caso SN4: considera-se, como medido, o mesmo grupo de quantidades do caso SN2, aumentando-se, em 20%, as referentes à B1 e D8.

A Tabela 4.3 apresenta os valores resultantes da aplicação do RETIF aos diversos casos do sistema de síntese de ácido nítrico e a seguir são feitas algumas observações.

Caso SN1:

-Função-objetivo:  $J(\text{todos valores medidos}) = 63,98 > JT(v=43) = 59,28$

-Estatística **de** medida: Há uma única estatística de medida cujo valor absoluto é maior do que o crítico  $z_c$  (**3** estatísticas diferentes) = 3,172:

$$Za(B1) = -7,43$$

-A única **medida**, cuja remoção causa uma redução suficiente na função-objetivo (J1), é B1:

$$J1(B1 \text{ removida}) = 8,76 < JT(v=42) = 58,11$$

-A remoção de B1 faz com que os valores de todas as estatísticas de medida das quantidades remanescentes sejam inferiores ao valor crítico.

-Todos os **valores** absolutos das estatísticas de desbalanço são inferiores ao crítico, após a remoção de **B1**.

-Conclusão: **foi** detectado e identificado corretamente o erro grosseiro da quantidade B1, através dos três testes estatísticos.

#### Caso SN2:

-Função-objetivo: (todos valores medidos) = 3,90 < JT(v=11) = 19,67

-Estatística **de** medida: Os valores absolutos de todas as estatísticas de medida são menores do que o crítico.

-Os valores **absolutos** de todas as estatísticas de desbalanço (nodais) são inferiores ao valor crítico.

-Conclusão: **não** foi detectado erro grosseiro usando os três testes estatísticos, o que está correto pois só **havia** erros aleatórios.

#### Caso SN3:

-Função-objetivo: J(todos valores medidos) = 51,56 > JT(v=11) = 19,67

-Estatística **de** medida: Há uma única estatística de medida cujo valor absoluto é maior do que o crítico  $z_c$  (**18** estatísticas diferentes) = 2,977:

$$Za(B1) = -6,91$$

-A única **medida**, cuja remoção causa uma redução suficiente na função-objetivo (J1), é B1:

$$J1(B1 \text{ removida}) = 3,88 < JT(v=10) = 18,31$$

-A remoção de B1 faz com que os valores de todas as estatísticas de medida das quantidades remanescentes sejam inferiores ao valor crítico.

-Todos os valores absolutos das estatísticas de desbalanço são inferiores ao crítico, após a remoção de B1.

-Conclusão: foi detectado e identificado corretamente o erro grosseiro da quantidade B1, através dos três testes estatísticos.

Caso SN4:

-Função-objeto:  $J(\text{todos valores medidos}) = 101,37 > JT(v=11) = 19,67$

-Estatística de medida: Há 6 estatísticas de medida cujo valor absoluto é maior do que o crítico  $z_c(16 \text{ estatísticas diferentes}) = 2,94$ :

$$Za(B1) = -5,14$$

$$Za(C7) = 4,21$$

$$Za(D8) = -7,15$$

$$Za(T8) = 3,40$$

$$Za(E13) = 3,39$$

$$Za(D15) = 6,47$$

-Não há uma única medida, cuja remoção cause uma redução suficiente na função-objeto (J1). Entretanto, a remoção de dois pares de medidas faz com que isto aconteça:

$$J2(\text{removendo B1 e D8}) = 3,79 < JT(v=9) = 16,92$$

$$J2(\text{removendo B1 e D15}) = 14,99 < JT(v=9) = 16,92$$

-Somente a remoção do primeiro par faz com que os valores absolutos de todas as estatísticas de medida das quantidades remanescentes sejam inferiores ao valor crítico.

Tabela 4.3. Resultados da Aplicação do RETIF aos Casos SN1, SN2, SN3 e SN4 da Síntese de Amônia (quantidades em kgmol/h)

Quantidade	SN1	SN2	SN3	SN4
B1	798,31**	796,29	797,07**	795,63**
A2	321,27	321,45*	321,71*	321,24*
A3	1699,22	1716,13*	1716,47*	1716,56*
B3	1953,09	1955,29*	1956,17*	1956,22*
T3	3652,31	3671,43*	3672,65*	3672,78*
A4	2020,49	2037,58*	2038,19*	2037,80*
B4	2751,40	2751,58*	2753,25*	2751,86*
T4	4771,89	4789,16*	4791,43*	4789,66*
A5	1788,65	1806,46*	1806,81*	1806,91*
B5	2055,88	2058,20*	2059,13*	2059,18*

\* indica quantidades não-medidas

\*\* indica quantidades removidas

continuação da Tabela 4.3 (quantidades em kgmol/h)

Quantidade	SN1	SN2	SN3	SN4
C5	463,68	462,25*	462,74*	461,78*
T5	4308,22	4326,91*	4328,69*	4327,87*
A6	1788,65	1806,46*	1806,81*	1806,91*
B6	2055,88	2058,20*	2059,13*	2059,18*
T6	3844,54	3864,66*	3865,94*	3866,09*
C7	463,68	462,25	462,74	461,78
A8	4814,45	4823,83	4824,45	4829,38
D8	1272,83	1270,06	1270,99	1268,74**
T8	6087,28	6093,89	6095,44	6098,12
A9	4814,45	4823,83*	4824,45*	4829,38*
C9	463,68	462,25*	462,74*	461,78*
D9	1272,83	1270,06*	1270,99*	1268,74*
T9	6550,96	6556,14*	6558,19*	6559,90*
A10	4814,45	4823,83*	4824,45*	4829,38*
C10	60,23	60,34	60,34	60,36
D10	465,94	466,22*	466,19*	465,88*
E10	403,45	401,92*	402,40*	401,43*
F10	403,45	401,92*	402,40*	401,43*
T10	6147,52	6154,22	6155,79	6158,48
A11	4814,45	4823,83*	4824,45*	4829,38*
D11	390,65	390,80*	390,76*	390,44*
E11	403,45	401,92*	402,40*	401,43*
F11	493,79	492,42*	492,92*	491,96*
G11	60,23	60,34*	60,34*	60,36*
T11	6162,57	6169,31*	6170,87*	6173,57*
F12	391,78	391,75	391,74	391,74
E13	403,45	401,92	402,40	401,43
F13	885,57	884,17	884,66	883,70
T13	1289,02	1286,08	1287,06	1285,13
A14	4814,45	4823,83*	4824,45*	4829,38*
D14	390,65	390,80*	390,76*	390,44*
G14	60,23	60,34*	60,34*	60,36*
T14	5265,33	5274,97*	5275,56*	5280,18*
A15	4493,18	4502,38	4502,74	4508,15
D15	390,65	390,80	390,76	390,44
G15	60,23	60,34	60,34	60,36
T15	4944,06	4953,52	4953,84	4958,94
A16	89,43	90,32	90,34	90,35
B16	102,79	102,91	102,96	102,96
T16	192,23	193,23	193,30	193,30
R1	231,84*	231,13*	231,37*	230,89*
R2	403,45*	401,92*	402,40*	401,43*
R3	60,23*	60,34*	60,34*	60,36*

\* indica quantidades não-medidas

\*\* indica quantidades removidas

-Todos os valores absolutos das estatísticas de desbalanço são inferiores ao crítico, após a remoção deste par.

-Conclusão = embora o teste global inicial tenha detectado a presença de erro grosseiro, o teste de medida não indicou as medidas certas como responsáveis; todavia, após a remoção de duas medidas somente um dos pares de medidas (B1 e D8 ) foi designado como responsável, através dos três testes estatísticos; este par é justamente o que contém os erros e dá o menor valor para a função-objetivo após duas remoções.

### c) Circuito de Moagem

O circuito de moagem (CM) é o mesmo descrito no item 3.8. Entretanto, nos casos aqui relatados, considerou-se que o desvio-padrão é igual a 2,5% dos valores reais dados na Tabela.3.8. Os dados medidos foram recalculados e são dados na Tabela 4.4, junto com os resultados da retificação:

A seguir, são apresentados os seguintes casos:

CM1: só contém erros aleatórios;

CM2: no qual acrescentou-se 20% ao valor medido da corrente C5;

CM3: em que reduziu-se em 20% o valor medido da corrente A2.

#### Caso CM1:

-Função-objetivo:  $J(\text{todos valores medidos}) = 17,22 < JT(v=10) = 18,31$

-Estatística de medida: todas as estatísticas de medida têm valor absoluto menor do que o crítico.

-Estatísticas nodais: os valores absolutos de todas as estatísticas de desbalanço são inferiores ao crítico.

-Conclusão: não há erro grosseiro, o que está de acordo com o conjunto de dados.

#### Caso CM2:

-Função-objetivo:  $J(\text{todos valores medidos}) = 51,97 > JT(v=10) = 18,31$

-Estatística de medida: Há 5 estatísticas de medida cujo valor absoluto é maior do que o crítico  $z_c(23 \text{ estatísticas diferentes}) = 3,048$ :

$$Z_a(A1) = -5,00$$

$$Z_a(C5) = -6,18$$

$$Z_a(D5) = -4,62$$

$$Z_a(T5) = 5,63$$



$$Z_a(C7) = 3,05$$

-Todavia, a única medida, cuja remoção causa uma redução suficiente na função-objetivo (J1), é C5:

$$J1(C5 \text{ removida}) = 13,79 < JT(v=9) = 16,92$$

-A remoção de C5 faz com que os valores de todas as estatísticas de medida das quantidades remanescentes sejam inferiores ao valor crítico.

-Todos os valores absolutos das estatísticas de desbalanço são inferiores ao crítico, após a remoção de C5.

-Conclusão: foi detectado e identificado corretamente o erro grosseiro da quantidade C5, através dos três testes estatísticos.

### Caso CM3:

-Função-objetivo:  $J(\text{todos valores medidos}) = 33,62 > JT(v=10) = 18,31$

-Estatística de medida: Há 3 estatísticas de medida cujo valor absoluto é maior do que o crítico  $z_c(23 \text{ estatísticas diferentes}) = 3,049$ :

$$Z_a(A2) = 4,06$$

$$Z_a(A3) = 4,04$$

$$Z_a(T7) = -3,07$$

-Não há uma única medida, cuja remoção cause uma redução suficiente na função-objetivo (J1), embora seus valores após a remoção de A2 ou A3 não excedam em muito o valor tabelado correspondente; observou-se, ainda, que a remoção individual de qualquer destas medidas faz com que as estatísticas de medida das quantidades remanescentes sejam inferiores ao valor crítico correspondente.

-Entretanto, a remoção de 14 pares de medidas faz com que a função-objetivo (J2) tenha valores menores do que  $JT(v=8) = 15,51$ :

Par	J2	Par	J2
A1 e A2	13,83	A2 e T8	11,48
A1 e A3	14,81	A2 e C9	14,72
A2 e C5	13,72	A3 e C5	14,32
A2 e D5	1,95	A3 e D5	11,63
A2 e C7	12,69	A3 e T5	13,77
A2 e T7	13,59	A3 e D6	14,46
A2 e C8	12,61	A3 e T7	10,31

-A remoção de qualquer um desses pares faz com que os valores absolutos de todas as estatísticas de medida das quantidades remanescentes sejam inferiores ao valor crítico.

-Todos os valores absolutos das estatísticas de desbalanço são inferiores ao crítico, após a remoção de qualquer um dos pares, com exceção do segundo (A1 e A3).

-Conclusão: o teste global inicial detectou a presença de erro grosseiro e o teste de medida indicou três medidas como culpadas; todavia, a remoção individual de cada medida não fez com que as três estatísticas fossem aceitas; somente a remoção alternativa de vários pares tornou isto possível. Conclui-se que não houve a indicação esperada da medida com erro grosseiro e, por isso, não são fornecidos os resultados da retificação.

Tabela 4.4 - Resultados da Aplicação do RETIF aos Casos CM1 e CM2 do Circuito de Moagem (em toneladas métricas/h)

Quantidade	Valores medidos	CM1	CM2
A1	100,19	98,82	100,13
A2	305,31	306,60	306,32
A3	24,53	24,55	24,55
B4	nm	no	no
C4	nm	no	no
D4	nm	no	no
T4	261,72	261,72	261,72
A5	nm	98,82*	100,13*
B5	24,08	24,30	24,16
C5	148,16	147,33	135,46**
D5	94,80	94,31	94,19
T5	352,96	364,76	353,94
A6	nm	272,40*	272,45*
B6	11,58	11,36	11,41
C6	524,41	513,90	520,68
D6	318,18	307,66	307,84
T6	1142,63	1105,32	1112,39
A7	nm	677,83*	678,91*
B7	35,43	35,66	35,57
C7	637,10	661,23	656,15
D7	394,96	401,97	402,03
T7	1809,29	1776,68	1772,66
A8	nm	247,85*	247,91*
B8	35,72	35,52	35,44
C8	628,70	611,92	606,80
D8	194,08	194,63	194,21
T8	1060,01	1089,93	1084,35
A9	nm	429,97*	431,00*
B9	0,1366	0,14	0,14
C9	48,78	49,31	49,35
D9	203,41	207,33	207,82
T9	674,19	686,75	688,31

nm = não-medida

\* = observável

no = não-observável

\*\* = medida removida

### Conclusões sobre as aplicações deste programa:

O programa de retificação de dados foi aplicado a um processo de síntese de amônia, a um processo de síntese de ácido nítrico e a um circuito de moagem, gerando-se diferentes casos para cada processo pela adição de erros (10 ou 20%) a alguns dos valores medidos. Os resultados da aplicação do programa RETIF a cada um deles, relatados anteriormente, levaram às conclusões específicas apresentadas ao final da descrição de cada um, que deram origem às seguintes conclusões:

-nos casos (SA1 e SA2) do processo de síntese de amônia, em que havia somente uma medida contendo erro grosseiro, houve a indicação correta da presença dessa medida, sendo ela a única medida cuja remoção fazia com que todas as estatísticas fossem aceitas; nos casos (SA3 e SA4) em que havia duas medidas com erros grosseiros, mais do que um par de medidas foi indicado como culpado em cada um deles, porém ao par em que foram introduzidos os erros correspondia o menor valor da função-objetivo em qualquer dos casos; no caso (SA5) que continha três medidas com erros grosseiros, indica-se que deve haver mais de duas medidas com erros grosseiros, o que está correto.

-nos casos (SN1 e SN3) do processo de síntese de ácido nítrico, em que havia uma medida contendo erro grosseiro, houve a indicação correta da presença dessa medida, sendo ela a única medida cuja remoção fazia com que todas as estatísticas fossem aceitas; no caso (SN4) que continha dois erros grosseiros também foi indicado de forma correta o par culpado e somente a sua remoção fazia com que todas os três testes estatísticos fossem aceitos; no caso (SN2) que continha somente erros aleatórios, não foi indicada a presença de erros grosseiros, que é o resultado correto.

-num dos casos (CM2) do circuito de moagem, onde havia um único erro grosseiro houve a identificação correta da fonte de erro, porém no outro (CM3) não; no caso (CM1) que continha somente erros aleatórios, não foi indicada a presença de erros grosseiros, que é o resultado correto.

Além disso, pelo fato de tratar os três processos diferentes, contendo reações químicas ou não, pode-se dizer que este programa se mostra capaz de retificar dados de vários outros processos.

## CAPÍTULO 5. CLASSIFICAÇÃO DE VARIÁVEIS DE PROCESSO

## 5.1 Introdução

De forma **g**eral, nem todas as variáveis de um processo são medidas, em função, por exemplo, dos **c**ustos envolvidos ou da exequibilidade técnica. Porém, dado um conjunto de variáveis medidas, **é** importante saber quais as variáveis cujos valores podem ser unicamente determinados, a **p**artir dele, para melhor acompanhar a operação de um processo. Por outro lado, no caso de **v**ariáveis medidas, pode ser fundamental conhecer os seus valores, mesmo se os instrumentos **r**esponsáveis por sua medição estiverem fora de uso.

A **c**lassificação de variáveis de processo trata de buscar respostas para estas questões, de modo que as **v**ariáveis não-medidas são separadas em dois grupos: observáveis, que inclui aquelas que **p**odem ser calculadas de forma única a partir das que foram medidas e não-observáveis, **o**nde **s**e incluem as demais. Por sua vez, as variáveis medidas são agrupadas de acordo com a sua **r**edundância ou não.

De acordo **c**om Mah (1990), se for possível mudar o valor de uma variável sem violar as restrições de **c**onservação, então a mudança é chamada factível (feasible). Comumente, tal mudança pode **i**mplicar que os valores de algumas outras variáveis relacionadas tenham de variar de forma **a**dequada, para que as restrições continuem sendo satisfeitas.

Dada uma rede de processo, com uma certa distribuição de um conjunto de instrumentos, se **f**or feita uma mudança factível numa variável, sem que ela seja detectada (ou observada) pelos **i**nstrumentos, então diz-se que a variável é não-observável. Por esta definição, uma **v**ariável medida é certamente observável, mas as não-medidas podem, ou não, ser observáveis. **U**ma variável é dita redundante, quando ela permanecer observável, ao se eliminar uma **m**edida a ela associada.

Em geral, a redundância leva a conflitos entre os dados e os balanços e à necessidade de reconciliá-los. **P**ossibilita, ainda, verificar a fidedignidade de uma dada medida.

Diz-se que um balanço de espécies químicas conservadas, em torno de uma seção de um processo, é **r**edundante se todas as vazões das espécies que nela entram ou saem forem medidas. Neste **c**aso, todas as medidas são redundantes. Se um balanço for não-redundante, não há possibilidade de conflito com as leis de conservação, já que a(s) medida(s) faltante(s) pode(m) ser **e**scolhida(s) para satisfazer os balanços. A redundância também ocorre entre as medidas de uma **c**orrente, quando todas as vazões mássicas dos componentes e a vazão total forem medidas. **U**m outro caso de redundância ocorre quando o número de concentrações medidas em cada **c**orrente for ao menos igual ao número de correntes incidentes em unidade de processo e somente uma vazão total for medida, ou tiver um valor atribuído.

Em vista **d**isso, a classificação da observabilidade e da redundância fornece uma base racional para **p**rojetar o desempenho de um sistema de monitoramento, definir a colocação de instrumentos de **d**iferentes tipos, custos e precisão.

## 5.2 Procedimentos Para Classificação de Variáveis de Processo

Para **c**lassificar, de forma ampla, as variáveis de uma rede de processo é necessário conhecer a **e**strutura do gráfico do processo (ou a topologia das correntes do processo) e os

tipos de arcos do processo (fluxo de massa e energia, fluxo de calor) e a presença de medições. Além disso, deve-se ter um conjunto de restrições impostas sobre quantidades extensivas em que se incluem as leis de conservação e os coeficientes estequiométricos de reações químicas; pode haver, também, um conjunto de restrições impostas sobre variáveis intensivas, como as chamadas equações de normalização (a soma das frações mássicas ou molares dos componentes de uma corrente é igual a um) ou como a igualdade de concentrações das espécies das correntes que entram e saem de um divisor de fluxo ou como o equilíbrio entre fases ou entre reações químicas.

No caso de variáveis de balanços de massa, as medições podem ser referentes à vazão mássica e às frações das espécies químicas (no caso de balanços multicomponentes), havendo três tipos de variáveis: vazão total, vazão dos componentes e frações dos componentes da corrente. As restrições se constituem dos balanços de massa totais e das espécies químicas, que são restrições nodais, e das equações de normalização, que são restrições de corrente. Podem haver, ainda, outras restrições aplicadas a quantidades intensivas.

Os procedimentos para classificação de variáveis descritos na literatura consultada se baseiam na análise teórico-gráfica simples (Václavek, em 1969 e Mah em 1976), nos subgráficos perturbação e nas perturbações factíveis não-medidas (Stanley e Mah, em 1981a-b), na resolubilidade de equações de balanços nodais (Romagnoli e Stephanopoulos, 1980), na teoria gráfica combinada com a resolubilidade de equações restritivas (Kretsovalis e Mah, em 1987 e 1988a-b) e em projeção matricial (Crowe, em 1989).

Os estudos de Václavek (1969) para classificação de variáveis, aplicados a redes de fluxo de massa de um único componente, baseados em análise teórico-gráfica, foram, posteriormente, estendidos para sistemas multicomponentes, por ele e seus colaboradores (1976). Estes apresentam certas limitações, pois são aplicáveis aos casos em que as correntes têm a composição de todos os seus componentes medida ou então nenhuma concentração é medida. O conjunto de restrições é limitado, não incluindo reações químicas; consideram a restrição de corrente, que diz que a soma das vazões dos componentes de uma dada corrente é igual à vazão total, eliminando o balanço de uma espécie química por nó, ao invés de acrescentar uma restrição por nó, reduzindo a redundância; chamam as quantidades medidas de não-redundantes, quando seu ajuste é igual a zero. Entretanto, uma quantidade não-redundante pode ter ajuste diferente de zero devido à covariância.

Os estudos de Romagnoli e Stephanopoulos (1980) tratam de redes multicomponentes, sendo aplicáveis aos casos em que as correntes têm a composição de todos os seus componentes medida ou então nenhuma concentração é medida. Também classificam as quantidades medidas como não-redundantes, quando seu ajuste é igual a zero. Todavia, uma quantidade não-redundante pode ter ajuste diferente de zero devido à covariância. De acordo com Kretsovalis e Mah (1987b), sua classificação da observabilidade de uma variável não-medida se limita ao fato dela poder ou não ser escrita como uma saída de uma equação de balanço nodal e que isto pode levar a uma classificação incorreta, visto que não é examinada a possibilidade dela ser obtida através de balanços de agregados de nós.

Os estudos de Stanley e Mah (1981a) estabelecem princípios teóricos para classificação da observabilidade e redundância; os algoritmos por eles desenvolvidos (1981b) se destinavam ao tratamento de fluxos totais de massa-energia, sem reações químicas. A partir destes, Kretsovalis e Mah (1987b) desenvolveram princípios de classificação, baseando-se na resolubilidade de equações restritivas e na teoria gráfica, aplicáveis a redes de fluxo massa-energia multicomponentes, em que fluxos de energia pura e medições de temperatura não

estavam presentes. Posteriormente, Kretsovalis e Mah (1988a-b) estenderam este tratamento a redes de processo generalizadas, incluindo balanços de energia, divisores de fluxo e reações químicas. Neste tratamento, variáveis de corrente que não caem em ciclos, com vazões mássicas não-medidas, são classificadas com base em critérios teórico-gráficos, somente. As vazões mássicas não-medidas e que caem em ciclos são classificadas pela análise da possibilidade de obter estimativas de seus valores, através de medidas de composição e de balanços de massa dos componentes. Para tratar deste problema, são examinadas seções do gráfico do processo, para escolher o subconjunto apropriado de equações restritivas e para avaliar sua resolubilidade. Como há um número grande de seções, uma implementação computacional bem sucedida depende da geração seletiva daqueles poucos que são necessários para a classificação.

Utilizando projeção matricial, como já havia feito anteriormente (Crowe et alii. em 1983 e Crowe em 1986), para reconciliar dados de processo, Crowe (1989a) desenvolveu a teoria e métodos diretos para classificação de dados de processo, no estado estacionário. Eles se baseiam nas matrizes que descrevem a estrutura da planta e em qualquer conjunto de valores consistentes com as restrições. É permitida a colocação arbitrária de medições, reações químicas, divisores de fluxo e fluxos de energia pura.

Os últimos estudos citados foram escolhidos para servir de base para o procedimento computacional para classificação das variáveis desenvolvido neste trabalho.

### 5.3 Classificação de Variáveis de Processo Através de Projeção Matricial

De acordo com Crowe (1986), para uma distribuição arbitrária das medições, a classificação deve se referir aos componentes das correntes, então há três categorias de variáveis:

1. Vazão total e concentração medidas e ajustáveis.
2. Concentração medida e ajustável.
3. Vazão total medida ou desconhecida; vazão do componente não-medida.

Nos seus estudos (1989) sobre observabilidade e redundância de dados de processo, para reconciliação no estado estacionário, apresentados a seguir, adota essa classificação.

Define observabilidade por:

-Uma quantidade não-medida em um processo no estado estacionário é observável se e somente se ela puder ser unicamente determinada a partir de um conjunto fixo de valores, correspondente a vazões e concentrações medidas, que são consistentes com todas as restrições em questão. Toda quantidade não-medida que não for assim determinável é não-observável.

Observa que o conjunto de valores correspondente às medições pode ser qualquer conjunto fixo desde que seja consistente com as restrições. Isto garante que pelo menos uma solução seja encontrada para as quantidades não-medidas. Tal conjunto consistente de valores não precisa ser obtido pela aplicação da reconciliação de dados.

Adota a definição de Stanley e Mah (1981a), para redundância:

-Uma quantidade medida é redundante se, e somente se, ela fosse observável se não fora medida. Caso contrário, a quantidade medida é não-redundante.

A forma geral das restrições de conservação e outras, impostas sobre quantidades extensivas, é:

$$B_1 \bar{x} + B_2 N_2 d + Pv = 0 \quad (5.1)$$

onde os três termos do lado esquerdo correspondem, respectivamente, às variáveis das categorias 1, 2 e 3, sendo:

$B_1$  = matriz das restrições das variáveis da Categoria 1,  $m \times q$

$B_2$  = matriz das restrições das variáveis da Categoria 2,  $m \times nb2$

$N_2$  = é uma matriz diagonal, que contém as vazões totais não-medidas das variáveis da Categoria 2, com tantos elementos para cada corrente, quantas forem as concentrações medidas em  $d$ , de modo que  $N_2 d$  é o vetor das vazões dos componentes

$P$  = matriz das restrições das variáveis da Categoria 3,  $m \times n$ , definida na eq.3.6

$\bar{x}$  = vetor das vazões medidas da Categoria 1,  $q \times 1$

$d$  = vetor das concentrações medidas da Categoria 2,  $nb2 \times 1$

$v$  = vetor das variáveis da Categoria 3,  $n \times 1$

$q$  = número de variáveis da Categoria 1

$nb2$  = número de variáveis da Categoria 2

$n$  = número de variáveis da Categoria 3

O segundo termo do lado esquerdo da equação (5.1) pode ser substituído por:

$$C(d) = B_2 N_2 d \quad (5.2)$$

onde  $C$  é uma matriz tal que cada coluna corresponde a uma corrente. Cada coluna contém a soma das colunas de  $B_2$ , correspondentes àquela corrente, sendo previamente multiplicadas pelas concentrações respectivas.

Observa que, em adição às restrições acima, pode haver restrições impostas sobre variáveis intensivas. Como exemplos, cita a igualdade de concentrações das espécies das correntes, que entram e saem de um divisor de fluxo, ou o equilíbrio entre fases ou entre reações químicas, às quais se refere como restrições intensivas.

A classificação das quantidades não-medidas se fundamenta nos quatro lemas seguintes, de acordo com Crowe (1989). Os dois primeiros dão condições suficientes para



encontrar vazões não-observáveis das Categorias 3 e 2, respectivamente. O Lema 3 apresenta condições necessárias e suficientes para vazões não-observáveis. No Lema 4, são dadas condições necessárias e suficientes para concentrações observáveis das espécies da Categoria 3.

Lema 1: Se existir um vetor  $t$  não-zero tal que  $Pt = 0$ , então cada quantidade extensiva não-medida correspondente a um elemento não-zero de  $t$  é não-observável, a menos que ela seja determinada através de uma restrição intensiva.

Lema 2: Se existir um vetor  $w$  não-zero, tal que  $C(d^+)w = 0$ , onde  $d^+$  é consistente com as restrições, então cada elemento de  $n$  correspondente a um elemento não-zero de  $w$  é não-observável, a menos que seja observável através de uma restrição intensiva.

Lema 3: Se e somente se existir um vetor

$$\begin{bmatrix} w \\ \dots \\ t \end{bmatrix} \quad (5.3)$$

tal que

$$C(d^+)w + Pt = 0 \quad (5.4)$$

onde  $d^+$  é um vetor fixo de concentrações consistentes; cada elemento de  $n$  ou  $v$  correspondente a um elemento não-zero de  $w$  ou  $t$ , respectivamente, é não-observável, desde que ele também seja não-observável através de restrições intensivas.

Lema 4: Uma concentração não-medida,  $c_{ij}$ , da espécie  $i$  na corrente  $j$  é observável se, e somente se, uma das três condições seguintes for válida:

a) a vazão da espécie  $i$ ,  $v_{ij}$ , é observável e a vazão total,  $v_{oj}$ , da corrente  $j$  é observável ou medida, ou

b) nem a vazão da espécie  $i$ , nem a vazão total da corrente  $j$  são observáveis e quaisquer mudanças factíveis nessas vazões (isto é, mudanças que são consistentes com as restrições) forem restringidas por

$$v_{ij} = \beta \delta v_{ij} \quad \text{ou} \quad v_{oj} = \beta \delta v_{oj} \quad (5.5)$$

para o escalar  $\beta$ . Aqui  $\delta v$  é uma mudança factível na vazão de uma espécie ( $v_{ij}$ ) ou na vazão total ( $v_{oj}$ ), que mantém as restrições em vigor, ou

c) há uma restrição intensiva, a partir da qual a concentração da espécie  $i$  na corrente  $k$  pode ser unicamente determinada como uma função de quantidades intensivas observáveis ou medidas.

A observabilidade consistente com a condição b do Lema 4 não pode envolver extensões de reações não-observáveis, pois uma reação varia a vazão de uma espécie, mas deixa a vazão mássica total inalterada, tornando inválida a equação 5.5. Para que esta equação se mantenha para as diversas espécies químicas de uma corrente, a variabilidade das vazões das

espécies deve ter quanto muito um grau de liberdade, que frequentemente se encontra associado com uma vazão não-observável da Categoria 2. Define a vazão de uma espécie da Categoria 2 como observável se, e somente se, a vazão total daquela corrente for observável.

A redução das restrições, como no problema de reconciliação (Crowe et alii.1983), é feita eliminando as variáveis não-medidas presentes nas equações restritivas (eq.5.1). Isto é feito usando matrizes ortogonais à matriz  $P$ , chamadas matrizes-projeção.

A matriz  $Y^T$  é construída para ter característica máxima e ser ortogonal às colunas de  $P$ , isto é,

$$Y^T P = 0 \quad (5.6)$$

A matriz  $P$  é separada em

$$P = \begin{bmatrix} P_1 & \vdots & P_3 \\ \dots & \dots & \dots \\ P_2 & \vdots & P_4 \end{bmatrix} \quad (5.7)$$

onde  $P_1$  é uma matriz não-singular quadrada máxima. Não há perda de generalidade ao assumir que ela esteja na divisão superior esquerda, pois as colunas podem ser reordenadas e as trocas de linha irão acarretar, simplesmente, trocas das colunas correspondentes de  $Y^T$ . Pode-se escolher

$$Y^T = [-P_2 P_1^{-1} \quad \vdots \quad I] \quad (5.8)$$

As colunas da divisão direita de  $P$  são dependentes das colunas da divisão esquerda, o que implica que a eq.5.6 é válida para a  $Y^T$  dada pela eq.5.8.

As restrições das eq.5.1 e 5.2 são multiplicadas por  $Y^T$ , dando, em geral,

$$Y^T B_1 x + Dn = 0 \quad (5.9)$$

onde

$$D = Y^T C \quad (5.10)$$

A matriz  $D$  é, então, dividida em

$$D = \begin{bmatrix} D_1 & \vdots & D_3 \\ \dots & \dots & \dots \\ D_2 & \vdots & D_4 \end{bmatrix} \quad (5.11)$$

onde  $D_1$  é uma matriz não-singular quadrada máxima, contida em  $D$ . A matriz  $Z$  é definida como

$$Z^T D = 0 \quad (5.12)$$

com

$$Z^T = [-D_2 D_1^{-1} \quad \vdots \quad I] \quad (5.13)$$

a fim de eliminar os termos da Categoria 2 das restrições.

Para determinar quais quantidades não-medidas são não-observáveis, sem o uso de restrições intensivas, busca-se uma solução geral para  $w$  e  $t$  na eq.(5.4). A solução geral usando somente as colunas de  $P$  é

$$w = 0; \quad t = \begin{bmatrix} -P_1^{-1} P_3 \\ \dots\dots\dots \\ I \end{bmatrix} u_j \quad (5.14)$$

para qualquer vetor unitário  $u_j$ . Cada elemento de  $v$  correspondente a um elemento não-zero de  $t$  (e conseqüentemente a uma linha não-zero da matriz que multiplica  $u_j$ ) é não-observável pelo Lema 1, já que  $P t = 0$  pela eq.5.4. Evidentemente, os elementos de  $v$  correspondentes às colunas de  $P_3$  são sempre não-observáveis. Lembra que há exatamente tantas dependências distintas entre as colunas de  $P$  quantas são as colunas de  $P_3$ .

A solução da eq.5.4 incluindo as colunas de  $C$ , conforme Crowe (1989), é

$$w = \begin{bmatrix} -(D_1^+)^{-1} D_3^+ \\ \dots\dots\dots \\ I \end{bmatrix} u_j \quad (5.15)$$

para qualquer vetor unitário conformável  $u_j$  e

$$t = \begin{bmatrix} -P_1^{-1} C_1^+ w \\ \dots\dots\dots \\ 0 \end{bmatrix} \quad (5.16)$$

onde  $C_1^+$  é uma divisão das linhas de  $C$  correspondentes às linhas de  $P_1$ . Observa que  $C$  é avaliada com valores consistentes de  $d$ , de modo que a separação de  $D$  pode ser diferente daquela usando valores medidos de  $D$ . Novamente, as linhas não-zero da matriz que multiplica  $u_j$  dão os elementos não-observáveis de  $n$ . Os elementos não-zero de  $t$  indicam os elementos não-observáveis de  $v$  a eles correspondentes, de acordo com o Lema 3. Todos os elementos de  $n$  e  $v$ , que não foram classificados como não-observáveis, são então observáveis. A partir das soluções gerais (eq. 5.14-5.16), é óbvio que todas quantidades não-medidas são observáveis se  $P$  e  $D$  têm característica de coluna total, pois  $P_3$  e  $D_3$  têm colunas vazias. Então, as quantidades intensivas podem ser introduzidas, para denominar alguma quantidade não-observável como observável.

O caso especial, onde  $D=0$ , deve ser notado. Aqui,  $D_1$  é vazia, então sua inversa é vazia e  $D_3$  tem uma linha vazia, mas colunas não-vazias. Então,  $w$  da eq.5.15 se torna qualquer coluna da matriz identidade, com ordem igual ao número de colunas de  $C$ .

As concentrações observáveis satisfazem pelo menos uma das três condições do Lema 4. Soluções da eq.5.4, dadas pelas eq.5.15 e 5.16, são examinadas para encontrar vazões da

Categoria 3, que satisfazem a eq.5.5. As concentrações que são classificadas como observáveis, através de restrições intensivas são especialmente marcadas, pois elas podem levar à classificação de quantidades não-observáveis como observáveis.

Distingue quatro casos de redundância, uma vazão de uma espécie, uma concentração ou uma vazão total da Categoria 1 e uma concentração da Categoria 2. Os quatro lemas seguintes fornecem os testes necessários para decidir quais medidas de cada um dos quatro casos são não-redundantes.

Lema 5: Uma vazão de uma espécie de  $x$  da Categoria 1 é não-redundante se, e somente se,

$$(Z^+)^T Y^T b = 0 \quad (5.17)$$

onde  $b$  representa a coluna de  $B_1$  correspondente àquela espécie e  $Z^+$  é avaliada a partir de  $D(d^+)$ , e sua concentração não pode ser unicamente determinada a partir de restrições intensivas.

Lema 6: A concentração de uma espécie  $i$  da Categoria 1 correspondente a sua vazão na corrente  $j$  (a saber, o elemento  $x_{ij}$  de  $x$ ) é redundante se, e somente se,

a) a vazão correspondente da espécie  $i$  for redundante, ou

b) ela for determinada de forma única, a partir de outras quantidades intensivas observáveis ou medidas, através de restrições intensivas.

Lema 7: A vazão total medida de uma corrente  $j$  é não-redundante se, e somente se, ambos

$$a) (Z^+)^T Y^T B_{1,ij} x_j^+ = 0 \quad (5.18)$$

onde  $B_{1,ij}$  é o conjunto de colunas de  $B_1$  para a corrente  $j$  e  $x_j^+$  é um vetor de vazões consistentes das espécies para aquela corrente, e

b) ela não pode ser unicamente determinada a partir de outras quantidades medidas ou observáveis, através de restrições intensivas.

Lema 8: A concentração medida  $d_{ij}$  de uma espécie  $i$  da Categoria 2 na corrente  $j$  é não-redundante se, e somente se, ambos

$$a) (Z^+)^T Y^T B_{2,ij} = 0 \quad (5.19)$$

onde  $B_{2,ij}$  é a coluna de  $B_2$  para a espécie  $i$  na corrente  $j$  e

b) ela não é unicamente determinada através de restrições intensivas.

Os casos especiais em que as matrizes  $P$  ou  $D$  têm característica igual ao número de linhas levam, respectivamente, a  $Y=0$  ou a  $D=0$ . Nestes casos, quaisquer valores medidos são não-redundantes, pois as condições dos lemas 5-8 para não-redundância são trivialmente satisfeitas.

O conjunto de valores consistentes correspondentes às medidas da Categoria 1 é usado, explicitamente, somente no Lema 7. Se a reconciliação houver sido feita, os cálculos necessários para verificar a redundância já terão sido feitos.

#### 5.4 Bases do Procedimento Computacional de Classificação de Variáveis de Processo

Desenvolveu-se um procedimento computacional, utilizando Linguagem Turbo C, denominado TCLASS, cuja listagem é dada no Anexo E, aplicável à classificação de variáveis medidas e não-medidas, envolvendo balanços de massa de sistemas multicomponentes de processos no estado estacionário, onde podem ocorrer ou não reações químicas. Foi empregada projeção matricial, de acordo com princípios descritos no item anterior.

Para aplicar este programa é preciso conhecer a estrutura ou topologia da rede de processo e saber em que correntes ocorrem medições e o tipo de medição. Em redes de fluxo multicomponentes, há dois tipos de medições que são a vazão total e a composição de cada corrente e três tipos de variáveis delas decorrentes: vazão total, vazão das espécies químicas e composição de cada corrente. Nas redes onde ocorrem reações químicas, é preciso saber quais são elas, os seus coeficientes estequiométricos e os reatores onde ocorrem. Não são impostas restrições sobre a situação das correntes, cuja vazão total ou cuja composição é medida, nem quais concentrações são medidas em cada corrente.

As variáveis são agrupadas em três categorias:

- 1) a vazão do componente é medida, o que implica que a vazão total e a concentração deste componente são medidas;
- 2) somente a concentração do componente é medida;
- 3) nenhuma medição é feita numa corrente desta categoria.

A vazão total é tratada como a vazão de um componente adicional. Ela é colocada na Categoria 1, quando for medida; na Categoria 2, quando ela não for medida e pelo menos a concentração de uma das espécies nela presente for medida e na Categoria 3, quando não for medida e não tiver a composição de nenhuma espécie química nela presente medida.

A conversão de cada reação é tomada como uma variável não-medida (Categoria 3).

O conjunto de restrições aplicado sobre quantidades extensivas, inclui os balanços de massa das correntes totais e das espécies químicas, incluindo reações químicas quando houver reatores.

O conjunto de restrições aplicados sobre quantidades intensivas se limita às chamadas equações de normalização, que estabelecem que a soma das frações das espécies químicas contidas em cada corrente é igual a um.

