



Ministério da  
**Ciência e Tecnologia**



[sid.inpe.br/mtc-m19/2011/03.30.12.48-NTC](http://sid.inpe.br/mtc-m19/2011/03.30.12.48-NTC)

## OPERACIONALIZAÇÃO DA ASSIMILAÇÃO DE DADOS GNSS NO PSAS: PARTE I

Dérek Schubert Valério de Souza  
Luiz Fernando Sapucci  
Flávio Santos de Cerqueira

URL do documento original:  
<<http://urlib.net/8JMKD3MGP7W/3945RG8>>

INPE  
São José dos Campos  
2011

**PUBLICADO POR:**

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@sid.inpe.br

**CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE (RE/DIR-204):****Presidente:**

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

**Membros:**

Dr<sup>a</sup> Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr<sup>a</sup> Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr<sup>a</sup> Regina Célia dos Santos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Dr. Ralf Gielow - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr. Wilson Yamaguti - Coordenação Engenharia e Tecnologia Espacial (ETE)

Dr. Horácio Hideki Yanasse - Centro de Tecnologias Especiais (CTE)

**BIBLIOTECA DIGITAL:**

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Deicy Farabello - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

**REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

**EDITORAÇÃO ELETRÔNICA:**

Vivéca Sant´Ana Lemos - Serviço de Informação e Documentação (SID)



Ministério da  
**Ciência e Tecnologia**



[sid.inpe.br/mtc-m19/2011/03.30.12.48-NTC](http://sid.inpe.br/mtc-m19/2011/03.30.12.48-NTC)

## OPERACIONALIZAÇÃO DA ASSIMILAÇÃO DE DADOS GNSS NO PSAS: PARTE I

Dérek Schubert Valério de Souza  
Luiz Fernando Sapucci  
Flávio Santos de Cerqueira

URL do documento original:  
<<http://urlib.net/8JMKD3MGP7W/3945RG8>>

INPE  
São José dos Campos  
2011



## **AGRADECIMENTOS**

Os autores agradecem a contribuição do meteorologista João Gerd Zell de Mattos no desenvolvimento de algumas ferramentas descritas no trabalho. Agradecimentos especiais são dados a Fundação de Apoio a Pesquisa do Estado de São Paulo (FAPESP), processo N. 2006/04008-2, pelo apoio financeiro a esse trabalho e ao Ministério de Ciências e Tecnologia pelo auxílio financeiro na forma de bolsa PCI (Programa de Capacitação Institucional), Processo N° 381.0\*8\*4/2010-7.

## RESUMO

Atualmente, o sistema de assimilação operacional do CPTEC é o PSAS (*Physical-space Statistical Assimilation System*), que desde sua primeira versão no CPTEC (Centro de Previsão de Tempo e Estudo Climáticos) tem sido beneficiado por significativos avanços nos últimos anos. Há diversas fontes de observações atmosféricas, que apesar de serem bastante exploradas em outros centros, não têm sido utilizadas no CPTEC. Esse é caso dos perfis de rádio ocultação GNSS (*Global Navigation Satellite System*) e dos valores do IWV (*Integrated Water Vapor*) provenientes de redes de receptores GNSS em terra. Com a inclusão desses novos dados no sistema de assimilação, espera-se a obtenção de melhores análises e previsões mais acertadas. Os perfis de altura geopotencial e de umidade provenientes da rádio ocultação GPS são uma fonte de dados importante na busca de melhores previsões numéricas de tempo. A fim de implementar a assimilação desses dados no modelo global do CPTEC, utilizando o PSAS, algumas ferramentas e procedimentos se fizeram necessários e serão aqui descritos. A descrição detalhada de tais procedimentos compõe o principal objetivo do presente relatório. Nesse relatório, são também apresentados os conceitos de rádio ocultação, IWV-GPS, e uma breve discussão sobre a análise de modelos de temperatura média, utilizada para obter os valores do IWV. Por último serão discutidos alguns outros tópicos que futuramente serão desenvolvidos na continuidade dos trabalhos.

## ÍNDICE DE FIGURAS

Figura 1 – Satélites GPS enviando sinais para os receptores dos satélites de baixa órbita.....	06
Figura 2 – Fluxograma que representa o fluxo de dados e as principais ações do programa conversor de arquivos NETCDF para ODS. As caixas brancas representam processos em Shell Script e as cinzas em Fortran.....	11
Figura 3 – Fluxograma que representa o fluxo de dados e as principais ações do programa que gera as estatísticas dos dados de radiossonda. As caixas brancas representam processos em Shell Script e as cinzas em Fortran.....	15
Figura 4 – Fluxograma que representa o fluxo de dados e as principais ações dos procedimentos para fazer uma comparação entre modelos de Temperatura Média. As caixas brancas representam processos em Shell Script, as cinzas em Fortran e as cinzas claro são processos em Matlab.....	23
Figura 5 – Figura representando o gráfico de dispersão das medidas de temperatura para os modelos avaliados.....	29
Figura 6 – Figura representando as estatísticas de VIÉS, RMS e Desvio Padrão dos modelos avaliados.....	30

## LISTA DE SIGLAS E ABREVIATURAS

PNT	Previsão Numérica do Tempo
NOAA	National Oceanic and Atmospheric Administration
ESA	European Spatial Agency
LETKF	Local Ensemble Transform Kalman Filter
CPTEC	Centro de Previsão de Tempo e Estudos Climáticos
INPE	Instituto Nacional de Pesquisas Espaciais
IWV	Integrated Water Vapor
IASI	Infrared Atmospheric Sounding Interferometer
GNSS	Global Navigation Satellite System
GLONASS	Global Navigation Satellite System
GPS	Global Positioning System
NASA	National Aeronautics and Space Administration
HDF	Hierarchical Data Format
PSAS	Physical-space Statistical Analysis System
GPSAS	Global Physical-Space Statistical Analysis System
FORTTRAN	IBM Mathematical Formula Translation System
GEMPAK	General Meteorology Package
ERS-2	European Remote Sensing satellite
ENVISAT	Environmental Satellite
LEO	Low Earth Orbits
EUA	Estados Unidos da América



RBMC	Rede Brasileira de Monitoramento Contínuo do Sistema GPS
IARS-AMSU	Atmospheric InfraRed Sounder/Advanced Microwave Sounding Unit
ODS	Observation Data Stream
BUFR	Binary Universal Form for the Representation of meteorological data
RMS	Root Mean Square
MATLAB	MATrix LABoratory
GRIB	Binary Grid
COSMIC	Constellation Observing System for Meteorology Ionosphere e Climate
BDM	Banco de dados Meteorológicos

## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>01</b>
1.1 Objetivos do trabalho.....	03
1.2 Conteúdos do Trabalho.....	04
<b>2. REVISÃO BIBLIOGRÁFICA.....</b>	<b>04</b>
2.1 Assimilação de Dados.....	05
2.2 A Rádio Ocultação GNSS.....	05
2.3 IWV-GPS.....	07
<b>3. DESCRIÇÃO DETALHADA DAS IMPLEMENTAÇÕES REALIZADAS.....</b>	<b>09</b>
3.1 Desenvolvimento do Conversor de NETCDF para ODS.....	09
3.2. Cálculo das estatísticas dos dados de Rádio Ocultação.....	14
3.3 Organizar e completar os dados ODS da REANALISE.....	17
3.4 Adaptar o conversor de dados Binários em Grib para saídas do modelo Global T213L42.....	19
3.5 Extrair dados de Radiossonda do BDM e criar gráficos de comparação para diversos modelos de Temperatura Média (TM) usando matlab.....	21
3.6 Criação de um interpolador de dados numéricos para as coordenadas das estações.....	26
<b>4. AVALIAÇÃO DOS MODELOS TEÓRICOS DE TM UTILIZANDO RADIOSSONDAS OPERACIONAIS.....</b>	<b>28</b>
<b>5. CONSIDERAÇÕES FINAIS E PERSPECTIVAS FUTURAS NO DESENVOLVIMENTO DO TRABALHO.....</b>	<b>31</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>37</b>
<b>APÊNDICE A – Programa feito em Shell script para ajudar a converter arquivos do tipo NETCDF para ODS.....</b>	<b>43</b>

<b>APÊNDICE B – Programa feito em Fortran que serve para fazer a conversão dos valores dos arquivos do tipo NETCDF para o tipo ODS.....</b>	<b>50</b>
<b>APÊNDICE C – Rotina feita em Shell Script em para criar o namelist e também para executar o Fortran.....</b>	<b>56</b>
<b>APÊNDICE D – Namelist que será lido pelo programa em Fortran na atividade 3.2 .....</b>	<b>57</b>
<b>APÊNDICE E – Código em Fortran para fazer o cálculo das estatísticas.....</b>	<b>58</b>
<b>APÊNDICE F – Rotina que foi feita em Shell Script que confere os arquivos ODS presentes em um determinado diretório.....</b>	<b>63</b>
<b>APÊNDICE G – Template que ajuda a gerar um matlab para cada estação.....</b>	<b>65</b>

## 1 . INTRODUÇÃO

A Previsão Numérica de Tempo (PNT) tem-se desenvolvido significativamente nos últimos anos e o grau de acerto tem superado as expectativas mais otimistas feitas décadas atrás. Essa evolução tem sido impulsionada pelo aumento da capacidade computacional disponível e principalmente pelo desenvolvimento de novas técnicas de coleta de dados. O desenvolvimento de novas tecnologias de processamento tem ampliado significativamente a capacidade computacional dos centros de previsão e por consequência a utilização de modelos com parametrizações mais sofisticadas e de maior resolução tem sido viabilizada, o que permite um melhor detalhamento espacial das variáveis prognósticas. No que se refere às novas técnicas de coleta de dados observa-se que nas últimas décadas houve um avanço significativo da tecnologia espacial o que tem permitido o monitoramento contínuo da atmosfera terrestre, incluindo regiões de difícil acesso ao homem. Atualmente, satélites como os da série NOAA, o AQUA e o TERRA (da NASA), e os ERS-2 e ENVISAT da ESA (Agência Espacial Europeia), entre outros, realizam medidas frequentes da distribuição espacial e temporal da umidade na atmosfera, da cobertura de nuvens, ventos próximos à superfície oceânica, temperatura da superfície do mar, índice de vegetação, medidas da concentração de ozônio, entre outras. Agregar a evolução das técnicas de observação à evolução dos modelos de PNT é de responsabilidade da Assimilação de Dados, pois essa tem por finalidade corrigir as condições iniciais dos modelos de PNT utilizando as observações disponíveis. Cabe ressaltar que, entre os fatores que tem contribuído para a evolução da PNT, o aumento do número de dados atmosféricos assimilados (em especial os provenientes de satélites) apresenta um predomínio sobre os demais, especialmente quando se trata de regiões oceânicas localizadas no Hemisfério Sul (KALNAY, 2002).