Ao conjunto de restrições dado pela eq.5.1, adicionou-se a matriz denominada de balanço das correntes individuais (BCI), sendo que cada uma de suas linhas corresponde a uma corrente. Os elementos correspondentes às vazões ou frações dos componentes daquela

corrente são iguais a 1; os elementos correspondentes às vazões totais ou ao componente total, daquela corrente, são iguais a -1 e os elementos das colunas correspondentes às demais correntes são iguais a zero. O número de colunas é igual à soma do número de variáveis de cada categoria.

Uma linha dessa matriz representa o balanço de uma corrente individual, se a corrente correspondente tiver vazão total e composição medidas. No caso de uma corrente onde somente a composição é medida, a linha correspondente representa uma equação de normalização.

O procedimento desenvolvido se baseia em lemas enunciados no item 5.2 e na solução geral dada pela eq. 5.14, usando somente as colunas de P (w=0 ou D tem característica de coluna total).

Entretanto, a solução geral para a eq.5.4, incluindo-se as colunas de C, foi obtida separando as colunas de C, da mesma forma em que foram separadas as colunas de P (eq. 5.7). Então, buscam-se w e t que satisfaçam

$$\begin{bmatrix} C_1 \equiv P_1:P_3 \\ \dots \\ C_2 \equiv P_2:P_4 \end{bmatrix} = \begin{bmatrix} w \\ \dots \\ t_1 \\ \dots \\ t_3 \end{bmatrix} \quad (5.20)$$

Substituindo a porção inferior desta equação, pelo produto da eq.5.4 por  $Y^T$ , obtém-se:

$$C_1 + P_1 t_1 + P_3 t_3 = 0 \quad (5.21)$$

$$Y^T C w = 0$$

Portanto, w deve estar no espaço nulo de D, devido à eq. 5.10 e, então, pode ser qualquer coluna de

$$\begin{bmatrix} -(D_1)^{-1} D_3 \\ \dots \\ I \end{bmatrix} \quad (5.22)$$

Então,  $t_1$  é obtido a partir da porção superior da eq.5.20

$$t_1 = -P_1^{-1} (C_1 w + P_3 t_3) \quad (5.22)$$

onde  $t_3$  é um vetor arbitrário de elementos não-zero ou uma matriz de elementos não-zero (pois corresponde às colunas linearmente dependentes de P, ou seja, às colunas de  $P_3$ ), se o número de colunas de w for igual, respectivamente, a 1 ou > 1.  $C_1$  é uma divisão das linhas de C, correspondente às linhas de  $P_1$ . As linhas de C devem ser trocadas da mesma forma que as de P antes de obter  $C_1$ .

## 5.5 O Programa de Classificação de Variáveis de Processo

Este programa, designado Programa TCLASS, pode ser aplicado para a classificação de variáveis de processo medidas e não-medidas, de sistemas no estado estacionário, de acordo com as bases dadas no item 5.4. É aplicável a balanços de massa multicomponentes, onde ocorrem ou não, reações químicas.

As variáveis (vazão total, vazão do componente e concentração medida do componente) são classificadas com base em restrições aplicadas sobre quantidades extensivas e nas equações de normalização. Portanto, a inclusão de outras restrições, baseadas em quantidades intensivas (como relações de equilíbrio), pode alterar os resultados da classificação.

Considerou-se que:

- o número máximo de nós é 10;
- o número máximo de correntes totais é 20;
- o número máximo de componentes é 10;
- o número máximo de reações químicas é 5;

Os dados de entrada podem entrar através do teclado ou através de um arquivo de dados tipo texto, cujas informações devem ser gravadas na ordem dada a seguir:

- número de correntes totais ( $n_i$ );
- número de nós ( $m_i$ );
- número de componentes ( $c$ );
- número de reações químicas ( $r_q$ );
- número de correntes totais da Categoria 2, que têm só concentração medida, ( $n_c$ );
- número de correntes totais da Categoria 1, cuja vazão é medida, ( $n_{ct}$ );
- número de vazões da Categoria 2 (que têm só a fração medida), ( $n_{b2}$ );
- número de vazões medidas (dos componentes e total),  $q$ ;

-matriz de informações das correntes  $M(n_i \times 2)$ ; esta matriz tem número de linhas igual ao número de correntes totais e duas colunas. A cada linha corresponde uma corrente, tomada na ordem numérica, sendo que o primeiro elemento de cada linha deve conter o número do nó no qual a corrente entra e o segundo elemento, o número do nó do qual ela sai. Quando uma corrente entrar ou sair do meio-ambiente deve ser colocado o número 100 no elemento correspondente;

-matriz estequiometria  $TE(r_q \times (c+2))$ ; que tem número de linhas igual ao número de reações químicas que ocorrem no sistema, acrescido de duas unidades. No primeiro elemento de cada linha, deve constar o número do nó, onde a reação ocorre, seguido dos coeficientes estequiométricos da reação unitária. São colocados com sinal negativo se forem reagentes, positivo se forem produtos. Coloca-se o algarismo 0 no elemento correspondente aquele(s) que não participar(em) da reação e também naquele referente a corrente total (que ocupa o último elemento de cada linha);

-vetor de informações das variáveis ( $viv$ ), que informa a categoria à qual a variável pertence. Quando se refere a um sistema com  $c$  componentes,  $n_i$  correntes totais e  $r_q$  reações químicas, ele deve conter um número de elementos igual a  $n_i \times (c+1) + r_q$ . Então a cada corrente correspondem  $c+1$  elementos de  $viv$ , que se referem às vazões dos componentes, seguidas da vazão total daquela corrente. Como, em geral, nem todos os elementos estão

presentes em todas as correntes, coloca-se o algarismo 0 no(s) elemento(s) de viv correspondente(s) à(s) espécie(s) química(s) que não faz(em) parte daquela corrente; coloca-se o algarismo 1 quando pertencer e se tratar da vazão medida de um componente, o algarismo 2 quando pertencer e somente a concentração for medida e o algarismo 3, quando pertencer e não tiver nenhum parâmetro medido (nem a concentração, nem a vazão total da corrente correspondente). No elemento correspondente à corrente total, coloca-se 0, quando ela contiver uma única espécie. Quando dela fizerem parte mais de um componente, coloca-se 1, quando for medida e pelo menos a concentração de um deles for medida, coloca-se 2, quando a vazão total não for medida, mas tiver a composição de pelo menos um componente medida e 3, quando não for medida e nenhuma composição for medida. Os últimos  $n$  elementos de viv, correspondem às conversões das reações químicas e são, portanto, iguais a 3, pois elas são não-medidas.

-vetor de informações dos equipamentos (vie), que informa quais os elementos que contatam com cada nó (equipamento ou junção do sistema). No caso de um sistema com  $m$  nós, ele tem  $m \times (c+1)$  elementos. Cada grupo de  $c+1$  elementos corresponde a um nó, atribuindo-se a eles o algarismo 0, quando a espécie química correspondente não fizer contato com aquele nó, ou corresponder à corrente total e o algarismo 1, quando a espécie química correspondente entrar ou sair daquele nó.

-vetor dos valores das frações (fra), que informa os valores numéricos das frações medidas. Quando a concentração de pelo menos um dos componentes de uma corrente total for medida, deve haver um elemento correspondente à corrente total, colocando-se 1 no elemento correspondente à fração conhecida da corrente total, posicionada logo após as frações medidas dos seus componentes.

-vetor do número de frações medidas por corrente (vfra), que informa o número de frações medidas em cada corrente. Inclue exclusivamente as correntes que têm somente concentração medida, contando-se a corrente total como um dos componentes quando pelo menos a concentração de um deles for medida.

-vetor dos valores das vazões medidas (vaz), que informa os valores numéricos das vazões medidas dos componentes, seguidos da vazão total daquela corrente, na ordem crescente do número das correntes;

-vetor do número de vazões medidas (vvaz), que informa o número de vazões medidas (da Categoria 1), em cada corrente total, que é incluída na contagem. O número de elementos desse vetor é igual ao número de correntes totais que têm quantidades nessa categoria.

## 5.6 Aplicações do Programa TCLASS

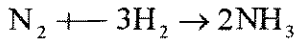
Para avaliar o desempenho do programa TCLASS, utilizou-se um processo de síntese de amônia e um processo de evaporação-cristalização, em situações diversas, em função da existência, ou não, de medições em suas correntes.

### a) Processo de Síntese de Amônia

No processo de síntese de amônia, o número de espécies químicas envolvidas é 4, a saber nitrogênio, hidrogênio, amônia e argônio, designados, respectivamente, pelas letras A, B, C e D. Os componentes presentes em cada corrente são mostrados na Figura 3.2.



A amônia é gerada no nó número dois, através da seguinte reação:



O nó número 4 é um divisor de fluxo, mas a restrição correspondente não consta do conjunto de restrições.

As vazões de cada corrente são apresentadas na Tabela 5.1, onde a letra T, seguida de um número representa a vazão total daquela corrente. As quantidades medidas são consistentes para a classificação da redundância.

Tabela 5.1 - Vazões do Processo de Síntese de Amônia (kgmol/s)

Corrente	Vazão dos Componentes				Vazão Total
	A	B	C	D	
1	32,70	98,04	0	0,40	131,14
2	100,56	298,99	0	20,15	419,70
3	69,23	205,00	62,66	20,15	357,04
4	0	0	62,66	0	62,66
5	69,23	205,00	0	20,15	294,38
6	1,37	4,05	0	0,40	5,82
7	67,86	200,95	0	19,75	288,56

As tabelas 5.2 a 5.4 apresentam quais as quantidades medidas nos oito diferentes casos (CSA1, CSA2, CSA3, CSA4, CSA5, CSA6, CSA7 e CSA8). As tabelas 5.5 a 5.7 apresentam os resultados de classificação de acordo com o programa TCLASS, onde R1 representa a conversão da reação química.

Tabela 5.2 - Quantidades Medidas do Caso CSA1 da Síntese de Amônia

Caso	Corrente	Vazão do Componente				Vazão Total
		A	B	C	D	
CSA1	1	X	nm	-	X	X
	2	X	nm	-	X	X
	3	X	nm	nm	nm	X
	4	-	-	X	-	X
	5	X	X	-	nm	X
	6	X	nm	-	nm	X
	7	nm	nm	-	nm	nm

nm = não-medida

X = medida

- = este componente não faz parte da corrente

Tabela 5.3 – Quantidades Medidas dos Casos CSA2, CSA3, CSA4 CSA5 e CSA6 da Síntese de Amônia

Caso	Corrente	Vazão do Componente				Vazão Total
		A	B	C	D	
CSA2	1	X	nm	-	X	X
	2	X	nm	-	X	X
	3	X	nm	nm	nm	X
	4	-	-	X	-	X
	5	(X)	(X)	-	nm	nm
	6	(X)	nm	-	nm	nm
	7	nm	nm	-	nm	nm
CSA3	1	(X)	nm	-	nm	nm
	2	X	nm	-	X	X
	3	X	nm	nm	nm	X
	4	-	-	X	-	X
	5	(X)	nm	-	nm	nm
	6	nm	nm	-	nm	nm
	7	nm	nm	-	(X)	nm
CSA4	1	nm	nm	-	nm	X
	2	X	nm	-	X	X
	3	X	nm	nm	nm	X
	4	-	-	X	-	X
	5	nm	nm	-	nm	nm
	6	nm	nm	-	nm	nm
	7	nm	nm	-	nm	nm
CSA5	1	nm	nm	-	X	X
	2	X	nm	-	X	X
	3	X	nm	nm	nm	X
	4	-	-	X	-	X
	5	(X)	nm	-	nm	nm
	6	nm	nm	-	nm	nm
	7	nm	nm	-	nm	nm
CSA6	1	X	X	-	X	X
	2	nm	nm	-	nm	nm
	3	X	nm	nm	nm	X
	4	-	-	X	-	X
	5	nm	nm	-	nm	nm
	6	X	X	-	X	X
	7	nm	nm	-	nm	nm

nm = não-medida

X = medida

(X) = somente a concentração é medida

- = este componente não faz parte da corrente

Tabela 5.4 - Quantidades Medidas dos Casos CSA7 e CSA8 da Síntese de Amônia

Caso	Corrente	Vazão do Componente				Vazão Total
		A	B	C	D	
CSA7	1	(X)	nm	-	nm	nm
	2	X	nm	-	X	X
	3	X	nm	nm	nm	X
	4	-	-	X	-	X
	5	(X)	nm	-	nm	nm
	6	(X)	nm	-	nm	nm
	7	nm	nm	-	nm	nm
CSA8	1	nm	nm	-	nm	nm
	2	X	nm	-	nm	X
	3	nm	nm	nm	nm	nm
	4	-	-	X	-	X
	5	(X)	nm	-	nm	nm
	6	(X)	nm	-	nm	nm
	7	(X)	nm	-	nm	nm

nm = não-medida

X = medida

(X) = somente a concentração é medida

- = este componente não faz parte da corrente

Tabela 5.5 - Resultados dos Casos CSA1 e CSA2 da Síntese de Amônia

## Caso CSA1

<i>Quantidades Medidas</i>	Redundantes	Não-Redundantes
Vazões e Concentrações da Categoria 1	A1,T1,A2,D2,T2,A3,T3,C4,T4,A5,B5,T5,A6,T6	D1
Concentrações da Categoria 2	não há	não há
<i>Quantidades Não-Medidas</i>	Observáveis	Não-Observáveis
Vazões da Categoria 2	não há	não há
Vazões da Categoria 3 e Conversão	B1,B2,B3,C3,D3,D5,B6,D6,A7,B7,D7,T7,R1	nenhuma

## Caso CSA2

<i>Quantidades Medidas</i>	Redundantes	Não-Redundantes
Vazões e Concentrações da Categoria 1	A1,T1,A2,D2,T2,A3,T3,C4,T4	D1
Concentrações da Categoria 2	A5,B5,A6	nenhuma
<i>Quantidades Não-Medidas</i>	Observáveis	Não-Observáveis
Vazões da Categoria 2	A5,B5,T5,A6,T6	nenhuma
Vazões da Categoria 3 e Conversão	B1,B2,B3,C3,D3,D5,B6,D6,A7,B7,D7,T7,R1	nenhuma

Tabela 5.6 — Resultados dos Casos CSA3, CSA4, CSA5, CSA6 da Síntese de Amônia

Caso CSA3		
<i>Quantidades Medidas</i>	Redundantes	Não-Redundantes
Vazões e Concentrações da Categoria 1	A2, T2, A3, T3, C4, T4	D2
Concentrações da Categoria 2	A5	A1, D7
<i>Quantidades Não-Medidas</i>	Observáveis	Não-Observáveis
Vazões da Categoria 2	A5, T5	A1, T1, D7, T7
Vazões da Categoria 3 e Conversão	B2, B3, C3, D3, B5, D5, R1	B1, D1, A6, B6, D6, T6, A7, B7
Caso CSA4		
<i>Quantidades Medidas</i>	Redundantes	Não-Redundantes
Vazões e Concentrações da Categoria 1	A2, T2, A3, T3, C4, T4	T1, D2
Concentrações da Categoria 2	não há	não há
<i>Quantidades Não-Medidas</i>	Observáveis	Não-Observáveis
Vazões da Categoria 2	não há	não há
Vazões da Categoria 3 e Conversão	B2, B3, C3, D3, A5, B5, D5, T5, T6, T7, R1	A1, B1, D1, A6, B6, D6, A7, B7, D7
Caso CSA5		
<i>Quantidades Medidas</i>	Redundantes	Não-Redundantes
Vazões e Concentrações da Categoria 1	A2, T2, A3, T3, C4, T4	D1, T1, D2
Concentrações da Categoria 2	A5	nenhuma
<i>Quantidades Não-Medidas</i>	Observáveis	Não-Observáveis
Vazões da Categoria 2	A5, T5	nenhuma
Vazões da Categoria 3 e Conversão	B2, B3, C3, D3, B5, D5, D6, T6, D7, T7, R1	A1, B1, A6, B6, A7, B7
Caso CSA6		
<i>Quantidades Medidas</i>	Redundantes	Não-Redundantes
Vazões e Concentrações da Categoria 1	A1, B1, D1, T1, C4, T4, A6, B6, D6, T6	A3, T3
Concentrações da Categoria 2	não há	não há
<i>Quantidades Não-Medidas</i>	Observáveis	Não-Observáveis
Vazões da Categoria 2	não há	não há
Vazões da Categoria 3 e Conversão	A2, T2, C3, A5, T5, A7, T7, R1	B2, D2, B3, D3, B5, D5, B7, D7

Tabela 5.7 - Resultados dos Casos CSA7 e CSA8 da Síntese de Amônia

Caso CSA7		
<i>Quantidades Medidas</i>	Redundantes	Não-Redundantes
Vazões e Concentrações da Categoria 1	A2, T2, A3, T3, C4, T4	D2
Concentrações da Categoria 2	A5	A1, A6
<i>Quantidades Não-Medidas</i>	Observáveis	Não-Observáveis
Vazões da Categoria 2	A1, T1, A5, T5, A6, T6	nenhuma
Vazões da Categoria 3 e Conversão	B2, B3, C3, D3, B5, D5, A7, T7, R1	B1, D1, B6, D6, B7, D7
Caso CSA8		
<i>Quantidades Medidas</i>	Redundantes	Não-Redundantes
Vazões e Concentrações da Categoria 1	A2, T2, C4, T4	nenhuma
Concentrações da Categoria 2	A5, A6, A7	nenhuma
<i>Quantidades Não-Medidas</i>	Observáveis	Não-Observáveis
Vazões da Categoria 2	A5, T5	A6, T6, A7, T7
Vazões da Categoria 3 e Conversão	A3, C3, T3, R1	A1, B1, D1, T1, B2, D2, B3, D3, B5, D5, B6, D6, B7, D7

b) Processo de Evaporação-Cristalização de Nitrato de Potássio

Para testar o desempenho do programa TCLASS nos casos onde não há reações químicas, utilizou-se um processo de evaporação-cristalização de uma solução de nitrato de potássio (A) em água (B). Os componentes presentes em cada corrente são mostrados no fluxograma da Figura 5.1, a onde evaporação ocorre no nó número 1 e a cristalização no número 2.

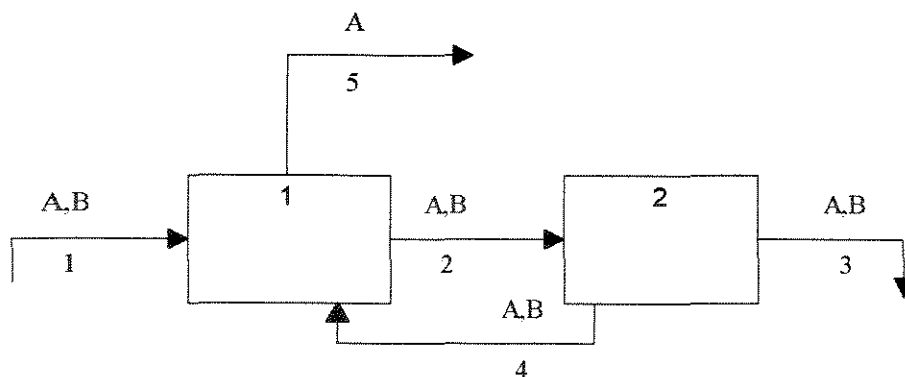


Figura 5.1 - Fluxograma do Processo de Evaporação-Cristalização de Nitrato de Potássio.

A Tabela 5.8 mostra as vazões totais (T) e dos componentes (A e B) de cada corrente, a Tabela 5.9 indica as quantidades medidas em cada um dos casos (EV1 e EV2) e as tabelas 5.10 e 5.11 mostram os resultados obtidos com o programa TCLASS.

Tabela 5.8 — Vazões das Correntes do Caso de Evaporação-Cristalização

Corrente	A	B	T
1	200,00	800,00	1000,00
2	487,45	487,45	947,90
3	199,968	8,332	208,30
4	287,475	479,125	766,60
5	0	791,70	791,70

Tabela 5.9 — Quantidades Medidas nos Casos EV1 e EV2 de Evaporação-Cristalização

Caso	Corrente	Vazão do Compon.		Vazão Total
		A	B	
EV1	1	X	X	X
	2	(X)	nm	nm
	3	(X)	nm	nm
	4	(X)	nm	nm
	5	-	X	X
EV2	1	X	X	X
	2	(X)	(X)	nm
	3	(X)	(X)	nm
	4	(X)	(X)	nm
	5	-	X	X

nm = não-medida

X = medida

(X) = somente a concentração é medida

- = este componente não faz parte da corrente

Tabela 5.10 - Resultados do Caso EV1 de Evaporação-Cristalização

Caso EV1

<i>Quantidades Medidas</i>	Redundantes	Não-Redundantes
Vazões e Concentrações da Categoria 1	A1,B1,T1,B5,T5	nenhuma
Concentrações da Categoria 2	A3	A2,A4
<i>Quantidades Não-Medidas</i>	Observáveis	Não-Observáveis
Vazões da Categoria 2	A2,T2,A3,T3,A4,T4	nenhuma
Vazões da Categoria 3	B2,B3,B4	nenhuma

Tabela 5.11 - Resultados do Caso EV2 de Evaporação-Cristalização

Caso EV2		
<i>Quantidades Medidas</i>	Redundantes	Não-Redundantes
Vazões e Concentrações da Categoria 1	A1,B1,T1,B5,T5	nenhuma
Concentrações da Categoria 2	A2,B2,A3,B3,A4,B4	nenhuma
<i>Quantidades Não-Medidas</i>	Observáveis	Não-Observáveis
Vazões da Categoria 2	A2,B2,T2,A3,B3,T3,A4,B4,T4	nenhuma
Vazões da Categoria 3	não há	não há

Conclusões sobre as aplicações deste programa:

Para avaliar o desempenho do programa de classificação de dados foram utilizados dois processos, sendo que num deles (síntese de amônia) ocorre uma reação química, havendo, portanto, pelo menos uma quantidade da Categoria 3 (a conversão) e no outro (evaporação-cristalização de nitrato de potássio) não. Para cada um deles, foram gerados casos diferentes, em função da existência, ou não, de medições em suas correntes.

Ao gerar os diversos casos, procurou-se que houvesse casos:

- com correntes das três categorias;
- com correntes das categorias 1 e 3;
- com correntes das categorias 1 e 2.

Além disso, procurou-se que fossem percorridas todas as saídas possíveis do programa, havendo, entre os resultados da classificação dos diversos casos:

- vazões e concentrações da Categoria 1 redundantes e não-redundantes;
- concentrações da Categoria 2 redundantes e não-redundantes;
- vazões da Categoria 2 observáveis e não-observáveis;
- vazões da Categoria 3 e conversão observáveis e não-observáveis.

Em vista disto, e considerando que foram classificadas de forma correta as variáveis dos casos relatados, acredita-se que este programa de classificação de variáveis de processo poderá ser aplicado com sucesso a diversos outros processos.

## 6. CONCLUSÕES E SUGESTÕES



Os estudos feitos para realizar este trabalho e as aplicações dos programas desenvolvidos permitiram traçar as seguintes conclusões e sugestões.

## 6.1 Conclusões

Este trabalho objetivava o estudo e desenvolvimento de metodologias e procedimentos computacionais que permitissem fazerem-se correções nos dados medidos de um processo, de modo que as leis de conservação e outras restrições de um processo fossem obedecidas, e que possibilitassem a obtenção de estimativas para quantidades não-medidas observáveis.

Este objetivo foi atingido, pois resultaram, como fruto deste trabalho, procedimentos, que deram origem ao programa de reconciliação de dados de processo (Programa RECON) e ao programa de retificação de dados de processo (Programa RETIF). O primeiro deve ser aplicado naqueles casos em que se tem certeza de que as quantidades medidas são sujeitas somente a erros aleatórios, já o segundo pode ser empregado quando houver suspeita de que há medidas contendo erros grosseiros, desde que os processos estejam no estado estacionário e sejam sujeitos a restrições lineares. Em ambos, a vazão total de cada corrente é sempre considerada, garantindo que a soma das vazões corrigidas dos componentes de uma corrente seja igual a sua vazão total corrigida, e contribuindo para aumentar a redundância.

Além disso, deste trabalho também resultou um procedimento para classificação de variáveis de processo, que originou o Programa TCLASS, no qual se emprega projeção matricial, que classifica as quantidades medidas em redundantes e em não-redundantes e as quantidades não-medidas em observáveis e não-observáveis. A classificação de dados é empregada quando se quer saber quais as quantidades não-medidas que podem ser estimadas e quais das quantidades medidas que podem ser estimadas mesmo quando falharem os instrumentos responsáveis por sua medição. Ela pode, portanto, ser usada na análise e no projeto de sistemas de monitoramento de processo.

Na reconciliação de dados, emprega-se o método dos mínimos quadrados ponderados e os multiplicadores de Lagrange, para a obtenção dos ajustes das quantidades medidas. Quando há quantidades não-medidas, no sistema a reconciliar, emprega-se a projeção matricial para diminuir a dimensão do problema de minimização. Estes mesmos métodos são empregados para o obtenção dos ajustes na retificação de dados.

Para detectar e identificar medidas contendo erros grosseiros, empregam-se, na retificação de dados, três testes estatísticos, que são o teste global, o teste de desbalanço nodal e o teste de medida, pois de acordo com vários pesquisadores, como Madron (1985b) e Crowe (1983), por exemplo, é aconselhável o uso de mais de um teste estatístico para melhorar a possibilidade de encontrar corretamente estas medidas.

Tanto o Programa RECON como o Programa RETIF foram aplicados, com sucesso, a diferentes processos, envolvendo balanços de um ou de vários componentes, ocorrendo ou não reações químicas, com adição ou não de restrições de corrente, como a razão de divisão conhecida de um divisor de fluxo, assegurando que eles poderão ser aplicados a tantos outros casos.

Para avaliar o funcionamento do Programa RECON, estudou-se a reconciliação de dados, sujeitos a restrições lineares, contendo somente erros aleatórios, de cinco sistemas

diferentes, que são um sistema de síntese de amônia, um reator químico, onde ocorrem duas reações químicas, um sistema de síntese de ácido nítrico, uma rede de vapor e de um circuito de moagem. Através desta diversificação, procurou-se fazer com que este programa esteja apto para fornecer estimativas para as quantidades medidas e para as quantidades não-medidas observáveis de diferentes sistemas, envolvendo balanços de um único ou multicomponentes, com reação química ou não, com ou sem restrições adicionais, e até com duas reações ocorrendo num mesmo vaso.

O programa RECON possibilitou a obtenção dos ajustes, fornecendo valores corrigidos para as quantidades medidas redundantes ou que tinham covariância com outra quantidade medida e estimativas para as quantidades não-medidas observáveis.

Além disso, este programa foi aplicado aos mesmos dados de entrada do caso de síntese de amônia (que não inclui as vazões das correntes totais) relatados por Crowe (1983 e 1988) e do caso relatado por Ripps (1965), obtendo-se, em cada um, os mesmos resultados que estes pesquisadores.

Para avaliar o funcionamento do Programa RETIF, estudou-se a retificação de dados de:

- cinco casos diferentes de um sistema de síntese de amônia, dois deles contendo somente um erro grosseiro, dois contendo dois erros grosseiros e um contendo três erros deste tipo;

- quatro casos de um sistema de síntese de ácido nítrico, um deles contendo somente erros aleatórios, dois contendo um erro grosseiro, e outro contendo erros grosseiros em duas quantidades; além disso, variou-se o número de quantidades medidas;

- três casos de um circuito de moagem, incluindo um ou nenhum erro grosseiro.

Embora os valores dos erros grosseiros tenham sido limitados a 10 ou 20% dos valores medidos, foi possível detectar e identificar as fontes deste tipo de erro na maioria dos casos estudados, graças ao uso em conjunto dos três testes estatísticos. Quando isto não aconteceu as medidas com esses erros estavam entre as possíveis responsáveis, com exceção de um caso do circuito de moagem. Conforme Madron (1985b), Holly e Crowe (1989), Verneuil et alii.(1992), entre outros, este tipo de resultado era esperado.

O programa de retificação de dados, quando aplicado ao mesmos dados de entrada do caso de síntese de amônia relatado por Crowe (1988), que não inclui as vazões das correntes totais, gera os mesmos resultados por ele obtidos.

Para verificar o funcionamento do Programa TCLASS, estudou-se a classificação de variáveis medidas e não-medidas de oito casos diferentes de um sistema de síntese de amônia, onde ocorre uma reação química, e de dois casos de um sistema de evaporação-cristalização de nitrato de potássio, onde não há reação química. Os oito casos do primeiro sistema abrangem situações diferentes, onde há quantidades não-medidas das Categorias 2 e 3 ou somente da Categoria 3 e os casos do segundo sistema se referem a situações diferentes, onde há, somente, quantidades não-medidas da Categoria 2. Procurou-se, através desses casos, gerar um conjunto tal que varresse as diferentes rotas do fluxograma deste programa, até a solução ou fim.

Este programa pode ser usado tanto para análise de sistemas de monitoramento de processos existentes, como para o projeto de novos sistemas, pois sua aplicação consecutiva, a um mesmo sistema com as vazões colocadas em diferentes categorias, possibilita definir qual o conjunto de quantidades que deve ser medido para que se tenha o conjunto desejado de vazões medidas redundantes e de vazões não-medidas observáveis.

É válido observar que os valores máximos arbitrados para o número de nós, correntes totais, espécies químicas, reações químicas e restrições adicionais dos programas desenvolvidos, podem ser aumentados, desde que haja disponibilidade de memória do computador e que sejam previstas a ampliação dos valores estatísticos tabelados e a modificação da função que dá os nomes das vazões, quando for o caso.

Por fim, num contexto mais amplo, pode-se dizer, que, ao realizar este trabalho, além de se desenvolver capacitação técnica no País, nesta área, está-se participando do esforço que a comunidade científica internacional vem realizando para obter procedimentos racionais e confiáveis para o aprimoramento de dados de processo e, conseqüentemente, das ações neles fundamentadas.

Além disso, têm-se a convicção de que este trabalho se constitui de ponto de partida importante para outros estudos nesta área e que a aplicação dos programas RECON, RETIF e TCLASS a casos reais ou simulados possibilitarão a realização de estudos mais específicos como sobre o problema da detecção e identificação de erros grosseiros, por exemplo.

## 6.2 Sugestões

-Aplicar os procedimentos de reconciliação e/ou de retificação de dados a casos reais, obtendo a matriz variância-covariância, a partir de conjuntos de dados medidos.

-Incluir o balanço de energia no conjunto de restrições, considerando a vazão de entalpia como a vazão de um componente adicional.

-Inserir modificações nos programas RECON e RETIF, que possibilitem o tratamento de sistemas sujeitos a restrições não-lineares.

-Incluir outros testes estatísticos, além dos presentes no Programa RETIF, como o teste de razão de probabilidade condicional, que, de acordo com Narasimham e Mah (1987), permite diferenciar as diversas fontes de erros grosseiros e um teste de potência máxima aplicado às restrições originais descrito por Crowe (1992), por exemplo.

-Ampliar o Programa TCLASS para que permita a classificação de concentrações não-medidas.

## APÊNDICE A. MÉTODOS MATEMÁTICOS

A reconciliação de dados de um processo, no estado estacionário, em que todos os parâmetros das correntes em questão são medidos, sujeito a restrições lineares, em que os erros de medida são variáveis aleatórias, que seguem uma distribuição normal (com média zero e matriz variância-covariância conhecida, positiva por definição), é resolvida, via de regra, utilizando o método dos mínimos quadrados ponderados para a obtenção dos ajustes, solucionado pelo método dos multiplicadores de Lagrange.

Nos casos em que nem todos os parâmetros das correntes são medidos, é usual utilizarem-se estratégias para diminuir a dimensão do problema. Neste trabalho, será utilizada a projeção matricial para obter o conjunto reduzido de equações de balanço, a partir dos balanços iniciais. A matriz projeção pode ser construída diretamente e remove as quantidades não-medidas, obtendo-se o conjunto reduzido de balanços, que deverá ser resolvido. As medidas não-redundantes são, em geral, ajustadas se elas tiverem covariância não-zero com uma medida redundante.

Esta abordagem permite tratar casos em que há correntes parcialmente medidas, reações químicas e mesmo conjuntos de equações restritivas lineares com matrizes arbitrárias.

Quando há erros grosseiros ou sistemáticos, eles devem ser eliminados, preferencialmente, antes de efetuar os procedimentos de reconciliação. Para tanto, utilizam-se estatísticas-teste que servem para detectar sua presença e identificar a(s) fonte(s) de erros extremos.

### A.1. Método dos Mínimos Quadrados

Em livros de álgebra linear aplicada, de métodos numéricos e de estatística aplicada à engenharia, como os que constam das referências bibliográficas, é comum se utilizar o método dos mínimos quadrados para ajustamento de dados experimentais, contendo erros aleatórios, a um modelo. Em outras palavras, tem-se uma relação

$$\text{observação} = \text{modelo} + \text{erro}$$

Mais especificamente, considere-se que o modelo tem a seguinte forma

$$b_i \approx x_1 f_1(t_i) + x_2 f_2(t_i) + \dots + x_n f_n(t_i)$$

onde as funções  $f_j(t_i)$  são as funções de modelo dadas. No caso onde o modelo é uma linha reta, supõe-se que

$$b_i \approx x_1 + x_2 t_i$$

de forma que  $f_1(t) = 1$  e  $f_2(t) = t$ . Os coeficientes  $x_j$  são chamados de parâmetros do modelo e são esses números que devem ser determinados. Se o modelo fosse perfeito e se não houvesse erros de medida, poder-se-ia substituir  $\approx$  por  $=$ , mas isso raramente ocorre.

O modelo aqui é chamado linear, porque ele é uma combinação linear das funções de modelo. Isto não significa que as funções de modelo sejam lineares. A idéia crucial é que o

modelo é uma função linear dos parâmetros  $x$ . Pode-se expressar o problema de ajustamento de dados de forma mais concisa, usando notação matricial/vetorial. Define-se a matriz  $A$ ,  $n \times m$ , por

$$a_{ij} = f_j(t_i)$$

e consideram-se  $b$  e  $x$  os vetores das observações e parâmetros, respectivamente. Então, a situação acima pode ser escrita como

$$b \approx Ax \quad \text{ou} \quad b - Ax \approx 0$$

onde  $b - Ax$  é chamado de vetor dos resíduos.

Os parâmetros serão escolhidos para fazer os resíduos  $b - Ax$  tão pequenos quanto possível. Um modo comum para isto é resolver

$$\min_x \sum_{j=1}^m \left[ (b - Ax)_j \right]^2$$

Como se está minimizando a soma dos quadrados, este procedimento é chamado ajustamento de dados por mínimos quadrados. Usando normas vetoriais, o problema é equivalente a resolver

$$\min_x \|b - Ax\|_2^2 = (b - Ax)^T (b - Ax)$$

onde é usada a norma-2 Euclidiana.

Kahaner et alii. (1988) observam que se o número de dados experimentais e de funções do modelo forem iguais, a matriz  $A$  é quadrada. Se  $A$  é, além disso, não-singular, a solução do problema de mínimos quadrados é o interpolante, isto é, o resíduo é zero. Então, apresentado na forma matricial, o ajustamento de dados por mínimos quadrados inclui um caso especial do problema de resolução de equações lineares  $Ax=b$ , se  $A$  for quadrada e não-singular.

O problema de mínimos quadrados pode ser interpretado graficamente como a minimização da distância vertical dos pontos experimentais ao modelo. Subjacente à esta idéia está a suposição de que todos os erros da aproximação correspondam a erros nas observações  $b_i$ . Se houver erros nas variáveis independentes  $t_i$ , então pode ser mais apropriado minimizar a distância Euclidiana dos dados ao modelo, que corresponde à minimização da distância ortogonal dos pontos experimentais ao modelo.

Em muitas aplicações, os pontos experimentais não têm a mesma importância. Isto se deve ao fato de se saber que uns são mais precisos que os outros. Esta informação pode ser incorporada ao problema de mínimos quadrados, ponderando os dados experimentais. O problema a resolver passa a ser

$$\min_x \sum_{j=1}^m \left[ p_j (b - Ax)_j \right]^2$$

que é um problema de mínimos quadrados ponderados, onde  $p_j$  é um peso que reflete a importância do ponto experimental. Quanto maior a importância do dado, maior a magnitude

do peso. Se o erro no  $j$ -ésimo ponto experimental é  $e_j$ , então se escolhe  $p_j=1/e_j$ , por exemplo. Então quanto menor o erro, maior o peso. A ponderação pode melhorar as propriedades estatísticas da solução.

A aplicação do método dos mínimos quadrados ponderados, para o ajustamento de dados de processos químicos, tem sido relatada, em diversos estudos publicados em periódicos. Em todos os estudos de reconciliação de dados de processos químicos citados no Capítulo 2, emprega-se como fator ponderante o inverso de suas variâncias, de modo que as medidas com maior dispersão em torno de média (e, portanto, mais passíveis de erro) têm menor influência na determinação dos ajustes. Himmelblau (1969) observa que este ou outro esquema de ponderação pode ser empregado.

Existe muitos algoritmos diferentes para calcular um conjunto de coeficientes, que dá a soma mínima dos quadrados. De acordo com Kahaner et alii.(1988), pode ser difícil de se tomar uma decisão antecipada sobre qual o método a ser empregado, pois a solução correta pode depender de fatores tais como a condição da matriz, o tamanho dos resíduos na solução e a forma dos dados. Acrescenta que programas computacionais baseados na fatoração ortogonal são considerados a melhor escolha para uso em biblioteca de programas de utilidade geral.

Quando o número de dados experimentais é maior do que o número de parâmetros, diz-se que o sistema de equações é superdeterminado. Se o problema é subdeterminado e existem múltiplas soluções para o sistema de equações, então pode ser importante determinar uma solução particular do sistema.

Uma aplicação importante é na otimização condicionada. Considere-se a obtenção de uma solução para

$$\begin{array}{l} \text{minimizar} \\ \text{sujeita à } x_1 + x_2 = 1 \end{array} \quad f(x_1, x_2)$$

Esta condição permite simplificar este problema, fazendo  $x_2=1-x_1$ , que dá

$$\begin{array}{l} \text{minimizar} \\ x_1 \end{array} \quad f(x_1, 1-x_1)$$

Além desse problema ser não condicionado, ele tem o número de variáveis diminuído de um.

O caso geral, que se constitui num problema  $n$ -dimensional, é representado por

$$\begin{array}{l} \text{minimizar} \\ \text{sujeita à } Ax=b \end{array} \quad f(x)$$

Este tipo de problema é semelhante ao que ocorre no ajustamento de dados de processos químicos, em que a função a minimizar é a soma ponderada dos quadrados dos ajustes e as restrições podem ser dadas por equações de conservação de massa ou de energia, por exemplo. A solução é obtida, geralmente, utilizando o método dos multiplicadores de Lagrange.

## A.2. Método dos Multiplicadores de Lagrange

Um modo conveniente de encontrar o extremo (não sobre uma fronteira) de uma função sujeita a condições de igualdade é usar os multiplicadores Lagrangianos, conforme descreve Himmelblau (1969). No cálculo do mínimo (ou do máximo) de uma função de várias variáveis independentes  $f(x_1, x_2, \dots, x_n)$ , sabe-se que a solução de

$$\frac{\partial f}{\partial x_1} = 0, \quad \frac{\partial f}{\partial x_2} = 0, \quad \dots, \quad \frac{\partial f}{\partial x_n} = 0 \quad (\text{a.1})$$

dá os valores de  $x_j$ , que estão nos extremos. Considerando, entretanto, que os valores de  $x_j$  estejam unidos por algumas relações de igualdade (condições), que são expressas em geral como

$$\begin{aligned} g_1(x_1, x_2, \dots, x_n) &= 0 \\ g_2(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ g_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \quad (\text{a.2})$$

Por exemplo, considere-se que a soma das frações molares de uma mistura é igual à unidade. Procurar resolver cada uma das eq.a.2 para isolar um único  $x_j$ , de forma que ele possa ser eliminado por substituição na eq.a.1, não é adequado, usualmente. Ao invés disso, proceda-se conforme descrito a seguir

Sabe-se que num extremo a derivada total de  $f(x_1, x_2, \dots, x_n)$  vai a zero, mesmo se as variáveis  $x_1, x_2, \dots, x_n$  não são independentes:

$$df = 0 = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 + \dots + \frac{\partial f}{\partial x_n} dx_n \quad (\text{a.3})$$

Assim como,

$$\begin{aligned} \frac{\partial g_1}{\partial x_1} dx_1 + \frac{\partial g_1}{\partial x_2} dx_2 + \dots + \frac{\partial g_1}{\partial x_n} dx_n &= 0 \\ \frac{\partial g_2}{\partial x_1} dx_1 + \frac{\partial g_2}{\partial x_2} dx_2 + \dots + \frac{\partial g_2}{\partial x_n} dx_n &= 0 \\ &\vdots \\ \frac{\partial g_p}{\partial x_1} dx_1 + \frac{\partial g_p}{\partial x_2} dx_2 + \dots + \frac{\partial g_p}{\partial x_n} dx_n &= 0 \end{aligned} \quad (\text{a.4})$$

Se as eq.a.4 forem multiplicadas, respectivamente, por  $\lambda_1, \lambda_2, \dots, \lambda_p$  e os produtos resultantes forem adicionados à eq.a.3, obtém-se





$$f(x_1, x_2, \dots, x_n)$$

cujas variáveis são sujeitas a  $p$  relações restritivas:

$$g_i(x_1, x_2, \dots, x_n) = 0, \quad i = 1, 2, \dots, p \quad (\text{a.8})$$

forma-se a função

$$F = f + \sum_{i=1}^p \lambda_i g_i$$

e se determinam os parâmetros  $\lambda_i$  e os valores de  $x_1, x_2, \dots, x_n$ , através de  $n$  equações:

$$\frac{\partial F}{\partial x_j} = 0, \quad j = 1, 2, \dots, n$$

e das  $p$  eq.a.8.

### A.3. Aplicação dos Mínimos Quadrados Ponderados Resolvidos Pelos Multiplicadores de Lagrange a um Problema de Reconciliação de Dados de Processo

Considere-se a reconciliação de dados de vazão total de um processo constituído por  $n$  equipamentos, tanques ou junções, onde todas as correntes são medidas. As vazões medidas são representadas por  $\tilde{x}_j$  e as vazões reais por  $x_j$ , correspondentes à corrente  $j$  ( $j=1,2,\dots,m$ ). A partir dos valores medidos, quer-se obter estimativas ( $\hat{x}_j$ ) desses valores, de forma que os balanços materiais nodais sejam satisfeitos. Supondo que o modelo é o seguinte:

$$e_j = \tilde{x}_j - x_j$$

$$d_j = \hat{x}_j - \tilde{x}_j$$

onde  $e_j$  é o erro de medida e  $d_j$  é o ajuste da medida.

O problema, na forma matemática, é minimizar

$$f(x_j) = \sum_j \frac{1}{\sigma_j^2} (\hat{x}_j - \tilde{x}_j)^2 \quad (\text{a.9})$$

sujeita às restrições:

$$g_i(x_j) = \sum_i a_{ij} \hat{x}_j = 0 \quad (\text{a.10})$$

onde:

$\hat{x}_j$  = valor corrigido da  $j$ -ésima medida

$\tilde{x}_j$  = valor medido da  $j$ -ésima medida

$\sigma_j^2$  = a variância do erro da  $j$ -ésima medida

$a_{ij}$  = o coeficiente da  $j$ -ésima medida na  $i$ -ésima equação de balanço de massa (ou calor)

Então, forma-se a função

$$F = f + \sum_{i=1}^p \lambda_i g_i$$

que é minimizada em relação a  $x_j$  e  $\lambda_i$ . Determinam-se os valores de  $x_j$  e de  $\lambda_i$ , resolvendo as seguintes equações como um conjunto de equações simultâneas:

$$\begin{aligned} \frac{\partial F}{\partial x_j} \left[ f(x_j) + \sum_i \lambda_i g_i(x_j) \right] &= 0 \\ \frac{\partial F}{\partial \lambda_i} \left[ f(x_j) + \sum_i \lambda_i g_i(x_j) \right] &= g_i(x_j) \end{aligned} \quad \text{(a.11)}$$

onde  $\lambda_i$  representam os multiplicadores de Lagrange.

Em notação matricial, o problema a resolver é

$$\left[ \begin{array}{c|c} D^{-1} & A^T \\ \hline A & 0 \end{array} \right] \begin{bmatrix} \hat{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} a' \\ 0 \end{bmatrix}$$

onde:

$D$  = matriz diagonal com elementos  $1/2\sigma_j^2$

$A$  = matriz dos coeficientes  $a_{ij}$

$\hat{x}$  = vetor dos  $\hat{x}_j$

$\lambda$  = vetor dos  $\lambda_i$

$a'$  = vetor  $[2\tilde{x}_j/\sigma_j^2]$

cuja solução, após certas transformações, pode ser expressa como

$$\begin{aligned} \lambda &= (ADA^T)^{-1} A\tilde{x} \\ \hat{x} &= \tilde{x} - DA^T \lambda \end{aligned}$$

onde  $\tilde{x}$  é o vetor dos  $\tilde{x}_j$ , conforme apresentado por Kuehn e Davidson (1961).

## APÊNDICE B. CONCEITOS E TESTES ESTATÍSTICOS

Como as características estatísticas dos erros de medida são de fundamental importância para validar a base estatística do procedimento de reconciliação de dados de processo, julgou-se necessária a inclusão de alguns conceitos básicos de probabilidade e princípios de estatística pertinentes. Soma-se a isso, o fato da maioria dos esquemas de detecção e identificação de erros grosseiros serem baseados em testes estatísticos.

O conteúdo deste item se limita a buscar uma melhor compreensão dos assuntos relevantes ao objetivo desse trabalho. Foi extraído de livros textos de estatística e de estatística aplicada à engenharia, listados nas referências bibliográficas e de artigos sobre retificação de dados de processo.

### B.1. Conceitos Básicos de Probabilidade

**Experimento:** alguma coisa que resulta numa observação. Pode ou não exigir ação da parte do experimentador.

**Resposta (evento simples, ponto amostral):** um dos conjuntos de observações possíveis, que resulta de um experimento. Uma e somente uma resposta resulta para uma realização do experimento.

**Espaço amostral:** o conjunto de todas as possíveis respostas que podem resultar de um experimento.

**Evento:** um subconjunto de um espaço amostral. Diz-se que um evento ocorre se qualquer resposta do evento ocorre.

**Variável aleatória:** é um número (ou vetor) determinado por uma resposta, isto é, uma função definida em pontos do espaço amostral. Uma variável aleatória pode ser discreta (como no lançamento de um dado) ou contínua (como na medição de temperatura).

**Distribuição ou função contínua de probabilidade (ou função densidade):** é a função  $f(x)$  cuja integral (tomada entre  $-\infty$  e  $A$ ) dá a probabilidade de uma variável aleatória  $X$  ser menor ou igual a algum número  $A$ , num espaço amostral, representada por  $P(X \leq A)$ . A função  $f(x)$  tem as seguintes propriedades: é positiva em todo o espaço amostral e é igual a 1, quando somada em todo o espaço amostral, pois é distribuída sobre todo este espaço. Pode ser relatada na forma de um gráfico, mas, na prática, calculam-se seu primeiro e segundo momentos, que são, respectivamente, o valor esperado e a variância, pois são considerados suficientes para descrever uma distribuição.

**Função distribuição acumulativa:** é a função  $F(x)$ , onde  $x$  é uma variável aleatória (discreta ou contínua), definida por  $F(A) = P(x \leq A)$ .

**Valor esperado (expectativa, significado):** é a média ponderada dos possíveis valores de  $X$ , cada valor ponderado por sua probabilidade. É representada por  $E(X)$  ou  $\mu_x$ , sendo definida por

$$E(X) = x_1 f(x_1) + x_2 f(x_2) + \dots + x_n f(x_n)$$

$$E(X) = \sum_i x_i P(X = x_i), \quad \text{para } i=1, 2, \dots, n.$$

As propriedades do valor esperado são descritas por Beck e Arnold (1977).

*Variância:* é o conceito que mede a expansão ou dispersão de X. É representada por  $\text{Var}(X)$  e por  $\sigma^2$  e é definida por

$$\text{Var}(X) = \sum (x_i - \mu)^2 f(x_i) = E[(X - \mu)^2] \text{ ou}$$

$$\text{Var}(X) = E(X^2) - \mu_x^2$$

As propriedades da variância podem ser encontradas em Beck e Arnold (1977).

*Desvio-padrão:* é a raiz quadrada não-negativa da variância. É representado por  $\sigma$ .

*Variável aleatória padronizada:* seja x a variável aleatória com significado  $\mu$  e desvio-padrão  $\sigma$ . A variável aleatória padronizada (z), que corresponde a x é definida por  $z = (x - \mu)/\sigma$ . Uma variável padronizada tem valor esperado igual a 0 e variância igual a 1 (desvio-padrão igual a 1).

*Teste amostral simples:* constitui-se de uma sucessão de leituras tomadas sob condições idênticas (mesmo observador e mesmo instrumento de medição), exceto o tempo.

*Teste multiamostrai:* ocorre quando são feitas medições repetidas de uma quantidade, utilizando diferentes instrumentos de medição e diferentes observadores. As trocas, tanto de observador como de instrumento, causam mudanças na distribuição dos erros e, conseqüentemente, tem-se um conjunto de variáveis aleatórias.

*Distribuição multivariada:* é a distribuição conjunta de duas ou mais variáveis aleatórias definidas no mesmo espaço amostral, própria de um teste multiamostrai. Um exemplo é a distribuição associada com a observação simultânea de temperatura, pressão, direção e velocidade do vento. A função distribuição de uma distribuição bivariada é  $P(X \leq x, Y \leq y)$  e é, usualmente, simbolizada por  $F_{X,Y}(x, y)$

*Valor verdadeiro de uma variável:* é aquele que seria obtido na medição se não houvesse característica estocástica (aleatória) associada com a medição.

*Erro aleatório:* é um erro que representa a diferença entre o valor medido da variável aleatória e seu valor verdadeiro.

*Erro sistemático:* é um erro introduzido continuamente, devido, por exemplo, a erro de calibração, desvios instrumentais, erro de técnica e, também, devido à representação inadequada de um processo (como a não-consideração da ocorrência de vazamentos, depósitos).

*Variáveis aleatórias independentes:* um número finito de variáveis aleatórias X, Y, ..., Z, num espaço amostral definido, são consideradas independentes se:

$$\begin{aligned} P(X = x_i, Y = y_j, \dots, Z = z_k) &= \\ &= P(X = x_i)P(Y = y_j) \dots P(Z = z_k) \end{aligned}$$

para quaisquer valores de  $x_i, y_j, \dots, z_k$ . Elas têm as seguintes propriedades:

$$i) E(XY) = E(X) E(Y)$$

$$ii) \text{Var}(X+Y) = \text{Var}(X) + \text{Var}(Y)$$

$$iii) \text{Cov}(X, Y) = 0$$

onde  $\text{Cov}(X, Y) = \text{covariância}$ .

*Covariância*: é o conceito que relaciona duas variáveis aleatórias, X e Y, definidas num mesmo espaço amostral, representada por  $\text{Cov}(X, Y)$ . É definida por

$$\text{Cov}(X, Y) = E[(X - \mu_x)(Y - \mu_y)]$$

ou equivalentemente,

$$\text{Cov}(X, Y) = E(XY) - \mu_x \mu_y$$

*Coefficiente de correlação*: fornece a correlação entre duas variáveis aleatórias, X e Y, definidas num mesmo espaço amostral, representada por  $\rho_{X,Y}$ . É definido por

$$\rho_{X,Y} = \text{Cov}(X, Y) / \sigma_x \sigma_y$$

As propriedades da covariância e do coeficiente de correlação são encontradas em Beck e Arnold (1977).

*Autocovariância*: é a covariância entre variáveis aleatórias conjuntas  $X(t_1)$  e  $X(t_2)$ , que representam pontos sobre um mesmo caminho em diferentes tempos.

*Covariância cruzada*: é a covariância entre dois processos estocásticos contínuos, sendo que um deles é referente a um ponto no tempo  $X(t_1)$  e o outro a um ponto e tempo possivelmente diferentes,  $Y(t_2)$ .

## B 2. A Variância e a Distribuição de Erros Aleatórios

Se uma medição experimental é repetida um número de vezes, os valores registrados das quantidades medidas diferem, quase sempre, uns dos outros. A dispersão de medições sucessivas de uma quantidade  $x$  é comumente expressa em termos da variância ou do desvio-padrão do conjunto de medidas. Essas quantidades são definidas de forma que sejam úteis na estimativa da probabilidade de ocorrência de erros aleatórios de intensidade definida nas medidas.

A variância da amostra é simplesmente o desvio quadrado médio, de  $n$  valores medidos de  $x$ , em relação à média da amostra  $\bar{x}$ :

$$\begin{aligned}\sigma_a^2 &= \frac{\sum^n (x - \bar{x})^2}{n} \\ &= \frac{\sum^n x^2 - \left[ \left( \sum x \right)^2 / n \right]}{n}\end{aligned}\quad (b.1)$$

onde a média ( $\bar{x}$ ) é obtida por

$$\bar{x} = \frac{\sum^n x}{n} \quad (b.2)$$

O desvio-padrão é obtido através da variância:

$$\sigma_a = \sqrt{\sigma_a^2} \quad (b.3)$$

Na definição da variância das eq.b.1, desvios positivos e negativos em torno da média não se cancelam uns aos outros. A última forma desta equação, que define a variância, é a mais conveniente, quando cálculos reais estão sendo feitos.

O valor da variância se torna mais confiável quanto mais medições forem obtidas e a exatidão real do procedimento de medição é indicada pelo valor da variância, calculada a partir de uma quantidade muito grande de dados. Quando o número de pontos experimentais obtido se tornar infinito, o conjunto infinito é chamado uma população de valores. Para tal população, a média da população  $\mu$  é definida como:

$$\mu = \lim_{n \rightarrow \infty} \frac{\sum^n x}{n} \quad (b.4)$$

e a variância da população  $\sigma^2$  é definida como

$$\begin{aligned}\sigma^2 &= \lim_{n \rightarrow \infty} \frac{\sum^n (x - \mu)^2}{n} \\ &= \lim_{n \rightarrow \infty} \frac{\sum^n x^2 - \left( \sum x \right)^2 / n}{n}\end{aligned}\quad (b.5)$$

O desvio-padrão da população é a raiz quadrada (não-negativa) da variância da população. A média da população é o melhor, ou mais provável, valor de  $x$ , desde que as variações de  $x$  sejam resultantes de efeitos pequenos, aleatórios, independentes e aditivos. Se, entretanto, erros de método e/ou erros não-aleatórios forem inerentes às medidas, então a média pode diferir substancialmente do valor verdadeiro de  $x$ . Se em um conjunto infinito de dados, as variações de  $x$  forem aleatórias, diz-se que a distribuição de valores de  $x$  em torno da média é uma Distribuição Normal ou Gaussiana.



*Distribuição Normal ou Gaussiana.* A variância da população, definida pela eq.b.5, é baseada numa amostra hipotética, contendo um número infinito de replicatas de uma medida. Todavia, para objetivos práticos, é necessário lidar com um número finito de valores da quantidade em questão. A média da amostra ( $\bar{x}$ ) é a melhor estimativa da média da população ( $\mu$ ). Entretanto, a variância da amostra ( $\sigma_a^2$ ) não é a melhor estimativa da variância da população ( $\sigma^2$ ). A melhor estimativa da variância da população é dada por

$$s^2 = \frac{\sum^n (x - \bar{x})^2}{n - 1} = \frac{\sum^n x^2 - \left(\sum^n x\right)^2 / n}{n - 1} = \frac{n}{n - 1} \sigma_a^2 \quad (\text{b.6})$$

Entretanto, é importante notar que os dados de processo, a serem ajustados, muitas vezes não são obtidos através de medidas diretas. Como exemplo, pode-se citar o caso da medida de vazão de um componente de uma corrente, como sendo dada pelo produto de duas quantidades (a concentração do componente na corrente pela vazão total desta corrente). Para estimar a variância da vazão de um determinado componente, pode ser empregado o método descrito por Kline e McClintock, que de acordo com Beck e Arnold (1977), vem sendo usado há décadas para estimar a variância de uma função, que não é linear, de duas ou mais variáveis aleatórias independentes, descrito a seguir.

Considere-se uma variável aleatória  $Z = XY$ , dada pelo produto de duas variáveis aleatórias independentes. Estas variáveis podem ser escritas como a soma de seu valor esperado com o erro:  $X$  como  $\mu_x + e_x$ ,  $Y$  como  $\mu_y + e_y$  e  $Z$  como  $\mu_z + e_z$ . Então,

$$\begin{aligned} \mu_z + e_z &= (\mu_x + e_x)(\mu_y + e_y) = \\ &= \mu_x \mu_y + \mu_y e_x + \mu_x e_y + e_x e_y \end{aligned} \quad (\text{b.7})$$

Como  $X$  e  $Y$  são independentes,  $\mu_z = \mu_x \mu_y$ . Subtraindo  $\mu_z$  dos dois lados, tem-se

$$e_z = \mu_y e_x + \mu_x e_y + e_x e_y \quad (\text{b.8})$$

Se os erros ( $e_x$  e  $e_y$ ) forem, com certeza, muito pequenos em comparação com os valores esperados ( $\mu_x$  e  $\mu_y$ ), tem-se, aproximadamente,

$$\begin{aligned} e_z &= \mu_x e_y + \mu_y e_x \\ \sigma_z^2 &= E(e_z^2) = \mu_x^2 \sigma_y^2 + \mu_y^2 \sigma_x^2 \quad \text{ou} \\ \left(\frac{\sigma_z}{\mu_z}\right)^2 &= \left(\frac{\sigma_x}{\mu_x}\right)^2 + \left(\frac{\sigma_y}{\mu_y}\right)^2 \end{aligned} \quad (\text{b.9})$$

Tudo isso sugere que, se  $z = f(x,y)$ ,

$$dz = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy \quad (\text{b.10})$$

pode-se considerar como uma aproximação para  $\sigma_z^2$

$$\sigma_z^2 \approx \left[ \frac{\partial f(x, y)}{\partial x} \right]_{\substack{x=\mu_x \\ y=\mu_y}}^2 \sigma_x^2 + \left[ \frac{\partial f(x, y)}{\partial y} \right]_{\substack{x=\mu_x \\ y=\mu_y}}^2 \sigma_y^2 \quad (\text{b.11})$$

Além disso, quando as variáveis a serem ajustadas não forem independentes, é necessário conhecer sua covariância.

Madron et alii.(1977) descrevem um método, sugerido por Box, para o cálculo aproximado da matriz covariância ( $\Sigma$ ). Consideram as variáveis medidas indiretamente ( $n_i^+$ ), escritas como função das variáveis primárias medidas ( $t_k^+$ ). O método consiste na linearização das funções  $f_i$  em relação às variáveis primárias. Então, aproxima-se o erro  $e_i$  pelas expressões

$$e_i = \sum_k \left( \frac{\partial f_i}{\partial t_k^+} \right) e_{ik} = \sum_k b_{ik} e_{ik} \quad (\text{b.12})$$

onde:

$$b_{ik} = \left( \frac{\partial f_i}{\partial t_k^+} \right)$$

e  $e_{ik}$  são os erros das variáveis primárias. Então, a matriz covariância  $\Sigma$  é dada por

$$\Sigma = \mathbf{B} \Sigma^+ \mathbf{B}^T$$

onde  $\Sigma^+$  é a matriz diagonal das variâncias das variáveis primárias. Observam que os resultados obtidos por este método são tanto melhores quanto mais aproximada for a linearização através da eq.b.12.

No caso mais simples, em que todos os dados são independentes, a matriz variância-covariância é uma matriz diagonal.

### B.3. Distribuição Normal e Qui-quadrado

O comportamento de uma variável aleatória (ou, em outras palavras, a função densidade de probabilidade à que está associada) fica bastante bem caracterizado por dois parâmetros de sua distribuição: a média e a variância (ou o desvio-padrão). A primeira dá idéia da posição da distribuição ao longo do eixo da variável e a segunda informa sobre a dispersão dos valores em torno do valor médio.

Existem várias funções-distribuição teóricas descritas na literatura. São tratadas, aqui, somente aquelas que se relacionam com os testes estatísticos utilizados para detecção e identificação de erros grosseiros, que são a distribuição normal e a distribuição qui-quadrado.

#### B.3.1. Distribuição Normal

As distribuições de fenômenos observados assumem, frequentemente, uma forma simétrica em torno da média. Uma função desse tipo, de extrema importância em estatística, é

chamada de Distribuição Normal ou Distribuição de Gauss, que é descrita pela função densidade

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\left[\frac{(x-\mu)}{\sigma}\right]^2\right\}$$

De forma abreviada, se a variável aleatória  $x$  apresenta distribuição normal, com média  $\mu$  e variância  $\sigma^2$ , diz-se, simplesmente,

$$x \sim N(\mu, \sigma^2)$$

Do exame da função densidade, verifica-se que:

- i) ela é simétrica em torno do eixo vertical que passa por  $\mu$ ;
- ii) tem ordenada máxima, medindo  $1/(\sigma\sqrt{2\pi})$ , para  $x = \mu$ ;
- iii) tem por assintota o eixo das abscissas;
- iv) tem pontos de inflexão em  $x = \mu \pm \sigma$ .

Quando a variável aleatória  $x$  é padronizada ela passa a ter uma Distribuição Normal Padronizada, que tem média zero e desvio-padrão unitário, ou equivalentemente, uma distribuição  $N(0,1)$ . Os valores desta função distribuição são tabelados e com eles se tem um meio de obter as probabilidades associadas a qualquer variável normal, conhecidos a média e o desvio-padrão e a definição de variável aleatória padronizada ( $z$ ), descrita no item B.1

Considerando, agora, o conjunto de variáveis aleatórias independentes  $(x_1, x_2, \dots, x_n)$ , cada uma normalmente distribuída, com mesma média ( $\mu$ ) e mesma variância ( $\sigma^2$ ), então a média de uma amostra de  $n$  indivíduos, extraída de uma população  $N(\mu, \sigma^2)$ , representada por

$$\bar{x} = \frac{x_1 + x_2 + \dots + x_n}{n}$$

é normalmente distribuída, com média  $\mu$  e variância  $\sigma^2/n$ . Este resultado é muito importante e é chamado de Teorema do Limite Central. Em consequência,

$$z = \frac{\bar{x} - \mu}{\sigma/\sqrt{n}}$$

tem uma distribuição normal padronizada. Esta situação corresponde a um teste amostral simples.

Considerando, agora um teste multiamostrais envolvendo  $n$  variáveis aleatórias  $X_1, X_2, \dots, X_n$ , que são normalmente distribuídas, com suas médias  $\mu_1, \mu_2, \dots, \mu_n$  e variâncias  $\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2$ , respectivamente. A variável aleatória  $Z$  é uma combinação linear dessas variáveis

$$Z = a_1 X_1 + a_2 X_2 + \dots + a_n X_n$$

Na hipótese de  $Z$  ser normalmente distribuída, sua média  $\mu_z$  é a média ponderada das médias das variáveis aleatórias individuais, que podem ou não ser independentes, que é dada por

$$\mu_z = (a_1\mu_1 + a_2\mu_2 + \dots + a_n\mu_n)$$

A variância é dada por

$$\sigma_z^2 = \sum_{i=1}^n a_i^2 \sigma_i^2 + \sum_{i \neq j} a_i a_j \text{Cov}(X_i, X_j)$$

onde a  $\text{Cov}(X_i, X_j)$  é zero, se as variáveis  $X_i$  e  $X_j$  são independentes uma da outra.

Os chamados testes nodal e de medida, muito utilizados para detectar e identificar fontes de erros grosseiros, são testados contra uma distribuição normal padronizada.

### B.3.2. Distribuição Qui-quadrado

A variável aleatória, formada pela soma dos quadrados de  $n$  variáveis aleatórias independentes  $Z_i$ , é definida por

$$\chi^2 = \sum_i^n Z_i^2, \quad (Z_i = N(0,1))$$

tem uma distribuição  $\chi^2$ . O valor  $n$  é também conhecido como o número de graus de liberdade da  $\chi^2$ . A média e a variância dessa distribuição são expressas em graus de liberdade, sendo iguais, respectivamente, a  $n$  e  $2n$ .

A função densidade de probabilidade qui-quadrado é representada por  $\chi^2_v$ , onde  $v$  designa o número de graus de liberdade da  $\chi^2$ . Se  $v$  observações de uma variável são independentes, então o número de graus de liberdade é igual a  $v$ . Entretanto, um grau de liberdade é perdido para cada restrição sobre as  $v$  observações. A função distribuição  $F(\chi^2_\alpha)$  é tabulada de forma a fornecer valores  $\chi^2_{\alpha,v}$ , que são disponíveis na maioria dos livros de estatística, para os quais

$$P(\chi^2 \geq \chi^2_{\alpha,v}) = \alpha$$

onde  $\alpha$  é a probabilidade de somas dos quadrados iguais ou superiores ao valor correspondente tabelado. O nível de significância  $\alpha$  é, geralmente, fixado em torno do valor 0,05. Quanto maior  $\alpha$ , maior é o risco de rejeitar hipóteses boas; inversamente, o risco de aceitar hipóteses falsas aumenta, na medida que o valor de  $\alpha$  diminua.

Se o valor calculado da variável aleatória  $\chi^2$  for maior do que o valor tabelado  $\chi^2_{\alpha,v}$ , rejeita-se a hipótese de que as variáveis  $Z_i$  sejam aleatórias, entretanto, se o for menor ou igual a hipótese é aceita.

Esta distribuição é usada para a detecção de erros grosseiros em dados de processo, onde se verifica se a variável aleatória constituída pela soma ponderada (pelo inverso das variâncias) dos quadrados dos erros das variáveis individuais segue uma distribuição qui-quadrado. Se isto for verdade, é válida a hipótese de que os erros de medida são variáveis aleatórias normalmente distribuídas e, portanto, o conjunto de medidas testadas não contém erro sistemático. Caso isto não seja verdade, deve(m) ser identificada(s) a(s) medida(s) que contém(em) erros grosseiros, submetendo-se o conjunto resultante ao teste em questão. Para realizar o teste, deve-se calcular o valor de  $\chi^2$  e compará-lo com valores tabelados, uma vez definido um nível de significância  $\alpha$ , que exprime a probabilidade de somas dos quadrados iguais ou superiores ao valor correspondente da tabela. Se o valor calculado for maior do que o valor tabelado, rejeita-se a hipótese e se for menor ou igual ao valor tabelado, aceita-se a hipótese. A distribuição qui-quadrado é usada no chamado teste global de Almásy e Sztanó (1975) e também no teste do desbalanço nodal de Romagnoli et alii. (1980).

#### B.4. Testes de Hipóteses

Uma hipótese estatística é uma suposição concernente à função distribuição de uma variável aleatória. Um teste de hipóteses estatístico é um procedimento para decidir se deve ser aceita ou rejeitada a hipótese  $H_0$  contra uma ou mais hipóteses alternativas. Conhecida a função densidade de probabilidade  $f$  de uma estimativa (que é uma estimativa não-tendenciosa de  $\theta$ , uma variável aleatória qualquer), faz-se a suposição de que a representação desta variável aleatória  $\theta$  por  $f(\hat{\theta})$  é correta e que a média (ou valor esperado) de  $\theta$  é  $\theta_0$ . Coloca-se, então a seguinte questão: Se é verdadeira a hipótese de que  $\theta = \theta_0$ , por quanto  $\theta$  pode diferir de  $\theta_0$ , antes que esta hipótese seja rejeitada por parecer errada? Se a hipótese  $\theta = \theta_0$  for verdadeira,  $E(\theta) = \theta_0$  e a probabilidade de que o valor de  $\hat{\theta}$  seja menor ou igual a  $\theta_{\alpha/2}$  é

$$P(\hat{\theta} \leq \theta_{\alpha/2}) = \frac{\alpha}{2}$$

e devido à simetria da curva da Distribuição Normal

$$P(\hat{\theta} > \theta_{1-\alpha/2}) = \frac{\alpha}{2}$$

Para tomar uma decisão concernente à hipótese, seleciona-se um valor de  $\alpha$ , que é chamado nível de significância, para o teste, antes de coletar a amostra. Usualmente,  $\alpha$  é escolhido de forma arbitrária, para ser suficientemente pequeno, a fim de que, à vista do analista, seja bastante improvável que  $\hat{\theta}$  exceda o valor selecionado de  $\theta_{1-\alpha/2}$  ou seja menor do que  $\theta_{\alpha/2}$ . Por exemplo,  $\alpha$  pode ser 0,01 ou 0,05. Então, a amostra é coletada e  $\hat{\theta}$  é calculado. Se  $\hat{\theta}$  for maior do que  $\theta_{1-\alpha/2}$  ou menor do que  $\theta_{\alpha/2}$ , a hipótese é rejeitada. Caso contrário, ela é aceita. A faixa de valores de  $\hat{\theta}$  em que a hipótese é rejeitada é chamada região de rejeição; a faixa de  $\hat{\theta}$  em que a hipótese é aceita é chamada de região de aceitação. O teste descrito é um teste de dois lados, também chamado bilateral ou bicaudal. Um teste unilateral pode ser baseado em duas possibilidades, sendo que numa delas  $\hat{\theta}$  seria maior do que algum  $\theta_{1-\alpha}$ , onde a hipótese seria rejeitada, se  $\hat{\theta}$  fosse de fato maior do que  $\theta_{1-\alpha}$ ; a outra seria supor que  $\hat{\theta}$  seja

menor do que  $\theta_\alpha$ . Himmelblau (1978) observa que a rejeição da hipótese não implica num resultado definitivo, mas indica que os dados e o procedimento experimental devem ser submetidos a um exame cauteloso para averiguar se ocorreu alguma coisa errada com a coleta de medidas ou com a instrumentação.

A estrutura de teste mais simples é imaginar que haja uma dicotomia de estado para as variáveis aleatórias:

- i)  $H_0$ :  $x$  é o valor verdadeiro da variável aleatória (que é a hipótese nula);
- ii)  $H_1$ :  $x$  não é o valor verdadeiro da variável (que é a hipótese alternativa).

No teste de hipóteses, a decisão é tomada da seguinte forma: com base na suposição de que a hipótese nula é verdadeira, se a estatística calculada a partir da amostra experimental aleatória cair fora da região de aceitação, a hipótese nula é rejeitada e a hipótese alternativa é aceita. Caso contrário, a hipótese  $H_0$  é aceita e  $H_1$  é rejeitada.

Podem-se distinguir dois tipos de erros ao testar uma hipótese:

i) um Erro de Primeiro Tipo (*Erro Tipo I*), que é o risco de declarar-se falsa uma hipótese verdadeira;

ii) um Erro do Segundo Tipo (*Erro Tipo II*), que é o risco de não rejeitar uma hipótese, quando ela é falsa.

Isto pode ser resumido no esquema abaixo, onde a hipótese que está sendo testada é a hipótese  $H$ .

Hipótese	Aceitar $H$	Rejeitar $H$
$H$ é verdadeira	Decisão correta	Erro do Tipo I
$H$ é falsa	Erro do Tipo II	Decisão correta

Certamente, o erro do Tipo I existe porque  $\alpha$  é selecionado para ser um valor não-zero. Quando a hipótese é verdadeira e  $\alpha = 0,05$ , por exemplo, em 5% dos testes a hipótese será rejeitada, o que é uma decisão errada.

A probabilidade  $\beta$  é a probabilidade de não rejeitar uma diferença quando ela existe. Existem curvas, chamadas curvas de operação características, para determinar a probabilidade  $\beta$ . A probabilidade  $(1 - \beta)$  é chamada *potência do teste* e representa a probabilidade de tomar-se a decisão correta (rejeitar a hipótese), quando ela é realmente errada. Quando a diferença entre as médias ( $\delta$ ) aumenta,  $(1 - \beta)$  aumenta e  $\beta$  diminui. A seguir, apresenta-se de forma resumida um exemplo:

Se	Probabilidade de concluir que	
	$\mu = \mu_A$	$\mu \neq \mu_A$
$\mu = \mu_A$	$1 - \alpha$	$\alpha$
$\mu = \mu_A + \delta$	$\beta$	$1 - \beta$

Himmelblau (1978) relata que, pela descrição dos dois tipos de erros, pode-se observar que a tentativa de diminuir um tipo de erro resulta em um aumento no outro tipo de erro. O

único modo de diminuir os dois tipos de erro, simultaneamente, é aumentar o tamanho da amostra, o que pode ser caro, na prática. Observa que talvez um tipo de erro tenha consequências menos sérias do que o outro e, neste caso, há alguma decisão adequada referente à seleção de valores de  $\alpha$  e ao número de observações a ser feito. A experiência leva em conta os instrumentos, o projeto do processo e os custos, de modo a tomar-se uma decisão econômica para  $\alpha$  e  $\beta$ . Em geral, os estudos descritos na literatura para detecção e identificação de erros grosseiros, na reconciliação de dados de processo, consideram a probabilidade de ocorrência de erro Tipo I.