Atualmente, o sistema de assimilação operacional do CPTEC é o PSAS (*Physical-space Statistical Analysis System*) (DA SILVA et al., 1995; COHN et al., 1998), que desde sua primeira versão no CPTEC tem sido beneficiado pelas evoluções destacadas acima e significativos avanços nessa área são observados nos últimos anos (CINTRA et al., 2005, SAPUCCI et al., 2006 e ANDREOLI et al., 2007). Devido a limitações, esse sistema será substituído por um novo sistema denominado *Local Ensemble Transform Kalman Filter* (LETKF), o qual atualmente está sendo implantado. Diversos artigos sobre o desempenho desse novo sistema e suas vantagens em relação a outros sistemas podem ser encontrados em KALNAY (2008). Há diversas fontes de observações atmosféricas, que apesar de serem bastante exploradas em outros centros, não têm sido utilizadas no CPTEC, tanto no PSAS, para fins operacionais, como no LETKF, em fase de implantação. Esse é caso dos perfis de rádio ocultação GNSS (*Global Navigation Satellite System*) e dos valores do IWV (*Integrated Water Vapor*) provenientes de redes de receptores GNSS em terra.

A técnica de rádio ocultação passou a ser aplicada na observação da atmosfera terrestre com o desenvolvimento e a operacionalização dos sistemas de posicionamento baseados em sinais de radiofrequência emitidos por satélites artificiais. O GNSS é um sistema que integra os principais sistemas de posicionamento por satélite, como o GPS (*Global Positioning System*) americano, o GLONASS (*Global Navigation Satellite System*) russo e o GALILEO, sistema da União Europeia em desenvolvimento. A rádio ocultação GNSS (daqui para frente denominada como RO-GNSS) está relacionada com dois satélites ocultos pela Terra, a saber: um satélite GNSS, que emite o sinal, e um satélite LEO (*Low Earth Orbits*), que recebe o sinal através de um receptor GNSS a bordo. Essa comunicação sofre os efeitos da troposfera terrestre, afetando a fase e a amplitude das ondas dos sinais envolvidos. Assim, a propagação do sinal entre o transmissor e o receptor se dá na forma de uma linha curva. Isso faz com que os satélites envolvidos, mesmo não estando visíveis entre si, consigam enviar e receber sinais.

Como essa alteração é função da densidade da atmosfera no ponto de ocultação, ao varrer verticalmente a mesma, torna-se possível a obtenção de um perfil atmosférico de temperatura e umidade (KURSINSKI, 1996).

As redes de receptores GNSS, implantadas inicialmente para fins geodésicos, têm contribuído para obtenção de dados meteorológicos, pois são capazes de gerar valores do IWV com boa qualidade (BEVIS et al. 1992). Diversos países, como por exemplo: os EUA (WARE et al., 1997), a Suécia (EMARDSON, 1998), o Japão (TSUDA et al., 1998), Alemanha (REIGBER et al., 2001; MAREL 2001) e a Itália (PACIONE et al., 2001a) já tem explorado suas redes locais para esse fim. Com as redes de receptores localizadas no Brasil, integradas à RBMC (Rede Brasileira de Monitoramento Contínuo dos satélites GNSS), bem como as outras em implantação, e dispendo de sensores meteorológicos de pressão atmosférica, valores do IWV de boa qualidade podem ser obtidos com boa resolução temporal. Tanto os perfis gerados pela técnica de RO-GNSS como os valores integrados de umidade obtidos a partir de receptores GNSS, ao serem assimilados nos modelos de PNT, apresentam grande potencial de minimizar os erros na previsão de tempo.

### **1.1 Objetivos do trabalho**

Este relatório técnico tem como objetivo a descrição dos procedimentos realizados para a assimilação operacional dos dados de RO-GNSS fornecidos pela constelação COSMIC no sistema de assimilação PSAS, bem como alguns preparativos visando à operacionalização da assimilação do IWV provenientes de redes de receptores GPS instalados no Brasil. Diversas implementações foram realizadas as quais de forma direta ou indiretamente estão ligadas com o objetivo descrito a cima. Os objetivos específicos que contribuem para se alcançar a meta acima podem ser descritos da seguinte forma:

- Implementar um método para fazer a conversão dos arquivos de formato netcdf para arquivos ODS, o qual é o formato de escrita do arquivo de dados a serem assimilados no PSAS;
- Desenvolver os procedimentos necessários para fazer com que os arquivos de rádio ocultação sejam assimilados de forma operacional pelo PSAS;
- Baixar, converter e organizar os dados ODS de outras fontes de dados para que possam ser utilizados nos experimentos para a avaliação dos resultados;
- Desenvolver procedimentos para extrair os dados de radiossonda do BDM (Banco de Dados Meteorológicos) do CPTEC e também fazer a comparação de valores da temperatura média (TM) obtidos utilizando diversos modelos teóricos. Os valores da TM são utilizados na obtenção de valores do IWV a partir de observações feitas por receptores GNSS.
- Desenvolver os procedimentos para realizar a interpolação dos campos do modelo de previsão para pontos de observações, visando à avaliação dos mesmos ao comparar com valores observados;
- Realizar a conversão dos arquivos binários de saída do PSAS para dados no formato grib. Essa implementação permite que os dados de saída ocupem menos espaço em disco, possibilitando que as rodadas sejam mais longas e que um maior número de experimentos sejam realizados.

## **1.2 Conteúdo do trabalho**

O segundo capítulo contém uma breve revisão bibliográfica sobre assimilação de dados, sobre a técnica de rádio ocultação e sobre a forma de se obter medidas de IWV a partir de observações GNSS. O terceiro capítulo descreve as implementações realizadas para fazer com que os dados de rádio ocultação fossem assimilados no PSAS, bem como as necessárias para tornar operacional a

assimilação desses dados ou aperfeiçoar as versões do modelo disponível para a pesquisa. No quarto capítulo é apresentada uma análise dos resultados obtidos na avaliação dos modelos teóricos ao compará-los com os valores obtidos utilizando as radiossondas. O quinto e último capítulo apresenta os comentários finais e trás alguns pontos que devem ser mais bem desenvolvidos na continuação dessa pesquisa e desenvolvimento. Nos apêndices desse trabalho encontram-se algumas rotinas desenvolvidas em cada uma das metas específicas listadas acima.

## **2. REVISÃO BIBLIOGRÁFICA**

No presente trabalho três principais pontos comparecem: assimilação de dados, a técnica de radio ocultação GNSS e a obtenção de valores de IWV a partir de observações feitas por receptores GNSS. Nesse capítulo são apresentadas algumas informações relevantes sobre cada um deles, bem como a citação de trabalhos nos quais ricos detalhamentos podem ser obtidos. A ideia básica do texto, é introduzir o leitor de forma sucinta nos temas abordados sem desviar o foco do trabalho que é descrever detalhadamente as implementações realizadas.

### **2.1 Assimilação de Dados**

A Assimilação de Dados tem a responsabilidade de adicionar à evolução das técnicas de observação nos modelos de PNT, pois essa tem por finalidade corrigir as condições iniciais dos modelos de PNT utilizando as observações disponíveis. Cabe ressaltar que entre os fatores que tem contribuído para a evolução da PNT, o aumento do número de dados atmosféricos assimilados (em especial os provenientes de satélites) apresenta um predomínio sobre os demais, especialmente quando se trata de regiões oceânicas localizadas no Hemisfério Sul (KALNAY, 2002).



## 2.2 A Rádio Ocultação GNSS

Por definição, GNSS é o termo genérico e padrão utilizado para designar sistema de navegação por satélites capazes de fornecer posicionamento geo-espacial autônomo com cobertura global. Dentre os sistemas que compõem o GNSS, destacam-se: o GPS (*Global Position System*), originalmente denominado NAVSTAR, dos EUA, em completa operação; o GLONASS, Sistema Russo de Navegação por Satélite, atualmente em processo de restauração; o GALILEO, sistema de posicionamento da União Europeia, em fase de desenvolvimento, e o COMPASS, projeto de expansão do atual sistema de navegação regional Chinês.

Atualmente o GPS é o sistema de posicionamento mais importante do mundo, funcionando com 32 satélites. Os satélites GPS estão em uma altitude aproximada de 20.000 km, com inclinação de 55 graus. A transmissão de sinal GPS permite determinar o posicionamento preciso de um receptor sobre a superfície terrestre, bem como estabelecer a posição exata do próprio satélite.

O fenômeno da ocultação acontece quando um corpo celeste deixa de ser visto por causa da sobreposição de outro corpo (FJELDB et al.; 1971). Em meados dos anos sessenta teve início a utilização da técnica de rádio ocultação, sendo usada posteriormente com sucesso para medir perfis verticais de temperatura e densidade das atmosferas de muitos planetas de nosso sistema solar. Nestes últimos anos houve um grande avanço nesta área e este avanço foi impulsionado pela evolução das tecnologias espaciais e também pelo lançamento de vários novos satélites.

A rádio ocultação GPS é viabilizada através da instalação de receptores GPS a bordo de satélites LEO. Os satélite LEO (*Low Earth Orbit*) são satélite de orbita polar, localizados a uma altitude aproximada de 800km. O processo de ocultação pode durar de 1 a 3 minutos. Durante este processo, com a variação de posição

entre o transmissor e o receptor GPS a bordo do satélite LEO, consegue-se um perfil vertical de refração do sinal eletromagnético (Figura 1). O sinal ao atravessar a atmosfera terrestre sofre refração, em função da variação de densidade do ar na camada atmosférica.