APÊNDICE C. LISTAGEM DO PROGRAMA RECON



O Programa RECON, conforme listado abaixo, permite a entrada de dados diretamente através do teclado ou de arquivo de dados. Os resultados são enviados para um arquivo de dados, cujo nome se informa no início.

```
/* Programa RECON - Reconciliacao de Dados de Processo e Coaptacao */
```

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<alloc.h>
#include<ctype.h>
#include<string.h>
#include<conio.h>

#define MFM(i,j) (* (mfm+c1 *(i)+j))
#define MB(i,j) (* (mb+nctm *(i)+j))
#define ST(i,j) (* (st+nctm *(i)+j))
#define MS(i,j) (* (ms+nctm *(i)+j))
#define M(i,k) (* (fd+2 *(i)+ k ))
#define A(i,j) (* (a+-mi *(i)+ j ))
#define AC(i,j) (* (ac+nic1 *(i)+j))
#define TE(i,j) (* (te+c2 *(i)+j))
#define ME(i,j) (* (me+mic1 *(i)+j))
#define Bi(i,j) (* (bi+q *(i)+j))
#define Pi(i,j) (* (ppi+n *(i)+j))
#define ACC(i,j) (* (acc+(nic1+rq) *(i)+j))
#define AR(i,j) (* (ar+(nic1+rq) *(i)+j))
#define BCI(i,j) (* (bci+(nic1+rq) *(i)+j))
#define ACB(i,j) (* (acb+(nic1+rq) *(i)+j))
#define TR(i,j) (* (tr+ni *(i)+j))
#define MR(i,j) (* (mr+nic1 *(i)+j))
#define B1(i,j) (* (b1+q *(i)+j))
#define P(i,j) (* (pp+n *(i)+j))
#define PL(i,j) (* (pl+n *(i)+j))
#define P1(i,j) (* (p1+r *(i)+j))
#define P2(i,j) (* (p2+r *(i)+j))
#define P3(i,j) (* (p3+(n-r) *(i)+j))
#define UP3(i,j) (* (up3+(n-r) *(i)+j))
#define VNO(i,j) (* (vno+(n-r) *(i)+j))
#define VI(i,j) (* (vi+(n-r) *(i)+j))
#define U(j,z) (* (u+r *(j)+z))
#define P2U(i,z) (* (pu+r *(i)+z))
#define H(j,z) (* (h+m *(j)+z))
#define YT(i,j) (* (y+m *(i)+j))
#define YHT(i,z) (* (yh+m *(i)+z))
#define YTB1(i,j) (* (yb+q *(i)+j))
#define S(j,z) (* (pps+q *(j)+z))
#define SBT(i,z) (* (sb+m *(i)+z))
#define SBY(i,j) (* (sby+p *(i)+j))
#define G(i,j) (* (g+p *(i)+j))
```

```

#define I(i,j) (*(ui+P*(i)+j))
#define X(i,j) *(x+1*(i)+j)
#define YTX(i,j) *(yx+1*(i)+j)
#define L(i,j) *(l+1*(i)+j)
#define AJ(i,j) *(aj+1*(i)+j)
#define UP(i,j) *(up+p*(i)+j)
#define B1m(z,j) *(b1m+q*(z)+j)
#define BX(i,j) *(bx+1*(i)+j)
#define VO(i,j) *(vo+1*(i)+j)
#define ALC(i,j) *(alc+3*(i)+j)
#define CM(i,j) *(cm+3*(i)+j)
#define CNM(i,j) *(cnm+3*(i)+j)

#define TM(size) (float *)malloc(size*sizeof(float))
#define TMI(size) (int *)malloc(size*sizeof(int))
#define TMC(size) (char *)malloc(size*sizeof(char))

#define MNO 10
#define MCOR 20
#define MCOM 10
#define MREA 5
#define MRES 3

int ni;
int mi;
int c;
int rq;
int nr;
int q;
int M[MCOR][2];
float varct[MCOR];
float MFM[MCOR][MCOM+1];
float X[MCOR*(MCOM+1)][1];
float TE[MREA][MCOM+2];
float TR[MRES][MCOR];
int viv[MCOR*(MCOM+1)+MREA];
int vie[MNO*(MCOM+1)+MRES*(MCOM+1)+MCOR];

void entrar(void), mostrar(void), gravar(void),ler(void);
char menu(void);
void reconciliar(void);
void backs(int r,int rs,int *ir,float *p1,float *u);
void prod_m(int s,int r,int w,float *y,float *h,float *yh);
int lu(int r, int *ir, float *p1);
void vnom(int c,int rq,int ni,int nicl,int *viv,char *cm,char *cnm );
void variancia( int rai,int c,int q,float *varct,float *mfm,float *pps);

main( )
{
char ch;

```

```

clrscr( );
printf("\nPROGRAMA RECON\n");
printf("\nRECONCILIACAO DE DADOS E COAPTACAO\n");
printf("\nElaborado por: TALITA FURLANETTO MENDES\n");
printf("\nOrientador: Joao Alexandre F. R. Pereira\n");
printf("\n\n\n\n\n\n\n\n\n\t\t digite qualquer letra\n");
ch = getche( );
clrscr( );

```

```

for (;;) {
ch=menu( );
switch(ch) {
case 'E' : entrar( );
break;
case 'M' : mostrar( );
break;
case 'G' : gravar( );
break;
case 'L' : ler( );
break;
case 'R' : reconciliar( );
break;
case 'S' : exit(0);
}
}
}

```

```

void reconciliar(void)

```

```

{
float *vnob;
int *fd;
float *a, *ac;
int c1, c2;
int nic1,mic1,nrc1,n,m,m1;
int r, p, *ig;
float *te, *me, *acc, *mfm, *bci,*acb;
float *bi, *ppi, *b1, *pp;
float *tr, *mr,*ar;
int *ip, *ic, *ir;
float *pl, *p1, *p2, *u, *pu, *up;
float *p3, *up3, *vno, *vi, *h, *y, *yh;
float *yb, *pps, *sb, *sby, *g, *ui, *x ;
float *yx, *l, *aj, *blm, *bx, *vo;
char *cm, *cnm, *comptroc;
float a1, a2;
double fabs( );

```

```

register int i,j,k, ki,kj,s,t,z;

```

```

FILE *fp;

```

```

char pa[20];
printf("Nome do Arquivo dos Resultados:");
gets(pa);
if( ( fp = fopen(pa, "w") ) == NULL ) {
printf("Nao posso abrir arquivo\n");
return;
}
printf("%d", fprintf(fp, "\nRESULTADOS DA RECONCILIACAO DE DADOS E DA
COAPTACAO\n" ));

```

```

c1 = c+1;
c2 = c+2;
mic1 = mi*c1;
nic1 = ni*c1;
nrcl = nr*c1;
clrscr( );

```

```

comptroc = TMC(3);
fd = TMI(ni*2);
mfm = TM(ni*c1 );
x = TM(q*1);

```

```

for ( i=0; i<ni; i++) for ( k=0; k<2; k++) M(i,k) = M[i][k];

```

```

for ( i=0; i<ni; i++) for ( j=0; j<c1; j++) MFM(i,j) = MFM[i][j];

```

```

for ( j=0, i=0; i<q; i++) X(i,j) = X[i][j];

```

```

a = TM(ni*mi);
for ( i=0; i<ni; i++) for ( j=0; j<mi; j++) {
    A(i,j) = ( M(i,0) == j+1 ? 1 : 0 );
    if ( A(i,j) == 0 ) A(i,j) = ( M(i,1) == j+1 ? -1 : 0 );
}

```

```

free(fd);

```

```

ac = TM(mic1*ni*c1);
for ( i=0; i<mi; i++) for ( ki=0; ki<c1; ki++)
    for ( j=0; j<ni; j++) for ( kj=0; kj<c1; kj++)
        AC( (c1*i+ki), (c1*j+kj) ) = ( ki == kj ? A(j,i) : 0 );

```

```

te = TM(rq*c2);
me = TM(rq*mic1);

```

```

for ( i=0; i<rq; i++) for ( j=0; j<c2; j++) TE(i,j) = TE[i][j];

```

```

for ( i=0; i<rq; i++) for ( j=0; j<mi; j++)
    for ( kj=0; kj<c1; kj++)
        ME(i, c1*j+kj) = ( c1*j+kj == (TE(i,0)-1)*c1+kj ? TE(i, kj+1) : 0 );

```

```

free(te);

```

```

acc=TM(mic1*(nrc1+rq));
for ( i=0; i<mic1; i++) for ( j=0; j<nic1; j++) ACC(i,j) = AC(i,j);
for ( i=0; i<mic1; i++)
  for ( j=nic1; j<nic1+rq; j++) ACC(i,j) = ME(j-nic1,i);

free(ac);

tr = TM(nr*ni);
mr = TM(nrc1*nic1);
for ( i=0; i<nr; i++) for ( j=0; j<ni; j++) TR(i,j) = TR[i][j];

for ( i=0; i<nr; i++) for ( ki=0; ki<c1; ki++)
  for ( j=0; j<ni; j++) for ( kj=0; kj<c1; kj++)
    MR( (c1*i+ki),(c1*j+kj) ) = ( ki == kj ? TR(i,j) : 0 );

free(tr);

ar=TM((mic1+nr)*nic1);
for ( i=0; i<mic1; i++) for ( j=0; j<nic1+rq; j++) AR(i,j) = ACC(i,j);

for ( i=mic1; i<mic1+nr; i++)
  for ( j=0; j<nic1; j++) AR(i,j) = MR(i-mic1,j);

for ( i=mic1; i<mic1+nr; i++)
  for ( j=nic1; j<nic1+rq; j++) AR(i,j) = 0;

free(mr);
free(acc);

bci = TM(ni*(nic1+rq));
for ( i=0; i<ni; i++) { for ( j=0; j<nic1+rq; j++)
  BCI(i,j) = ( j >= (c1*i) && j < (c1*i)+c1 ? 1 : 0 );
}

for ( i=0; i<ni; i++) { for ( j=0; j<nic1+rq; j++)
  if ( BCI(i,j) && j == i*c1+(c1-1) ) BCI(i,j) = -1;
}

acb = TM((mic1+nrc1+ni)*(nic1+rq));
for ( i=0; i<mic1+nrc1; i++)
  for ( j=0; j<nic1+rq; j++) ACB(i,j) = AR(i,j);
for ( i=mic1+nrc1; i<mic1+nrc1+ni; i++)
  for ( j=0; j<nic1+rq; j++) ACB(i,j) = BCI(i-(mic1+nrc1),j);

free(ar);

for ( q=0, n=0, j=0; j<nic1+rq; j++) if ( viv[j] == 1 ) q++;
  else if ( viv[j] == 3 ) n++;

bi = TM((mic1+nrc1+ni)*q);
ppi = TM((mic1+nrc1+ni)*n);

```

```

for ( kj=0, ki=0, j=0; j<nic1+rq; j++ ) if ( viv[j] == 1 ) {
  for ( i=0; i<mic1+nrc1+ni; i++ ) Bi(i,kj) = ACB(i,j); kj++;
}
else if ( viv[j] == 3 ) {
  for ( i=0; i<mic1+nrc1+ni; i++ ) Pi(i,ki) = ACB(i,j); ki++;
}

for ( m=0, i=0; i<mic1+nrc1+ni; i++ ) if ( vie[i] == 1 ) m++;

b1 = TM(m*q);
for ( ki=0, i=0; i<mic1+nrc1+ni; i++ ) if ( vie[i] == 1 ) {
  for ( j=0; j<q; j++ ) B1(ki,j) = Bi(i,j); ki++;
}

free(bi);

cm = TMC(q*3);
cnm = TMC(n*3);

vnom(c,rq,ni,nic1,viv,cm,cnm);

yh = TM(m*m);

if ( n ) {

pp=TM(m*n);
for ( ki=0, i=0; i<mic1+nrc1+ni; i++ ) if ( vie[i] == 1 ) {
  for ( j=0; j<n; j++ ) P(ki,j) = Pi(i,j); ki++;
}

free(ppi);
pl = TM(m*n);
for ( i=0; i<m; i++ ) for ( j=0; j<n; j++ ) PL(i,j) = P(i,j);

ic = TMI(n);
ip = TMI(m);
for ( i=0; i<m; i++ ) ip[i] = i;
for ( i=0; i<n; i++ ) ic[i] = i;

u = TM(m*n);
pu = TM(m*n);
up = TM(m*m);
h = TM(m*m);

for ( i=0; i<m; i++ ) for ( j=0; j<m; j++ ) H(i,j) = ( i == j ? 1 : 0 );
for ( i=0; i<n; i++ ) {
  for ( j = (k=i)+1; j<m; j++ ) if ( fabs(PL(j,i)) > fabs(PL(k,i))) k = j;
  if ( k != i ) for ( j=i; j<n; j++ ) {
    a1 = PL(i,j); PL(i,j) = PL(k,j); PL(k,j) = a1;
  }
}

```

```
a2 = PL(i,i); t = 0;
```

```
if ( a2 == 0 && t == 0 ) for ( j=i+1; j<n; j++ )
  for ( s=i; s<m; s++ )
    if ( PL(s,j) != 0 && t == 0 ) for ( j=0; j<n; j++ ) {
      a1 = PL(i,j); PL(i,j) = PL(s,j); PL(s,j) = a1;
      t = 1; k = s;
    }
```

```
a2 = PL(i,i); t = 0;
```

```
if ( a2 == 0 && t == 0 ) for ( s=i+1; s<n; s++ )
  if ( PL(i,s) != 0 && t == 0 ) for ( j=0; j<m; j++ ) {
    a1 = PL(j,i); PL(j,i) = PL(j,s); PL(j,s) = a1;
    t = 1; ic[i] = s;
  }
```

```
a2 = PL(i,i);
for ( ip[i]=k, j=i+1; j<m && a2 != 0; j++ ) {
  a1 = PL(j,i) / a2; PL(j,i) = 0;
  for ( k=i+1; k<n; k++ ) PL(j,k) += a1 * PL(i,k);
}
}
```

```
m1=m-1;
for ( ip[m1] = m1, i=0; i<m1; i++ )
  for ( j=0; j<n; j++ ) {
    a1 = P(i,j); P(i,j) = P(ip[i],j); P(ip[i],j) = a1;
  }
```

```
for ( j=0; j<n-1; j++ ) if ( j<ic[j] )
  for ( i=0; i<m; i++ ) {
    a1 = P(i,j); P(i,j) = P(i,ic[j]); P(i,ic[j]) = a1;
  }
```

```
for ( a1=0, r=0, i=0; i<n && i<m; i++ ) if ( fabs( PL(i,i) ) > 0.0001 ) r++;
```

```
p = m-r;
p1 = TM(r*r);
p2 = TM(p*r);
ir = TMI(r);
```

```
for ( i=0; i<r; i++ ) ir[i] = i;
```

```
for ( i=0; i<r; i++ ) for ( j=0; j<r; j++ ) P1(i,j) = P(i,j);
```

```
for ( i=0; i<p; i++ ) for ( j=0; j<r; j++ ) P2(i,j) = -P(i+r,j);
```

```
for ( j=0; j<r; j++ ) for ( z=0; z<r; z++ ) U(j,z) = ( j == z ? 1 : 0 );
```

```
lu(r,ir,p1);
```

```

backs(r,r,ir,p1,u);
prod_m(p,r,r,p2,u,pu);

for ( i=0; i<p ; i++ ) for ( j=0; j<p ; j++ ) UP(i,j) = ( i == j ? 1 : 0 );

y = TM(m*m);
for ( i=0; i<p ; i++ ) for ( j=0; j<r ; j++ ) YT(i,j) = P2U(i,j);

for ( i=0; i<p ; i++ ) for ( j=r; j<m ; j++ ) YT(i,j) = UP(i,j-r);

for ( ip[m1]=m1, i=0; i<m1 ; i++ ) for ( j=0; j<m ; j++ ) {
    a1 = H(i,j); H(i,j) = H(ip[i],j); H(ip[i],j) = a1;
}

prod_m(p,m,m,y,uh,yh);

}

if ( !n ) {

p=m;
for ( i=0; i<m ; i++ ) for ( j=0; j<m ; j++ ) YHT(i,j) = ( i == j ? 1 : 0 );

}

yb = TM(m*q);
sb = TM(q*m);
sby = TM(q*m);
g = TM(m*m);
ui = TM(m*m);
yx = TM(m*1);
l = TM(m*1);
aj = TM(q*1);

for ( a1=0, i=0; i<p ; i++ ) for ( j=0; j<q ; j++ ) {
    for ( z=0; z<m ; z++ ) {
        a1 += YHT(i,z)*B1(z,j) ;
    }
    YTB1(i,j) = a1; a1 = 0;
}

pps = TM(q*q);
variancia(ni,c,q,varct,mfm,pps);

for ( a1=0, i=0; i<q ; i++ ) for ( z=0; z<m ; z++ ) {
    for ( j=0; j<q ; j++ ) {
        a1 += S(i,j) * B1(z,j) ;
    }
    SBT(i,z) = a1;
    a1 = 0;
}

```



```

for ( a1=0, i=0; i<Q; i++ ) for ( j=0; j<p; j++ ) {
    for ( z=0; z<m; z++ ) {
        a1 += SBT(i,z) * YHT(j,z);
    }
    SBY(i,j) = a1; a1 = 0;
}

for ( a1=0, i=0; i<P; i++ ) for ( z=0; z<p; z++ ) {
    for ( j=0; j<q; j++ ) {
        a1 += YTB(i,j) * SBY(j,z);
    }
    G(i,z) = a1; a1 = 0;
}

ui = TM(p*p);
for ( i=0; i<p; i++ ) for ( j=0; j<p; j++ ) I(i,j) = ( i == j ? 1 : 0 );

ig = TMI(p);
for ( i=0; i<p; i++ ) ig[i] = i;

lu(p,ig,g);

backs(p,p,ig,g,ui);

for ( a1=0, i=0; i<P; i++ ) { for ( j=0, z=0; z<q; z++ ) {
    a1 += YTB(i,z) * X(z,j);
}
    YTX(i,j) = a1; a1 = 0;
}

for ( a1=0, i=0; i<P; i++ ) { for ( j=0, z=0; z<p; z++ ) {
    a1 += I(i,z) * YTX(z,j);
}
    L(i,j) = -a1; a1 = 0;
}

for ( a1=0, i=0; i<Q; i++ ) { for ( j=0, z=0; z<p; z++ ) {
    a1 += SBY(i,z) * L(z,j);
}
    AJ(i,j) = a1; a1 = 0;
}

for ( j=0, i=0; i<q; i++ ) X(i,j) += AJ(i,j);

printf("%d", fprintf(fp, "\nVazoes Medidas:\n"));
for ( i=0; i<q; i++ ) {
    for ( j=0; j<3; j++ ) printf("%d", fprintf(fp, "%c", CM(i,j)));
    printf("%d", fprintf(fp, " "));
}

printf("%d", fprintf(fp, "\n"));

```

```

printf("%d", fprin f(fp, "\nValores Corrigidos\n"));
for ( j=0, i=0; i<q ; i++ ) printf("%d", fprintf(fp, "%10.4f", X(i,j)));
printf("\n");

if ( n ) {

vnob = TM(n);
for ( i=0; i<n; i++ ) vnob[i] = 0;

for ( j=0; j<n-1; j++ ) if ( j < ic[j] )
  for ( i=0; i<3; i++ ) {
    comptroc[ i ] = CNM(j,i); CNM(j,i) = CNM(ic[j],i);
    CNM(ic[j] , i) = comptroc[i];
  }

printf("%d", fprin f(fp, "\n\nVazoes nao-medidas apos troca\n"));
for ( i=0; i<n; i++ ) {
  for ( j=0; j<3; j++ ) printf("%d", fprintf(fp, "%c", CNM(i,j)));
  printf("%d" , fprintf(fp, " "));
}
printf("%d", fprin f(fp, "\n"));

if ( r == n )
  printf("%d", fprin f(fp, "\nTodos valores nao medidos sao observaveis\n"));
else {

p3 = TM(r*(n-r));
for ( i=0; i<r; i++ ) for ( j=r; j<n; j++ ) P3(i,j-r) = P(i,j);

up3 = TM(r*(n-r));
for ( a1=0, i=0; i<r; i++ ) for ( j=0; j<n-r; j++ ) {
  for ( z=0; z<r; z++ ) {
    a1 += U( i, z)*P3(z,j);
  }
  UP3(i,j) = a1; a1 = 0;
}

vi = TM((n-r)*(n-r));
for ( i=0; i<n-r; i++ )
  for ( j=0; j<n-r; j++ ) VI(i,j) = ( i == j ? 1 : 0 );

vno = TM(n*(n-r));

for ( i=0; i<r; i++ ) for ( j=0; j<n-r; j++ ) VNO(i,j) = UP3(i,j);
for ( i=r; i<n; i++ ) for ( j=0; j<n-r; j++ ) VNO(i,j) = VI(i-r,j);

for ( a1=0, i=0; i<n; i++ ) { for ( j=0; j<n-r; j++ ) {
  a1 += fabs(VNO(i,j));
}
  vnob[i] = a1; a1 = 0;
}

```

```

printf("%d", fprintf(fp, "\nValores nao-observaveis: "));
for ( i=0; i<n ; i++ ) if ( vnob[i] != 0 ) {
    for ( j=0; j<3; j++ ) printf("%d", fprintf(fp, "%c", CNM(i,j)));
    printf("%d", fprintf(fp, " "));
}
printf("%d", fprintf(fp, "\n"));
}

m1=m-1;
for ( ip[m1]=m1, i=0; i<m1; i++ ) for ( j=0; j<q; j++ ) {
    a1 = B1(i,j); B1(i,j) = B1(ip[i],j); B1(ip[i],j) = a1;
}

.....

b1m = TM(r*q);
for ( i=0; i<r; i++ ) for ( j=0; j<q; j++ ) B1m(i,j) = -B1(i,j);

bx = TM(r*1);
for ( a1=0, i=0; i<r; i++ ) { for ( z=0, j=0; j<q; j++ ) {
    a1 += B1m(i,j) * X(j,z);
}
    BX(i,z) = a1; a1 = 0;
}

vo = TM(n*1);
for ( a1=0, i=0; i<r; i++ ) if ( vnob[i] == 0 ) {
    for ( z=0, j=0; j<r; j++ ) {
        a1 += U(i,j) * BX(j,z);
    }
    VO(i,z) = a1; a1 = 0;
}
else VO(i,z) = 0;

for( z=0, i=r; i<n; i++ ) VO(i,z) = 0;

printf("%d", fprintf(fp, "\nValores Observaveis\n"));
for ( j=0, i=0; i<n ; i++ ) printf("%d", fprintf(fp, "%10.4f", VO(i,j)));
}
else printf("%d", fprintf( fp, "\nTodos valores sao medidos\n"));
fclose(fp);
}

```

```

int lu(int r,int *ir,float *p1)
{
    int r1 = r-1;
    register int i,j,k;
    float a1,a2;double fabs( );

    for ( ir[r1]=r1, i=0 ; i<r1 ; i++ ) {
        for ( j = ( k=i ) +1 ; j<r ; j++ )
            if ( fabs( P1(j,i) ) > fabs( P1(k,i) ) ) k = j;
    }
}

```

```

    if ( k != i ) for ( j=i ; j<r ; j++ ) {
        a1 = P1(i,j) ; P1(i,j) = P1(k,j) ; P1(k,j) = a1 ;
    }
if ( !(a2 = P1(i,i), a2 ) ) return (-1);

for ( ir[i]=k, j=i+1 ; j<r ; j++ ) {
    a1 = ( P1(j,i) /=- a2 );
    for ( k = i+1 ; k<r ; k++ ) P1(j,k) += a1 * P1(i,k) ;
}
}
if ( !P1(r1,r1) ) return (-1) ;
return(0);
}

```

```

void backs (int r, int rs, int *ir, float *p1, float *u)
{
    register int i, j, k;
    int r1 = r - 1;
    float a1;
    for ( i = 0; i < r1; i++ ) {
        if ( (j = ir[i]) != i ) for ( k = 0; k < rs; k++ ) {
            a1 = U(i,k); U(i,k) = U(j,k); U(j,k) = a1;
        }
        for ( j = i + 1; j < r; j++ )
            for ( k = 0; k < rs; k++ ) U(j,k) += P1(j,i) * U(i,k);
    }
    for ( k = 0; k < rs; k++ ) U(r1,k) /= P1(r1,r1);
    for ( i = r1 - 1; i >= 0; i-- ) {
        for ( a1 = P1(i,i), j = i + 1; j < r; j++ )
            for ( k = 0; k < rs; k++ ) U(i,k) -= P1(i,j) * U(j,k);
        for ( k = 0; k < rs; k++ ) U(i,k) /= a1;
    }
}

```

```

void prod_m(int s, int r, int w, float *p2, float *u, float *pu)
{
    register int i, j, z;
    float a1;

    for ( a1=0, i=0; i<s; i++ ) for ( z=0; z<w; z++ ) {
        for ( j=0; j<r; j++ ) {
            a1 += P2(i,j) * U(j,z) ;
        }
        P2U(i,z) = a1; a1 = 0;
    }
}

```

```

char menu(void)
{
    char ch;
    do {
        printf("\n\n( E )ntrar dados\n");
        printf("( M )Ostrar dados\n");
        printf("( G )ravar em arquivo\n");
        printf("( L )e r um arquivo de dados\n");
        printf("( R )e c conciliar dados\n");
        printf("( S )a i r\n");
        printf("\nEs colha uma das opcoes(E,M,G,L,R,S):");
        ch = toupper(getche( ));
    }
    while( ch != 'E' && ch != 'M' && ch != 'G' && ch != 'S' && ch != 'L'
        && ch != 'R');
    printf("\n");
    return ch;
}

```

```

void entrar( )
{
    int i,j,nviv,nvie;

    printf("Numero de correntes globais: ");
    scanf("%d%c",&ni);

    printf("Numero de nos: ");
    scanf("%d%c",&mi);

    printf("Numero de componentes: ");
    scanf("%d%c",&c);

    printf("Numero de reacoes quimicas: ");
    scanf("%d%c",&rq);

    printf("Numero de restricoes: ");
    scanf("%d%c",&nr);

    printf("Numero de vazoes medidas: ");
    scanf("%d%c",&q);

    nviv = ni*(c+1)+rq;
    nvie = mi*(c+1)+nr*(c+1)+ni;

    printf("Entre com informacoes das correntes\n");
    for ( i=0; i<ni; ++i ) {
        printf("Corrente %d:",i+1);
        scanf("%d%c%d%c",&M[i][0],&M[i][1]);
    }
    printf("Entre com as variancias das correntes totais:");
}

```

```

for ( i=0; i<ni; i++ ) {
    printf("Entre com a variancia da corrente %d:",i+1);
    scanf("%f",&varct[i]);
}
printf("Entre com as fracoes parciais\n");
for ( i=0; i<ni; i++ ) {
    printf("Corrente %d:",i+1);
    for ( j=0; j<c+1; j++ ) scanf("%f",&MFM[i][j]);
}
printf("Entre com informacoes sobre reacoes\n");
for ( i=0; i<rq; i++ ) {
    printf("Reacao %d:",i+1);
    for ( j=0; j<c+2; j++ ) {
        printf("Elemento %d:",j+1);
        scanf("%f",&TE[i][j]);
    }
}

printf("Entre com informacoes sobre restricoes\n");
for ( i=0; i<nr; i++ ) {
    printf("Restricao %d:",i+1);
    for ( j=0; j<ni; j++ ) scanf("%f",&TR[i][j]);
}

printf("Entre com as vazoes medidas:\n");
for ( j=0, i=0; i<q; i++ ) scanf("%d",&X[i][j]);

printf("Entre com informacoes das variaveis:\n");
for ( i=0; i<nviv; i++ ) scanf("%d",&viv[i]);

printf("Entre com informacoes dos equipamentos:\n");
for ( i=0; i<nvie; i++ ) scanf("%d",&vie[i]);
}

void mostrar(void)
{
    int i,j;

    printf("\nNumero de correntes globais:%d\n",ni);
    printf("Numero de nos:%d\n",mi);
    printf("Numero de componentes:%d\n",c);
    printf("Numero de reacoes quimicas:%d\n",rq);
    printf("Numero de restricoes:%d\n",nr);
    printf("Numero de vazoes medidas:%d\n",q);
    printf("\nInformacoes das correntes:\n");
    for ( i=0; i<ni; ++i ) printf("Corrente %d:%d,%d\n",i+1,M[i][0],M[i][1]);
    printf("\n");

    printf("\nVariancia das correntes globais:\n");

```

```

for( i=0; i<ni; i++ ) printf("%8.2f\t",varct[i]);
printf("\n");

printf("Fracoes parciais\n");
for ( i=0; i<ni; i++ ) {
    printf("Corrente %d:",i+1);
    for ( j=0; j<c+1; j++ ) printf("%8.4f",MFM[i][j]);
    printf("\n");
}

printf("\nMatriz estequiometria:\n");
for ( i=0; i<rq; i++ ) {
    for ( j=0; j<c+2; j++ ) .....
        printf("%6.2f",TE[i][j]);
    printf("\n");
}

printf("\nMatriz das restricoes:\n");
for ( i=0; i<nr; i++ ) {
    for ( j=0; j<ni; j++ )
        printf("%6.2f",TR[i][j]);
    printf("\n");
}

printf("\nVazoes Medidas:\n");
for ( j=0,i=0; i<q; i++ ) printf("%8.4f ",X[i][j]);
printf("\n");

printf("\nInformacoes das variaveis:\n");
for ( i=0; i<(ni*(c+1))+rq; i++ ) printf("%d ",viv[i]);

printf("\nInformacoes dos equipamentos:\n");
for ( i=0; i<mi*(c+1)+nr*(c+1)+ni; i++ ) printf("%d ",vie[i]);
}

void gravar(void)
{
    FILE *fp;
    int i,j;
    char s[20];

    printf("Nome do arquivo:");
    gets(s);

    if( ( fp = fopen(s,"w")) == NULL ) {
        printf("Nao posso abrir arquivo\n");
        return;
    }
    printf("%d",fprintf(fp, "%d %d %d %d %d %d\n",ni,mi,c,rq,nr,q));
}

```

```

for ( i=0; i<ni; i++ )
    printf("%d",fprintf(fp,"%d %d ",M[i][0],M[i][1]));
    printf("%d",fprintf(fp,"\n"));
printf("%d",fprintf(fp,"\n"));
for ( i=0; i<ni; i++ ) printf("%d",fprintf(fp,"%ft",varct[i]));
printf("%d",fprintf(fp,"\n"));

for ( i=0; i<ni; i++ ) {
    for ( j=0; j<c+1; j++ ) printf("%d",fprintf(fp,"%ft",MFM[i][j]));
    printf("%d",fprintf(fp,"\n"));
}
for ( i=0; i<rq; i++ ) {
    for ( j=0; j<c+2; j++ )
        printf("%d",fprintf(fp,"%6.2f",TE[i][j]));
    printf("%d",fprintf(fp,"\n"));
}

for ( i=0; i<nr; i++ ) {
    for ( j=0; j<ni; j++ )
        printf("%d",fprintf(fp,"%6.2f",TR[i][j]));
    printf("%d",fprintf(fp,"\n"));
}

for ( j=0, i=0; i<q; i++ )
    printf("%d",fprintf(fp,"%6.2f",X[i][j]));

printf("%d",fprintf(fp,"\n"));

for ( i=0; i<(ni*(c+1))+rq; i++ )
    printf("%d",fprintf(fp,"%d",viv[i]));

printf("%d",fprintf(fp,"\n"));

for ( i=0; i<(mi*(c+1))+nr*(c+1)+ni; i++ )
    printf("%d",fprintf(fp,"%d",vie[i]));

printf("%d",fprintf(fp,"\n"));

fclose(fp);
}

void ler(void)
{
    FILE *fp;
    int i,j;
    char s[20];

    printf("\nNome do arquivo: ");
    gets(s);

```



```

if( ( fp = fopen(s, "r") ) == NULL ) {
    printf("Nao posso abrir arquivo\n");
    exit(1);
}
fscanf(fp, "%d%*c%d%*c%d%*c%d%*c%d%*c%d%*c", &ni, &mi, &c, &rq, &nr, &q);
for ( i=0; i<ni; i++ ) fscanf(fp, "%d%*c%d%*c", &M[i][0], &M[i][1]);
for ( i=0; i<ni; i++ ) fscanf(fp, "%f%*c", &varct[i]);
for ( i=0; i<ni; i++ )
for ( j=0; j<c+1; j++ ) fscanf(fp, "%f%*c", &MFM[i][j]);
for ( i=0; i<rq; i++ )
for ( j=0; j<c+2; j++ ) fscanf(fp, "%f%*c", &TE[i][j]);

for ( i=0; i<nr; i++ )
for ( j=0; j<ni; j++ ) fscanf(fp, "%f%*c", &TR[i][j]);
for ( j=0, i=0; i<q; i++ ) fscanf(fp, "%f%*c", &X[i][j]);
for ( i=0; i<(ni*(c+1))+rq; i++ ) fscanf(fp, "%d%*c", &viv[i]);
for ( i=0; i<(mi*(c+1))+nr*(c+1)+ni; i++ ) fscanf(fp, "%d%*c", &vie[i]);

fclose(fp);
}

```

```

void vnom(int c, int rq, int ni, int nic1, int *viv, char *cm, char *cnm)

```

```

{
    char *compo;
    char *numcor1;
    char *numcor2;
    char *alc;
    int c1 = c+1;
    register int i, j, ki, kj;

    compo = TMC(MCOM+1);
    numcor1 = TMC(MCOR);
    numcor2 = TMC(MCOR);
    alc = TMC(MCOR*(MCOM+1)*3);
    for ( i=0; i<c; i++ ) compo[i] = 'A'+i;
    for ( i=c; i<c1; i++ ) compo[i] = 'T';

    for ( i=0; i<9; i++ ) {
        numcor1[i] = '-'; numcor2[i] = '1'+i;
    }

    for ( j=0, i=9; i<ni; i++, j++ ) {
        numcor1[i] = '1'; numcor2[i] = '0'+j;
    }

    if ( ni == 20 ) {
        numcor1[ni-1] = '2'; numcor2[ni-1] = '0';
    }

    for ( i=0; i<ni; i++ ) for ( j=0; j<c1; j++ ) {

```

```

ALC(j+i*c1,0) = compo[j]; ALC(j+i*c1,1) = numcor1[i];
ALC(j+i*c1,2) = numcor2[i];
}

for ( i=nic1; i<nic1+rq; i++ ) {
    ALC(i,0) = 'R'; ALC(i,1) = '1'+(i-nic1); ALC(i,2) = '';
}

for ( ki=0, kj=0, i=0; i<nic1+rq; i++ ) if ( viv[i] == 1 ) {
    for ( j=0; j<3; j++ ) CM(ki,j) = ALC(i,j); ki++;
}

else if ( viv[i] == 3 ) {
    for ( j=0; j<3; j++ ) CNM(kj,j) = ALC(i,j); kj++;
}
}

void variancia(int ni,int c,int q,float *varct,float *mfm,float *pps)
{
    int nctm, c1 = c+1;
    float *mb, *st, *ms;
    float *varm;
    float a1;
    register int i,j,t,k;

    for ( nctm=0, i=0; i<ni; i++ ) if ( varct[i] != 0 ) nctm++;

    mb = TM(q*nctm);
    for ( i=0; i<q; i++ ) for ( j=0; j<nctm; j++ ) MB(i,j) = 0;

    for ( t=0, k=0, j=0; j<ni; j++ ) if ( varct[j] != 0 ) {
        for ( i=0; i<c1; i++ ) if ( MFM(j,i) != 0 && MFM(j,i) != 2 ) {
            MB(k,t) = MFM(j,i); k++;
        }
        t++;
    }

    varm = TM(nctm);
    for ( t=0, i=0; i<ni; i++ ) if ( varct[i] != 0 ) {
        varm[t] = varct[i]; t++;
    }

    st = TM(nctm*nctm);
    for ( i=0; i<nctm; i++ )
        for ( j=0; j<nctm; j++ ) ST(i,j) = ( i == j ? varm[i] : 0 );

    ms=TM(q*nctm);
    for ( a1=0, i=0; i<q; i++ ) for ( j=0; j<nctm; j++ ) {
        for ( t=0; t<nctm; t++ ) {

```

```

a1 += MB(i, t) * ST(t, j);
}
MS(i, j) = a1; a1 = 0;
}

```

```

for ( a1=0, i=0; i<q; i++) for ( t=0; t<q; t++) {
  for ( j=0; j<q; j++) {
    a1 += MS(i, j) * MB(t, j);
  }
  S(i, t) = a1; a1 = 0;
}

```

```

for ( i=0; i<q; i++) for ( j=0; j<q; j++) if ( i != j ) S(i, j) = S(i, j) / 2 ;
}

```

APÊNDICE D. LISTAGEM DO PROGRAMA RETIF

Este programa, conforme listado abaixo, prevê a entrada de dados através de arquivo de dados. Entretanto, pode ser usado o programa RECON para entrar com dados pelo teclado, que devem então ser gravados em arquivo de dados

/\* Programa RETIF – Retificacao de Dados de Processo e Coaptacao \*/

```
#include<stdio.h>
#include<ctype.h>
#include<string.h>
#include<conio.h>
#include<stdlib.h>
#include<math.h>
#include<alloc.h>

#define MFM(i,j) (* (mfm+c1*(i)+j))
#define MB(i,j) (* (mb+nctm *(i)+j))
#define SCT(i,j) (* (sct+nctm *(i)+j))
#define MS(i,j) (* (ms+nctm *(i)+j))
#define M(i,k) (* (fd+2 *(i)+ k ))
#define A(i,j) (* (a+mi *(i)+ j ))
#define AC(i,j) (* (ac+nic1 *(i)+j))
#define TE(i,j) (* (te+c2 *(i)+j))
#define ME(i,j) (* (me+mic1 *(i)+j))
#define Bi(i,j) (* (bi+q *(i)+j))
#define Pi(i,j) (* (ppi+n *(i)+j))
#define ACC(i,j) (* (acc+(nic1+rq) *(i)+j))
#define AR(i,j) (* (ar+(nic1+rq) *(i)+j))
#define BCI(i,j) (* (bci+(nic1+rq) *(i)+j))
#define ACB(i,j) (* (acb+(nic1+rq) *(i)+j))
#define TR(i,j) (* (tr+ni *(i)+j))
#define MR(i,j) (* (mr+nic1 *(i)+j))
#define B1(i,j) (* (b1+q *(i)+j))
#define P(i,j) (* (pp+n *(i)+j))
#define PL(i,j) (* (p1+n *(i)+j))
#define P1(i,j) (* (p1+r*(i)+j))
#define P2(i,j) (* (p2+r*(i)+j))
#define P3(i,j) (* (p3+(n-r)*(i)+j))
#define UP3(i,j) (* (up3+(n-r)*(i)+j))
#define VNO(i,j) (* (vno+(n-r)*(i)+j))
#define VI(i,j) (* (vi+(n-r)*(i)+j))
#define U(j,z) (* (u+r*(j)+z))
#define P2U(i,z) (* (pu+r *(i)+z))
#define H(j,z) (* (h+m *(j)+z))
#define YT(i,j) (* (y+m *(i)+j))
#define YHT(i,z) (* (yh+m *(i)+z))
#define YTB1(i,j) (* (yb+q *(i)+j))
#define S(j,z) (* (pps+q *(j)+z))
#define SBT(i,z) (* (sb+m *(i)+z))
#define SBY(i,j) (* (sby+p *(i)+j))
#define G(i,j) (* (g+p*(i)+j))
```

```

#define I(i,j) (* $u_i - p^{i+j}$ )
#define X(i,j) (* $x - 1^{i+j}$ )
#define YTX(i,j) (* $\subset y_{x+1}^{i+j}$ )
#define L(i,j) (* $l - 1^{i+j}$ )
#define AJ(i,j) (* $a_j - 1^{i+j}$ )
#define UP(i,j) (* $u_p + p^{i+j}$ )
#define B1m(z,j) (* $\subset b_{1m+q}^{z+j}$ )
#define BX(i,j) (* $b_{x+1}^{i+j}$ )
#define VO(i,j) (* $\cup o + 1^{i+j}$ )
#define J(i,z) (* $p_j - 1 + 2^{i+z}$ )
#define NU(k,j) (* $\subset nu + q^{k+j}$ )
#define YI(j,i) (* $y_i - p^{j+i}$ )
#define DE(i,j) (* $d_e + q^{i+j}$ )
#define Za(i,j) (* $z_a + q^{i+j}$ )
#define GL(i,j) (* $g_l + q^{i+j}$ )
#define IB(i,j) (* $i_b - 1 + q^{i+j}$ )
#define M1(i,j) (* $m_{m1+p}^{i+j}$ )
#define YM(i,j) (* $y_{m+p}^{i+j}$ )
#define DN(i,j) (* $d_n + q^{i+j}$ )
#define DD(i,j) (* $d_d + q^{i+j}$ )
#define ZA1(i,j) (* $z_{a1+q}^{i+j}$ )
#define ZAM(i,j) (* $\subset z_{am+1}^{i+j}$ )
#define J1(i,j) (* $j_1 - 1 + 1^{i+j}$ )
#define VZAD(i,j) (* $\subset v_{zad+q}^{i+j}$ )
#define J2(i,j) (* $j_2 + q^{i+j}$ )
#define NY(i,j) (* $\subset n_y + q^{i+j}$ )
#define YM1(i,j) (* $\subset y_{m1+p}^{i+j}$ )
#define GA(i,j) (* $\subset g_a + q^{i+j}$ )
#define ZDA(i,j) (* $\subset z_{da+q}^{i+j}$ )
#define PAR(i,j) (* $\subset par + 2^{i+j}$ )
#define ZDAP(i,j) (* $\subset z_{dap+q}^{i+j}$ )
#define VZD(i,j) (* $\subset v_{zd+q}^{i+j}$ )
#define ZAM2(i,j) (* $\subset z_{am2+npar}^{i+j}$ )
#define GI(i,j) (* $g_i + p^{i+j}$ )
#define ZE(i,j) (* $z_e + p^{i+j}$ )
#define ZE1(i,j) (* $\subset z_{e1+pm1}^{i+j}$ )
#define CE1(i,j) (* $\subset c_{e1+1}^{i+j}$ )
#define MG(i,j) (* $\subset mg + pm1^{i+j}$ )
#define RTT(i,j) (* $\subset r_{tt+p}^{i+j}$ )
#define RT(i,j) (* $\subset r_{t+p}^{i+j}$ )
#define RE(i,j) (* $\subset r_e + 1^{i+j}$ )
#define RG(i,j) (* $\subset r_g + p^{i+j}$ )
#define RR(i,j) (* $\subset r_{r+pm1}^{i+j}$ )
#define RGR(i,j) (* $\subset r_{gr+pm1}^{i+j}$ )
#define CE2(i,j) (* $\subset c_{e2+1}^{i+j}$ )
#define MG2(i,j) (* $\subset mg2 + pm2^{i+j}$ )
#define RTT2(i,j) (* $\subset r_{tt2+pm1}^{i+j}$ )
#define RT2(i,j) (* $\subset r_{t2+pm1}^{i+j}$ )
#define R2T(i,j) (* $\subset r_{2t+p}^{i+j}$ )
#define RE2(i,j) (* $\subset r_{e2+1}^{i+j}$ )
#define RG2(i,j) (* $\subset r_{g2+p}^{i+j}$ )

```

```

#define R2R(i,j) (*C r2r+pm2*(i+j))
#define RGR2(i,j) (*C (rgr2+pm2*(i+j))
#define ZE2(i,j) (*C (ze2+pm2*(i+j))
#define ZEM2(i,j) (*C (zem2+npar *(i+j))
#define ZJ1(i,j) (*C (zj1+1 *(i+j))
#define ZEM1(i,j) (*C (zem1+q *(i+j))
#define ST(i,j) (*C (st+p *(i+j))
#define M1E(i,j) (*C (m1e+1 *(i+j))
#define AJ1(i,j) (*C (aj1+q *(i+j))
#define X1(i,j) (*C (x1+q *(i+j))
#define X12(i,j) (*C (x12+npart *(i+j))
#define SNY(i,j) (*C (sny+npart *(i+j))
#define P2R(i,j) (*C (p2r+(n+2)*(i+j))
#define P2RL(i,j) (*C (p2rl+(n+2)*(i+j))
#define P2R1(i,j) (*C (p2r1+r2*(i+j))
#define U2(i,j) (*C (u2+r2 *(i+j))
#define B12Rm(i,j) (*C (b12rm+q *(i+j))
#define B2X(i,j) (*C (b2x+1 *(i+j))
#define VO2(i,j) (*C (vo2+npart *(i+j))
#define P2R3(i,j) (*C (p2r3+(nm2-r2) *(i+j))
#define UP2R3(i,j) (*C (up2r3+(nm2-r2) *(i+j))
#define VI2(i,j) (*C (vi2+(nm2-r2) *(i+j))
#define VNO2(i,j) (*C (vno2+(nm2-r2) *(i+j))
#define P1R(i,j) (*C (p1r+(n+1)*(i+j))
#define P1RL(i,j) (*C (p1rl+(n+1)*(i+j))
#define P1R1(i,j) (*C (p1r1+r1*(i+j))
#define U1(i,j) (*C (u1+r1 *(i+j))
#define B11Rm(i,j) (*C (b11rm+q *(i+j))
#define B1X(i,j) (*C (b1x+1 *(i+j))
#define VO1(i,j) (*C (vo1+q *(i+j))
#define P1R3(i,j) (*C (p1r3+(nm1-r1) *(i+j))
#define UP1R3(i,j) (*C (up1r3+(nm1-r1) *(i+j))
#define VI1(i,j) (*C (vi1+(nm1-r1) *(i+j))
#define VNO1(i,j) (*C (vno1+(nm1-r1) *(i+j))
#define PARF(i,j) (*C (parf+2 *(i+j))
#define PART(i,j) (*C (part+2 *(i+j))
#define JT(i,j) (*C (tj+50 *(i+j))
#define Zc(i,j) (*C (zc+50 *(i+j))

#define ALC(i,j) (*C (alc+3 *(i+j))
#define CM(i,j) (*C (cm+3 *(i+j))
#define CNM(i,j) (*C (cnm+3 *(i+j))
#define CNM2R(i,j) (*C (cnm2r+3 *(i+j))
#define CNM1R(i,j) (*C (cnm1r+3 *(i+j))

#define TMI(size) (int *)malloc(size*sizeof(int))
#define TM(size) (float *)malloc(size*sizeof(float))
#define TMC(size) (char *)malloc(size*sizeof(char))

#define MNO 10
#define MCOR 20

```

```

#define MCOM 10
#define MREA 5
#define MRES 3
int ni;
int mi;
int c;
int rq;
int nr;
int q;
int M[MCOR][2];
float varct[MCOR];
float MFM[MCOR][MCOM+1];
float X[MCOR*(MCOM+1)][1];
float TE[MREA][MCOM+2];
float TR[MRES][MCOR];
int viv[MCOR*(MCOM+1)+MREA];
int vie[MNO*(MCOM+1)+MRES*(MCOM+1)+MCOR];

void reconciliar(void);
void backs(int r,int rs,int *ir,float *p1,float *u);
void prod_m(int s,int r,int w,float *y,float *h,float *yh);
int triang(int m,int n,float *pl,float *pp,int *ip,int *ic);
int lu(int r,int *ir,float *p1);
void IEG(int p,int q,float *ui,float *yx,float *l,float *pj,float *yb,float *yi,float *de,float *nu,
float *za);
int emdif(int q,int *pvza,float *za,float *zc, FILE *fp);
int comp_J(int p,float *pj,float *tj);
void R1VM(int q,float *za,float *pj,int *pvza);
void comp_J1(int p,int q,float *pj,int *j1,float *tj);
void VZaD(int p,int q,float *yb,float *yi,float *ui,float *ib,float *mm1,float *yx,float *nu,
float *de,float *za1,int *pvza,int *vza1,int *zam,float *zc);
void R2VM(int q,float *pj,float *za1,int *vzad,float *j2,int *pvza);
int comp_J2(int p,int q,int *pvza,int *vzad,float *za1,float *j2,float *tj);
void Mos_par(int p,int q,int *pvza,int *vzad,float *za1,float *j2,int *par,float *tj,FILE *fp);
void VZDA(int p,int q,float *ib,float *yi,float *ui,float *mm1,float *yb,float *ny,float *za1,
int *vzad,int npar,int *par,int *zam2,float *zc);
int ZEduas(int p,int q,int pml,float *yb,float *cel,float *mg,float *rtt,float *rt,float *yx,
float *gi,int npar,int *par,int *zam2,int *pvza,int *part,float *zc, FILE *fp);
int ZEuma(int p,int q,int pml,float *yb,float *cel,float *mg,float *rtt,float *rt,float *yx,
float *gi,int *zj1,int *zem1,float *zc);
int ZEzero(int p,float *yx,float *gi,float *zc, FILE *fp);
void Valcorr1(int p,int q,float *pps,float *yb,float *st,float *mm1,float *ib,float *yi,
float *mle,float *yx,float *aj1,float *x,float *x1,int *zem1,int *pvza);
void Valcorr2(int p,int q,int npart,float *pps,float *st,float *mm1,float *ib,float *yi,
float *mle,float *yx,float *aj1,float *x,float *x12,int *part,float *ui,float *yb,
float *ny,float *za1);
void Val_obs2(int m,int n,int q,int npart,int *part,float *pp,float *b1,float *x12,char *cm,
char *cnm, char *cnm2r,float *vo2, FILE *fp);
void Val_obs1(int m,int n,int q,int *zem1,int *pvza,float *pp,float *b1,float *x1,char *cm,
char *cnm, char *cnm1r,float *vo1, FILE *fp);

```



```
void vnom (int c, int ni, int rq, int ni, int *viv, char *cm, char *cnm);
void variancia(int ni, int c, int q, float *varct, float *mfm, float *pps);
```

```
main()
{
    FILE *fp;
    int i,j;
    char s[20];

    clrscr();
    printf("\nPROGRAMA RETIF\n");
    printf("\nRETIFICACAO DE DADOS DE PROCESSO E COAPTACAO\n");
    printf("\nElaborado por: TALITA FURLANETTO MENDES\n");
    printf("\nOrientado por: Joao Alexandre F. R. Pereira\n");
    printf("\n\n\n\n\n\n\n\n\n\n\t\tdigite qualquer letra\n");
    getch();
    clrscr();

    printf("\nNome do arquivo de dados: ");
    gets(s);
    if ( (fp=fopen(s,"r")) == NULL ) {
        printf("Nao posso abrir arquivo\n");
        exit(1);
    }

    fscanf(fp,"%d%*c%d%*c%d%*c%d%*c%d%*c%d%*c",&ni,&mi,&c,&rq,&nr,&q);

    for ( i=0; i<ni; i++ ) fscanf(fp,"%d%*c%d%*c",&M[i][0],&M[i][1]);

    for ( i=0; i<ni; i++ ) fscanf(fp,"%f%*c",&varct[i]);

    for ( i=0; i<ni; i++ )
        for ( j=0; j<c+1; j++ ) fscanf(fp,"%f%*c",&MFM[i][j]);

    for ( i=0; i<rq; i++ )
        for ( j=0; j<c+2; j++ ) fscanf(fp,"%f%*c",&TE[i][j]);

    for ( i=0; i<nr; i++ ) for ( j=0; j<ni; j++ )
        fscanf(fp,"%f%*c",&TR[i][j]);

    for ( j=0, i=0; i<q; i++ ) fscanf(fp,"%f%*c",&X[i][j]);

    for ( i=0; i<(ni*(c+1))+rq; i++ ) fscanf(fp,"%d%*c",&viv[i]);

    for ( i=0; i<(mi*(c+1))+(nr*(c+1))+ni; i++ ) fscanf(fp,"%d%*c",&vie[i]);

    fclose(fp);

    reconciliar();
}
```

```

void reconciliar(void d)
{
char *comptroc;
float *vnob;
int miel, niel, nrc 1;
int m, ml, n;
int c1,c2, r, p;
int *ig, *fd;
float *a, *ac, *te, *me, *acc, *mfm;
float *bci,*acb, *bi, *ppi,*bl, *pp;
float *tr, *mr,*ar;
int *ip, *ic, *ir;
float *pl, *p1, *p2, *u, *pu, *up;
float *p3, *up3, *vi, *vno, *h, *y, *yh;
float *yb, *pps, *sb, *sby, *g, *ui, *x;
float *yx, *l, *aj, *blm, *bx, *vo;
char *cm, *cnm;
float a1;
double fabs( );
int c3,c4;
float *pj, *yi, *ib, *mml;
float *de, *nu, *za, *zal ;
int *zam, *vzad, *j1;
float *j2, *ny;
int *par, npar=0, rpart, *part, *zam2;
int resposta, resp__zj1=0, respostal, pml ;
float *cel, *rtt, *mg, *rt, *gi;
int *zj1, *zem1;
float *st,*mle, *aj1, *x1, *x12, *vo1, *vo2;
int *pvza;
float *tj,*zc;
char *cnmlr, *cnm2r;

float JT[50]={ 3.841, 5.991, 7.815, 9.488, 11.070, 12.592, 14.067,
15.507, 16.919, 18.307, 19.675, 21.026, 22.362, 23.685, 24.996,
26.296, 27.587, 28.869, 30.144, 31.410, 32.672, 33.924, 35.172,
36.515, 37.652, 38.885, 40.113, 41.337, 42.557, 43.773, 44.985,
46.194, 47.400, 48.602, 49.802, 50.998, 52.192, 53.384, 54.572,
55.758, 56.933, 58.107, 59.282, 60.457, 61.631, 62.806, 63.981,
65.156, 66.330, 67.505 };

float Zc[50] = { 1.96, 2.235, 2.39, 2.49, 2.57, 2.63, 2.68, 2.73, 2.76,
2.79, 2.825, 2.855, 2.88, 2.905, 2.93, 2.94, 2.959, 2.977, 2.995,
3.021, 3.028, 3.043, 3.048, 3.053, 3.077, 3.101, 3.105, 3.109, 3.132,
3.135, 3.137, 3.139, 3.172, 3.175, 3.177, 3.179, 3.211, 3.213, 3.214,
3.216, 3.218, 3.219, 3.220, 3.262,3.263,3.264,3.266,3.267,3.268,3.269 };

register int i,j,k, ki,kj,z;

FILE *fp;
char pa[20];

```

```

printf("Nome do arquivo dos resultados:");
gets(pa);
if ( ( fp = fopen(pa, "w")) == NULL ) {
    printf("Nao posso abrir arquivo\n");
    return;
}

c1 = c+1; mic1 = mi*c1; nic1 = ni*c1;
nr1 = nr*c1; c2 = c+2;
tj = TM(50);
zc = TM(50);
clrscr( );

for ( i=0, j=0; j<50; j++) JT(i,j) = JT[j];
for ( i=0, j=0; j<50; j++) Zc(i,j) = Zc[j];

fd = TMI(mi*2);
mfm = TM(mi*c1);
x = TM(q*1);
comptroc = TMC(3);

for ( i=0; i<ni; i++ ) for ( k=0; k<2; k++ ) M(i,k) = M[i][k];

for ( i=0; i<ni; i++ ) for ( j=0; j<c1; j++ ) MFM(i,j) = MFM[i][j];

for ( j=0, i=0; i<q; i++) X(i,j) = X[i][j];

a = TM(mi*mi);
for ( i=0; i<ni; i++ ) for ( j=0; j<mi; j++ ) {
    A(i,j) = ( M(i,0) == j+1 ? 1 : 0 );
    if ( A(i,j) == 0 )
        A(i,j) = ( M(i,1) == j+1 ? -1 : 0 );
}

ac = TM(mic1*nic1);
for ( i=0; i<mi; i++ ) for ( ki=0; ki<c1; ki++ )
    for ( j=0; j<ni; j++ ) for ( kj=0; kj<c1; kj++ )
        AC( (c1*i+ki), (c1*j+kj) ) = ( ki == kj ? A(j,i) : 0 );

free(a);
te = TM(rq*c2);
me = TM(rq*mic1);

for ( i=0; i<rq; i++ ) for ( j=0; j<c2; j++) TE(i,j) = TE[i][j];

for ( i=0; i<rq; i++ )
    for ( j=0; j<mi; j++ )
        for ( kj=0; kj<c1; kj++ )
            ME(i,c1*j+kj) = ( c1*j+kj == (TE(i,0)-1)*c1+kj ? TE(i,kj+1) : 0 );

free(te);

```

```

acc = TM(mic1*(mic1+rq));
for ( i=0; i<mic1; i++) for ( j=0; j<nic1; j++) ACC(i,j) = AC(i,j);
for ( i=0; i<mic1; i++)
  for ( j=nic1; j<nic1+rq; j++) ACC(i,j) = ME(j-nic1,i);

free(ac);
tr = TM(nr*ni);
mr = TM(nrcl*nic1);

for ( i=0; i<nr; i++) for ( j=0; j<ni; j++) TR(i,j) = TR[i][j];

for ( i=0; i<nr; i++) for ( ki=0; ki<c1; ki++)
  for ( j=0; j<ni; j++) for ( kj=0; kj<c1; kj++)
    MR( (c1*i+ki),(c1*j+kj) ) = ( ki == kj ? TR(i,j) : 0 );

free(tr);

ar = TM((mic1+nrcl)*(nic1+rq));
for ( i=0; i<mic1; i++) for ( j=0; j<nic1+rq; j++) AR(i,j) = ACC(i,j);

for ( i=mic1; i<mic1+nrcl; i++)
  for ( j=0; j<nic1; j++) AR(i,j) = MR(i-mic1,j);

for ( i=mic1; i<mic1+nrcl; i++) for ( j=nic1; j<nic1+rq; j++) AR(i,j)= 0;

free(mr); free(acc);

bci = TM(ni*(nic1+rq));
for ( i=0; i<ni; i++) { for ( j=0; j<nic1+rq; j++)
  BCI(i,j) = ( j >= (c1*i) && j < (c1*i)+c1 ? 1 : 0 );
}

for ( i=0; i<ni; i++) { for ( j=0; j<nic1+rq; j++)
  if ( BCI(i,j) && j == i*c1+(c1-1) ) BCI(i,j) = -1;
}

acb = TM((mic1+nrcl+ni)*(nic1+rq)),
for ( i=0; i<mic1+nrcl; i++)
  for ( j=0; j<nic1+rq; j++) ACB(i,j) = AR(i,j);
for ( i=mic1+nrcl; i<mic1+nrcl+ni; i++)
  for ( j=0; j<nic1+rq; j++) ACB(i,j) = BCI(i-(mic1+nrcl),j);

free(ar); free(bci);

for ( q=0, n=0, j=0; j<nic1+rq; j++)
  if ( viv[j] == 1 ) q++;
  else if ( viv[j] == 3 ) n++;

bi = TM((mic1+nrcl+ni)* q);
ppi = TM((mic1+nrcl+ni)* n);

```

```

for ( kj=0, ki=0, j=0; j<nicl+rq ; j++ ) if ( viv[j] == 1 ) {
    for ( i=0; i<micl+nrc1+ni; i++ ) Bi(i,kj)=ACB(i,j); kj++;
    }
else if ( viv[j] == 3 ) {
    for ( i=0; i<micl+nrc1+ni; i++ ) Pi(i,ki)=ACB(i,j); ki++;
    }

for ( m=0, i=0; i<micl+nrc1+ni; i++ ) if ( vie[i] == 1 ) m++;

b1 = TM(m*q);
for ( ki=0, i=0; i<micl+nrc1+ni; i++ ) if ( vie[i] == 1 ) {
    for ( j=0; j<q; j++ ) B1(ki,j) = Bi(i,j); ki++;
    }
}

free(bi);
cm = TMC(q*3);
cnm = TMC(n*3);

vnom(c,rq,ni,viv,cm,cnm);

yh=TM(m*m);

if ( n ) {

pp = TM(m*n);
for ( ki=0, i=0; i<micl+nrc1+ni; i++ ) if ( vie[i] == 1 ) {
    for ( j=0; j<n; j++ ) P(ki,j) = Pi(i,j); ki++;
    }

free(ppi);
pl = TM(m*n);
for ( i=0; i<m; i++ ) for ( j=0; j<n; j++ ) PL(i,j) = P(i,j);

ic = TMI(n);
for ( i=0; i<n; i++ ) ic[i] = i;
ip = TMI(m);
for ( i=0; i<m; i++ ) ip[i] = i;

r = triang(m,n,pl,pp,ip,ic);

printf("%d", fprintf(fp,"RESULTADOS DA RETIFICACAO DE DADOS E DA
COAPTACAO:\n"));

printf("%d", fprintf(fp,"\nn=%d r=%d\n",n,r));

u = TM(m*n);
pu = TM(m*n);
up = TM(m*m);
h = TM(m*m);

for ( i=0; i<m; i++ ) for ( j=0; j<m; j++ ) H(i,j) = ( i == j ? 1 : 0 );

```

```

p = m-r;
pm1 = p-1;
p1 = TM(r*r);
p2 = TM(p*r);

ir = TMI(r);
for ( i=0; i<r; i++) r r[i] = i;

for ( i=0; i<r; i++) for ( j=0; j<r; j++) P1(i,j) = P(i,j);
for ( i=0; i<p; i++) for ( j=0; j<r; j++) P2(i,j) = -P(i+r,j);
for ( j=0; j<r; j++) for ( z=0; z<r; z++) U(j,z) = ( j == z ? 1 : 0 );

lu(r,ir,p1);
backs(r,r,ir,p1,u);
prod_m(p,r,r,p2,u,uu);

for ( i=0 ; i<p ; i++ )
    for ( j=0 ; j j<p ; j++ ) UP(i,j) = ( i == j ? 1 : 0 );

y = TM(p*m);
for ( i=0; i<p; i++ ) {
    for ( j=0; j<r; j++ ) YT(i,j) = P2U(i,j);
    for ( j=r; j<m; j++ ) YT(i,j) = UP(i,j-r);
}

for ( m1=m-1, ip[m-1]=m1, i=0; i<m1; i++ ) for ( j=0; j<m; j++ ) {
    a1 = H(i,j); H(i,j) = H(ip[i],j); H(ip[i],j) = a1;
}

prod_m(p,m,m,y,hyh);
}

if ( !n ) {

p = m;
for ( i=0; i<m; i++ ) for ( j=0; j<m; j++ ) YHT(i,j) = ( i == j ? 1 : 0 );
}

yb = TM(p*q);
sb = TM(q*m);
sby = TM(q*m);
g = TM(p*p);
gi = TM(p*p);
aj = TM(q*1);

for ( a1=0, i=0; i<p; i++ ) for ( j=0; j<q; j++ ) {
    for ( z=0; z<m; z++ ) {
        a1 += YHTYHT(i,z)*B1(z,j);
    }
}

```

```

    YTB1(i,j) = a1;    a1 = 0;
  }

pps = TM(q*q);
variancia(ni,c,q,varct,mfm,pps);

for ( a1=0, i=0; i<q; i++ ) for ( z=0; z<m; z++ ) {
  for ( j=0; j<q; j++ ) {
    a1 += S(i,j)*B1(z,j);
  }
  SBT(i,z) = a1; a1 = 0;
}

.....

for ( a1=0, i=0; i<q; i++ ) for ( j=0; j<p; j++ ) {
  for ( z=0; z<m; z++ ) {
    a1 += SBT(i,z) * YHT(j,z);
  }
  SBY(i,j) = a1; a1 = 0;
}

for ( a1=0, i=0; i<p; i++ ) for ( z=0; z<p; z++ ) {
  for ( j=0; j<q; j++ ) {
    a1 += YTB1(i,j) * SBY(j,z);
  }
  G(i,z) = a1; a1 = 0;
}

for ( i=0; i<p; i++ ) for ( j=0; j<p; j++ ) GI(i,j) = G(i,j);

ui = TM(p*p);
for ( i=0; i<p; i++ ) for ( j=0; j<p; j++ ) I(i,j) = ( i == j ? 1 : 0 );

ig = TMI(p);
for ( i=0; i<p; i++ ) ig[i] = i;

lu(p,ig,g);
backs(p,p,ig,g,ui);

yx = TM(p*1);
for ( i=0; i<p; i++ ) { for ( a1=0, j=0, z=0; z<q; z++ ) {
  a1 += YTB1(i,z) * X(z,j);
}
  YTX(i,j) = a1;
}

l = TM(p*1);
for ( i=0; i<p; i++ ) { for ( a1=0, j=0, z=0; z<p; z++ ) {
  a1 += I(i,z) * YTX(z,j);
}
  L(i,j) = -a1;
}

```

```

for ( i=0; i<q; i++ ) { for ( a1=0, j=0, z=0; z<p; z++ ) {
    a1 += SBY(i, z) * L(z, j);
    }
    AJ(i, j) = a1;
}

for ( j=0, i=0; i<q; i++ ) X(i, j) += AJ(i, j);

printf("%d", fprintf(fp, "\nNomes das Vazoes Medidas\n"));
for ( i=0; i<q; i++ ) {
    for ( j=0; j<3; j++ ) printf("%d", fprintf(fp, "%c", CM(i, j)));
    printf("%d", fprintf(fp, " "));
}

printf("%d", fprintf(fp, "\n\nValores Corrigidos das Vazoes Medidas\n"));
for ( j=0, i=0; i<q; i++ ) printf("%d", fprintf(fp, "%8.2f", X(i, j)));
printf("%d", fprintf(fp, "\n"));

if ( n ) {
    vnob = TM(n);
    for ( i=0; i<n; i++ ) vnob[i] = 0;

    for ( j=0; j<n-1; j++ ) if ( j < ic[j] )
        for ( i=0; i<3; i++ ) {
            comptroc[i] = CNM(j, i); CNM(j, i) = CNM(ic[j], i);
            CNM(ic[j], i) = comptroc[i];
        }

    printf("%d", fprintf(fp, "\n\nNomes das Quantidades Nao-Medidas\n"));
    for ( i=0; i<n; i++ ) {
        for ( j=0; j<3; j++ ) printf("%d", fprintf(fp, "%c", CNM(i, j)));
        printf("%d", fprintf(fp, " "));
    }
    printf("%d", fprintf(fp, "\n"));

    if ( r == n )
        printf("%d", fprintf(fp, "\nTodas quantidades nao-medidas sao observaveis\n"));
    else {

        p3 = TM(r*(n-r));
        for ( i=0; i<r; i++ ) for ( j=r; j<n; j++ ) P3(i, j-r) = P(i, j);

        up3 = TM(r*(n-r));
        for ( i=0; i<r; i++ ) for ( j=0; j<n-r; j++ ) {
            for ( a1=0, z=0; z<r; z++ ) {
                a1 += -U(i, z) * P3(z, j);
            }
            UP3(i, j) = a1;
        }

        vi = TM((n-r)*(n-r));

```



```
for ( i=0; i<n-r; i++) for ( j=0; j<n-r; j++) VI(i,j) = ( i == j ? 1 : 0 );
```

```
vno = TM(n*(n-r));
for ( i=0; i<r; i++) for ( j=0; j<n-r; j++) VNO(i,j) = UP3(i,j);
for ( i=r; i<n; i++) for ( j=0; j<n-r; j++) VNO(i,j) = VI(i-r,j);
```

```
for ( i=0; i<n; i++) { for ( a1=0, j=0; j<n-r; j++) {
    a1 += fabs(VNO(i,j));
}
    vnob[i] = a1;
}
printf("%d", fprintf(fp, "\nQuantidades Nao-Observaveis: "));
for ( i=0; i<n; i++) if ( vnob[i] != 0 ) {
    for ( j=0; j<3; j++)
        printf("%d", fprintf(fp, "%c", CNM(i,j)));
    printf("%d", fprintf(fp, " "));
}
printf("%d", fprintf(fp, "\n"));
}
```

```
for ( m1=m-1, ip[m-1]=m1, i=0; i<m1; i++)
    for ( j=0; j<q; j++) {
        a1 = B1(i,j); B1(i,j) = B1(ip[i],j); B1(ip[i],j) = a1;
    }
```

```
b1m = TM(r*q);
for ( i=0; i<r; i++) for ( j=0; j<q; j++) B1m(i,j) = -B1(i,j);
```

```
bx = TM(r*1);
for ( i=0; i<r; i++) { for ( a1=0, z=0, j=0; j<q; j++) {
    a1 += B1m(i,j) * X(j,z);
}
    BX(i,z) = a1;
}
```

```
vo = TM(n*1);
for ( i=0; i<r; i++) if ( vnob[i] == 0 ) {
    for ( a1=0, z=0, j=0; j<r; j++) {
        a1 += U(i,j) * BX(j,z);
    }
    VO(i,z) = a1;
}
else VO(i,z) = 0;
```

```
for ( z=0, i=r; i<n; i++) VO(i,z) = 0;
```

```
printf("%d", fprintf(fp, "\nValores das Quantidades Observaveis:\n"));
for ( j=0, i=0; i<n; i++) printf("%d", fprintf(fp, "%8.2f", VO(i,j)));
printf("%d", fprintf(fp, "\n"));
}
```

```
else printf("%d", fprintf(fp, "\nNao ha quantidades nao-medidas\n"));
```

```
ib = TM(p*q);
mm1 = TM(p*p);
za1 = TM(q*q);
zam = TMI(q*1);
vzad = TMI(q*q);
j2 = TM(q*q);
ny = TM(p*q);
ce1 = TM(p*1);
rtt = TM(pm1*p);
mg = TM(pm1*pm1);
rt = TM(pm1*p);
zj1 = TMI(q*1);
zem1 = TMI(1*q);
st = TM(q*p);
mle = TM(p*1);
aj1 = TM(q*q);
x1 = TM(q*q);
pj = TM(q*2);
nu = TM(1*q);
yi = TM(q*p);
de = TM(1*q);
za = TM(1*q);
j1 = TMI(q*1);
```

```
IEG(p,q,ui,yx,l,pj,yb,yi,de,nu,za);
```

```
printf("%d", fprintf(fp, "\nJ(calculado) =%8.3f, p=%d\n", J(0,0),p));
printf("%d", fprintf(fp, "\nJT(s/rem.) =%8.3f\n", JT(0,p-1)));
printf("%d", fprintf(fp, "\nEstatisticas de Medida\n\n"));
```

```
for ( i=0, j=0; j<q; j++) if ( DE(i,j) )
    printf("%d", fprintf(fp, "Za(%d)=%6.3ft", j, Za(i,j)));
    else { Za(i,j) = 0 ;
        printf("%d", fprintf(fp, "Za(%d)= na\t", j ));
    }
```

```
pvza=TMI(q);
c4 = emdif(q,pvza,za,zc,fp);
c3 = comp_J(p,pj,tj);
```

```
if ( !c4 && !c3 ) { resposta = ZEzero(p,yx,gi,zc,fp);
if ( !resposta ) {
printf("%d", fprintf(fp, "\n\nNao ha indicios de haver erros extremos\n"));
goto fim;
}
}
```

```
R1VM(q,za,pj,pvza);
```

```

printf("%d", fprintf(fp, "\n\nValores de J apos uma remocao\n\n"));
for ( j=0; j<q; j++ )
  printf("%d", fprintf(fp, "J(medida %d)=%8.3fn", j, J(j,1)));
printf("%d", fprintf(fp, "\n"));

comp_J1(p,q,pj,jl,tj);
printf("%d", fprintf(fp, "\nJT(1 rem.)=%8.3fn", JT(0,p-2)));

VZaD(p,q,yb,yi,ui,ib,mm1,yx,nu,de,za1,pvza,vzad,zam,zc);
for ( z=0, k=0; k<q; k++ ) {
  if( !J1(k,z) && !ZAM(k,z) ) {
    ZJ1(k,z)=0; resp_zj1 ++;
  }
  else ZJ1(k,z) = 1;
}

if( resp_zj1 ) {
  resposta1=ZEUma(p,q,pm1,yb,ce1,mg,rtt,rt,yx,gi,zj1,zem1,zc);

  printf("%d", fprintf(fp, "\nZEM1:"));
  for ( j=0, k=0; k<q; k++ ) printf("%d", fprintf(fp, "\t%d", ZEM1(j,k)));

  if( resposta1 ) {
    Valcorr1(p,q,pps,yb,st,mm1,ib,yi,m1e,yx,aj1,x,x1,zem1,pvza);

    printf("%d", fprintf(fp, "\n\nValores Corrigidos apos 1 remocao\n\n"));

    for ( j=0, k=0; k<q; k++ ) if( !ZEM1(j,k) ) {
      printf("%d", fprintf(fp, "\nMedida removida: "));
      for ( z=0; z<3; z++ ) printf("%d", fprintf(fp, "%c", CM(k,z)));
      printf("%d", fprintf(fp, "\n"));
      for ( i=0; i<q; i++ ) printf("%d", fprintf(fp, "%8.2ft", X1(i,k)));
    }

    cnm1r=TMC((n+1)*3);
    vol=TM((n+1)*q);

    Val_obs1(m,n,q,zem1,pvza,pp,b1,x1,cm,cnm,cnm1r,vol,fp);

    printf("%d", fprintf(fp, "\n\nQuantidades nao-medidas apos 1 remocao\n\n"));
    for ( i=0; i<n+1; i++ ) {
      for ( j=0; j<3; j++ ) printf("%d", fprintf(fp, "%c", CNM1R(i,j)));
      printf("%d", fprintf(fp, " "));
    }

    for ( j=0, k=0; k<q; k++ ) if( !ZEM1(j,k) || !ZEM1(j,pvza[k]) ) {
      printf("%d", fprintf(fp, "\n\nMedida removida: "));
      for ( z=0; z<3; z++ ) printf("%d", fprintf(fp, "%c", CM(k,z)));
      printf("%d", fprintf(fp, "\n"));
      for ( i=0; i<n+1; i++ ) printf("%d", fprintf(fp, "%8.2ft", VO1(i,k)));
    }

```

```

    goto fim;
}
}

R2VM(q,pj,za1,vzad,j2,pvza);
npar=comp_J2(p,q,pvza,vzad,za1,j2,tj);
printf("%d", fprintf(fp, "\nnpar=%d\n", npar));
printf("%d", fprintf(fp, "\nJT(2 rem.)=%8.3f\n", JT(0,p-3)));

par=TMI(npar*2);
Mos_par(p,q,pvza,vzad,za1,j2,par,tj,fp);
if ( !npar )
    printf("%d", fprintf(fp, "\nHa mais de dois erros grosseiros\n"));
    else {
        zam2=TMI(1*npar);
        VZDA(p,q,ib,yb,ui,mm1,yb,ny,za1,vzad,npar,par,zam2,zc);

        printf("%d", fprintf(fp, "\nZAM2:"));
        for ( z=0, k=0; k<npar; k++ ) printf("%d", fprintf(fp, "%3d", ZAM2(z,k)));
        printf("%d", fprintf(fp, "\n"));

        part = TMI(npar*3);
        npart=ZEduas(p,q,pm1,yb,ce1,mg,rtt,rt,yx,gi,npar,par,zam2,pvza,part,zc,fp);

        printf("%d", fprintf(fp, "\nPares suspeitos\n"));
        for ( i=0; i<npart; i++ ) {
            printf("%d", fprintf(fp, "Par numero %d:", i+1));
            for ( k=0; k<2; k++ ) {
                for ( j=0; j<3; j++ )
                    printf("%d", fprintf(fp, "%c", CM(PART(i,k),j) ));
                printf("%d", fprintf(fp, " "));
            }
            printf("%d", fprintf(fp, "\n"));
        }

        x12=TM(q*npart);
        Valcorr2(p,q,npart,pps,st,mm1,ib,yi,m1e,yx,aj1,x,x12,part,ui,yb,ny,za1);

        printf("%d", fprintf(fp, "\nValores Corrigidos apos 2 remocoes\n"));
        for ( z=0; z<npart; z++ ) {
            printf("%d", fprintf(fp, "\nMedidas removidas: Par numero %3d\n", z+1));
            for ( i=0; i<q; i++ ) printf("%d", fprintf(fp, "%8.2f", X12(i,z)));
            printf("%d", fprintf(fp, "\n"));
        }
        cnm2r = TMC((npart-2)*3);
        vo2 = TM((npart-2)*npart);
        Val_obs2(m,n,q,npart,part,pp,b1,x12,cm,cnm,cnm2r,vo2,fp);
    }
}
fim: ;
fclose(fp);
}

```

```

int triang(m,n,pl,pp,ip,ic)
int m,n,*ip,*ic;
float *pl,*pp;
{
  int m1 = m-1,r;
  float a1,a2;
  double fabs( );
  register int i,j,k,s,t;

  for ( i=0; i<n; i++ ) {
    for ( j=(k=i)+1; j<m; j++ )
      if ( fabs(PL(j,i)) > fabs( PL(k,i)) ) k = j;
    if ( k != i ) for ( j=i; j<n; j++ ) {
      a1 = PL(i,j); PL(i,j) = PL(k,j); PL(k,j) = a1;
    }

    a2 = PL(i,i); t = 0;
    if ( a2 == 0 && t == 0 ) for ( j=i+1; j<n; j++ )
      for ( s=i; s<m; s++ )
        if ( PL(s,j) != 0 && t == 0 ) for ( j=0; j<n; j++ ) {
          a1 = PL(i,j); PL(i,j) = PL(s,j); PL(s,j) = a1;
          t = 1; k = s;
        }

    a2 = PL(i,i); t = 0;
    if ( a2 == 0 && t == 0 ) for ( s=i+1; s<n; s++ )
      if ( PL(i,s) != 0 && t == 0 )
        for ( j=0; j<m; j++ ) {
          a1 = PL(j,i); PL(j,i) = PL(j,s); PL(j,s) = a1;
          t = 1; ic[i] = s;
        }

    a2 = PL(i,i);
    for ( ip[i]=k, j=i+1; j<m && a2 != 0; j++ ) {
      a1 = PL(j,i) / -a2; PL(j,i) = 0;
      for ( k=i+1; k<n; k++ ) PL(j,k) += a1 * PL(i,k);
    }
  }

  for ( m1=m-1, ip[m1]=m1, i=0; i<m1; i++ )
    for ( j=0; j<n; j++ ) {
      a1 = P(i,j); P(i,j) = P(ip[i],j); P(ip[i],j) = a1;
    }

  for ( j=0; j<n-1; j++ ) if ( j < ic[j] )
    for ( i=0; i<m; i++ ) {
      a1 = P(i,j); P(i,j) = P(i,ic[j]); P(i,ic[j]) = a1;
    }

  for ( a1=0, r=0, i=0; i<n && i<m; i++ )
    if ( fabs( PL(i,i) ) > 0.0001 ) r++;
}

```

```

return r;
}

```

```

int lu(int r,int *ir,float *p1)
{
int r1 = r-1;
register int i,j,k;
float a1,a2,double fabs( );

for ( ir[r1]=r1, i=0 ; i<r1 ; i++ ) {
for ( j = ( k=i ) +1 ; j<r ; j++ )
if ( fabs( P1(j,i)) > fabs( P1(k,i))) k = j;
if ( k != i ) for ( j=i ; j<r ; j++ ) {
a1 = P1(i,j) ; P1(i,j) = P1(k,j) ; P1(k,j) = a1 ;
}
if ( !(a2 = P1(i,i), a2 )) return (-1);

for ( ir[i]=k, j=i+1 ; j<r ; j++ ) {
a1 = ( P1(j,i) /=-a2 );
for ( k = i+1 ; k<r ; k++ ) P1(j,k) += a1 * P1(i,k) ;
}
}
if ( !P1(r1,r1) ) return (-1) ;
return(0);
}

```

```

void backs (int r, int rs, int *ir, float *p1, float *u )
{
register int i, j, k;
int r1 = r - 1;
float a1;
for ( i = 0; i < r1; i++ ) {
if ( ( j = ir[i] ) != i )
for ( k = 0; k < rs; k++ )
{
a1 = U(i,k); U(i,k) = U(j,k); U(j,k) = a1;
}
for ( j = i + 1; j < r; j++ )
for ( k = 0; k < rs; k++ ) U(j,k) += P1(j,i) * U(i,k);
}
for ( k = 0; k < rs; k++ ) U(r1,k) /= P1(r1,r1);
for ( i = r1 - 1; i >= 0; i-- ) {
for ( a1 = P1(i,i), j = i + 1; j < r; j++ )
for ( k = 0; k < rs; k++ ) U(i,k) -= P1(i,j) * U(j,k);
for ( k = 0; k < rs; k++ ) U(i,k) /= a1;
}
}
}

```

```

void prod_m(int s,int r,int w,float *p2,float *u,float *pu)
{
  register int i,j,z;
  float a1;

  for ( a1=0, i=0; i<s; i++ ) for ( z=0; z<w; z++ ) {
    for ( j=0; j<r; j++ ) {
      a1 += P2(i,j) * U(j,z) ;
    }
    P2U(i,z) = a1; a1 = 0;
  }
}

```

```

void IEG(p,q,ui,yx,l,pj,yb,yi,de,nu,za)
int p,q;
float *ui,*yx,*l,*pj,*yb,*yi,*za,*de,*nu;
{
  register int i,j,z,k;
  double a1;

  for ( a1=0, i=0; i<l; i++ ) for ( z=0; z<1; z++ ) {
    for ( j=0; j<p; j++ ) {
      a1 += YTX(j,i) * -L(j,z);
    }
    J(i,z) = a1; a1 = 0;
  }

  for ( j=0; j<q; j++ ) {
    for ( k=0, a1=0, i=0; i<p; i++ ) {
      a1 += YTB1(i,j) * L(i,k) ;
    }
    NU(k,j) = a1;
  }

  for ( j=0; j<q; j++ ) {
    for ( a1=0, z=0; z<p; z++ ) {
      for ( i=0; i<p; i++ ) {
        a1 += YTB1(i,j) * I(i,z) ;
      }
      YI(j,z) = a1; a1 = 0;
    }
  }

  for ( i=0, a1=0, z=0; z<p; z++ ) {
    a1 += YI(j,z) * YTB1(z,j);
  }
  DE(i,j) = sqrt( a1);
  a1 = 0;
}

for ( i=0, j=0; j<q; j++ ) if ( DE(i,j) )

```

```

        Za(i,j) = NU(i,j)/DE(i,j);
    else Za(i,j) = 0 ;
}

int emdif(q,pvza,za,zc,fp)
int q, *pvza;
float *za, *zc;
FILE *fp;
{
    register int i,j,k;
    int count=0, nmm=0;

    for ( j=0; j<q; j++) pvza[j] = j;
    for ( k=0; k<q-1; k++ ) {
        for ( i=0, j=k+1; j<q; j++ )
            if ( fabs( fabs(Za(i,k)) - fabs(Za(i,j))) <= 0.01 )
                if ( pvza[j] == j ) pvza[j] = k;
    }

    for ( count=0, j=0; j<q; j++ ) if ( pvza[j] == j ) count++ ;

    for ( nmm=0, i=0, j=0; j<q; j++ ) {
        if( fabs(Za(i,j) ) > Zc(0,count-1) ) nmm++;
    }
    printf("%d", fprintf(fp, "\n\nZc(estat. de medida)= %8.3fn", Zc(0,count-1)));
    return nmm;
}

int comp_J(p,pj,tj)
int p;
float *pj,*tj;
{
    if ( J(0,0) > JT(0,p-1) ) return 1;
    return 0;
}

int ZEzero(p,yx,gi,zc,fp)
int p;
float *yx,*gi,*zc;
FILE *fp;
{
    float *ze;
    register int i,j,k;
    int count = 0 , nmm = 0, pml = p-1;
    int *vze;

    ze = TM(1*p);
    for ( j=0, i=0; i<p; i++ ) ZE(j,i) = YTX(i,j) / sqrt( GI(i,i));
}

```



```

printf("%d", fprintf(fp, "\n\nEstatisticas Nodais\n"));
printf("%d", fprintf(fp, "\nZE\n"));
for ( j=0, i=0; i<p; i++) printf("%d", fprintf(fp, "%8.3f\t", ZE(j,i)));

vze = TMI(p);
for ( i=0; i<p; i++ ) vze[i] = i;

for ( k=0; k<pm1; k++ ) for ( i=0, j=k+1; j<p; j++ )
    if ( fabs( fabs(ZE(i,k) ) - fabs( ZE(i,j))) <= 0.01 )
        if ( vze[j] == j ) vze[j] = k;

for ( i=0; i<p; i++ ) if ( vze[i] == i ) count++;

for ( j=0, i=0; i<p; i++ ) if ( fabs(ZE(j,i) ) > Zc(0,count-1) ) nmm++;

printf("%d", fprintf(fp, "\n\nZc(estadistica nodal)= %8.3f\n", Zc(0,count-1)));

return nmm;
}

```

```

void R1VM(q,za,pj,pvza)
int q, *pvza;
float *za, *pj;
{
    register int i, j;
    double pow( ), a1=0, a2=2, a3=0;

    for ( i=0, j=0; j<q; j++ ) { if ( pvza[j] == j ) {
        a1 = J(0,0); a3 = Za(i,j);
        a3 = pow(a3,a2); a1 -= a3;
        J(j,1) = a1;
    }
    else J(j,1) = J(pvza[j],1);
}
}

```