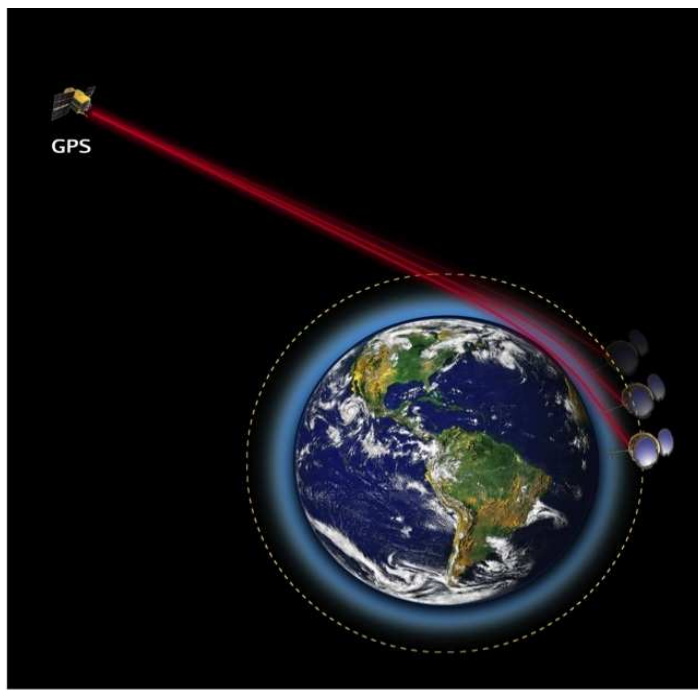


Figura 1 – Satélites GPS enviando sinais para os receptores dos satélites de baixa órbita.

A partir dos dados coletados por esse receptor, utilizando-se da relação existente entre a refração e a densidade do ar, é possível, através da técnica de Rádio Ocultação GPS, recuperar perfis de temperatura, pressão e vapor d'água da atmosfera terrestre (KURSINSKI et al.; 1996). Este processo é descrito em detalhes em (FJELDBO et al., 1971) e (KURSINSKI et al., 1997).

As medidas de RO - GNSS são capazes de representar o estado da atmosfera de uma maneira quase instantânea. Com 32 satélites GPS, um único receptor GPS instalado em um satélite LEO irá observar aproximadamente 500 ocultações por dia,

distribuídas quase que uniformemente ao redor do globo, e com todas essas ocultações provendo dados atmosféricos úteis.

Com a implementação dos receptores GPS COSMIC (*Constellation Observing System for Meteorology Ionosphere & Climate*), a meta é de que sejam viabilizados em torno de 4000 ocultações diárias, fornecendo em média um perfil a cada 360 quilômetros quadrados por dia.

Um único evento de ocultação que dura de 1 a 3 minutos é capaz de fornecer cerca de 4000 medidas por perfil, que tipicamente começa em 100 km e se estende com dados úteis até aproximadamente dois quilômetros de altura.

Os perfis de pressão e de umidade de radio ocultação GPS são uma fonte de dados importante na busca de melhores previsões numéricas de tempo. A fim de implementar a assimilação desses dados no modelo global do CPTEC, utilizando o PSAS, várias ferramentas de tratamento de dados foram necessárias e serão descritas no presente trabalho.

### **2.3 IWV-GPS**

A quantificação do vapor d'água na atmosfera, ao contrário de outras variáveis meteorológicas, é algo que ainda se apresenta como um grande desafio para as Ciências Atmosféricas. Diversos mecanismos, envolvendo diferentes técnicas, têm sido empregados e testados para esse fim em diferentes regiões do globo por pesquisadores das mais variadas áreas da ciência. (SAPUCCI, 2005)

IWV-GPS consiste no emprego de receptores GPS instalados em bases terrestres que possibilitam a obtenção do vapor d'água integrado na coluna atmosférica. Essas medidas são extraídas tendo como base o atraso zenital troposférico, ou seja, a interferência dos gases atmosféricos na propagação do sinal GPS, ao percorrer o caminho entre a estação terrestre e o satélite. O atraso zenital consiste na diferença entre: a distância geométrica retilínea de um receptor GPS na

Terra a um satélite GPS no espaço, e a real trajetória efetivamente percorrida pelo sinal eletromagnético. As ondas eletromagnéticas sofrem refração, cuja intensidade depende da quantidade de vapor de água na atmosfera, do perfil atmosférico de temperatura e da densidade do ar.

O atraso zenital troposférico ( $Z_{TD}$ ), dado pela equação(1), é composto por duas componentes: componente úmida ( $Z_{WD}$ ), que é influenciada pelo vapor de água presente na atmosfera, e a componente hidrostática ( $Z_{HD}$ ), que é influenciada pelos demais gases.

$$Z_{TD} = Z_{HD} + Z_{WD}. \quad (1)$$

A conversão do atraso zenital troposférico em medidas de conteúdo integrado de umidade atmosférica é generosamente detalhada em SAPUCCI (2001).

As sensibilidades dos sinais GPS às alterações da concentração de vapor d'água na atmosfera tornam a técnica IWV-GPS uma forma eficiente e barata de obtenção desses dados. Vários países vêm utilizando as redes de estações já existentes para o emprego da técnica IWV-GPS, nos EUA (BEVIS et al., 1992; DUAN et al., 1996; WARE et al., 1997; ROCKEN et al., 1997), na Suécia (EMARDSON, 1998), no Japão (TSUDA et al., 1998), na Alemanha (REIGBER et al., 2001; MAREL 2001) e na Itália (PACIONE et al., 2001a), entre outros.

Testes em assimilação dos valores obtidos por GPS têm sido feitos em modelos de previsão numérica de tempo com o objetivo de analisar o potencial dessa nova forma de obtenção de dados. O desenvolvimento de uma metodologia apropriada para a inclusão de valores do IWV em modelos de PNT de mesoescala (KUO et al., 1993) tornou possível a assimilação destas informações de várias fontes, tais como: satélites sondadores de umidade (SMITH, 1983); Raman Lidar (MELFI et al., 1989); dariômetro de microondas (HOGG et al., 1983); e valores IWV-GPS (BEVIS et al., 1992). Em alguns trabalhos os valores IWV-GPS foram utilizados para validar sistemas de PNT (YANG et al., 1999; CUCURULL et al., 2000). Para

que os dados IWW-GPS possam ser utilizados na previsão numérica do tempo, algumas exigências devem ser seguidas, dentre elas estão a estimativa do atraso zenital em tempo quase real. Outra não menos importante questão é existência de sensores de superfície próximos as bases GPS terrestres, a fim de obter dados de temperatura e pressão, que serão utilizados na conversão do atraso zenital em IWW.

### **3. DESCRIÇÃO DETALHADA DAS IMPLEMENTAÇÕES REALIZADAS**

Nessa seção são descritas as implementações realizadas para se atingir os objetivos do trabalho. Essa descrição compreende a apresentação de fluxogramas de todo o processo no qual estão envolvidos scripts em Shell, Matlab e rotinas Fortran, dos quais, os mais importantes, são apresentados na íntegra como apêndice desse trabalho. Uma breve descrição de cada implementação é apresentada no início de cada subseção na qual é mostrada sua importância dentro do contexto do trabalho.

#### **3.1 Desenvolvimento do Conversor de NETCDF para ODS**

O PSAS utiliza um formato de arquivos de dados de observação chamado ODS (*Observation Data Stream*) (CINTRA, 2004) com isso, há a necessidade de se criar conversores de dados, pois o formato de disseminação de dados quase sempre são baseados em bibliotecas CDF. Para o caso dos dados de rádio ocultação GNSS o formato de distribuição é o NETCDF. Nesse caso, o conversor foi criado utilizando a linguagem de programação Fortran, o qual é acionado por um Shell Script.

Os dados de RO-GNSS foram obtidos no site do *Cosmic (Constellation Observing System for Meteorology Ionosphere e Climate)*, site: [www.cosmic.ucar.edu/index.html](http://www.cosmic.ucar.edu/index.html), e são perfis atmosféricos de refratividade, temperatura, vapor de água e pressão.

O conversor foi criado e executado utilizando a máquina UNA no diretório: /home/flaviosc/gera\_ods\_ROGNSS. Pode ser encontrado também no caminho: /home2/dereks/Backup/TRAB2010/DESK\_210 na máquina paranoa, utilizando o usuário dereks. Neste caminho, encontra-se dois diretórios: *run* e *source*. O *run* contém o Shell Script *nc2ods\_temp.ksh* usado para executar o Fortran que se encontra na pasta *source* com o nome *nc2cdl\_temp.f90*. O local que se encontra os dados de entrada é: na máquina UNA /mpp/flaviosc/pathin, contendo as pastas com os arquivos. Para compilar este programa é necessário que a biblioteca do netcdf (netcdf-3.6.2) esteja instalada na máquina. A compilação foi realizada através do *Makefile* localizada na pasta *source*.

O Fluxograma apresentado na Figura 2 abaixo representa os processos mais importantes na conversão do arquivo e também o fluxo de dados que acontece dentro do programa. Após o fluxograma há uma breve explicação do que é abordado em cada processo dentro do fluxo.