```

void comp_J1(p,q,pj,j1,tj)
int p,q,*j1;
float *pj,*tj;
{
    register int j,z;

    for ( z=0, j=0; j<q; j++ ) if ( J(j,1) < JT(z,p-2) ) J1(j,z) = 0;
    else J1(j,z) = 1;
}

```

```

void VZaD(p,q,yb, yi,ui,ib,mml,yx,nu,de,za1,pvza,vzad,zam,zc)

```

```

int p, q, *pvza, *vza, *d, *zam;
float *yb, *yi, *ui, *ib, *mm1, *yx, *nu, *de, *zal, *zc;
{
float *gl, *ym, *dn, *dd;
int i,k,j,z=0;
int nmm1=0, emd1 =0;
float a1;

gl = TM(1*q);
for ( a1=0, z=0, k=0; k<q; k++ ) {
for ( i=0; i<p; i++ ) {
a1 += YI(k,i) * YTB1(i,k);
}
if ( a1 ) GL(z,k) = 1/a1;
else GL(z,k) = 0; a1 = 0;
}

for ( k=0; k<q; k++ ) for ( a1=0, i=0; i<p; i++ ) {
for ( j=0; j<p; j++ ) {
a1 += I(i,j) * YTB1(j,k);
}
IB(i,k) = a1 * GL(z,k); a1 = 0;
}

ym = TM(q*p);
for ( k=0; k<q; k++ ) { for ( a1=0, i=0; i<p; i++ )
for ( j=0; j<p; j++ ) {
a1 = IB(i,k) * YI(k,j);
M1(i,j) = a1; a1 = 0;
}
}

for ( a1=0, i=0; i<q; i++ ) for ( z=0; z<p; z++ ) {
for ( j=0; j<p; j++ ) {
a1 += YTB1(j,i) * M1(j,z);
}
YM(i,z) = a1; a1=0;
}

dn = TM(1*q);
for ( a1=0, i=0; i<q; i++ ) for ( z=0; z<p; z++ ) {
for ( j=0; j<p; j++ ) {
a1 += YM(i,j) * YTX(j,z);
}
DN(z,i) = a1; a1 = 0;
}

dd = TM(1*q);
for ( z=0, a1=0, i=0; i<q; i++ ) {
for ( j=0; j<p; j++ ) {
a1 += -YM(i,j) * YTB1(j,i);
}
}

```

```

DD(z,i) = a1; a1=0;
}

for ( a1=2, z=0, i=0; i<q; i++ ) if( k != i )
  if ( pvza[i] != k && pvza[k] != i && DE(z,i) ) {
    DN(z,i) += NU(z,i);
    DD(z,i) += pow( DE(z,i), a1 );
    DD(z,i) = sqrt ( DD(z,i) );
    if( !DD(z,i) ) DD(z,i) = 0.01;
    ZA1(k,i) = DN(z,i)/DD(z,i);
  }
  else ZA1(k,i)=8;
  else ZA1(k,k)=8;
}

for ( k=0; k<q; k++ ) {
  for ( j=0; j<q; j++ ) VZAD(k,j) = j;

  for ( i=0; i<q-1; i++ ) for ( j=i+1; j<q; j++ )
    if ( fabs( fabs(ZA1(k,i) ) - fabs(ZA1(k,j))) <= 0.01 )
      if ( VZAD(k,j) == j && ZA1(k,j) != 8 ) VZAD(k,j) = i;

  for ( emd1=0, j=0; j<q; j++ )
    if ( VZAD(k,j) == j && ZA1(k,j) != 8 ) emd1++;

  for ( nmm1=0, z=0, j=0; j<q; j++ ) {
    if ( ZA1(k,j) != 8 )
      if( fabs(ZA1(k,j) ) > Zc(0,emd1-1) ) nmm1++;
  }
  ZAM(k,z) = nmm1;
}
}

void R2VM(q,pj,za1,vzad,j2,pvza)
int q, *vzad, *pvza;
float *za1,*j2,*pj;
{
  register int k,j,z;
  double pow( ),a1=0, a2=2, a3=0;

  for ( k=0; k<q; k++ ) { for ( z=1, j=0; j<q; j++ )
    if ( pvza[k] == k ) if ( VZAD(k,j) == j && ZA1(k,j) != 8 && k < j ) {
      a1 = J(k,z); a3 = ZA1(k,j);
      a3 = pow(a3,a2); a1 -= a3;
      J2(k,j) = a1;
    }
    else J2(k,j)=70;
  else J2(k,j)=80;
}
}
}

```

```

int comp_J2(p,q,pvza,vzad,za1,j2,tj)
int p,q, *pvza, *vzad;
float *j2, *za1, *tj;
{
  register int i,k ;
  int npar=0;

  for ( k=0; k<q; k++ ) if ( pvza[k] == k )
    for ( i=0; i<q; i++ ) if ( VZAD(k,i) == i && ZA1(k,i) != 8 && k < i )
      if ( J2(k,i) < JT(0,p-3) ) {
        npar += 1;
      }
  return npar;
}

void Mos_par(p,q,pvza,vzad,za1,j2,par,tj,fp)
int p,q, *pvza, *vzad, *par ;
float *j2, *za1, *tj;
FILE *fp;
{
  register int i,k ;
  int npar=0;

  for ( k=0; k<q; k++ ) if ( pvza[k] == k )
    for ( i=0; i<q; i++ ) if ( VZAD(k,i) == i && ZA1(k,i) != 8 && k < i )
      if ( J2(k,i) < JT(0,p-3) ) {
        PAR(npar,0) = k; PAR(npar,1) = i;
        npar += 1;
      }

  printf("%d", fprintf(fp, "\nValores de J apos duas remocoes\n"));
  for(i=0;i<npar;i++)
    printf("%d", fprintf(fp, "J2(%d,%d)=%6.2f\n", PAR(i,0), PAR(i,1), J2(PAR(i,0), PAR(i,1))));
}

void VZDA(p,q,ib,yi,ui,mm1,yb,ny,za1,vzad,npar,par,zam2,zc)
int p,q,npar;
int *par,*vzad,*zam2;
float *ib,*yi,*ui,*mm1,*yb,*ny,*za1,*zc;
{
  int *vzd;
  float *ym1,*ga,*zda,*zdap;
  register int i,j,k,t,s,z;
  int emd2=0, nmm2=0;
  float a1,a2,a3=0,a4;
  ym1=TM(q*p);
  ga=TM(q*q);
  zda=TM(q*q);
  zdap=TM(npar*q);
  vzd=TMI(npar*q);
}

```

```

for ( z=0; z<npar; z++ ) {
  k = PAR(z,0); i = PAR(z,1);
  for ( a1=0, t=0; t<p; t++ ) for ( s=0; s<p; s++ ) {
    a1 = IB(t,k) * YI(k,s);
    M1(t,s) = a1; a1=0;
  }

  for ( s=0; s<p; s++ ) for ( t=0; t<p; t++ ) M1(s,t) = I(s,t) - M1(s,t);

  for ( a1=0, s=0; s<p; s++ ) {
    for ( t=0; t<p; t++ ) {
      a1 += M1(s,t) * YTB1(t,i);
    }
    NY(s,i) = a1; a1=0;
  }
  for ( a1=0, s=0; s<p; s++ ) {
    a1 += YTB1(s,i) * NY(s,i);
  }
  a2=a1;

  for ( a1=0, j=0 ; j<q; j++ ) {
    if( ZA1(k,j) != 8 && VZAD(k,i) == i && VZAD(k,j) == j && i != j &&
      ZA1(k,i) != 8 ) {
      a1=0;
      for ( s=0; s<p; s++ ) {
        a1 += YTB1(s,j) * NY(s,i);
      }
      a3 = a1;

      for ( a1=0, t=0; t<p; t++ ) {
        for ( s=0; s<p; s++ ) {
          a1 += YTB1(s,j) * M1(s,t);
        }
        YM1(j,t)=a1; a1=0;
      }

      for ( a1=0, t=0; t<p; t++ ) {
        a1 += YM1(j,t) * YTB1(t,j);
      }
      a4=a1; a1=0;

      GA(i,j)= a3/sqrt(a2*a4);

      ZDA(i,j) = (ZA1(k,j)-GA(i,j)*ZA1(k,i))/sqrt(1-pow(GA(i,j),2));
    }
    else ZDA(i,j) = 25;
    ZDAP(z,j) = ZDA(i,j);
  }
}

for ( k=0; k<npar; k++ ) {

```

```

for ( j=0; j<q; j++ ) VZD(k,j) = j;

for ( i=0; i<q-1; i++ ) for ( j=i+1; j<q; j++ )
    if ( fabs( fabs(ZDAP(k,i) ) - fabs(ZDAP(k,j)) ) <= 0.01 )
        if ( VZD(k,j) == j && ZDAP(k,j) != 25 ) VZD(k,j) = i;

for ( emd2=0, j=0; j<q; j++ )
    if ( VZD(k,j) == j && ZDAP(k,j) != 25 ) emd2 += 1 ;

for ( nmm2=0, z=0, j=0; j<q; j++ ) {
    if ( VZD(k,j) == j && ZDAP(k,j) != 25 )
        if ( fabs(ZDAP(k,j)) > Zc(0,emd2-1)) nmm2++;
    }
    ZAM2(z,k) = nmm2;
}
}
int ZEuma(p,q,pm1 ,yb,cel,mg,rtt,rt,yx,gi,zj1,zem1,zc)
int p,q,pm1,*zj1,*zem1;
float *yb, *cel, *mg, *rtt, *rt,*yx,*gi, *zc;
{
    register int i,j,k,t;
    float a1;
    float *re,*zel,*rg,*rr,*rgr;
    int count1=0,nmm1=0;
    int *vzel,*pcel;
    int respostal=0;

    pcel=TM(p);
    re=TM(pm1*1);
    zel=TM(q*pm1);
    rg=TM(pm1*p);
    rr=TM(p*pm1);
    rgr=TM(pm1*pm1);

    for ( t=0, k=0; k<q; k++ ) if ( !ZJ1(k,0) ) {
        for ( i=0; i<p; i++ ) pcel[i] = i;
        for ( j=0, i=0; i<p; i++ ) CE1(i,j) = YTB1(i,k);

        for ( j=0, i=j+1; i<p; i++ )
            if ( !CE1(j,j) && CE1(i,j) ) {
                pcel[j] = i ; pcel[i] = j ;
                a1 = CE1(j,j); CE1(j,j) = CE1(i,j); CE1(i,j) = a1;
            }
        for ( t=0, i=0; i<pm1; i++ ) for ( j=0; j<pm1; j++ ) MG(i,j) = (i == j ? -CE1(t,t) : 0);

        for ( i=0; i<pm1; i++ ) {
            for ( j=0; j<1; j++ ) RTT(i,j) = CE1(i+1,j);
            for ( j=1; j<p; j++ ) RTT(i,j) = MG(i,j-1);
        }

        for ( i=0; i<pm1; i++ )

```

```

for ( j=0; j<p; j++) RT(i,j) = RTT(i,pce1[j]);

for ( a1=0, i=0; i<pml; i++ )
  for ( t=0; t<1; t++) {
    for ( j=0; j<p; j++) {
      a1 += RT(i,j) * YTX(j,t);
    }
    RE(i,t) = a1; a1=0;
  }

for ( a1=0, i=0; i<pml; i++ ) for ( t=0; t<p; t++ ) {
  for ( j=0; j<p; j++) {
    a1 += RT(i,j) * GI(j,t);
  }
  RG(i,t) = a1; a1=0;
}
for ( i=0; i<pml; i++ ) for ( j=0; j<p; j++ ) RR(j,i) = RT(i,j);

for ( a1=0, i=0; i<pml; i++ ) for ( t=0; t<pml; t++ ) {
  for ( j=0; j<p; j++) {
    a1 += RG(i,j) * RR(j,t);
  }
  RGR(i,t) = a1; a1=0;
}

for ( j=0, i=0; i<pml; i++ ) ZE1(k,i) = RE(i,j)/sqrt( RGR(i,i));
}
else for ( j=0; j<pml; j++ ) ZE1(k,j) = 10;

vze1 = TMI(pml);
for ( k=0; k<q; k++ ) {
  for ( i=0; i<pml; i++ ) vze1[i] = i;

for ( t=0; t<pml-1; t++ ) for ( j=t+1; j<pml; j++ )
  if ( fabs ( fabs(ZE1(k,t) ) - fabs( ZE1(k,j))) <= 0.01 )
    if ( vze1[j] == j ) vze1[j] = t;

for ( count1=0, i=0; i<pml; i++ ) if ( vze1[i] == i ) count1++;

for ( nmm1=0, t=0, i=0; i<pml; i++ )
  if ( fabs(ZE1(k,i)) > Zc(0,count1-1) ) nmm1++;

ZEM1(t,k) = nmm1;
}

for ( t=0, k=0; k<q; k++ )
  if ( !ZEM1(t,k) ) respostal++;
return respostal;
}

```

```

int ZEduas(p,q,pm1,yb,cel,mg,rtt,rt,yx,gi,npar,par,zam2,pvza,part,zc,fp )
int p,q,pm1,npar;
int *par, *zam2, *pvza, *part;
float *yb,*cel,*mg,*rtt,*rt,*yx,*gi,*zc;
FILE *fp;
{
  register int i,j,k,s,t,z;
  float a1;
  int nparf=0, nparb=0, npart=0;
  int *parf;
  int pm2=p-2;
  float *ce2,*mg2,*rtt2,*rt2,*r2t,*re2;
  float *rg2,*r2r,*rgr2,*ze2;
  int count2=0,nmm2=0;
  int *vze2,*zem2;
  int *pcel,*pce2;
  pcel = TMI(p);
  pce2 = TMI(pm1);
  zem2 = TMI(1*npar);
  ce2 = TM(pm1*1);
  mg2 = TM(pm2*pm2);
  rtt2 = TM(pm2*pm1);
  rt2 = TM(pm2*pm1);
  r2t = TM(pm2*p);
  re2 = TM(pm2*1);
  ze2 = TM(npar*pm2);
  rg2 = TM(pm2*p);
  r2r = TM(p*pm2);
  rgr2 = TM(pm2*pm2);

  for ( z=0; z<npar; z++ ) if ( !ZAM2(t=0,z) ) {
    for ( j=0, k=PAR(z,0), s=0; s<p; s++ ) CE1(s,j) = YTB1(s,k);
    for ( s=0; s<p; s++ ) pcel[s] = s;
    for ( j=0, s=j+1; s<p; s++ ) if ( !CE1(j,j) && CE1(s,j) ) {
      pcel[j]=s; pcel[s]=j;
      a1 = CE1(j,j); CE1(j,j) = CE1(s,j); CE1(s,j) = a1;
    }

    for ( k=0, s=0; s<pm1; s++ )
      for ( j=0; j<pm1; j++ ) MG(s,j) = ( s == j ? -CE1(k,k) : 0 );
    for ( k=0, s=0; s<pm1; s++ ) {
      for ( j=0; j<1; j++ ) RTT(s,j) = CE1(s+1,j);
      for ( j=1; j<p; j++ ) RTT(s,j) = MG(s,j-1);
    }
    for ( s=0; s<pm1; s++ ) for ( j=0; j<p; j++ ) RT(s,j) = RTT(s,pcel[j]);

    for ( i=PAR(z,1), a1=0, j=0, s=0; s<pm1; s++ ) {
      for ( t=0; t<p; t++ ) {
        a1 += RT(s,t) * YTB1(t,i);
      }
      CE2(s,j) = a1; a1=0;
    }
  }
}

```



```

    }
for ( s=0; s<pm1; s++ ) pce2[s] = s;

for ( j=0, s=j+1; s<pm1; s++ ) if ( ! CE2(j,j) && CE2(s,j) ) {
    pce2[j] = s ;
    pce2[s] = j ;
    a1 = CE2(j,j); CE2(j,j) = CE2(s,j); CE2(s,j) = a1;
}

for ( t=0, s=0; s<pm2; s++ )
    for ( j=0; j<pm2; j++ ) MG2(s,j) = ( s == j ? -CE2(t,t) : 0 );

for ( k=0, s=0; s<pm2; s++ ) {
    for ( j=0; j<1; j++ ) RTT2(s,j) = CE2(s+1,j);
    for ( j=1; j<pm1; j++ ) RTT2(s,j) = MG2(s,j-1);
}
for ( s=0; s<pm2; s++ ) for ( j=0; j<pm1; j++ ) RT2(s,j) = RTT2(s,pce2[j]);

for ( a1=0, s=0; s<pm2; s++ ) for ( t=0; t<p; t++ ) {
    for ( j=0; j<pm1; j++ ) {
        a1 += RT2(s,j) * RT(j,t);
    }
    R2T(s,t) = a1; a1=0;
}

for ( a1=0, s=0; s<pm2; s++ ) for ( t=0; t<1; t++ ) {
    for ( j=0; j<p; j++ ) {
        a1 += R2T(s,j) * YTX(j,t);
    }
    RE2(s,t) = a1; a1 = 0;
}

for ( a1=0, s=0; s<pm2; s++ ) for ( t=0; t<p; t++ ) {
    for ( j=0; j<p; j++ ) {
        a1 += R2T(s,j) * GI(j,t);
    }
    RG2(s,t) = a1; a1 = 0;
}

for ( i=0; i<pm2; i++ ) for ( j=0; j<p; j++ ) R2R(j,i) = R2T(i,j);

for ( a1=0, s=0; s<pm2; s++ ) for ( t=0; t<pm2; t++ ) {
    for ( j=0; j<p; j++ ) {
        a1 += RG2(s,j) * R2R(j,t);
    }
    RGR2(s,t) = a1; a1 = 0;
}

for ( t=0, s=0; s<pm2; s++ ) ZE2(z,s) = RE2(s,t) / sqrt( RGR2(s,s));
}
else for ( t=0; t<pm2; t++ ) ZE2(z,t) = 10;

```

```

vze2 = TMI(pm2);
for ( z=0; z<npar; z++ ) {
    for ( i=0; i<pm2; i++ ) vze2[i] = i;

    for ( t=0; t<pm2- 1; t++ ) for ( j=t+1; j<pm2; j++ )
        if ( fabs( fabs( ZEM2(z,t) ) - fabs( ZEM2(z,j))) <= 0.01 )
            if( vze2[j] == j ) vze2[j] = t;

    for ( count2=0, i=0; i<pm2; i++) if ( vze2[i] == i ) count2++;

    for ( nmm2=0, t=0, i=0; i<pm2; i++ )
        if( fabs(ZEM2(z,i)) > Zc(0,count2-1) ) nmm2++;

    ZEM2(t,z) = nmm2;
}

printf("%d", fprintf(fp, "\nZEM2:"));
for ( t=0, z=0; z<npar; z++ ) printf("%d", fprintf(fp, "%3d", ZEM2(t,z)));
printf("%d", fprintf(fp, "\n"));

for ( nparf=0, i=0; i<npar; i++ ) for ( t=0, k=0; k<q; k++ )
    if ( pvza[k] != k && !ZEM2(t,i) )
        if ( pvza[k] == PAR(i,0) || pvza[k] == PAR(i,1) ) nparf++;

parf=TMI(nparf* 2);
for ( j=0, i=0; i<npar; i++ ) for ( t=0, k=0; k<q; k++ )
    if ( pvza[k] != k && !ZEM2(t,i) ) {
        if ( pvza[k] == PAR(i,0) ) {
            PARF(j,0) = k; PARF(j,1) = PAR(i,1); j++;
        }

        if ( pvza[k] == PAR(i,1) ) {
            PARF(j,0) = PAR(i,0); PARF(j,1) = k; j++;
        }
    }

for ( t=0, nparb=0, i=0; i<npar; i++ )
    if ( !ZEM2(t,i) ) nparb++;
npart = nparb+nparf;
for ( t=0, nparb=0, i=0; i<npar; i++ ) if ( !ZEM2(t,i) ) {
    PART(nparb,0) = PAR(i,0); PART(nparb,1) = PAR(i,1); nparb++;
}
for ( i=nparb; i<npart; i++ ) {
    PART(i,0) = PARF(i-nparb,0); PART(i,1) = PARF(i-nparb,1);
}
return npart;
}

```

```

void Valcorr1(p,q,pps,yb,st,mml,ib,yi,mle,yx,ajl,x,xl,zem1,pvza)
int p,q;int *zem1, *pvza;

```

```

float *pps,*yb,*st,*mml,*ib,*yi,*mle,*yx,*ajl,*x,*x1;
{
  register int s,t,j,k;
  float a1;

  for ( a1=0, s=0; s<q; s++ ) for ( t=0; t<p; t++ ) {
    for(j=0;j<q;j++) {
      a1 += S(s,j)* YTB1(t,j);
    }
    ST(s,t) = a1; a1 = 0;
  }

  for ( t=0, k=0; k<q; k++ ) if ( !ZEM1(t,k)|| !ZEM1(t,pvza[k]) ) {
    for ( a1=0, t=0; t<p; t++ ) for ( s=0; s<p; s++ ) {
      a1 = IB(t,k) * YI(k,s); M1(t,s) = a1; a1 = 0;
    }
    for ( a1=0, s=0; s<p; s++ ) for ( t=0; t<1; t++ ) {
      for ( j=0; j<p; j++ ) {
        a1 += M1(s,j) * YTX(j,t);
      }
      M1E(s,t) = a1; a1 = 0;
    }

    for ( a1=0, s=0; s<q; s++ ) for ( t=0; t<1; t++ ) {
      for ( j=0; j<p; j++ ) {
        a1 += ST(s,j) * M1E(j,t);
      }
      AJ1(s,k) = a1; a1 = 0;
    }

    for ( t=0, s=0; s<q; s++ )
      if ( s != k) X1(s,k) = X(s,t) + AJ1(s,k);
      else X1(s,k) = 0;
  }

  else for ( s=0; s<q; s++ ) X1(s,k) = 0;
}

void Valcorr2(p,q,npart,pps,st,mml,ib,yi,mle,yx,ajl,x,x12,part,ui,yb,ny,zal)
int p,q,npart, *part;
float *pps,*st,*mml,*ib,*yi,*mle,*yx,*ajl;
float *x,*x12,*ui,*yb,*ny,*zal;
{
  register int s,t,j,k,z,i;
  float a1, a2, *sny;

  sny = TM(q*npart);
  for ( a1=0, s=0; s<q; s++ ) for ( t=0; t<p; t++ ) { for ( j=0; j<q; j++ ) {
    a1 += S(s,j) * YTB1(t,j);
  }
}

```

```

ST(s,t) = a1;  a1 = 0;
}

for ( t=0, z=0; z<npart; z++ ) {
  k = PART(z,0);  i = PART(z,1);
  {
    for ( a1=0, t=0; t<p; t++ ) for ( s=0; s<p; s++ ) {
      a1 = IB(t,k) * YI(k,s);
      M1(t,s) = a1;  a1 = 0;
    }

    for ( a1=0, s=0; s<p; s++ ) for ( t=0; t<1; t++ ) {
      for ( j=0; j<p; j++ ) {
        a1 += M1(s,j) * YTX(j,t);
      }
      M1E(s,t) = a1;  a1=0;
    }

    for ( a1=0, s=0; s<q; s++ ) for ( t=0; t<1; t++ ) {
      for ( j=0; j<p; j++ ) {
        a1 += ST(s,j) * M1E(j,t);
      }
      AJ1(s,z) = a1;  a1 = 0;
    }

    for ( t=0, s=0; s<q; s++ )
      if ( s != k ) X12(s,z) = X(s,t) + AJ1(s,z);
      else X12(s,z) = 0;

    for ( s=0; s<p; s++ )
      for ( t=0; t<p; t++ ) M1(s,t) = I(s,t) - M1(s,t);

    for ( a1=0, s=0; s<p; s++ ) {
      for ( t=0; t<p; t++ ) {
        a1 += M1(s,t) * YTB1(t,i);
      }
      NY(s,z)=a1;  a1 = 0;
    }

    for ( a1=0, s=0; s<q; s++ ) {
      for ( t=0; t<p; t++ ) {
        a1 += ST(s,t) * NY(t,z);
      }
      SNY(s,z) = a1;      a1 = 0;
    }

    for ( a1=0, s=0; s<p; s++ ) {
      a1 += YTB1(s,i) * NY(s,z);
    }
    a2=a1;

    for ( s=0; s<q; s++ )

```

```

if ( s != i && s != k )
    X12(s,z) = X12(s,z) - SNY(s,z) * ZA1(k,i) / sqrt(a2);
else X12(s,z) = 0;
}
}
}

```

```

void vnom(int c,int rq,int ni,int *viv,char *cm,char *cnm)

```

```

{
char *compo;
char *numcor1;
char *numcor2;
char *alc;
int c1 = c+1, nic1 = ni*c1;
register int i,j,ki,kj;

compo = TMC(MCOM+1);
numcor1 = TMC(MCOR);
numcor2 = TMC(MCOR);
alc = TMC(MCOR*(MCOM+1)*3);

for ( i=0; i<c; i++ ) compo[i] = 'A'+i;
for ( i=c; i<c1; i++ ) compo[i] = 'T';

for ( i=0; i<9; i++ ) {
    numcor1[i] = '-'; numcor2[i] = '1'+i;
}

for ( j=0, i=9; i<ni && j<10; i++,j++ ) {
    numcor1[i] = '1'; numcor2[i] = '0'+j;
}

if ( ni == 20 ) { numcor1[ni-1] = '2';
    numcor2[ni-1] = '0';
}

for ( i=0; i<ni; i++ ) for ( j=0; j<c1; j++ ) {
    ALC(j+i*c1,0) = compo[j]; ALC(j+i*c1,1) = numcor1[i];
    ALC(j+i*c1,2) = numcor2[i];
}

for ( i=nic1; i<nic1+rq; i++ ) {
    ALC(i,0)='R'; ALC(i,1)='-'; ALC(i,2)='1'+(i-nic1);
}

for ( ki=0, kj=0, i=0; i<nic1+rq; i++ ) if ( viv[i] == 1 ) {
    for(j=0;j<3;j++) CM(ki,j) = ALC(i,j); ki++;
}
else if ( viv[i] == 3 ) {
    for ( j=0; j<3; j++ ) CNM(kj,j) = ALC(i,j); kj++;
}
}

```

```

    }
}

void variancia (int ni,int c,int q,float *varct, float *mfm, float *pps)
{
    int nctm, c1 = c+1 ;
    float *mb, *sct,*rns;
    float *varm, a1;
    register int i,j,t,k;

    for ( nctm=0, i=0; i<ni; i++ ) if ( varct[i] != 0 ) nctm++;

    mb = TM(q*nctm);
    for ( i=0; i<q; i++ ) for ( j=0; j<nctm; j++ ) MB(i,j) = 0 ;

    for ( t=0, k=0, j=0; j<ni; j++ ) if ( varct[j] != 0 ) {
        for ( i=0; i<c1; i++ ) if ( MFM(j,i) != 0 && MFM(j,i) != 2 ) {

            MB(k,t) = MFM(j,i); k++;
        }
        t++;
    }

    varm = TM(nctm);
    for ( t=0, i=0; i<ni; i++ ) if ( varct[i] != 0 ) {
        varm[t] = varct[i]; t++;
    }

    sct = TM(nctm*nctm);
    for ( i=0; i<nctm; i++ ) for ( j=0; j<nctm; j++ )
        SCT(i,j) = ( i == j ? varm[i] : 0 );

    ms=TM(q*nctm);
    for ( a1=0, i=0; i<q; i++ ) for ( j=0; j<nctm; j++ ) {
        for ( t=0; t<nctm; t++ ) {
            a1 += MB(i,t) * SCT(t,j);
        }
        MS(i,j) = a1; a1 = 0;
    }

    for ( a1=0, i=0; i<q; i++ ) for ( t=0; t<q; t++ ) {
        for ( j=0; j<nctm; j++ ) {
            a1 += MS(i,j) * MB(t,j);
        }
        if ( i == t ) S(i,t) = a1;
        else S(i,t) = a1/2;
        a1 = 0;
    }
}

```

```

void Val_obs2(m,n,q,npart,part,pp,b1,x12,cm,cnm,cnm2r,vo2,fp)
int m,n,q,npart,*part;
char *cm,*cnm,*cnm2r;
float *pp,*b1,*x12,*vo2;
FILE *fp;
{
float a1;
float *p2r,*p2rl,*p2r1,*u2,*b12rm,*b2x;
int nm2=n+2, m1=m-1, r2;
int *vip,*vic,*vir;
float *vnob2,*p2r3,*up2r3,*vi2,*vno2;
char *comptroc;
double fabs( );
register int i,j,z,k,t;

comptroc = TMC(3);
p2r = TM(m*nm2);
p2rl = TM(m*nm2);
vip = TMI(m);
vic = TMI(nm2);
b12rm = TM(m*q);
vnob2 = TM(nm2);

printf("%d",fprintf(fp,"\n\nQuantidades Nao-Medidas Apos 2 Remocoes\n"));

for ( z=0; z<npart; z++ ) { for ( i=0; i<m; i++ ) {
for ( j=0; j<n; j++ ) P2RL(i,j) = P2R(i,j) = P(i,j);
for ( j=n; j<n+1; j++ ) P2RL(i,j) = P2R(i,j) = B1(i,PART(z,0));
for ( j=n+1; j<nrm2; j++ ) P2RL(i,j) = P2R(i,j) = B1(i,PART(z,1));
}

for ( i=0; i<m; i++ ) for ( j=0; j<q; j++ ) B12Rm(i,j) = B1(i,j);

for ( i=0; i<m; i++ ) vip[i] = i;
for ( i=0; i<nm2; i++ ) {
vic[i] = i; vnob2[i] = 0;
}

r2=triang(m,nm2,p2rl,p2r,vip,vic);

vir = TMI(r2);
for ( i=0; i<r2; i++ ) vir[i] = i;

p2r1 = TM(r2*r2);
for ( i=0; i<r2; i++ ) for ( j=0; j<r2; j++ ) P2R1(i,j) = P2R(i,j);

u2 = TM(r2*r2);
for ( i=0; i<r2; i++ ) for ( j=0; j<r2; j++ ) U2(i,j) = ( i == j ? 1 : 0 );

lu(r2,vir,p2r1);
backs(r2,r2,vir,p2r1,u2);

```

```

free(vir); free(p2r-1);

for ( i=0; i<n; i++ ) for ( j=0; j<3; j++ ) CNM2R(i,j) = CNM(i,j);
for ( i=n; j=0; j<3; j++ ) CNM2R(i,j) = CM(PART(z,0),j);
for ( i=n+1; j=0; j<3; j++ ) CNM2R(i,j) = CM(PART(z,1),j);

if (r2 == nm2) {
    printf("%d", r2); printf(fp, "\nTodos os v n m sao observaveis\n");
    printf("%d", r2); printf(fp, "\nMedidas removidas: Par numero %d\n", z+1));
    for ( i=0; i<nm2; i++ ) {
        for ( j=0; j<3; j++ ) printf("%d", fprintf(fp, "%c", CNM2R(i,j)));
        printf("%d", fprintf(fp, " "));
    }
    printf("%d", fprintf(fp, "\n"));
}

else {
    p2r3 = TM(r2*(nm2-r2));
    for ( i=0; i<r2; i++ ) for ( j=r2; j<nm2; j++ ) P2R3(i,j-r2) = P2R(i,j);

    up2r3 = TM(r2*(nm2-r2));
    for ( a1=0, i=0; i<r2; i++ ) for ( j=0; j<nm2-r2; j++ ) {
        for ( k=0; k<r2; k++ ) {
            a1 += U2(i,k) * P2R3(k,j);
        }
        UP2R3(i,j) = a1; a1 = 0;
    }
    free(p2r3);

    vi2 = TM((nm2-r2)*(nm2-r2));
    for ( i=0; i<nm2-r2; i++ )
        for ( j=0; j<nm2-r2; j++ ) VI2(i,j) = ( i == j ? 1 : 0 );

    vno2 = TM(nm2*(nm2-r2));
    for ( i=0; i<r2; i++ ) for ( j=0; j<nm2-r2; j++ ) VNO2(i,j) = UP2R3(i,j);
    for ( i=r2; i<nm2; i++ ) for ( j=0; j<nm2-r2; j++ ) VNO2(i,j) = VI2(i-r2,j);

    free(up2r3); free(vi2);

    for ( j=0; j<nm2-1; j++ ) if ( j < vic[j] )
        for ( i=0; i<3; i++ ) {
            comptroc[i] = CNM2R(j,i); CNM2R(j,i) = CNM2R(vic[j],i);
            CNM2R(vic[j],i) = comptroc[i];
        }

    printf("%d", fprintf(fp, "\nMedidas removidas apos troca: Par numero %3d\n", z+1));
    for ( i=0; i<nm2; i++ ) {
        for ( j=0; j<3; j++ ) printf("%d", fprintf(fp, "%c", CNM2R(i,j)));
        printf("%d", fprintf(fp, " "));
    }
}

```



```

for ( a1=0, i=0; i<rm2; i++ ) {
  for ( j=0; j<nm2-r2; j++ ) {
    a1 += fabs(√VNO2(i,j));
  }
  vno2[i] = a1; a1 = 0;
}
free(vno2);

}

for ( vip[m1]=m1, i=0; i<m1; i++ ) for ( j=0; j<q; j++ ) {
  a1 = B12Rm(i,j); B12Rm(i,j) = B12Rm(vip[i],j);
  B12Rm(vip[i],j) = a1;
}

b2x = TM(r2);
for ( a1=0, i=0; i<r2; i++ ) {
  for ( t=0, j=0; j<q; j++ ) { a1 += B12Rm(i,j) * X12(j,z);
  }
  B2X(i,t) = -a1; a1 = 0;
}
for ( a1=0, i=0; i<r2; i++ ) if ( vno2[i] == 0 )
  for ( k=0; k<1; k++ ) { for ( j=0; j<r2; j++ ) {
    a1 += U2(i,j) * B2X(j,k);
  }
  VO2(i,z) = a1; a1 = 0;
}
else VO2(i,z) = 0;

for ( i=r2; i<nm2; i++ ) VO2(i,z) = 0;
free(u2); free(b2x);

printf("%d", fprintf(fp, "\nValores observaveis apos duas remocoes\n"));
for ( i=0; i<n+2; i++ )
  printf("%d", fprintf(fp, "%8.2f", VO2(i,z)));
printf("%d", fprintf(fp, "\n"));

}
}

void Val_obs1(m,n,q,zem1,pvza,pp,b1,x1,cm,cnm,cnm1r,vol,fp)
int m,n,q,*zem1,*pvza;
float *pp,*b1,*x1,*vol;
char *cm,*cnm,*cnm1r;
FILE *fp;
{
float *plr,*plr1,*plr1,*ul,*b11rm;
float *blx,*plr3,*uplr3,*vil,*vno1;
float *vno1, a1;
int nml = n+1, r1, m1 = m-1;

```

```

int *vip1,*vic1,*vir1;
char *comptroc;
double fabs( );
register int i,j,k,t;

comptroc = TMC(3);
plr = TM(m*nm1);
plr1 = TM(m*nm1);
vip1 = TMI(m);
vic1 = TMI(nm1);
b11rm = TM(m*q);
vnobl = TM(nm1);

for ( t=0, k=0; k<q; k++ ) { t = 0;
  if ( !ZEM1(t,k) || !ZEM1(t,pvza[k]) ) {
    for ( i=0; i<m; i++ ) {
      for ( j=0; j<n; j++ ) P1RL(i,j) = P1R(i,j) = P(i,j);
      for ( j=n; j<nm1; j++ ) P1RL(i,j) = P1R(i,j) = B1(i,k);
    }

for ( i=0; i<m; i++ ) for ( j=0; j<q; j++ ) B11Rm(i,j) = B1(i,j);

for ( i=0; i<m; i++ ) vip1[i] = i;
for ( i=0; i<nm1; i++ ) {
  vic1[i] = i; vnobl[i] = 0;
}

r1=triang(m,nm1,plr1,plr,vip1,vic1);

vir1 = TMI(r1);
for ( i=0; i<r1; i++ ) vir1[i] = i;

plr1 = TM(r1*r1);
for ( i=0; i<r1; i++ ) for ( j=0; j<r1; j++ ) P1R1(i,j) = P1R(i,j);

u1 = TM(r1*r1);
for ( i=0; i<r1; i++ ) for ( j=0; j<r1; j++ ) U1(i,j) = ( i==j ? 1 : 0 );

lu(r1,vir1,plr1);
backs(r1,r1,vir1,plr1,u1);
free(vir1); free(plr1);
for ( i=0; i<n; i++ ) for ( j=0; j<3; j++ ) CNM1R(i,j) = CNM(i,j);

for ( i=n, j=0; j<3; j++ ) CNM1R(i,j) = CM(k,j);

if(r1 == nm1) printf("%d", fprintf(fp, "\nTodos os v n m sao observaveis\n"));
else {
  plr3 = TM(r1*(nm1-r1));
  for ( i=0; i<r1; i++ ) for ( j=r1; j<nm1; j++ ) P1R3(i,j-r1)=P1R(i,j);
}

```

```

up1r3 = TM(r1*(nm1-r1));
for ( a1=0, i=0; i<r1; i++ ) for ( j=0; j<nm1-r1; j++ ) {
    for ( t=0; t<r1; t++ ) {
        a1 += U1(i,t) * P1R3(t,j);
    }
    UP1R3(i,j) = a1; a1 = 0;
}
free(p1r3);

vi1 = TM((nm1-r1)*(nm1-r1));
for ( i=0; i<nm1-r1; i++ )
    for ( j=0; j<nm1-r1; j++ ) VI1(i,j) = ( i==j ? 1 : 0 );

vno1=TM(nm1*(nm1-r1));
for ( i=0; i<r1; i++ ) for ( j=0; j<nm1-r1; j++ ) VNO1(i,j) = UP1R3(i,j);
for ( i=r1; i<nm1; i++ ) for ( j=0; j<nm1-r1; j++ ) VNO1(i,j) = VI1(i-r1,j);

free(up1r3); free(vi1);

for ( j=0; j<nm1-1; j++ ) if ( j < vic1[j] )
    for ( i=0; i<3; i++ ) {
        comptroc[i] = CNM1R(j,i); CNM1R(j,i) = CNM1R(vic1[j],i);
        CNM1R(vic1[j],i) = comptroc[i];
    }

for ( a1=0, i=0; i<nm1; i++ ) { for ( j=0; j<nm1-r1; j++ ) {
    a1 += fabs(VNO1(i,j));
}
    vnob1[i] = a1; a1 = 0;
}
free(vno1);

}

for ( vip1[m1]=m1, i=0; i<m1; i++ ) for ( j=0; j<q; j++ ) {
    a1 = B11Rm(i,j); B11Rm(i,j) = B11Rm(vip1[i],j);
    B11Rm(vip1[i],j) = a1;
}

b1x = TM(r1);
for ( a1=0, i=0; i<r1; i++ ) {
    for ( t=0, j=0; j<q; j++ ) {
        a1 += B11Rm(i,j) * X1(j,k);
    }
    B1X(i,t) = -a1; a1 = 0;
}

for ( a1=0, i=0; i<r1; i++ ) if ( vnob1[i] == 0 )
    for ( t=0; t<1; t++ ) {
        for ( j=0; j<r1; j++ ) {
            a1 += U1(i,j) * B1X(j,t);

```

```
    }  
    VO1(i,k) = a1; a1 = 0;  
  }  
  else VO1(i,k) = 0;  
  
  for ( i=r1; i<nm1; i++) VO1(i,k) = 0;  
  free(u1); free(b1);  
  }  
  else for ( i=0; i<nm1; i++) VO1(i,k) = 0;  
  }  
}
```

## APÊNDICE E. LISTAGEM DO PROGRAMA TCLASS

O Programa TCLASS, conforme listado abaixo, permite a entrada de dados pelo teclado ou através de arquivo de dados. Os resultados são gravados em arquivo de dados cujo nome é informado no início.