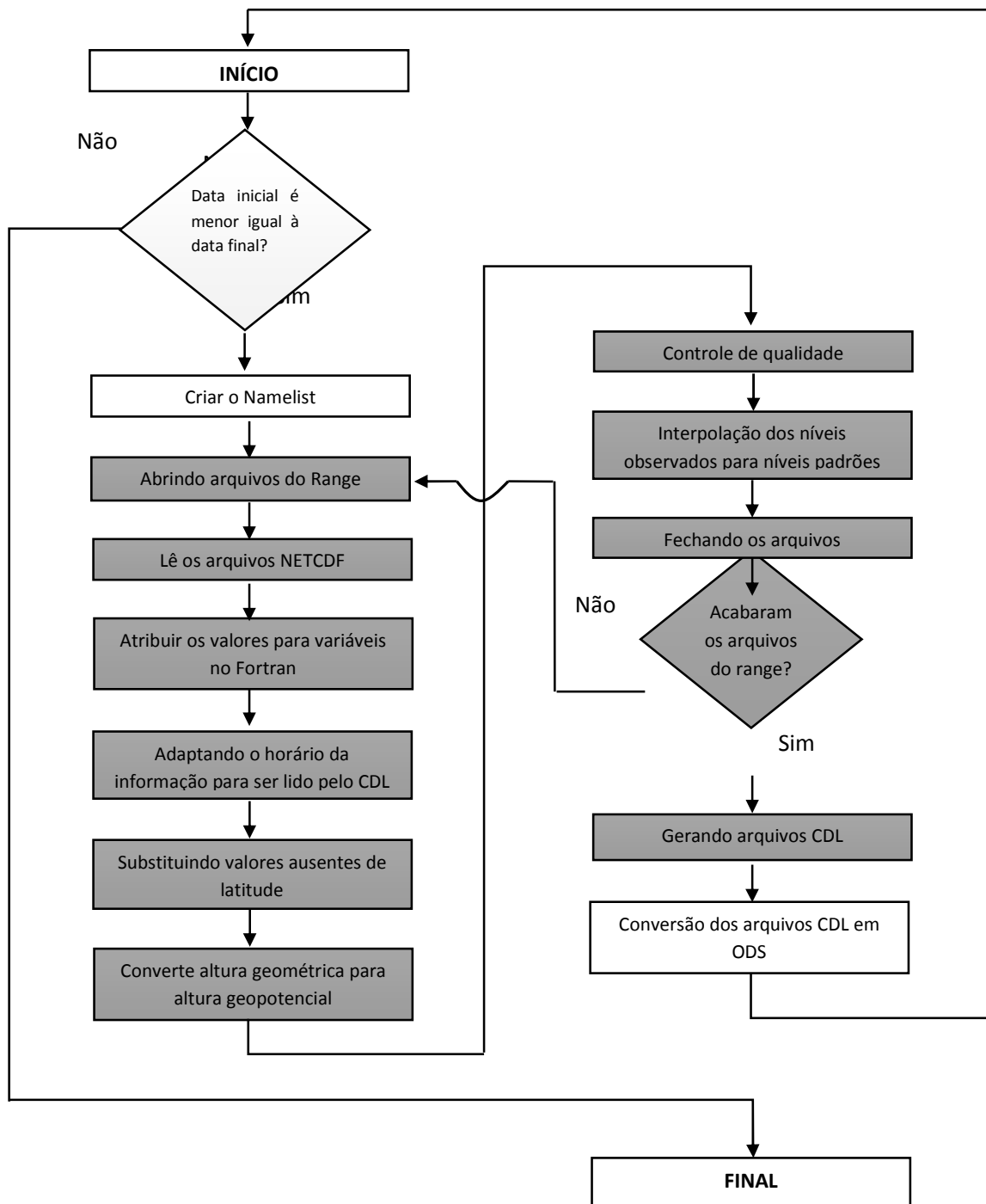


Figura 2 – Fluxograma que representa o fluxo de dados e as principais ações do programa conversor de arquivos NETCDF para ODS. As caixas brancas representam processos em Shell Script e as cinzas em Fortran.

As fases do fluxograma e suas funções são:

- Criar o Namelist - Neste passo o Shell Script gera o namelist contendo os nomes dos arquivos que fazem parte de cada range (00, 06, 12 ou 18) para que o Fortran possa lê-los;
- Abrindo arquivos do Range – Neste passo o Fortran abre os arquivos de entrada que contêm os dados de rádio ocultação no formato NETCDF;
- Lê os arquivos NETCDF - Neste passo, foi necessário criar uma forma de ler os arquivos NETCDF utilizando o Fortran, assim, foram utilizadas funções da biblioteca NETCDF ligadas ao Fortran para retirar os dados das variáveis contidas nos arquivos de entrada. As funções que controlam os arquivos NETCDF em Fortran estão no site: [www.unidata.ucar.edu/software/netcdf/docs/netcdf-f90](http://www.unidata.ucar.edu/software/netcdf/docs/netcdf-f90).
- Atribuir os valores para variáveis no Fortran - Neste passo, os valores das variáveis MSL, Temp, Lat, Lon, PV, Pres, Ref, Ref\_obs que estão contidos dentro dos arquivos de entrada, são atribuídos a vetores dentro do Fortran. Assim, são manipulados de forma a possibilitar a conversão. Caso tenham mais arquivos para um determinado range o Fortran refaz os passos de abrir e ler o arquivo e também atribuir os valores para as variáveis.
- Adaptando o horário da informação para ser lido pelo CDL - Nesse passo, foi feita uma transformação do horário em que o perfil foi obtido em minutos positivos ou negativos em torno da hora sinótica.
- Substituindo valores ausentes de latitude - Nesse passo, foi realizada uma substituição dos valores ausentes de latitude por valores globais de referência. Este recurso foi utilizado nos casos em que se têm os dados de pressão, temperatura e umidade em um determinado nível, mas o valor de



latitude se encontra ausente (-9999), situação esta em que a latitude igual a -9999 impedem as conversões de altura geométrica para altura geopotencial.

- Converte altura geométrica para altura geopotencial - Nesse passo, o Fortran fez a conversão das alturas geométricas contidas no perfil observado para altura geopotencial, que é o dado a ser assimilado pelo PSAS.
- Converte pressão de vapor em razão de mistura – Tal conversão se faz necessária, visto que o PSAS assimila razão de mistura como variável de umidade.
- Controle de qualidade – Este passo, exclui as observações abaixo de determinado nível onde o gradiente de refratividade for considerado anormal ( $DN/DZ < -50$ ). Isto para filtrar possíveis dados errôneos.
- Interpolação dos níveis observados para níveis padrões - Neste passo, foi realizada uma interpolação dos níveis observados para níveis padrões assimilados pelo PSAS.
- Fechando os arquivos – Neste passo, todos os arquivos abertos durante a conversão são fechados, apenas o arquivo que representa a namelist não será fechado, pois ele contém a lista de quais outros dados irão ser convertidos pelo Fortran.
- Gerando arquivos CDL – Neste passo, foi feita a criação dos arquivos CDL para os dados GEOP e também para a Umidade específica (Q). Utilizando todos os dados que foram obtidos através dos processos dentro do Fortran.
- Conversão dos arquivos CDL para ODS – Neste passo usou-se um Shell Script para fazer a conversão de arquivos CDL para arquivos ODS. Ele converte arquivos CDL criados pelo Fortran utilizando-se de uma biblioteca do HDF. O comando que foi usado neste processo para fazer a conversão foi:  
`/user/local/HDF4.2r1/Bin/ncgen -o <arquivo de saída> <arquivo de entrada>`

No Apêndice A, que se encontra no final deste trabalho, é apresentada a rotina em Shell Script que foi usada para efetuar a conversão de NETCDF para ODS. No Apêndice B, Encontra-se a rotina em Fortran que foi utilizada para ler os arquivos NETCDF e também para convertê-los em CDL, e posteriormente serem convertidos para ODS pelo Shell Script.

### **3.2. Cálculo das estatísticas dos dados de Rádio Ocultação.**

Após os dados NETCDF serem convertidos em ODS, eles passaram a serem introduzidos sistema GPSAS. As análises geradas em formato ODS contêm informações sobre os dados assimilados. Utilizando esses arquivos é possível verificar quais dados foram assimilados, bem como suas coordenadas, nível, e diferenças entre a os valores da observação e do *First guess* (O-F), para cada tipo de dados assimilado. Com o propósito de obter informações sobre os valores de O-F para os dados de RO - GNSS, mais especificamente dados de altura geopotencial e umidade, foi criado um programa que calcula o erro médio quadrático (RMS), o desvio padrão (DP) e Viés dos valores de O-F, tanto para as observações de RO - GNSS, como também para as demais observações assimiladas, como por exemplo, altura geopotencial provenientes de radiossonda e satélites. Essas estatísticas de O-F são importantes e o seu desvio padrão é uma informação utilizada como parâmetro dentro do PSAS, para cada tipo de dado observado.

Este programa foi criado no servidor pesquisa, diretório: /home/flaviosc, ou também pode ser encontrado no diretório: /home2/dereks/Backup/TRAB2010/DESK\_211. Ele foi compilado utilizando o ifort para gerar o seu executável. Este aplicativo se divide em quatro partes: o Shell script omf\_txt.ksh, que se encontra no apêndice C, é quem cria o *namelist* que contêm o diretório dos arquivos a serem lidos e seus respectivos nomes, e também serve para fazer o Fortran rodar. O *namelist\_omf.txt*, que se encontra no apêndice D, é o *namelist* que contêm o caminho e o nome dos arquivos que o Fortran irá abrir.

Nivel.txt é o arquivo que contém os níveis de pressão, e o omf.f90, que se encontra no apêndice E, é o programa em Fortran que tem todas as fórmulas para calcular as estatísticas e gerar os arquivos de saída com os respectivos resultados.

O Fluxograma na Figura 3 apresenta a sequência dos processos realizados nos cálculos das estatísticas e também em como se dá o fluxo de dados dentro da estrutura do programa.

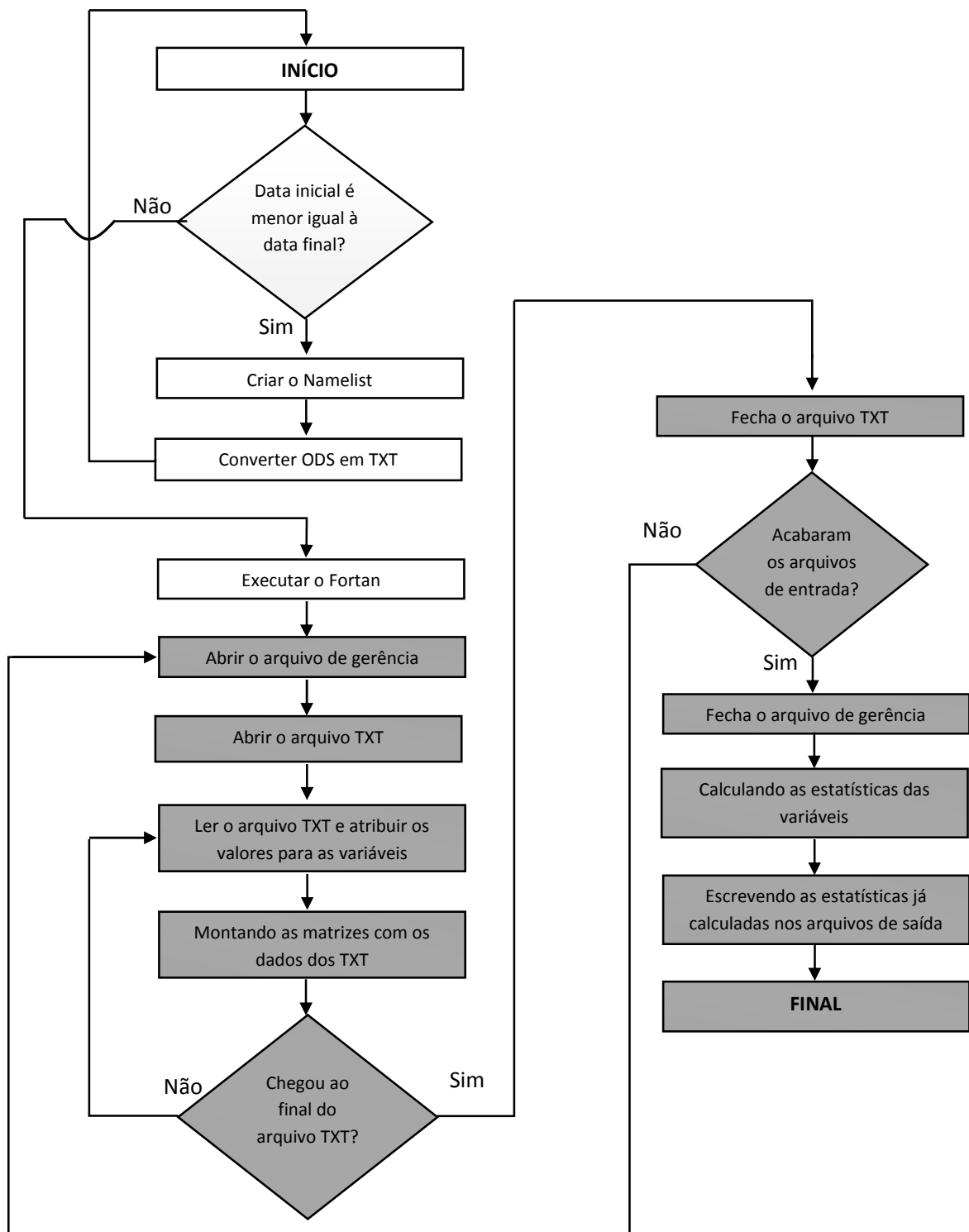


Figura 3 – Fluxograma que representa o fluxo de dados e as principais ações do programa que gera as estatísticas dos dados de radiossonda. As caixas brancas representam processos em Shell Script e as cinzas em Fortran.

- Criar o Namelist - Neste passo o Shell Script gera o *namelist* contendo o diretório onde se encontram os arquivos TXT que serão lidos pelo Fortran e lista o nome dos arquivos para o período desejado.
- Converter ODS em TXT – Neste passo o Shell converteu as saídas do PSAS que estão no formato ODS para TXT, assim os dados contidos dentro dos arquivos de saída podem ser lidos pelo Fortran.
- Executar o Fortan – O Script em sua última linha também executa o aplicativo em Fortran para dar início aos procedimentos de forma automática.
- Abrir o arquivo de gerência – Nesse passo o Fortran abriu o *namelist* e leu o seu conteúdo, assim, ele saberá onde estão os arquivos que ele irá ler para fazer os cálculos das estatísticas e também quais serão estes arquivos.
- Abrir o arquivo TXT – Neste passo o Fortran abriu o arquivo TXT que contém os dados para um determinado período de tempo.
- Ler o arquivo TXT e atribuir os valores às variáveis – O Fortran leu os arquivos de entrada e também atribuiu os valores lidos às matrizes que posteriormente servirão para fazer os cálculos das estatísticas.
- Montando as matrizes com os dados dos TXT – Neste passo, foram atribuídos os valores lidos nos arquivos TXT para dentro das matrizes de geopotencial e de umidade, assim, os cálculos puderam ser realizados. Depois disso, o programa voltará para a parte onde contém a leitura das linhas do arquivo para ler a próxima linha, caso não tenham mais linhas no arquivo o programa segue para fechar este arquivo e abrir o próximo.
- Fecha o arquivo TXT – Neste processo TXT foi fechado, e o programa verificou se ainda existem mais arquivos para serem lidos no *namelist*, caso não haja, o programa fecha o arquivo de *namelist*, do contrário, o fluxo de dados será remetido para abrir o arquivo novo TXT.

- Fecha o arquivo de gerência – Neste passo, o *namelist* foi fechado, pois todo o seu conteúdo já foi lido. Isso significa que todos os arquivos de entrada já foram completamente lidos pelo programa.
- Calculando as estatísticas das variáveis – Neste passo, foram calculadas todas as estatísticas (viés, rms e o desvio padrão) para todos os dados que foram lidos nas matrizes de geopotencial e de umidade.
- Escrevendo as estatísticas já calculadas nos arquivos de saída – Neste passo, os dados obtidos nos cálculos anteriores foram escritos nos arquivos de saída para serem guardados.

Com isso conclui-se o processo do cálculo das estatísticas para os dados de RO-GNSS.

### **3.3 Organizar e completar os dados ODS da REANALISE**

Uma forma de se verificar se os dados de rádio ocultação ou os dados de IWV-GPS adicionados na assimilação melhoraram a análise e a previsão é compará-los com os dados de reanálise do CPTEC. Estas previsões tradicionais são obtidas utilizando dados ODS que se localizam no caminho: `/rede/nas/gdad_reanalise/reanalisis/ods_files` na máquina tupay. Devido à falta de alguns dados ODS, para que pudessem ser feitas as comparações para um mesmo período, foi necessário criar um Shell Script para saber quais arquivos ODS estavam ausentes. Tal programa, denominado de `confere_ods.ksh` se encontra no diretório: `/rede/nas/gdad_reanalise/reanalisis/ods_files`. No Apêndice F são apresentados os comandos contidos no Shell Script `confere_ods.ksh`, que tem a função de conferir quais arquivos ODS estão presentes ou não em um determinado local escolhido do disco. O Shell foi executado e foi descoberto que havia muitos dados faltando, assim, foram iniciados os procedimentos para preencher todas estas lacunas dos anos de 2005 até 2009. Os passos realizados foram:

- Foi logado com o usuário Rita no servidor mangaba utilizando o comando: `ssh rita@mangaba -XC`;
- Entrou-se no diretório onde foi feito o download dos arquivos HDF com o comando: `cd /mangaba01/rita`;
- Foi criada uma pasta com o nome do período desejado para converter utilizando o nome padrão, YYYYMM ex.: 200503
- Foi copiado o Script `geralista.ksh` que se encontra na em: `mangaba01/rita` para dentro da nova pasta criada. Depois, editei este Script e alterei as datas de início e data de final para satisfazer o período dos arquivos que eu precisava. Este Script serve para criar a lista dos arquivos HDF que serão baixados para um determinado período;
- Foi executado o Shell Script `geralista.ksh` utilizando o comando: `./geralista.ksh`, ele gerou um arquivo com o nome `lista.airs.dat` a lista contendo os nomes dos arquivos HDF do período que foi escolhido;
- Após criar a lista foi necessário executar o comando: `nohup ./lista.airs.dat` para iniciar o download dos arquivos HDF no local onde estiver sendo executado;
- Depois de baixar os arquivos foi necessário move-los para outro lugar na rede, mas em uma pasta de mesmo nome da pasta de origem. Isto se vê necessário porque não é na mangaba onde ocorre a conversão dos arquivos HDF para ODS, o Shell que faz esta conversão se encontra no servidor pesquisa e para fazer com que no servidor pesquisa possa ter acesso aos HDF baixados, para usa-los como arquivos de entrada, é preciso mover os dados para outra localidade na rede. O novo local para onde os arquivos foram copiados é: `/rede/assimdados/psas/rita/AIRSreana/<nome da pasta destino>`. Foi copiado também para o local de destino o último dia do mês anterior ao mês em processamento, assim, o ODS do primeiro dia do mês

que esta sendo convertido ficará completo, pois ele utiliza uma parte dos arquivos do dia anterior para fazer a hora zero do ODS;

- Foi preciso logar no cluster pesquisa (teju) utilizando o comando: `ssh rita@pesquisa -XC`, pois o programa conversor de HDF para ODS esta compilado somente nesta máquina;
- Após entrar no servidor teju, foi preciso entrar na pasta: `/rede/assimdados/psas/rita/AIRSreana/gera_ods_airs`. Esta pasta contém os arquivos necessários para fazer a conversão dos dados AIRS em ODS;
- Foi necessário editar o Script `gera_ods_airs.sh`, arrumar a data inicial e a data final nas linhas: `set datai=<data inicial>` e `set dataf=<data final>` ;
- Foi preciso editar também o matlab `ler_hdf.m`. Neste Script em Matlab foi escrito o nome da pasta que foi criada onde se encontram os dados AIRS, foram editadas estas linhas:  
  
1- `cmd=['cd /rede/assimdados/psas/rita/AIRSreana/<nome-pasta>'];eval(cmd);`    2-  
`cmd=['airs_data_dir="/rede/assimdados/psas/rita/ARISreana/<nome-pasta>"/'];eval(cmd);`
- Por último, foi executado o Script `gera_ods_airs.sh` utilizando o comando: `./gera_ods_airs.sh` . Geralmente para converter um mês de dados são necessárias de sete a oito horas de processamento de dados.