```
/* Programa TCLASS - Classificacao de Variaveis de Processo */
```

```
#include <stdio.h>
#include <alloc.h>
#include <conio.h>
#include <math.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>

#define M(i,j) (*(fd+2 *(i)+j))
#define A(i,j) *(a+mi *(i)+j)
#define AC(i,j) *(ac+nic1 *(i)+j)
#define TE(i,j) *(te+c2 *(i)+j)
#define ME(i,j) *(me+mic1 *(i)+j)
#define ACC(i,j) *(acc+(nic1+rq)*(i)+j)
#define B1(i,j) *(b1+q*(i)+j)
#define B2(i,j) *(b2+nb2*(i)+j)
#define P(i,j) *(pp+n*(i)+j)
#define BF(i,j) *(bf+nb2*(i)+j)
#define C(i,j) *(cc+nc*(i)+j)
#define PL(i,j) *(p1+n*(i)+j)
#define P1(i,j) *(p1+r*(i)+j)
#define P2(i,j) *(p2+r*(i)+j)
#define U(i,j) *(u+r*(i)+j)
#define P2U(i,j) *(pu+r*(i)+j)
#define UP(i,j) *(up+p*(i)+j)
#define YT(i,j) *(y+m*(i)+j)
#define H(i,j) *(h+m*(i)+j)
#define YHT(i,j) *(yh+m*(i)+j)
#define D(i,j) *(d+nc*(i)+j)
#define P3(i,j) *(p3+(n-r)*(i)+j)
#define DL(i,j) *(d1+nc*(i)+j)
#define UP3(i,j) *(up3+(n-r)*(i)+j)
#define VI(i,j) *(vi+(n-r)*(i)+j)
#define VCN(i,j) *(vcn+(n-r)*(i)+j)
#define BCI(i,j) *(bci+(nic1+rq)*(i)+j)
#define ACB(i,j) *(acb+(nic1+rq)*(i)+j)
#define D1(i,j) *(d1+rd *(i)+j)
#define UD(i,j) *(ud+rd *(i)+j)
#define D3(i,j) *(d3+(nc-rd) *(i)+j)
#define UD3(i,j) *(ud3+(nc-rd) *(i)+j)
#define VIW(i,j) *(viw+(nc-rd) *(i)+j)
#define W(i,j) *(w+(nc-rd) *(i)+j)
#define C1(i,j) *(pc1+nc *(i)+j)
#define C1W(i,j) *(c1w+(nc-rd) *(i)+j)
```

```

#define P3T3(i,j) (*(p3t3+(nc-rd) *(i)+j))
#define T3(i,j) *(t3+(nc-rd) *(i)+j)
#define T(i,j) *(t+(nc-rd) *(i)+j)
#define CL(i,j) *(c1+nc *(i)+j)
#define D2(i,j) *(d2+rd *(i)+j)
#define D2U(i,j) *(du+rd *(i)+j)
#define UZ(i,j) *(uz+pmrd *(i)+j)
#define ZT(i,j) *(zt+p *(i)+j)
#define HZ(i,j) *(hz+p *(i)+j)
#define ZHT(i,j) *(zh+p *(i)+j)
#define ZY(i,j) *(zy+m *(i)+j)
#define ZB1(i,j) *(zb1+q *(i)+j)
#define ZB2(i,j) *(zb2+nb2 *(i)+j)
#define ZBF(i,j) *(zbf+q *(i)+j)
#define ZTO(i,j) *(zto+nct *(i)+j)
#define HC(i,j) *(hc+nc *(i)+j)
#define ALC(i,j) *(alc+3*(i)+j)
#define CM(i,j) *(cm+3 *(i)+j)
#define CNM2(i,j) *(cnm2+3 *(i)+j)
#define CNM3(i,j) *(cnm3+3 *(i)+j)

#define TMI(size) (int *)malloc(size*sizeof(int))
#define TM(size) (float *)malloc(size*sizeof(float))
#define TMC(size) (char *)malloc(size*sizeof(char))

```

```

#define MNO 10
#define MCOR 20
#define MCOM 10
#define MREA 5

```

```

int ni; /* correntes globais */
int mi; /* num de nos */
int c; /* num especies quimicas */
int rq; /* num de reacoes quimicas */
int nc; /* num cor globais cat 2 */
int nct; /* num cor globais cat 1 */
int nb2; /* num de so fracoas medidas */
int q; /* num correntes medidas */

```

```

int M[MCOR][2];
float TE[MREA][MCOM+2];
float fra[MCOR*(MCOM+1)]; /* lista so fracoas medidas */
int vfra[MCOR]; /* numero de so fracoas medidas em cada corrente */
float vaz[MCOR*(MCOM+1)]; /* lista vazoes medidas */
int vvaz[MCOR]; /* num de vazoes Cat 1 em cada corrente */
int viv[MCOR*(MCOM+1)+MREA];
int vie[MNO*(MCOM+1)];

```

```

void entrar(void), mostrar(void), gravar(void),ler(void);
void classificar(void);
char menu(void);

```

```

main( )
{
    char ch;
    clrscr( );

    printf("\nPROGRAMA T C L A S S\n");
    printf("\nCLASSIFICACAO DE VARIAVEIS DE PROCESSO\n");
    printf("\nElaborado por: TALITA FURLANETTO MENDES\n");
    printf("\nOrientador: Joao Alexandre.F.R. Pereira\n");
    printf("\n\n\n\n\n\n\n\n\n\t\tpara continuar digite qualquer letra\n");
    ch=getche( );

    clrscr( );
    printf("\nEscolha uma opcao\n");
    for(;;){
        ch=menu( );
        switch(ch) {
            case 'E' : entrar( );
                break;
            case 'M' : mostrar( );
                break;
            case 'G' : gravar( );
                break;
            case 'L' : ler( );
                break;
            case 'C' : classificar( );
                break;
            case 'S' : exit(0);
                }
        }
    }

void classificar(void)
{
    void backs (int r, int rs, int *ir, float *p1, float *u );
    void prod_m(int s, int r,int w,float *p2,float *u,float *pu);
    void vnom(int c,int rq,int ni,int nic1,int *viv, char *cm,char *cnm2,char *cnm3);
    int triang(int m,int n,float *pl,float *pp,int *ip,int *ic);
    int lu(int r,int *ir,float *p1);

    int c1,c2,nic1, mic 1;
    int m, m1, n, *fd;
    float *a, *ac, *te, *me, *acc;
    float *bci, *acb, *b1,*b2,*pp, *bf, *cc;
    int r,*ip, *ic, *ir, p;
    float *pl, *p1,*p2, *u,*pu;
    float a1, *up,*y,*h,*yh, *d, *p3, *dl;
    int *ipd,*icd, rd, *ird;
    float *up3,*vi,*vcn, *d1, *ud, *d3;
    float *ud3, *viw, *w, *pcl;

```



```

float *c1w, *p3t3, *t3, *t, *cl;
float *d2, *du, *uz, *zt, *hz, *zh, *zb1, *zb2;
int pmrd, *vvalmed1, *vvalnm2, *vvalnm2e, *vvalnm3, *vvalmed2;
float *zto, *zbf, *zy;
char *cm, *cnm2, *cnm3, *comptroc;
int *hc, ncl, ali, *vvalnm2t;
int i,j,k,ki,kj,kt;

```

```

FILE *fp;
char s[20];
printf("\nNome do Arquivo dos Resultados:");
gets(s);
if ( ( fp=fopen(s,"w") ) == NULL ) {
printf("Nao posso abrir arquivo\n");
return;
}

```

```

printf("%d", fprintf(fp,"RESULTADOS DA CLASSIFICACAO DE VARIAVEIS DE
PROCESSO\n"));

```

```

c1 = c+1;
nic1 = ni*c1;
mic1 = mi*c1;
c2 = c+2;
comptroc = TMC(3);
clrscr();
fd = TMI(ni*2);
a = TM(ni*mi);
for ( i=0 ; i<ni ; i++ ) for ( j=0 ; j<2 ; j++ ) M(i,j) = M[i][j];

```

```

for ( i=0 ; i<ni ; i++ ) for ( j=0 ; j<mi ; j++ ) {
A(i,j) = ( M(i,0) == j+1 ? 1 : 0 );
if ( A(i,j) == 0 ) A(i,j) = ( M(i,1) == j+1 ? -1 : 0 );
}

```

```

ac = TM(mic1*nic1);
for ( i=0 ; i<mi ; i++ ) for ( ki=0 ; ki<c1 ; ki++ )
for ( j=0 ; j<ni ; j++ ) for ( kj=0 ; kj<c1 ; kj++ )
AC( (c1*i+ki),(c1*j+kj) ) = ( ki == kj ? A(j,i) : 0 );

```

```

te = TM(rq*c2);
me = TM(rq*mic1);
for ( i=0 ; i<rq ; i++ ) for ( j=0 ; j<c2 ; j++ ) TE(i,j) = TE[i][j];

```

```

for ( i=0 ; i<rq ; i++ ) for ( j=0 ; j<mi ; j++ )
for ( kj=0 ; kj<c1 ; kj++ )
ME(i,c1*j+kj) = ( c1*j+kj == (TE(i,0)-1)*c1+kj ? TE(i,kj+1) : 0 );

```

```

acc = TM( mic1 * (nic1+rq) );
for ( i=0 ; i<mic1 ; i++ ) for ( j=0 ; j<nic1 ; j++ ) ACC(i,j) = AC(i,j);

```

```

for ( i=0 ; i<mic1 ; i++ )

```

```

for ( j=nic1 ; j<nic1+rq ; j++ ) ACC(i,j) = ME(j-nic1,i);

for ( m=0, i=0 ; i<ni ; i++ ) if ( vie[i] == 1 ) m++;

for ( ki=0, i=0 ; i<ni ; i++ ) if ( vie[i] == 1 ) {
    for ( j=0 ; j<nic1+rq ; j++ ) ACC(ki,j) = ACC(i,j); ki++;
}

bci = TM( ni * (nic1+rq) );
for ( i=0 ; i<ni ; i++ ) { for ( j=0 ; j<nic1+rq ; j++ )
    BCI(i,j) = ( j >= (c1*i) && j < (c1*i)+c1 ? 1 : 0 );
}

for ( i=0 ; i<ni ; i++ ) { for ( j=0 ; j<nic1+rq ; j++ )
    if ( BCI(i,j) && j == i*c1+(c1-1) ) BCI(i,j) = -1;
}

acb = TM((m+ni)* (nic1+rq));
for ( i=0 ; i<m ; i++ ) for ( j=0 ; j<nic1+rq ; j++ ) ACB(i,j) = ACC(i,j);
for ( i=m ; i<m+ni ; i++ ) for ( j=0 ; j<nic1+rq ; j++ ) ACB(i,j) = BCI(i-m,j);

for ( n=0, i=0 ; i<nic1+rq ; i++ ) { if ( viv[i] == 3 ) n++;
}

m = m+ni;
b1 = TM(m*q);
b2 = TM(m*nb2);
pp = TM(m*n);
for ( kj=0, ki=0, kt=0, j=0 ; j<nic1+rq ; j++ ) {
    if ( viv[j] == 1 ) {
        for ( i=0 ; i<m ; i++ ) B1(i,kj) = ACB(i,j); kj++;
    }
    if ( viv[j] == 2 ) {
        for ( i=0 ; i<m ; i++ ) B2(i,ki) = ACB(i,j); ki++;
    }
    if ( viv[j] == 3 ) {
        for ( i=0 ; i<m ; i++ ) P(i,kt) = ACB(i,j); kt++;
    }
}

cm = TMC(q*3);
cnm2 = TMC(nb2*3);
cnm3 = TMC(n*3);

vnom(c,rq,ni,nic1,viv,cm,cnm2,cnm3);

if ( nb2 ) {

cc = TM(m*nc);
bf = TM(m*nb2);
for ( i=0 ; i<m ; i++ ) for ( j=0 ; j<nc ; j++ ) C(i,j) = 0;
}

```

```

for ( j=0; j<nb2 ; j++ ) for ( i=0 ; i<m ; i++ ) {
    if ( fabs ( B2(i,j) ) == 1 ) BF(i,j) = B2(i,j) * fra[j];
    else BF(i,j) = B2(i,j);
}

for ( kj=0, j=0 ; j<nb2 ; j++ ) {
    for ( k=0; k<vfra[kj] ; k++ ) {
        for ( i=0; i<m; i++ ) C(i,kj) += BF(i,j+k) ;
    }
    kj++;
    j += k-1;
}

if ( n ) {

pl = TM(m*n);
ic = TMI(n);
ip = TMI(m);

for ( i=0 ; i<m ; i++ ) for ( j=0 ; j<n ; j++ ) PL(i,j) = P(i,j);

for ( i=0 ; i<m ; i++ ) ip[i] = i;
for ( i=0 ; i<n ; i++ ) ic[i] = i;

r=triang(m,n,pl,pp,ip,ic);

p = m-r;
p1 = TM(r*r);
for ( i=0 ; i<r ; i++ ) for ( j=0 ; j<r ; j++ ) P1(i,j) = P(i,j);

ir = TMI(r);
u = TM(r*r);
for ( i=0 ; i<r ; i++ ) ir[i] = i;

for ( i=0 ; i<r ; i++ )
    for ( j=0 ; j<r ; j++ ) U(i,j) = ( i == j ? 1 : 0 );

lu(r,ir,p1);
backs(r,r,ir,p1,u);

p3 = TM(r*(n-r));
for ( i=0 ; i<r ; i++ ) for ( j=r ; j<n ; j++ ) P3(i,j-r) = P(i,j);

p2 = TM(p*r);
for ( i=0 ; i<p ; i++ ) for ( j=0 ; j<r ; j++ ) P2(i,j) = - P(i+r,j);

pu=TM(p*r);
prod_m(p,r,r,p2,u,pu);

up=TM(p*p);

```

```

y=TM(p*m);
for ( i=0; i<p; i++ ) for ( j=0; j<p; j++ ) UP(i,j) = ( i == j ? 1 : 0 );

for ( i=0; i<p; i++ ) {
  for ( j=0; j<r; j++ ) YT(i,j) = P2U(i,j);
  for ( j=r; j<m; j++ ) YT(i,j) = UP(i,j-r);
}

h=TM(m*m);
for ( i=0; i<m; i++ ) for ( j=0; j<m; j++ ) H(i,j) = ( i == j ? 1 : 0 );

m1 = m-1;
for ( ip[m1]=m1, i=0; i<m1; i++ ) for ( j=0; j<m; j++ ) {
  a1 = H(i,j); H(i,j) = H(ip[i],j); H(ip[i],j) = a1;
}

yh = TM(p*m);
prod_m(p,m,m,y,h,yh);

if ( r == n && ! nb2 ) {

printf("%d", fprintf(fp, "\nClassificacao das Quantidades da Categoria 3\n"));
printf("%d", fprintf(fp, "\nTodas sao observaveis:"));

for ( i=0; i<n; i++ ) {
  for ( j=0; j<3; j++ ) printf("%d", fprintf(fp, "%c", CNM3(i,j)));
  printf("%d", fprintf(fp, " "));
}

printf("%d", fprintf(fp, "\n\nNao ha vazoes da Categoria 2\n"));
}

else {

if ( nb2 ) {
d=TM(p*nc);
for ( a1=0, i=0; i<p; i++ ) for ( k=0; k<nc; k++ ) {
  for ( j=0; j<m; j++ ) {
    a1 += YHT(i,j)*C(j,k);
  }
  D(i,k) = a1; a1 = 0;
}

dl = TM(p*nc);
icd = TMI(nc);
ipd = TMI(p);
for ( i=0; i<p; i++ ) for ( j=0; j<nc; j++ ) DL(i,j) = D(i,j);

for ( i=0; i<p; i++ ) ipd[i] = i;
for ( i=0; i<nc; i++ ) icd[i] = i;

```

```

rd = triang(p,nc,d1,d,ipd,icd);

d1 = TM(rd*rd);
for ( i=0 ; i<rd ; i++ ) for ( j=0 ; j<rd ; j++ ) D1(i,j) = D(i,j);
ird = TMI(rd);
ud = TM(rd*rd);
for ( i=0 ; i<rd ; i++ ) ird[i] = i;

for ( i=0 ; i<rd ; i++ )
    for ( j=0 ; j<rd ; j++ ) UD(i,j) = ( i == j ? 1 : 0 );

lu(rd,ird,d1);
backs(rd,rd,ird,d1,ud);

d3 = TM(rd*(nc-rd));
for ( i=0 ; i<rd ; i++ ) for ( j=rd ; j<nc ; j++ ) D3(i,j-rd) = D(i,j);

pmrd = p-rd;
d2 = TM(pmrd*rd);
for ( i=0 ; i<pmrd ; i++ ) for ( j=0 ; j<rd ; j++ ) D2(i,j) = -D(i+rd,j);

if ( rd == nc )
    if ( r == n ) {

printf("%d", fprintf(fp,"\\nClassificacao das Vazoes da Categoria 2\\n"));
printf("%d", fprintf(fp,"\\nTodas sao observaveis:"));
for ( i=0 ; i<nb2 ; i++ ) {
    for ( j=0 ; j<3 ; j++ ) printf("%d", fprintf(fp,"%c",CNM2(i,j)));
    printf("%d", fprintf(fp," "));
}
printf("%d", fprintf(fp,"\\n\\nClassificacao das Quantidades da Categoria 3\\n"));
printf("%d", fprintf(fp,"\\nTodas sao observaveis:"));
for ( i=0 ; i<n ; i++ ) {
    for ( j=0 ; j<3 ; j++ ) printf("%d", fprintf(fp,"%c",CNM3(i,j)));
    printf("%d", fprintf(fp," "));
}
goto fim;
}

else {

up3 = TM(r*(n-r));
for ( a1=0, i=0 ; i<r ; i++ ) for ( j=0 ; j<n-r ; j++ ) {
    for ( k=0 ; k<r ; k++ ) {
        a1 += -U(i,k) * P3(k,j);
    }
    UP3(i,j) = a1; a1 = 0;
}

vi = TM((n-r)*(n-r));
for ( i=0 ; i<n-r ; i++ )

```

```

    for (j=0; j<n-r; j++) VI(i,j) = (i==j ? 1 : 0);

vcn = TM(n*(n-r));
for (i=0; i<r; i++) for (j=0; j<n-r; j++) VCN(i,j) = UP3(i,j);

for (i=r; i<n; i++) for (j=0; j<n-r; j++) VCN(i,j) = VI(i-r,j);

vvalnm3 = TMI(n);
for (a1=0, i=0; i<n; i++) { for (j=0; j<n-r; j++) {
    a1 += fabs ( VCN(i,j) );
    }
    vvalnm3[i] = ( a1 ? 0 : 1 ); a1 = 0;
}

for (j=0; j<n-1; j++) if (j < ic[j])
    for (i=0; i<3; i++) {
        comptroc[i] = CNM3(j,i); CNM3(j,i) = CNM3(ic[j],i);
        CNM3(ic[j],i) = comptroc[i];
    }

printf("%d", fprintf(fp, "\nClassificacao das Vazoes da Categoria 2\n"));
printf("%d", fprintf(fp, "\nTodas sao observaveis:"));
for (i=0; i<nb2; i++) {
    for (j=0; j<3; j++) printf("%d", fprintf(fp, "%c", CNM2(i,j)));
    printf("%d", fprintf(fp, " "));
}
printf("%d", fprintf(fp, "\n\nClassificacao das Quantidades da Categoria 3\n"));
printf("%d", fprintf(fp, "\nQuantidades Observaveis:"));
for (i=0; i<n; i++) if (vvalnm3[i] == 1) {
    for (j=0; j<3; j++) printf("%d", fprintf(fp, "%c", CNM3(i,j)));
    printf("%d", fprintf(fp, " "));
}

printf("%d", fprintf(fp, "\n\nQuantidades Nao-Observaveis:"));
for (i=0; i<n; i++) if (vvalnm3[i] == 0) {
    for (j=0; j<3; j++) printf("%d", fprintf(fp, "%c", CNM3(i,j)));
    printf("%d", fprintf(fp, " "));
}
printf("%d", fprintf(fp, "\n"));
goto fim;
}

else {

ud3 = TM(rd*(nc-rd));
for (a1=0, i=0; i<rd; i++) for (j=0; j<nc-rd; j++) {
    for (k=0; k<rd; k++) {
        a1 += -UD(i,k) * D3(k,j);
    }
    UD3(i,j) = a1; a1 = 0;
}
}
}

```

```

viw = TM((nc-rd)*(nc-rd));

for ( i=0; i<nc-rd; i++ )
  for ( j=0; j<nc-rd ; j++ ) VIW(i,j) = ( i == j ? 1 : 0 );

w = TM(nc*(nc-rd));
for ( i=0; i<rd; i++ ) for ( j=0; j<nc-rd ; j++ ) W(i,j) = UD3(i,j);

for ( i=rd ; i<nc ; i++ ) for ( j=0 ; j<nc-rd ; j++ ) W(i,j) = VIW(i-rd,j);

vvalnm2 = TMI(nc);
for ( a1=0, i=0; i<nc ; i++ ) { for ( j=0 ; j<nc-rd ; j++ ) {
  a1 += fabs ( W(i,j) );
  }
  vvalnm2[i] = ( a1 ? 0 : 1 ); a1 = 0;
  }

for ( i=0 ; i<nc ; i++ ) if ( icd[i] != i ) {

hc=TMI(nc*nc);
for ( i=0 ; i<nc ; i++ )
  for ( j=0 ; j<nc ; j++ ) HC(i,j) = ( i == j ? 1 : 0 );

nc1 = nc-1;
for ( icd[nc1]=nc1, i=0 ; i<nc1 ; i++ ) for ( j=0 ; j<nc ; j++ ) {
  a1 = HC(j,i); HC(j,i) = HC(j,icd[i]) ; HC(j,icd[i]) = a1;
  }

vvalnm2t = TMI(nc);
for( i=0 ; i<nc ; i++ ) vvalnm2t[i] = vvalnm2[i];

for ( ali=0, j=0 ; j<nc ; j++ ) { for ( i=0 ; i<nc ; i++ ) {
  ali += HC(j,i) * vvalnm2t[i];
  }
  vvalnm2[j] = ali; ali = 0;
  }
  break;
  }
vvalnm2e = TMI(nb2);
for ( i=0, j=0 ; j<nc ; j++ ) { for ( k=0 ; k<vfra[j] ; k++ ) {
  vvalnm2e[i+j+k] = vvalnm2[j];
  }
  i += vfra[j]; i = i-1;
  }

printf("%d", fprintf(fp, "\nClassificacao das Vazoes da Categoria 2\n"));
printf("%d", fprintf(fp, "\nVazoes Observaveis:"));
for ( i=0 ; i<nb2 ; i++ ) if ( vvalnm2e[i] == 1 ) {
  for ( j=0; j<3 ; j++ ) printf("%d", fprintf(fp, "%c", CNM2(i,j)));
  printf("%d", fprintf(fp, " "));
  }

```

```

printf("%d", fprintf(fp, "\n"));
printf("%d", fprintf(fp, "\nVazoes Nao-Observaveis:"));
for ( i=0 ; i<nb2 ; i++ ) if ( vvalnm2e[i] == 0 ) {
    for ( j=0 ; j<3 ; j++ ) printf("%d", fprintf(fp, "%c", CNM2(i,j)));
    printf("%d", fprintf(fp, " "));
}

printf("%d", fprintf(fp, "\n"));

m1=m-1;
for ( ip[m1]=m1, i=0 ; i<m1 ; i++ ) for ( j=0 ; j<nc ; j++ ) {
    a1 = C(i,j); C(i,j) = C(ip[i],j); C(ip[i],j) = a1;
}

pcl = TM(r*nc);
for ( i=0 ; i<r ; i++ ) for ( j=0 ; j<nc ; j++ ) C1(i,j) = C(i,j);

clw = TM(r*(nc-rd));
for ( a1=0, i=0 ; i<r ; i++ ) for ( j=0 ; j<nc-rd ; j++ ) {
    for ( k=0 ; k<nc ; k++ ) {
        a1 += C1(i,k) * W(k,j);
    }
    C1W(i,j) = a1; a1 = 0;
}

t3 = TM((n-r)*(nc-rd));
p3t3 = TM(r*(nc-rd));
for ( i=0 ; i<n-r ; i++ ) for ( j=0 ; j<nc-rd ; j++ ) T3(i,j) = 1;

for ( a1=0, i=0 ; i<r ; i++ ) for ( j=0 ; j<nc-rd ; j++ ) {
    for ( k=0 ; k<n-r ; k++ ) {
        a1 += P3(i,k) * T3(k,j);
    }
    P3T3(i,j) = a1; a1 = 0;
}

for ( i=0 ; i<r ; i++ ) for ( j=0 ; j<nc-rd ; j++ ) C1W(i,j) += P3T3(i,j);

t = TM(n*(nc-rd));
for ( a1=0, i=0 ; i<r ; i++ ) for ( j=0 ; j<nc-rd ; j++ ) {
    for ( k=0 ; k<r ; k++ ) {
        a1 += -U(i,k) * C1W(k,j);
    }
    T(i,j) = a1; a1 = 0;
}

for ( i=r ; i<n ; i++ ) for ( j=0 ; j<nc-rd ; j++ ) T(i,j) = 1;

vvalnm3 = TMII(n);
for ( a1=0, i=0 ; i<n ; i++ ) { for ( j=0 ; j<nc-rd ; j++ ) { a1 += fabs( T(i,j) );
}
}

```



```

        vvalnm3[i] = ( a1 ? 0 : 1 ); a1 = 0;
    }
for ( j=0 ; j<n-1 ; j++ ) if ( j < ic[j] )
    for ( i=0; i<3; i++ ) {
        comptroc[i] = CNM3(j,i); CNM3(j,i) = CNM3(ic[j],i);
        CNM3(ic[j],i) = comptroc[i];
    }

printf("%d", fprintf(fp, "\nClassificacao das Quantidades da Categoria 3\n"));
printf("%d", fprintf(fp, "\nQuantidades Observaveis:"));
for ( i=0 ; i<n ; i++ ) if ( vvalnm3[i] == 1 ) {
    for ( j=0 ; j<3 ; j++ ) printf("%d", fprintf(fp, "%c", CNM3(i,j)));
    printf("%d", fprintf(fp, " "));
}

printf("%d", fprintf(fp, "\n"));

printf("%d", fprintf(fp, "\nQuantidades Nao-Observaveis:"));
for ( i=0 ; i<n ; i++ ) if ( vvalnm3[i] == 0 ) {
    for ( j=0 ; j<3 ; j++ ) printf("%d", fprintf(fp, "%c", CNM3(i,j)));
    printf("%d", fprintf(fp, " "));
}

printf("%d", fprintf(fp, "\n"));

goto fim;
}
}

printf("%d", fprintf(fp, "\nNao ha vazoes da Categoria 2\n"));

if ( r != n ) {

up3=TM(r*(n-r));
for ( a1=0, i=0 ; i<r ; i++ ) for ( j=0 ; j<n-r ; j++ ) {
    for ( k=0 ; k<r ; k++ ) {
        a1 += -U(i,k) * P3(k,j);
    }
    UP3(i,j) = a1; a1 = 0;
}

vi = TM((n-r)*(n-r));
for ( i=0 ; i<n-r ; i++ )
    for ( j=0 ; j<n-r ; j++ ) VI(i,j) = ( i == j ? 1 : 0 );

vcn = TM(n*(n-r));
for ( i=0 ; i<r ; i++ ) for ( j=0 ; j<n-r ; j++ ) VCN(i,j) = UP3(i,j);

for ( i=r; i<n ; i++ ) for ( j=0 ; j<n-r ; j++ ) VCN(i,j) = VI(i-r,j);

vvalnm3 = TMI(n);

```

```

for ( a1=0, i=0 ; i<n ; i++ ) { for ( j=0 ; j<n-r ; j++ ) {
    a1 += fabs( VCN(i,j) );
    }
    vvalnm3[i] = ( a1 ? 0 : 1 ); a1 = 0;
    }

for ( j=0 ; j<n-1 ; j++ ) if ( j < ic[j] )
    for ( i=0 ; i<3 ; i++ ) {
        comptroc[i] = CNM3(j,i); CNM3(j,i) = CNM3(ic[j],i);
        CNM3(ic[j],i) = comptroc[i];
    }

printf("%d", fprintf(fp, "\nClassificacao das Quantidades da Categoria 3\n"));
printf("%d", fprintf(fp, "\nQuantidades Observaveis:"));
for ( i=0 ; i<n ; i++ ) if ( vvalnm3[i] == 1 ) {
    for ( j=0 ; j<3 ; j++ ) printf("%d", fprintf(fp, "%c", CNM3(i,j)));
    printf("%d", fprintf(fp, " "));
    }

printf("%d", fprintf(fp, " "));

printf("%d", fprintf(fp, "\nQuantidades Nao-Observaveis:"));
for ( i=0 ; i<n ; i++ ) if ( vvalnm3[i] == 0 ) {
    for ( j=0 ; j<3 ; j++ ) printf("%d", fprintf(fp, "%c", CNM3(i,j)));
    printf("%d", fprintf(fp, " "));
    }

printf("%d", fprintf(fp, " "));

goto fim;

    }

printf("%d", fprintf(fp, "\nNao ha vazoes da Categoria 2\n"));
printf("%d", fprintf(fp, "\nClassificacao das Quantidades da Categoria 3\n"));
printf("%d", fprintf(fp, "\nTodas sao observaveis:"));
for ( i=0 ; i<n ; i++ ) {
    for ( j=0 ; j<3 ; j++ ) printf("%d", fprintf(fp, "%c", CNM3(i,j)));
    printf("%d", fprintf(fp, " "));
    }

    goto fim;

    }
}

else {

printf("\nNao ha quantidades da Categoria 3\n");
p = m;
yh = TM(p*m);

```

```

for ( i=0 ; i<p ; i++ )
  for ( j=0 ; j<m ; j++ ) YHT(i,j) = ( i == j ? 1 : 0 );

cl = TM(m*nc);
icd = TMI(nc);
ipd = TMI(m);

for ( i=0 ; i<m ; i++ ) for ( j=0 ; j<nc ; j++ ) CL(i,j) = C(i,j);

for ( i=0 ; i<m ; i++ ) ipd[i] = i;
for ( i=0 ; i<nc ; i++ ) icd[i] = i;

rd = triang(m,nc,c1,cc,ipd,icd);

if ( nc > rd || p > rd ) {

d1 = TM(rd*rd);
for ( i=0 ; i<rd ; i++ ) for ( j=0 ; j<rd ; j++ ) D1(i,j) = C(i,j);

ird = TMI(rd);
ud = TM(rd*rd);

for ( i=0 ; i<rd ; i++ ) ird[i] = i;

for ( i=0 ; i<rd ; i++ )
  for ( j=0 ; j<rd ; j++ ) UD(i,j) = ( i == j ? 1 : 0 );

lu(rd,ird,d1);

backs(rd,rd,ird,d1,ud);

pmrd=p-rd;
d2=TM(pmrd*rd);
for ( i=0 ; i<pmrd ; i++ ) for ( j=0 ; j<rd ; j++ ) D2(i,j) = -C(i+rd,j);

}

if ( rd == nc ) {

printf("%d", fprintf(fp,"nClassificacao das Vazoes da Categoria 2\n"));
printf("%d", fprintf(fp,"nTodas sao observaveis:"));
for ( i=0 ; i<nb2 ; i++ ) {
  for ( j=0 ; j<3 ; j++ ) printf("%d", fprintf(fp,"%c",CNM2(i,j)));
  printf("%d", fprintf(fp, " "));
}
}

else {

d3 = TM(rd*(nc-rd));
for ( i=0 ; i<rd ; i++ ) for ( j=rd ; j<nc ; j++ ) D3(i,j-rd) = C(i,j);

```

```

ud3 = TM(rd*(nc-rd));
for ( a1=0, i=0; i<rd ; i++ ) for ( j=0; j<nc-rd ; j++ ) {
    for ( k=0; k<rd ; k++ ) { a1 += -UD(i,k) * D3(k,j);
        }
    UD3(i,j) = a1; a1 = 0;
}

viw = TM((nc-rd)*(nc-rd));
for ( i=0; i<nc-rd ; i++ )
    for ( j=0; j<nc-rd ; j++ ) VIW(i,j) = ( i == j ? 1 : 0 );

w = TM(nc*(nc-rd));
for ( i=0; i<rd ; i++ ) for ( j=0; j<nc-rd; j++ ) W(i,j) = UD3(i,j);

for ( i=rd; i<nc; i++ ) for ( j=0; j<nc-rd; j++ ) W(i,j) = VIW(i-rd,j);

vvalnm2 = TMI(nc);
for ( a1=0, i=0; i<nc; i++ ) { for ( j=0; j<nc-rd; j++ ) {
    a1 += fabs( W(i,j) );
    }
    vvalnm2[i] = ( a1 ? 0 : 1 ); a1 = 0;
}

for ( i=0; i<nc; i++ ) if ( icd[i] != i ) {

hc=TMI(nc*nc);
for ( i=0; i<nc; i++ )
    for ( j=0; j<nc; j++ ) HC(i,j) = ( i == j ? 1 : 0 );

nc1 = nc-1;
for ( icd[nc1]=nc1, i=0; i<nc1; i++ ) for ( j=0; j<nc; j++ ) {
    a1 = HC(j,i); HC(j,i) = HC(j,icd[i]); HC(j,icd[i]) = a1;
}

vvalnm2t = TMI(nc);
for ( i=0; i<nc; i++ ) vvalnm2t[i] = vvalnm2[i];

for ( ali=0, j=0; j<nc; j++ ) { for ( i=0; i<nc; i++ ) {
    ali += HC(j,i)* vvalnm2t[i];
    }
    vvalnm2[j] = ali; ali = 0;
    }
break;
}

vvalnm2e = TMI(nc);
for ( i=0, j=0; j<nc; j++ ) { for ( k=0; k<vfra[j]; k++ ) {
    vvalnm2e[i+j+k] = vvalnm2[j];
    }
    i += vfra[j]; i = i-1;
}

```

```

printf("%d", fprintf(fp, "\nClassificacao das Vazoes da Categoria 2\n"));
printf("%d", fprintf(fp, "\nVazoes Observaveis:"));
for ( i=0; i<nb2; i++) if ( vvalnm2e[i] == 1 ) {
    for ( j=0; j<3; j++) printf("%d", fprintf(fp, "%c", CNM2(i,j)));
    printf("%d", fprintf(fp, " "));
}

printf("%d", fprintf(fp, "\n"));

printf("%d", fprintf(fp, "\nVazoes Nao-Observaveis:"));
for ( i=0; i<nb2; i++) if ( vvalnm2e[i] == 0 ) {
    for ( j=0; j<3; j++) printf("%d", fprintf(fp, "%c", CNM2(i,j)));
    printf("%d", fprintf(fp, " "));
}

printf("%d", fprintf(fp, "\n"));

}
}

fim:

/* Classificacao de Quantidades Medidas */

if ( r != m && rd != p ) {
    if ( nb2 ) {

du = TM(pmr*d*rd);
prod_m(pmr,d,rd,d2,ud,du);

uz = TM(pmr*d*pmr);
zt = TM(pmr*d*p);

for ( i=0; i<pmr; i++)
    for ( j=0; j<pmr; j++) UZ(i,j) = ( i == j ? 1 : 0 );

for ( i=0; i<pmr; i++) {
    for ( j=0; j<rd; j++) ZT(i,j) = D2U(i,j);
    for ( j=rd; j<p; j++) ZT(i,j) = UZ(i,j-rd);
}

hz=TM(p*p);
for ( i=0; i<p; i++)
    for ( j=0; j<p; j++) HZ(i,j) = ( i == j ? 1 : 0 );

m1 = p-1;
for ( ipd[m1]=m1, i=0; i<m1; i++) for ( j=0; j<p; j++) {
    a1 = HZ(i,j); HZ(i,j) = HZ(ipd[i],j) ; HZ(ipd[i],j) = a1;
}

zh = TM(pmr*d*p);

```

```

prod_m(pmr,d,p,p,zt,hz,zh);
}

else {
pmrd = p;
zh = TM(pmr*d*p);
for ( i=0; i<pmrd; i++)
  for ( j=0; j<p; j++) ZHT(i,j) = ( i == j ? 1 : 0 );
}

zy = TM(pmr*d*m);
for ( a1=0, i=0; i<pmrd; i++) for ( j=0; j<m; j++) {
  for ( k=0; k<p; k++) {
    a1 += ZHT(i,k) * YHT(k,j);
  }
  ZY(i,j) = a1; a1 = 0;
}

zbl = TM(pmr*d*q);
for ( a1=0, i=0; i<pmrd; i++) for ( j=0; j<q; j++) {
  for ( k=0; k<m; k++) {
    a1 += ZY(i,k) * B1(k,j);
  }
  ZB1(i,j) = a1; a1 = 0;
}

vvalmed1 = TMI(q);
for ( a1=0, j=0; j<q; j++) { for ( i=0; i<pmrd; i++) {
  a1 += fabs( ZB1(i,j) );
}
vvalmed1[j] = ( a1 ? 1 : 0 ); a1 = 0;
}

zto = TM(pmr*d*nct);
zbf = TM(pmr*d*q);

for ( i=0; i<pmrd; i++) for ( j=0; j<nct; j++) ZTO(i,j) = 0;

for ( j=0; j<q; j++) for ( i=0; i<pmrd; i++) {
  if ( fabs( ZB1(i,j) ) ) ZBF(i,j) = ZB1(i,j) * vaz[j];
  else ZBF(i,j) = 0;
}

for ( kj=0, j=0; j<q; j++) { for ( k=0; k<vvaz[kj]; k++) {
  for ( i=0; i<pmrd; i++)
    ZTO(i,kj) += ZBF(i,j+k);
  }
  kj++; j += k-1;
}
}

```