Todos os dados utilizados neste processo também se encontram no diretório de backup em: `/home2/dereks/Backup/TRAB2010/DESK_2000`

Executando todos estes procedimentos, realiza-se a conversão de dados HDF para ODS. Isso foi feito para todos os arquivos ODS que estavam faltando entre os anos de 2005 á 2009. Agora com todos os dados completos pode-se gerar análises e previsões para comparar com a assimilação dos dados obtidos através de rádio ocultação e IWV-GPS.



### **3.4 Adaptar o conversor de dados Binários em Grib para saídas do modelo Global T213L42**

Os arquivos binários são as saídas do modelo PSAS, mas estes dados têm um tamanho elevado, com isso, é uma tarefa desafiadora conseguir um local para armazená-los. Então, foi proposto fazer a conversão dos arquivos binários para arquivos do tipo grib, pois este tipo de dado contém um tamanho menor e mantém uma qualidade minimamente inferior das informações. Assim, em cada resultado obtido pela assimilação dos dados de IWV-GPS serão gerados dados binários que, visando o ganho de espaço em disco, serão convertidos para grib.

Para realizar tal tarefa, foi preciso adaptar uma ferramenta de conversão já existente de forma a trabalhar com arquivos de entrada no formato T213L42. Foi necessário alterar os arquivos `writgrib.f90`, `bin2grb.sh`, `common.f90` e `grib_mod.f90` que se encontram na máquina tupay no seguinte caminho: `/gfs/home2/asismila/BIN2GRIB/GRIB/BIN2GRIB`, ou também pode ser encontrado acessando a máquina paranoa com o usuário dereks no caminho: `/home2/dereks/Backup/TRAB2010/DESK_2004/BIN2GRIB`

Para compilar este programa em tupay foi utilizado o compilador IFORT.

- Primeiramente foi necessário alterar o Fortran `common.f90`. Foram adicionadas novas variáveis atmosféricas, pois o modelo anterior tinha um número menor de variáveis;
- O arquivo escrito em Fortran: `grib_mod.f90`, configura as várias variáveis grib usadas para escrever a saída do arquivo em grib. Neste Fortran foram adicionados os códigos de cada variável atmosférica descritos na tabela do CPTEC/INPE para se adequar ao padrão T213L42;

- O Fortran `writegrib.f90` foi feito com o objetivo de escrever os dados em grib, nele estão contidas as funções que servem para ler e escrever arquivos em grib. Neste caso foram adicionadas as novas variáveis para que o arquivo possa ser convertido corretamente;
- O Shell Script `bin2grib.sh` teve uma alteração onde foram adicionados os comandos para extrair o CTL dos dados em grib e o comando para gerar o IDX do grib;
- Agora com a ferramenta já implementada e totalmente configurada, é necessário que seja escolhida a data para fazer a conversão e altere o Script com o nome de `bin2grb.sh`. Foi necessário mudar o caminho da variável `pathi` (entrada) que é onde estão os arquivos do tipo binários e da `patho` (saída) que é para onde vão os arquivos já convertidos para grib. Também deve ser alterada a data para o período que se deseja converter, as linhas para alterar as datas são 58 e 59 do Script;
- Depois, foi executado o Shell Script usando o comando: `./bin2grib.sh .`. Após finalizar a conversão, os arquivos CTL que foram gerados pelo `bin2grib.sh` não estão escritos com os nomes das variáveis corretos. Assim, foi necessário a criação de um Shell Script com o nome de `arruma_ctl.sh` para substituir todos os arquivos CTL errados por arquivos corretos. Para roda-lo foi necessário apenas que seja selecionada a data inicial, a data final e o local onde estejam os arquivos que se deseja arrumar. Depois se deve executar o seguinte comando: `./arruma_ctl.sh`;

Este programa permitiu a economia de um grande espaço em disco, que pode ser utilizado para armazenar mais rodadas de testes em assimilação do IWV-GPS.

### **3.5 Extrair dados de Radiossonda do BDM e criar gráficos de comparação para diversos modelos de Temperatura Média (TM) usando matlab.**

Para extrair os dados das estações de radiossonda do BDM (Banco de Dados Meteorológicos) foi logar no servidor capivary e acessar o seguinte diretório:

```
/tmp_mnt/home/derek/Dados_Derek/Radio_Sonda/Extrair_dados_BDM
```

O Shell Script com o nome de: `extrator_estac2010.ksh` é o script responsável por fazer a extração dos dados dos arquivos do BDM. Foi preciso editá-lo e alterar a data inicial e a data final usando o padrão de ano com quatro números, mês com dois números, dia com dois números e hora com dois números ex: 2001123123. Também foi necessário alterar, na linha 50 do script, o nome do diretório destinado a receber tais dados, tipo DRS. Para fazer a conversão das informações contidas nos dados DRS, foi necessário, utilizar um Fortran com o nome de `radioOPERbrAvalMod_v4_REANA_derek.for`. Este utiliza como arquivo de controle o `gereRADIOoper.txt`, *namelist* que contém os diretórios de saída do log e dos arquivos, o range inicial e final dos números das estações de rádio sonda, e o caminho completo de cada arquivo DRS. Este programa além de converter as informações em formato de TXT também tem vários filtros para extrair informações ruins dos dados do BDM. Dessa forma, a chance de ter alguma informação que atrapalhe a comparação entre os dados torna-se muito baixa.

Após extrair as informações do BDM, filtrar os dados para tirar valores incorretos e também converter para TXT, deve-se utilizar estes dados para fazer as comparações entre os diferentes tipos de modelos de Temperatura Média (TM). Para fazer os gráficos dessas comparações e também gerar estatísticas, foi utilizado um Script em Matlab que se encontra no Apêndice G. Este Script carrega os valores do arquivo de entrada e os atribui às variáveis. Iniciam-se os cálculos de VIÉS, RMS e Desvio Padrão e por último são plotados os gráficos de comparação.

Para descrever estes processos foi criado um fluxograma, apresentado na Figura 4, que contém os passos mais importantes e como acontece o fluxo dos dados durante a execução do programa.

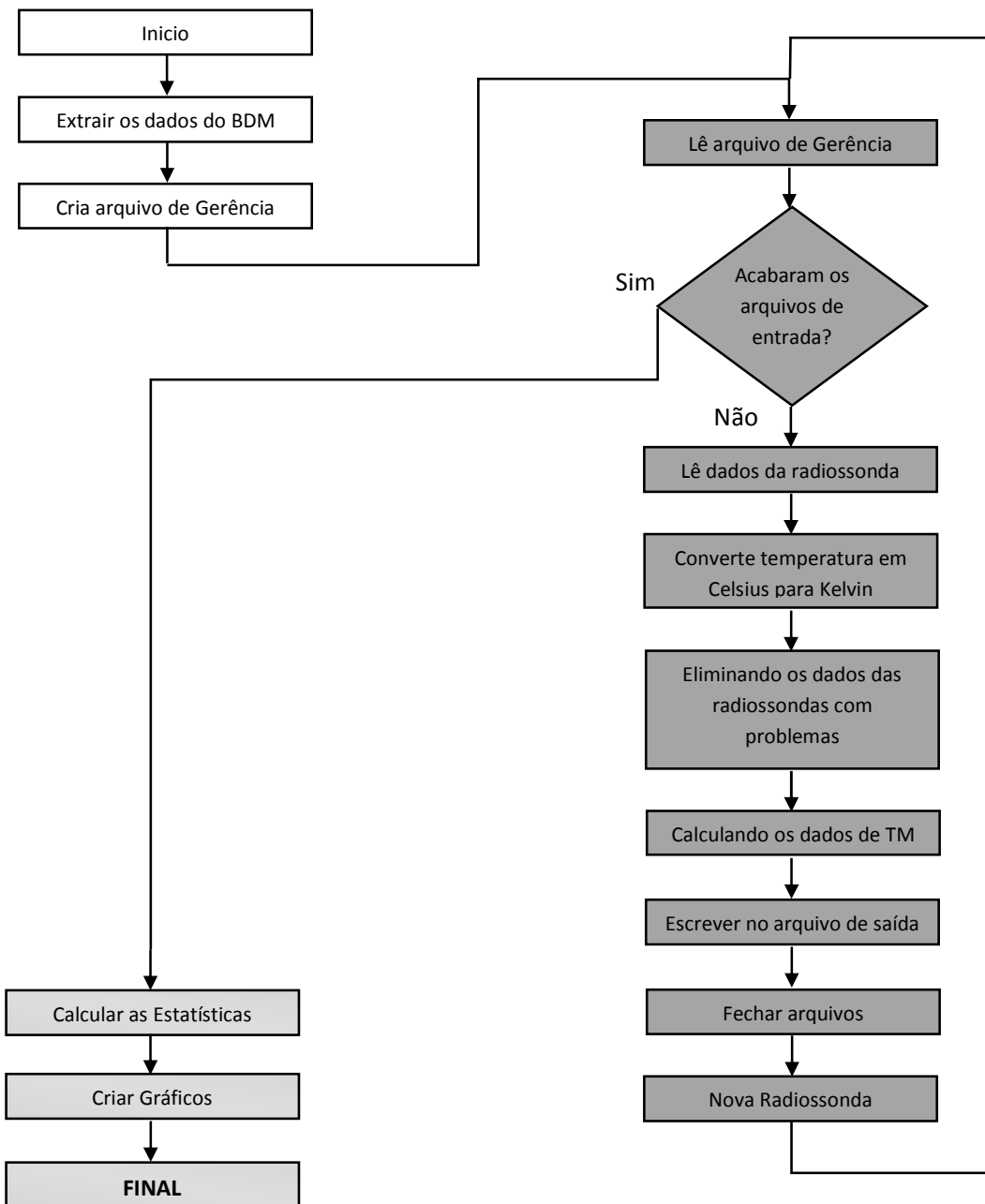


Figura 4 – Fluxograma que representa o fluxo de dados e as principais ações dos procedimentos para fazer uma comparação entre modelos de Temperatura Média. As caixas brancas representam processos em Shell Script, as cinzas em Fortran e as cinzas claro são processos em Matlab.

## FLUXOGRAMA:

- Extrair os dados do BDM - Neste passo deve-se logar na servidora capivary, depois deve-se executar o Shell Script denominado `extrator_estac2010.ksh` que tem a função de extrair os dados de radiossonda do banco de dados do BDM. Antes de executá-lo deve-se alterar a data inicial e a data final para o período desejado, em seguida, selecionar o range das estações de radiossonda para se obter os dados. Este processo pode ser um pouco demorado, dependendo das datas e da quantidade de estações selecionadas.
- Criar arquivo de gerencia – Nesse passo é necessário editar um arquivo chamado de `gereRADIOoper2.txt` encontrado no diretório:  

```
/home/derek/Dados_Derek/Radio_Sonda/RSOcalc/fonte
```

Sendo ele o *namelist* do programa, ele contém primeiramente o local e o nome do arquivo log que o Fortran criará. Na próxima linha, coloca-se o endereço dos arquivos de saída do programa (no caso os arquivos TXT), e por último, é necessário escrever uma lista com o caminho e o nome de todos os arquivos de entrada (os arquivos DRS do range obtidos com o processo anterior). Assim, com o *namelist* corretamente preenchido o Fortran não terá problemas em sua execução.
- Lê os arquivos de gerência – Nesse passo, o Fortran lê os nomes dos arquivos de entrada e os atribui a uma variável para que mais tarde possam ser abertos e lidos. Caso a lista chegue ao final isso fará com que o fluxo do programa vá direto para o passo de fechar os arquivos pulando todos os passos que estiverem entre eles.
- Lê dados da radiossonda – Nesse passo são feitas as atribuições dos valores que se encontram dentro dos arquivos que contêm os dados das radiossondas para as suas respectivas variáveis dentro do programa em Fortran.

- Converte temperatura em Celsius para Kelvin – Neste passo a temperatura é convertida de Celsius para Kelvin, possibilitando a realização dos cálculos pelos modelos.
- Eliminando os dados das radiossondas com problemas – Neste passo os dados das radiossondas obtidos são filtrados para que futuros problemas que envolvam os dados sejam evitados. Dessa forma garantindo uma maior integridade na informação que é utilizada para fazer as comparações.
- Calculando os dados de Temperatura Média (TM) – Neste passo utilizam-se as fórmulas para gerar todos os modelos de TM que serão usados para fazer as comparações de temperatura.
- Escrever no arquivo de saída – Nesse momento com os dados de temperatura média já calculados, filtrados e completos, é a hora de escrever estas informações no arquivo de saída no formato de TXT e posteriormente o Matlab possa ser utilizado para fazer os devidos gráficos de comparação.
- Fechar os arquivos – Agora os arquivos que foram usados para a leitura dos dados serão fechados. Logo após passar por este processo, o programa volta para a parte onde abre os arquivos DRS, de uma nova radiossonda, para refazer os mesmos procedimentos, assim, todos os arquivos serão lidos e convertidos corretamente.
- Calcular as estatísticas: Depois de concluído toda a parte de converter os dados DRS para dados de formato TXT, inicia-se o momento em que é feito o cálculo das estatísticas (RMS, VIES, Desvio padrão) utilizando o programa Matlab. Este script que foi feito em Matlab se encontra no diretório:

/home/Dados\_Derek/Radio\_Sonda/RSOcalc/RSOdados/radioOPERbeAvalMod

Para que se possa passar este script em matlab em todas as estações foi necessário a criação de um template com o nome de:

template\_estatisticas.txt

Para substituir os nomes de todas as estações no template, foi criado um Shell Script, visando passar por todas as estações e se tornar um processo automático, menos demorado e mais eficiente. O nome desse Shell é:

chama\_matlab\_estacao.ksh

- Criar gráficos – No mesmo Script, em Matlab, do processo anterior, têm-se algumas linhas de código para plotar um gráfico de dispersão. Nesse caso, é realizada uma comparação entre todos os modelos de temperatura média (TM) calculados, com a temperatura média que foi obtida pelas estações. Após fazer isso, também é plotado um gráfico contendo as estatísticas para cada modelo, dentro de um determinado período.

Todos estes processos permitiram que houvesse uma avaliação da qualidade dos valores da temperatura média troposférica para as estações do Brasil. Os arquivos utilizados para fazer este processo também podem ser encontrados na máquina paranoa usando o usuário dereks no caminho: /home2/dereks/Backup/TRA2010/DESK\_209/Radios\_Sonda .

### **3.6 Criação de um interpolador de dados numéricos para as coordenadas das estações.**

Nesta tarefa, foi realizada uma adaptação do programa interpZEDeta20\_v1.1.for para que ele pudesse ler arquivos do tipo binário. O interpolador faz uma interpolação por ponto de grade. Dada uma estação qualquer em algum lugar do globo, a partir de uma *namelist*, chamado de namelist.sh, que alimenta o interpolador com os dados de diretório de entrada, data inicial, data final, latitude, longitude e a altitude da estação, ele varre a grade para encontrar o local da estação e os quatro pontos de grade mais próximos a ela. Dessa forma, o programa realiza a interpolação dos valores das variáveis e obtêm um valor interpolado para



uma determinada variável de uma dada estação. Este Fortran foi compilado na máquina pesquisa e compilado com o compilador IFORT.

O programa se encontra na seguinte pasta:

/home/derek/Dados\_Derek/Interpolador/interpZWDeta20\_v1.v Ou também esta localizado na máquina paranoa no caminho: /home2/dereks/Backup/TRAB2010/DESK\_206/Intepolador . Seguem abaixo, alguns procedimentos que foram feitos para adaptar e rodar este aplicativo, bem como uma breve explicação de seu funcionamento:

- Primeiramente foi necessário alterar o Fortran interpZWDeta20\_v1.1.for para fazer com que ele conseguisse ler arquivos do tipo binário;
- Depois foi necessário adaptar este programa para fazer com que ele pudesse trabalhar com dados de superfície, pois originalmente ele trabalhava com dados que continham vários níveis. As variáveis que ele trabalha são: temperatura e pressão na superfície;
- No namelist foi necessário adicionar primeiramente o caminho do arquivo de *first guess* para o programa poder utilizá-lo na interpolação dos valores das variáveis desejadas. Em segundo, foi adicionado o nome dos arquivos de texto, onde o Fortran irá escrever a sua saída. Em terceiro, adicionou-se a data do período a ser rodado no programa. Em quarto, foi adicionado o nome do arquivo que servira de *first guess* a fim de ser feita a interpolação. E por ultimo adicionou-se os dados das estações, seu número, sua latitude, sua longitude e sua altitude;
- Realizados os procedimentos acima, o Fortran pôde ser executado no servidor pesquisa, através do comando:

```
./interpZWDeta20_v1.1.exe_pesquisa ;
```

- Todo o processo começou com o Shell passando pelas datas e criando o *namelist* para que o Fortran pudesse fazer a interpolação das estações. Assim, o Fortran foi executado pelo Shell Script e iniciou-se os processos para realizar as interpolações;
- Uma vez executado o Fortran abre o arquivo de *namelist* para ler o arquivo binário que servirá de base para fazer a interpolação. Depois disso, ele abre também o arquivo de topografia e lê toda a topografia do globo;
- Após abrir o arquivo binário lê as variáveis de pressão e temperatura na superfície;
- A partir desse ponto, começa ser feita a interpolação dos dados de temperatura e pressão;
- Depois do programa terminar de fazer a interpolação de todas as variáveis, ele inicia o processo de escrever no arquivo de saída os resultados do cálculos efetuados;
- Após o término de todas as estações, o Fortran fecha todos os arquivos que foram abertos e sai do processo.

#### **4. Avaliação dos modelos teóricos de TM utilizando radiossondas operacionais.**

A temperatura média (TM) da coluna vertical troposférica ao longo da altitude varia de acordo com o local e a época do ano (DAVIS et al., 1985). Os valores da TM podem ser determinados a partir da análise estatística de perfis de radiossondas (SAPUCCI, 2005). Existem vários modelos para calcular a temperatura média. Os modelos de Temperatura Média que foram utilizados para fazer as comparações

foram: TMsch1b, TMsch2, ERA40, TMsch3b, TMmend, TMsapu, TMregi. Cada um destes modelos tem uma formulação diferente para chegar a valores de temperatura média.

O modelo ERA40 que foi rodado utilizando dados de reanálise interpolados de um arquivo binário. O modelo TMsapu foi criado por SAPUCCI (2005) em sua tese de doutorado. O modelo TMmend foi desenvolvido por MENDES et al. (2000), usando os dados de radiossondas lançadas em vários locais em todo o mundo. Outro modelo é o de SCHUELER et al. (2001), ele utiliza perfis de temperatura preditos por modelos de previsão numérica de tempo.

Com todos estes modelos foram gerados duas figuras: a Figura 5, um gráfico de dispersão, contém dados de comparação da temperatura média entre os modelos, considerando todas as localidades, ou seja, todas as estações meteorológicas onde os dados foram medidos. E a Figura 6, que contém os dados das estatísticas (RMS, viés e Desvio Padrão) de cada modelo, considerando todas as estações de radiossondagens.

Na Figura 5 a temperatura média que foi calculada pelos modelos é comparada com temperatura média do perfil obtido pelas radiossondas. Podemos observar que em geral todos os modelos apresentam a mesma dispersão, porém com diferentes VIES. Destacando-se dentre eles o TMregi com menores valores, tanto de dispersão como de VIES.