```

for (a1=0, j=0, k=vvaz[j]-1; j<nct; j++) {
  for ( i=0; i<pmrd; i++) {
    a1 += fabs( ZTO(i,j) );
  }
  vvalmed1 [ k] = ( a1 > 1 : 0 );
  a1 = 0;
  k += vvaz [j+1];
}

printf("%d", fprintf(fp, "\n\nClassificacao das Quantidades Medidas\n"));
printf("%d", fprintf(fp, "\nVazoes e Concentracoes da Categoria 1\n"));
printf("%d", fprintf(fp, "\nRedundantes:"));
for ( i=0; i<q; i++) if ( vvalmed1[i] == 1 ) {
  for ( j=0; j<3; j++) printf("%d", fprintf(fp, "%c", CM(i,j)));
  printf("%d", fprintf(fp, " "));
}

printf("%d", fprintf(fp, "\nNao-Redundantes:"));
for ( i=0; i<q; i++) if ( vvalmed1[i] == 0 ) {
  for ( j=0; j<3; j++) printf("%d", fprintf(fp, "%c", CM(i,j)));
  printf("%d", fprintf(fp, " "));
}

printf("%d", fprintf(fp, "\n"));

if ( nb2 ) {
  zb2 = TM(pmr*d*nb2);
  for ( a1=0, i=0; i<pmrd; i++) for ( j=0; j<nb2; j++) {
    for ( k=0; k<m; k++) {
      a1 += ZY(i,k) * B2(k,j);
    }
    ZB2(i,j) = a1; a1 = 0;
  }

  vvalmed2 = TMI(nb2);
  for ( a1=0, j=0; j<nb2; j++) { for ( i=0; i<pmrd; i++) {
    a1 += fabs( ZB2(i,j) );
  }
  vvalmed2[j] = ( a1 > 0.0001 ? 1 : 0 ); a1 = 0;
}

for (i=0, k=vfra[i]-1, j=0; j<nb2; j++) if (j == k) {
  vvalmed2[j] = 5; i++; k += vfra[i];
}

printf("%d", fprintf(fp, "\nConcentracoes da Categoria 2\n"));
printf("%d", fprintf(fp, "\nRedundantes:"));
for ( i=0; i<nb2; i++) if ( vvalmed2[i] == 1 ) {
  for ( j=0; j<3; j++) printf("%d", fprintf(fp, "%c", CNM2(i,j)));
  printf("%d", fprintf(fp, " "));
}

```

```

printf("%d", fprintf(fp, "\n\nNao-Redundantes:"));
for ( i=0; i<nb2; i++) if ( vvalmed2[i] == 0 ) {
    for ( j=0; j<3; j++) printf("%d", fprintf(fp,"%c",CNM2(i,j)));
    printf("%d", fprintf(fp, " "));
}
}
else printf("%d", fprintf(fp, "\nNao ha concentracoes da Categoria 2\n"));

}

else
    printf("%d", fprintf(fp, "\nTodos valores medidos sao nao-redundantes\n"));
fclose(fp);
}

void vnom(int c,int rq,int ni,int nic1,int *viv,char *cm,char *cnm2,char *cnm3)
{
    char *compo;
    char *numcor1;
    char *numcor2;
    char *alc;
    int c1 = c+1;
    register int i,j,k,ki, kj;

    compo=TMC(MCOM+1);
    numcor1=TMC(MCOR);
    numcor2=TMC(MCOR);
    alc=TMC(MCOR*(MCOM+1)*3);

    for ( i=0; i<c; i++ ) compo[i] = 'A'+i;

    for ( i=c; i<c1; i++ ) compo[i] = 'T';

    for ( i=0; i<9; i++ ) {
        numcor1[i] = '-'; numcor2[i] = '1'+i;
    }

    for ( j=0, i=9; i<ni && j<10; i++, j++ ) {
        numcor1[i] = '1'; numcor2[i] = '0'+j;
    }

    if ( ni == 20 ) {
        numcor1[ni-1] = '2'; numcor2[ni-1] = '0';
    }

    for ( i=0; i<ni; i++ ) for ( j=0; j<c1; j++ ) {
        ALC(j+i*c1,0) = compo[j]; ALC(j+i*c1,1) = numcor1[i];
        ALC(j+i*c1,2) = numcor2[i];
    }
}

```



```

for ( i=nic1; i<nic1 +rq; i++ ) { ALC(i,0) = 'R'; ALC(i,1) = '1'+(i-nic1); ALC(i,2) = ' ';
}

for ( ki=0, kj=0, k==0, i=0; i<nic1+rq; i++ ) {
if ( viv[i] == 1 ) {
for ( j=0; j<3; j++ ) CM(ki,j) = ALC(i,j); ki++;
}
if ( viv[i] == 2 ) {
for ( j=0; j<3; j++ ) CNM2(kj,j) = ALC(i,j); kj++;
}

if ( viv[i] == 3 ) {
for ( j=0; j<3; j++ ) CNM3(k,j) = ALC(i,j); k++;
}
}
}

```

```
int triang(m,n,pl,pp,ip,ic)
```

```
int m,n,*ip,*ic;
```

```
float *pl,*pp;
```

```
{
```

```
int m1=m-1,r;
```

```
float a1,a2;
```

```
double fabs( );
```

```
register int i,j,k,s,t;
```

```
for ( i=0; i<n;i++) {
```

```
for ( j=(k=i)+1; j<m; j++)
```

```
if (fabs(PL(j,i)) > fabs(PL(k,i))) k = j;
```

```
if(k !=i) for ( j=i; j<n; j++){
```

```
a1=PL(i,j); PL(i,j)=PL(k,j); PL(k,j) = a1;
```

```
}
```

```
a2 = PL(i,i); t = 0;
```

```
if ( a2 == 0 && t == 0 ) for ( j=i+1; j<n; j++ )
```

```
for ( s=i; s<m ; s++ )
```

```
if ( PL(s,j) != 0 && t == 0 ) for ( j=0; j<n; j++ ) {
```

```
a1 = PL(i,j); PL(i,j) = PL(s,j); PL(s,j) = a1;
```

```
t = 1; k = s;
```

```
}
```

```
a2 = PL(i,i); t = 0;
```

```
if ( a2 == 0 && t == 0 ) for ( s = i+1; s<n; s++ )
```

```
if ( PL(i,s) != 0 && t == 0 )
```

```
for ( j=0; j<m; j++ ) {
```

```
a1 = PL(j,i); PL(j,i) = PL(j,s); PL(j,s) = a1;
```

```
t = 1; ic[i] = s;
```

```
}
```

```

a2 = PL(i,i);
for ( ip[i]=k, j=i+1; j<m && a2 != 0 ; j++ ) {
    a1 = PL(j,i) /= -a2 ; PL(j,i) = 0;
    for ( k = i+ 1; k<n ; k++ ) PL(j,k) += a1 * PL(i,k);
}
}

m1=m-1;
for ( ip[m1] = m1, i=0; i<m1; i++ ) for ( j=0; j<n; j++ ) {
    a1 = P(i,j); P(i,j) = P(ip[i],j); P(ip[i],j) = a1;
}

for ( j=0; j<n-1; j++ ) if ( j<ic[j] )
    for ( i=0; i<m; i++ ) {
        a1 = P(i,j); P(i,j) = P(i,ic[j]); P(i,ic[j]) = a1;
    }

for ( a1=0, r=0, i=0; i<n && i<m; i++ ) if ( fabs(PL(i,i) ) > 0.0001 ) r++;
return r;
}

```

```

int lu(int r,int *ir,float *p1)
{
    int r1 = r-1;
    register int i,j,k;
    float a1,a2;double fabs( );

    for ( ir[r1]=r1, i=0 ; i<r1 ; i++ ) {
        for ( j = ( k=i) +1 ; j<r ; j++ )
            if ( fabs( P1(j,i) ) > fabs( P1(k,i))) k = j;
        if ( k != i ) for ( j=i ; j<r ; j++ ) {
            a1 = P1(i,j) ; P1(i,j) = P1(k,j) ; P1(k,j) = a1 ;
        }
    }
    if ( !(a2 = P1(i,i), a2 ) ) return (-1);

    for ( ir[i]=k, j=i+1; j<r; j++ ) {
        a1 = ( P1(j,i) /= -a2 );
        for ( k = i+1 ; k<r ; k++ ) P1(j,k) += a1 * P1(i,k) ;
    }
}
if( !P1(r1,r1) ) return (-1) ;
return(0);
}

```

```

void backs (int r, int rs, int *ir, float *p1, float *u )
{
    register int i, j, k;
    int r1 = r - 1;
    float a1;

```

```

for ( i = 0; i < r1; i++ ) {
    if ( ( j = ir[ i ] ) != i ) for ( k = 0; k < rs; k++ ) {
        a1 = U(i,k); U(i,k) = U(j,k); U(j,k) = a1;
    }
    for ( j = i + 1; j < r; j++ )
        for ( k = 0; k < rs; k++ ) U(j,k) += P1(j,i) * U(i,k);
}
for ( k = 0; k < rs; k++ ) U(r1,k) /= P1(r1,r1);
for ( i = r1 - 1; i >= 0; i-- ) {
    for ( a1 = P1(i,i), j = i + 1; j < r; j++ )
        for ( k = 0; k < rs; k++ ) U(i,k) -= P1(i,j) * U(j,k);
    for ( k = 0; k < rs; k++ ) U(i,k) /= a1;
}
}

```

```

void prod_m(int s,int r,int w,float *p2,float *u,float *pu)
{
    register int i,j,z;
    float a1;

    for ( a1=0, i=0; i<s; i++ ) for ( z=0; z<w; z++ ) {
        for ( j=0; j<r; j++ ) {
            a1 += P2(i,j) * U(j,z);
        }
        P2U(i,z) = a1; a1 = 0;
    }
}

```

```

char menu(void)
{
    char ch;
    do {
        printf("\n(E)nterar dados\n");
        printf("(M)ostrar dados\n");
        printf("(G)ravar em arquivo\n");
        printf("(L)er um arquivo de dados\n");
        printf("(C)lassificar\n");
        printf("(S)air\n");
        printf("\nEscolha uma das opcoes(E,M,G,L,C,S):");
        ch = toupper(getche());
    }
    while ( ch != 'E' && ch != 'M' && ch != 'G' && ch != 'S' && ch != 'L'
           && ch != 'C' );

    return ch;
}

```

```

void entrar( )
{
    int i,j,nviv,nvie;

    printf("Numero de correntes globais: ");
    scanf("%d%c",&ni);

    printf("Numero de nos: ");
    scanf("%d%c",&mi);

    printf("Numero de componentes: ");
    scanf("%d%c",&c);
    printf("Numero de reacoes quimicas: ");
    scanf("%d%c",&rq);

    printf("Numero de correntes globais da Categoria 2: ");
    scanf("%d%c",&nc);

    printf("Numero de correntes globais da Categoria 1: ");
    scanf("%d%c",&nct);

    printf("Numero de so fracoes medidas: ");
    scanf("%d%c",&nb2);

    printf("Numero de vazoes medidas: ");
    scanf("%d%c",&q);

    printf("Entre com informacoes das correntes\n");
    for ( i=0; i<ni; ++i ) {
        printf("Corrente %d:",i+1);
        scanf("%d%c%d%c",&M[i][0],&M[i][1]);
    }

    printf("Entre com informacoes sobre reacoes\n");
    for ( i=0; i<rq; i++ ) {
        printf("Reacao %d:",i+1);
        for ( j=0; j<c+2; j++ ) {
            printf("Elemento %d:",j+1);
            scanf("%f",&TE[i][j]);
        }
    }

    nviv = ni*(c+1)+rq;
    nvie = mi*(c+1);

    printf("Entre com informacoes das variaveis:\n");
    for ( i=0; i<nviv; i++ ) scanf("%d",&viv[i]);

    printf("Entre com informacoes dos equipamentos:\n");
    for ( i=0; i<nvie; i++ ) scanf("%d",&vie[i]);

```

```

printf("Entre com as so fracoes medidas:");
for ( i=0; i<nb2; i++) {
    printf("Entre com a fra[%d]:",i+1);
    scanf("%f",&fra[i]);
}

printf("Entre com informacoes das so fracoes medidas:");
for ( i=0; i<nc; i++){
    printf("Entre com vfra[%d]:",i+1);
    scanf("%d%*c",&vfra[i]);
}

printf("Entre com as vazoes\n");
for ( i=0; i<q; i++ ) {
    printf("Entre com vaz[%d]:",i+1);
    scanf("%f",&vaz[i]);
}

printf("Entre com informacoes das vazoes medidas:");
for ( i=0; i<nct; i++ ) {
    printf("Entre com vvaz[%d]:",i+1);
    scanf("%d%*c",&vvaz[i]);
}
}

void mostrar(void)
{
    int i,j;

    printf("\nNumero de correntes globais:%d\n",ni);

    printf("Numero de nos:%d\n",mi);

    printf("Numero de componentes:%d\n",c);

    printf("Numero de reacoes quimicas:%d\n",rq);

    printf("Numero de vazoes globais da Categoria 2:%d\n",nc);

    printf("Numero de vazoes globais da Categoria 1:%d\n",nct);

    printf("Numero de so fracoes medidas:%d\n",nb2);

    printf("Numero de vazoes medidas:%d\n",q);

    printf("\nInformacoes das correntes:\n");
    for ( i=0; i<ni; ++i )
        printf("Corrente %d:%d,%d\n",i+1,M[i][0],M[i][1]);
    printf("\n");
}

```

```
printf("\nMatriz estequiometria:\n");
for ( i=0; i<rq; i++ ) {
    for ( j=0; j<c+2; j++ )
        printf("%6.2f", TE[i][j]);
    printf("\n");
}
```

```
printf("\nInformacoes das variaveis:\n");
for ( i=0; i<(ni*(c+1))+rq; i++ ) printf("%d ",viv[i]);
printf("\n");
```

```
printf("\nInformacoes dos equipamentos:\n");
for ( i=0; i<mi*(c+1); i++ ) printf("%d ",vie[i]);
printf("\n");
printf("\nSo fracoes medidas:\n");
for ( i=0; i<nb2; i++ ) printf("%8.4f\t",fra[i]);
printf("\n");
```

```
printf("Informacoes das so fracoes medidas:");
for ( i=0; i<nc; i++ ) printf("%d ",vfra[i]);
printf("\n");
```

```
printf("Vazoes medidas\n");
for ( i=0; i<q; i++ ) printf("%8.4f\t",vaz[i]);
printf("\n");
```

```
printf("Informacoes das vazoes medidas:");
for ( i=0; i<nct; i++ ) printf("%d ",vvaz[i]);
printf("\n");
}
```

```
void gravar(void)
```

```
{
    FILE *fp;
    int i,j;

    char s[20];

    printf("Nome do arquivo:");
    gets(s);

    if(( fp = fopen(s,"w")) == NULL ) {
        printf("Nao posso abrir arquivo\n");
        return;
    }

    printf("%d",fprintf(fp,"%d %d %d %d \n",ni,mi,c,rq));

    printf("%d",fprintf(fp,"%d %d %d %d \n",nc,nct,nb2,q));
}
```

```

for (i=0; i<ni; i++ )
    printf("%d",fprintf(fp,"%d %d ",M[i][0],M[i][1]));
printf("%d",fprintf(fp,"\n\n"));

for ( i=0; i<rq; i++ ) {
    for ( j=0; j<c+2; j++ )
        printf("%d",fprintf(fp,"%6.2f",TE[i][j]));
        printf("%d",fprintf(fp,"\n"));
    }

for ( i=0; i<(ni*(c+ 1))+rq; i++ ) printf("%d",fprintf(fp,"%d ",viv[i]));

printf("%d",fprintf(fp,"\n"));

for ( i=0; i<mi*(c+ 1); i++ ) printf("%d",fprintf(fp,"%d ",vie[i]));
printf("%d",fprintf(fp,"\n"));

for ( i=0; i<nb2; i++ ) printf("%d",fprintf(fp,"%f\t",fra[i]));

printf("%d",fprintf(fp,"\n"));

for ( i=0; i<nc; i++ ) printf("%d",fprintf(fp,"%d ",vfra[i]));

printf("%d",fprintf(fp,"\n"));

for ( i=0; i<q; i++ ) printf("%d",fprintf(fp,"%f\t",vaz[i]));

printf("%d",fprintf(fp,"\n"));

for ( i=0; i<nct; i++ ) printf("%d",fprintf(fp,"%d ",vvaz[i]));

printf("%d",fprintf(fp,"\n"));

fclose(fp);
}

```

```

void ler(void)
{
    FILE *fp;
    int i,j;

    char s[20];

    printf("\nNome do arquivo de dados." );
    gets(s);

    if(( fp = fopen(s,"r")) == NULL ) {
        printf("Nao posso abrir arquivo\n");
        exit(1);
    }
}

```

```

fscanf(fp,"%d%*c%o%d%*c%o%d%*c%o%d%*c",&ni,&mi,&c,&rq);
fscanf(fp,"%d%*c%o%d%*c%o%d%*c%o%d%*c",&nc,&nct,&nb2,&q);
for ( i=0; i<ni; i++ ) fscanf(fp,"%d%*c%o%d%*c",&M[i][0],&M[i][1]);
for ( i=0; i<rq; i++ ) for ( j=0; j<c+2; j++ ) fscanf(fp,"%f%*c",&TE[i][j]);
for ( i=0; i<(ni*(c+1))+rq; i++ ) fscanf(fp,"%d%*c",&viv[i]);
for ( i=0; i<(mi*(c+1)); i++ ) fscanf(fp,"%d%*c",&vie[i]);
for ( i=0; i<nb2; i++ ) fscanf(fp,"%f%*c",&fra[i]);
for ( i=0; i<nc; i++ ) fscanf(fp,"%d%*c",&vfra[i]);
for ( i=0; i<q; i++ ) fscanf(fp,"%f%*c",&vaz[i]);
for ( i=0; i<nct; i++ ) fscanf(fp,"%d%*c",&vvaz[i]);
fclose(fp);
}

```



## LISTA DE SÍMBOLOS

a	vetor dos <b>aj</b> ustes $a_j$
$a_j$	ajuste da <b>variável</b> j
$a_j$	ajuste total
$a_*$	vetor dos <b>aj</b> ustes mínimos
$a'$	vetor cujos componentes são $2\bar{x}/\sigma_j^2$
$a_{ij}$	elemento da matriz A
A	matriz incidência de um processo
b	uma coluna da matriz B
$b_0$	vetor das <b>vazões</b> exatamente conhecidas
$b$	coluna da matriz $B_1$
B	matriz dos <b>balanços</b> do processo
$B'$	matriz das <b>colunas</b> de B não-removidas
$B''$	matriz das <b>colunas</b> de B removidas
$B_0$	matriz das <b>colunas</b> correspondentes às vazões constantes
$B_1$	matriz das <b>colunas</b> correspondentes às vazões medidas
$B_{1m}$	matriz das <b>r</b> primeiras linhas da matriz $B_1$ após troca de linhas
$B_2$	matriz das <b>colunas</b> correspondentes a vazões que têm só a concentração medida
$B_3$	matriz das <b>vazões</b> não-medidas
c	definida pela eq.4.17
c	número de <b>componentes</b> de um sistema
$c_{ij}$	concentração do componente i na corrente j
C	matriz definida pela eq.5.2
d	vetor das <b>concentrações</b> medidas da Categoria 2
$d^+$	d com <b>valores</b> consistentes
D	matriz definida pela eq.5.10
$D_0$	matriz diagonal com elementos $1/\sigma_j^2$
e	vetor dos <b>desbalanços</b>
$e_r$	vetor dos <b>desbalanços</b> após a remoção das l colunas de $B''$
$e_j$	erro da <b>variável</b> j
E	nó meio-ambiente num gráfico de processo
f	função-objetivo
$G_1$	matriz definida pela eq.4.11
H	matriz $Y^T B_1 \Sigma B_1^T Y$
$H_r$	matriz definida pela eq.4.7
i	índice que indica o componente
I	matriz identidade
j	índice que indica o número da corrente
J	número de <b>correntes</b> do processo
l	número de <b>medidas</b> removidas
k	índice que indica o número do nó
K	número de <b>nós</b> do processo
M	matriz definida pela eq.4.10
$M_n$	matriz definida pela eq.4.26
n	vetor das <b>vazões</b> totais não-medidas da Categoria 2
N	matriz definida pela eq.4.25
$N_2$	matriz diagonal que contém as <b>vazões</b> totais da Categoria 2
0	matriz nula

$p_j$	fator ponderante do ajuste da variável $j$
$P$	matriz igual a $[B_3 : S^T]$
$Q$	matriz variância não-singular de $a$ , definida pela eq.4.18
$r$	reação química
$r$	característica da matriz $P$
$R$	matriz com $r$ colunas ortogonais a $B''$ definida pela eq.4.16
$R_k$	número de reações químicas do nó $k$
$s'$	número de estatísticas de medida com valores absolutos distintos
$S$	matriz estequiometria de todo o processo
$S_k$	matriz estequiometria do nó $k$
$t$	vetor das variações em $v$
$u$	vetor das vazões não-medidas
$u_j$	vetor unitário, $j$ -ésima coluna de $I$
$v$	vetor de quantidades não-medidas (da Categoria 3) definido por $[u^T : \zeta^T]^T$
$v_*$	vetor das estimativas das quantidades não-medidas
$v_{ij}$	vazão do componente $i$ na corrente $j$
$v$	vetor arbitrário
$w$	vetor das variações em $n$
$w$	vetor arbitrário
$x$	vetor das vazões dos componentes
$\tilde{x}$	vetor dos valores medidos
$\tilde{x}_j$	valor medido da variável $j$
$x_j$	valor verdadeiro da variável $j$
$\hat{x}_j$	valor estimado (ajustado) da variável medida $j$
$\hat{x}$	vetor dos valores ajustados definido por $\hat{x} = \tilde{x} + a$
$x_{ij}$	vazão total da corrente $j$
$z_a$	estatística-teste de medida
$z_{a,j}^*$	estatística-teste de medida de potência máxima da medida $j$ , definida pela eq.4.20
$z_{ar,j}^*$	estatística-teste de medida de potência máxima da medida remanescente $j$ , definida pela eq.4.21
$z_{an,j}^*$	estatística-teste de medida de potência máxima da medida remanescente $j$ , após a remoção adicional da coluna de $B$ correspondente à medida $q$ , definida pela eq.4.31
$z_e$	estatística-teste de desbalanço (nodal)
$Z$	matriz definida pela eq.5.12
$Z^+$	$Z$ com valores consistentes
$Y$	matriz cujas colunas varrem o espaço-nulo de $P^T$
$\lambda$	vetor dos multiplicadores de Lagrange
$\lambda_i$	componente do vetor $\lambda$
$\lambda_*$	$\lambda$ que corresponde à solução mínima
$\alpha$	nível de probabilidade global
$\alpha^*$	nível de probabilidade de cada estatística, definida pela eq.4.38
$\delta$	variação causada por uma única remoção adicional
$\gamma$	escalar não-zero
$\mu$	média da população
$\sigma$	desvio-padrão
$\sigma_j^2$	variância da quantidade medida $j$
$\zeta$	vetor das conversões das reações

- $\Delta$  variação causada pela remoção das medidas correspondentes às colunas de  $B''$
- $\Gamma$  matriz definida pela eq.4.32
- $\Sigma$  matriz variância-covariância das variáveis medidas

## REFERÊNCIAS BIBLIOGRÁFICAS

- ALDRICH, C. & VAN DEVENDER, J.S.J. The use of neural nets to detect systematic errors in process systems. *Int. J. Miner. Process.*, v.39, p.173-97, 1993.
- ALMÁSY, G.A. & MAH, R.S.H. Estimation of measurement error variances from process data. *Ind. Eng. Chem., Process Des. Develop.*, v.23, n.4, p.779-84, 1984.
- ANDERSON, T.W. & SLOVE, S.L. *An Introduction to the Statistical Analysis of Data*. Boston: Houghton Mifflin, 1978. 720p.
- ALMÁSY, G.A. & SZTANÓ, T. Checking and correction of measurement errors on the basis of linear system model. *Problems of Control and Information Theory*, v.4, n.1, p.57-69, 1975.
- BECK, J.V. & ARNOLD, K.J. *Parameter Estimation In Engineering and Science*. New York: John Wiley & Sons, 1977. 520p.
- CARDOSO, M. *Desenvolvimento de um Programa Executivo Para Análise de Plantas Químicas*. Tese de Mestrado. UNICAMP, Agosto 1991.
- CROWE, C.M. Observability and redundancy of process data for steady state reconciliation. *Chem. Eng. Sci.*, v.44, n.12, p.2909-17, 1989(a).
- CROWE, C.M. Reconciliation of process flow rates by matrix projection. Part II: The nonlinear case. *AIChE J.*, v.32, n.4, p.616-23, Apr. 1986.
- CROWE, C.M. Recursive identification of gross errors in linear data reconciliation. *AIChE J.*, v.34, n.4, p.541-50, Apr. 1988.
- CROWE, C.M. The maximum power test for gross errors in original constraints in data reconciliation. *Can. J. Chem. Eng.*, v.70, n.5, p.1030-6, Oct. 1992.
- CROWE, C.M. Test of maximum power for detection of gross errors in process constraints. *AIChE J.*, v.35, n.5, p.869-72, May 1989(b).
- CROWE, C.M., GARCIA CAMPOS, Y.A. & HRYMAK, A. Reconciliation of process flow rates by matrix projection. Part I: Linear case. *AIChE J.*, v.29, n.12, p.881-8, Dec. 1983.
- DAROUACH, M. & ZASADZINSKI, M. Data reconciliation in generalized linear dynamic systems. *AIChE J.*, v.37, n.20, p.193-201, Feb. 1991.
- FREUND, J.E.. *Modern Elementary Statistics*. New Jersey: Prentice-Hall, 1988. 590p.
- HIMMELBLAU, D.M. *Fault Detection and Diagnosis in Chemical and Petrochemical Processes*. Amsterdam: Elsevier Scientific Publishing Company, 1978. 421p.
- HIMMELBLAU, D. M. *Process Analysis by Statistical Methods*. New York: John Wiley & Sons, 1969. 425p.

- HLAVACEK, V. Analysis of a complex plant—steady state and transient behavior. *Comput. Chem. Eng.*, v. 1, n.1, p.75-100, 1977.
- HOLLY, W. & CROWE, C.M. Reconciliation of mass flow rate measurements in a chemical extraction plant. *Can. J. Chem. Eng.*, v.67, n.8, p.595-601, Aug. 1989.
- IODACHE, I., MAH, R.S.H. & TAMHANE, A.C. Performance studies of the measurement test for detection of gross errors in process data. *AIChE J.*, v.31, n.7, p.1187-201, Jul. 1985.
- KAHANER, D., MOLER, C. & NASH, S. *Numerical Methods and Software*. Prentice-Hall, 1988. 507p.
- KAO, C.S., TAMHANE, A.C. & MAH, R.S.H. Gross error detection in serially correlated process data. *Ind. Eng. Chem. Res.*, v.29, n.6, p.1004-12, Jun. 1990.
- KELLER, J.Y., ZASADZINSKI, M. & DAROUACH, M. Analytical estimator of measurement error variances in data reconciliation. *Comput. Chem. Engng.*, v.16, n.3, p.185-8, Mar. 1992.
- KELLY-BOOTLE, S. *Dominando o Turbo C*. Rio de Janeiro: Ciência Moderna, 1989. 635p.
- KNEPPER, J.C. & GORMAN, J.W. Statistical analysis of constrained data sets. *AIChE J.*, v.26, n.2, p.260-4, Mar. 1980.
- KRETISOVALIS, A. & MAH, R.S.H. Effect of redundancy on estimation accuracy in process data reconciliation. *Chem. Eng. Sci.*, v.42, n.9, p.2115-21, 1987(a).
- KRETISOVALIS, A. & MAH, R.S.H. Observability and redundancy classification in generalized process networks-I. Theorems. *Comput. Chem. Engng.*, v.12, n.7, p.671-87, 1988(a).
- KRETISOVALIS, A. & MAH, R.S.H. Observability and redundancy classification in multicomponent process networks. *AIChE J.*, v.33, n.1, p.70-82, 1987(b).
- KUEHN, D.R. & DAVIDSON, H. Computer control-II. Mathematics of control. *Chem. Eng. Progr.*, v.57, n. 6, p.44-7, Jun. 1961.
- KUMAR, A. *Chemical Process Synthesis and Engineering Design*. New Delhi: Tata McGraw-Hill, 1981. 482p.
- LEIBOVICI, C.F., VERNEUIL, V.S. & YANG, P. Improve prediction with data reconciliation. *Hydrocarbon Processing*, n.10, p.79-80, Oct. 1993.
- LIEBMAN, M.J., EDGAR, T.F. & LASDON, L.S. Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques. *Comput. Chem. Engng.*, v.16, n.10/11, p.963-86, 1992.
- LIPSCHUTZ, S. *Probabilidade*. São Paulo: McGraw-Hill do Brasil, 1972. 222p.

- MacDONALD, R.J. & HOWAT, C.S. Data reconciliation and parameter estimation in plant performance analysis. *AIChE J.*, v.34, n.10, p.1-8, Jan. 1988.
- MADRON, F. A new approach to the identification of gross errors in chemical engineering measurements. *Chem. Eng. Sci.*, v.40, n.10, p.1855-60, 1985(a).
- MADRON, F. Identification of gross errors in balance measurements. *Coll. Cze. Chem. Commun.*, v.50, p.428-45, 1985(b).
- MADRON, F., VEVERKA, V. & VANECEK, V. Statistical analysis of material balance of a chemical reactor. *AIChE J.*, v.23, n.4, p.482-6, Jul. 1977.
- MAH, R.S.H. *Chemical Process Structures and Information Flows*. Stoneham: Butterworth Publishers, 1990. 500p.
- MAH, R.S.H. Design and analysis of process performance monitoring systems. *Engineering Foundation Conf. on Chem. Proc. Control II*, p.525-40, 1981.
- MAH, R.S.H., STANLEY, G.M. & DOWNING, D.M. Reconciliation and rectification of process flow and inventory data. *Ind. Eng. Chem., Process Des. Dev.*, v.15, n.1, p.175-83, 1976.
- MAH, R.S.H. & TAMHANE, A.C. Detection of gross errors in process data. *AIChE J.*, v.28, n.5, p.828-30, 1982.
- MATHIESEN, N.L. Adjustment of inconsistent sets of measurements using linear programming. *Automatica*, v.10, p.431-5, 1974.
- MURTHY, A.K.S. A least-squares solution to mass balance around a chemical reactor. *Ind. Eng. Chem., Process Des. Develop.*, v.12, n.3, p.246-8, 1973.
- MICKLEY, H.S., SHERWOOD, T.K. & REED, C.E. *Applied Mathematics in Chemical Engineering*. New York: McGraw-Hill, 1957. 425p.
- MILLER, I. et alii. *Probability and Statistics for Engineers*. New Jersey: Prentice-Hall, 1990. 624p.
- MILLER, J, FREUND, J.E. & JOHNSON, R.A. *Probability and Statistics for Engineers*. Englewood Cliffs: Prentice Hall, 1990. 624p.
- MURTHY, A.K.S. Material balance around a chemical reactor II. *Ind. Eng. Chem., Process Des. Develop.*, v.13, n.4, p.347-9, 1974.
- NARASIMHAN, S. et alii. A composite test for detecting changes of steady states. *AIChE J.*, v.32, n.9, p.1409-18, 1986.
- NARASIMHAN, S. & MAH, R.S.H. Generalized likelihood ratio method for gross error identification. *AIChE J.*, v.33, n.9, p.1514-21, Sept. 1987.
- NARASIMHAN, S. & MAH, R.S.H. Generalized likelihood ratio for gross error identification in dynamic processes. *AIChE J.*, v.34, n.8, p.1321-31, Aug. 1988.

- NARASIMHAN, S. & MAH, R.S.H. Treatment of general steady state process models in gross error identification. *Comput Chem. Engng.*, v.13, n.7, p.851-3, 1989.
- NOGITA, S. Statistical test and adjustment of process data. *Ind. Eng. Chem., Process Des. Develop.*, v.11, n.2, p.197-200, 1972.
- PAI, C.C.D. & FISHER, G.R. Application of Broyden's method to reconciliation of nonlinearly constrained data. *AIChE J.*, v.34, n.5, p.873-876, 1988.
- PHILLIPS, A.G. & HARRISON, D.P. Gross error detection and data reconciliation in experimental kinetics. *Ind. Eng. Chem. Res.*, v.32, n.11, p.2530-6, Nov. 1993.
- PRASAD, B.V.K. & KUESTER, J.L. Sequential and nonsequential process data coaptation. *Ind. Eng. Chem. Res.*, v.27, n.2, p.294-303, Feb. 1988.
- RAMAMURTHI, T., SISTU, P.B. & BEQUETTE, B.W. Control relevant dynamic data reconciliation and parameter estimation. *Comput. Chem. Engng.*, v.17, n.1, p.41-59, Jan. 1993.
- RIPPS, D.L. Adjustment of experimental data. *Chem. Eng. Prog. Symp. Ser.*, v.61, n.55, p.8-13, 1965.
- ROLLINS, D.K. & DAVIS, J.F. Unbiased estimation of gross errors in process measurements. *AIChE J.*, v.38, n.4, p.563-72, Apr. 1992.
- ROLLINS, D.K. & ROELFS, S.D. Gross error detection when constraints are bilinear. *AIChE J.*, v.38, n.8, p.1295-8, Aug. 1992.
- ROMAGNOLI, J.A. On data reconciliation: constraints processing and treatment of bias. *Chem. Eng. Sci.*, v.38, n.7, p.1107-17, 1983.
- ROMAGNOLI, J.A. & STEPHANOPOULOS, G. On the rectification of measurements errors for complex chemical plants. *Chem. Eng. Sci.*, v.35, p.1067-81, 1980.
- ROMAGNOLI, J.A. & STEPHANOPOULOS, G. Rectification of process measurement data in the presence of gross errors. *Chem. Eng. Sci.*, v.36, n.11, p.1849-63, 1981.
- ROSEMBERG, J., MAH, R.S.H. & IORDACHE, C. Evaluation of schemes for detecting and identifying gross errors in process data. *Ind. Eng. Chem. Res.*, v.26, n.3, p.555-64, 1987.
- SANCHEZ, M., BANDONI, A. & ROMAGNOLI, J. PLADAT: A package for process variable classification and plant data reconciliation. *Comput. Chem. Engng.*, v.16, p.s499-s506, May 1992. (Supplement).
- SCHILD, H. *Turbo C Avançado*. São Paulo: McGraw-Hill, 1990. (Guia do Usuário). 568p.
- SCHILD, H. *TURBO C: Guia do Usuário*. São Paulo: McGraw-Hill, 1988. 427p.
- SECCHI, A. R. & LAGE, P.L.C. *Turbo C: Guia de Referência*. Relatório interno, 1988. 65p.

- SERTH, R.W. & HEENAN, W.A. Gross error detection and data reconciliation in steam-metering systems. *AIChE J.*, v.32, n.5, p.733-42, May 1986.
- SERTH, R.W., VALERO, C.M. & HEENAN, W.A. Detection of gross errors in nonlinearly constrained data: a case study. *Chem. Eng. Comm.*, v.51, p.89-104, 1987.
- SERTH, R.W., SRIKANTH, B. & MARONGA, S.J. Gross error detection and stage efficiency estimation in a separation process. *AIChE J.*, v.39, n.10, p.1726-31, Oct. 1993.
- STANLEY, G.M. & MAH, R.S.H. Estimation of flows and temperature in process networks. *AIChE J.*, v.23, n.5, p.642-50, Sep. 1977.
- STANLEY, G.M. & MAH, R.S.H. Observability and redundancy classification in process networks. Theorems and algorithms. *Chem. Eng. Sci.*, v.36, n.12, p.1941-54, 1981(a).
- STANLEY, G.M. & MAH, R.S.H. Observability and redundancy in process data estimation. *Chem. Eng. Sci.*, v.36, p.259-72, 1981(b).
- STEPHENSON, G.R. & SHEWCHUK, C.F. Reconciliation of process data with process simulation. *AIChE J.*, v.32, n.2, p.247-54, Feb. 1986.
- TAMHANE, A.C. A note on the use of residuals for detecting an outlier in linear regression. *Biometrika*, v.69, n.2, p.488-9, 1982.
- TAMHANE, A.C. & MAH, R.S.H. Data reconciliation and gross error detection in chemical process networks. *Technometrics*, v.27, n.4, p.409-22, Nov. 1985.
- TERRY, P.A. & HIMMELBLAU, D.M. Data rectification and gross error detection in a steady-state process via artificial neural networks. *Ind. Eng. Chem. Res.*, v.32, n.12, p.3020-8, Dec. 1993.
- TJOA, I.B. & BIEGLER, L.T. Simultaneous strategies for data reconciliation and gross error detection of nonlinear systems. *Comput. Chem. Engng.*, v.15, n.10, p.679-90, 1991.
- UMEDA, T., NISHIO, M. & KOMATSU, S. A method for plant analysis and parameter estimation. *Ind. Eng. Chem., Process Des. Develop.*, v.10, n.2, p.236-43, 1971.
- VÁCLAVEK, V., KUBICEK, M. & LOUCHKA, M. Calculation of material balances for chemical engineering systems with due allowance for measurement errors. *Theor. Found. of Chem. Eng.*, v.9, p.242-245, 1976(a).
- VÁCLAVEK, V., KUBICEK, M. & LOUCHKA, M. Calculation of material balances in chemical engineering systems with allowance for errors in measurement: classification of stream parameters. *Theor. Found. of Chem. Eng.*, v.10, p.256-60, 1976(b).
- VÁCLAVEK, V. Studies on system engineering II. On the application of the calculus of observations in calculations of chemical engineering balances. *Coll. Cze. Chem. Commun.*, v.34, p.364-72, 1969(a).



VÁCLAVEK, V. Studies on system engineering III. Optimal choice of the balance measurements in complicated chemical engineering systems. *Chem. Eng. Sci.*, v.24, n.6, p.947-55, 1969(b).

VÁCLAVEK, V. & LOUCHKA, M. Selection of measurements necessary to achieve multicomponent mass balances in chemical plant. *Chem. Eng. Sci.*, v.31, p.1199-205, 1976.

VERNEUIL, V.S., YANG, P. & MADRON, F. Banish bad plant data. *Chem. Eng. Prog.*, n.10, p.45-50, Oct.1992.

---

## ABSTRACT

Process data from industrial unities are the basis to take decisions which affects the product quality and industrial security, in certain situations. When they are obtained in laboratory scale, they are the beginning of new process design. By these reasons, the actions based on their analysis will be more correct when they are nearest of their real values.

In this thesis, a computational procedure to reconcile and rectify measured data and to obtain estimates of observable unmeasured quantities was developed. They are applied to multicomponent mass balances, subject to linear constraints, where can occur chemical reactions, or not. It is assumed that for any stream in which concentration is measured, the total flow rate is also measured.

The reconciliation and rectification procedures make use of matrix projection to obtain the adjustments of measured quantities, which is done by the weighted least squares method and the Lagrange multipliers, so that the conservation laws are obeyed. Beyond the mass balance constraints, other restrictions can be included as a known split fraction of a flow splitter.

In the data rectification procedure, three statistical tests are applied to the detection and/or identification of gross errors: the global test, that uses a chi-squared statistic constructed from the observed discrepancies in the constraints, and the measurement test and nodal imbalance test, whose statistics are tested against the standard normal distribution.

All the mathematical methods led to the development of computer software to data reconciliation and rectification and process variable classification, respectively, the RECON, RETIF and TCLASS programs.

The results of the application of the reconciliation data program (RECON) to data of five different processes are described, including a simple ammonia synthesis system, a chemical reactor where two chemical reactions occur, a steam-metering system, a nitric acid synthesis system and a grinding circuit. The use of the rectification data program (RETIF) is illustrated by its application to five different situations of the ammonia synthesis system, four of the nitric acid synthesis system and three of the grinding circuit. The results show the reliability of the mathematical method and software developed.

The TCLASS program to process variable classification can be applied to mass balance variables of multicomponent systems. The classification is based only on the constraints imposed on extensive quantities, except the normalization equations. The measured flow rates and concentrations are classified in redundant or nonredundant and the unmeasured flow rates and the extents of reaction are classified in observable or unobservable. This program can also be used for designing the placement of measurements points of monitoring systems. The results of the application of this program to eighth situations of the ammonia synthesis system and two of the potassium nitrate evaporation-crystallization are described.