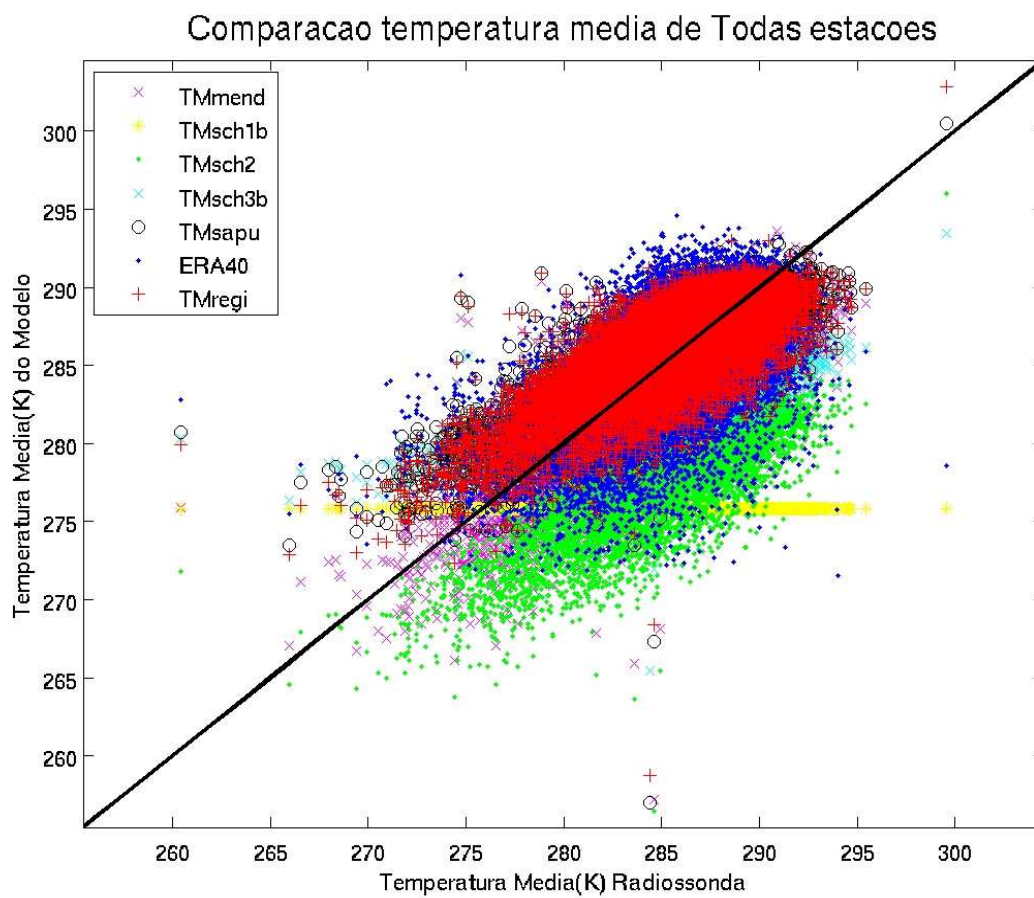


Figura 5 – Figura representando o gráfico de dispersão das medidas de temperatura para os modelos avaliados.

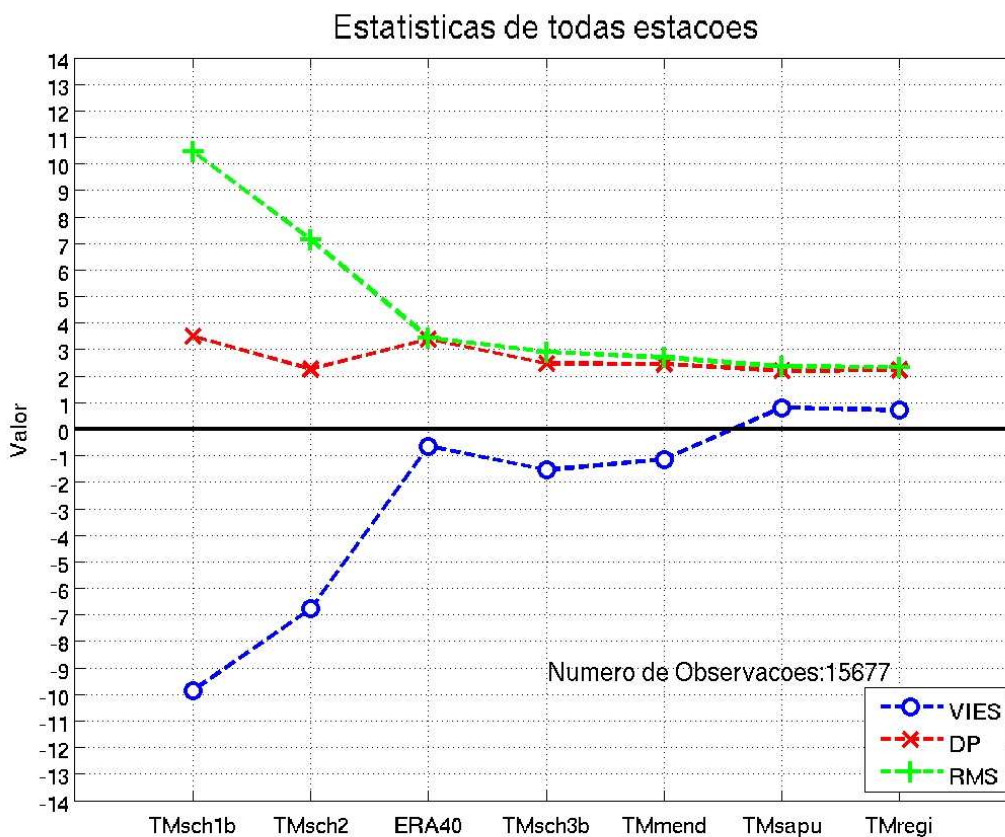


Figura 6 – Figura representando as estatísticas de VIÉS, RMS e Desvio Padrão dos modelos avaliados.

Analisando a Figura 6, podemos perceber que o modelo TMsch2 é tendencioso, pois a maioria de seus pontos de temperatura média estão abaixo da linha de máxima correlação, visto na Figura 5, bem como confirmado pelo alto viés da Figura 6. Assim sendo, o TMsch2 não é um dos melhores modelos para representar a real temperatura média troposférica. O modelo TMsch1b apresenta, praticamente, o mesmos valores de temperatura média para todas as estações, o que não é esperado. Desta forma, o TMsch1b não se apresenta confiável. O modelo TMschu3b apresenta uma boa precisão e um viés não muito alto, em comparação aos demais. O modelo TMmend é um modelo pouco tendencioso, com relativa baixa dispersão, tendo um desempenho semelhante ao TMsapu e TMregi, os quais se

apresentam como melhores, visto seu baixo viés, rms e desvio padrão. O modelo ERA40 é o modelo que apresenta o quase que ausência de VIÉS, isso demonstra que ele é pouco tendencioso, no entanto o valor do desvio padrão indicam que ele é o mais disperso, junto com o modelo TMsch2.

Todos estes modelos foram avaliados para várias estações e foi visto que os que obtiveram melhores desempenhos foram: TMregi e TMsapu, assim estes dois modelos destacam-se por ter um baixo viés, desvio padrão e rms.

## **5. Considerações finais e perspectivas futuras no desenvolvimento do trabalho.**

O presente trabalho apresenta uma lista de desenvolvimentos necessários à implementação da assimilação de dados de RO-GNSS e do IWV-GPS no modelo global do CPTEC. Diversos programas foram criados a fim de permitir a assimilação de perfis verticais de altura geopotencial e umidade globalmente distribuídos, bem como e valores integrados do IWV sobre o Brasil, no GPSAS. Também, procedimentos para comparação e avaliação dos modelos de temperatura média foram executados. Tais valores de TM são importantes na obtenção do IWV-GPS, dado que também apresenta um grande potencial de emprego na assimilação.

A maior parte das ações realizadas neste trabalho teve por objetivo fazer com que os dados de rádio ocultação e valores do IWV-GPS, acima mencionados, fossem assimilados pelo sistema de assimilação de dados PSAS. Com o sucesso das ações empregadas e melhoria da qualidade das previsões, através do uso de informações provenientes do GPS, novas medidas e trabalhos futuros devem ser realizados. Em virtude da substituição do PSAS pelo LETKF, vários outros trabalhos serão demandados, ilustrando novos procedimentos e processos, de forma a possibilitar o uso desse tipo de dado pelo novo sistema, na busca de maior eficiência e melhores resultados.

Dentre os possíveis trabalhos futuros, juntamente com o uso dos dados de RO-GPS no novo sistema de assimilação (LETKF), está a inclusão e operacionalização dos dados IWV-GPS nesse sistema. Para ajudar na construção de todos estes trabalhos, a ferramenta SVN (Subversion) poderá ser utilizada a fim de facilitar todo o processo de criação em conjunto de um novo programa e tornar mais acessível a todos que quiserem dar sua contribuição nos desenvolvimentos.

## REFERÊNCIAS BIBLIOGRÁFICAS

ANDREOLI, R. V.; SOUZA, R. F. A.; FERREIRA, S.F.S.; SAPUCCI, L. F.; HERDIES, D. L.; ARAVÉQUIA, J. A. **Inclusão das sondagens AIRS/AMSU no sistema global de assimilação/previsão de tempo do CPTEC/INPE: estudo de impacto.** Res. Bras. Meteorol., 2007, no prelo.

BEVIS, M. G., SUSINGER, S., HERRING, T., ROCKEN, C., ANTHES, R. A., WARE, R. H. **GPS Meteorology: Remote of Atmospheric Water Vapor Using the Global Positioning System.** *Journal of Geophysical Research*, Vol. 97, No. D14, Pages 15.787-15.801, October 20, 1992.

CINTRA, R. S. C. **Implementação do Sistema Estatístico de Assimilação de Dados em Espaço Físico para o Modelo Global do CPTEC.** Dissertação de Mestrado, 30-06-2004 (INPE-14223-TDI/1124) Instituto Nacional de Pesquisas Espaciais, São José dos Campos.2004.

\_\_\_\_\_. R. S. C.; LIMA, W. F.; ESPINOZA, E. S.; CAMPOS VELHO, H. F. **Inclusão de dados ATOVS no sistema de assimilação de dados GPSAS do CPTEC.** In: XII Simpósio Brasileiro de Sensoriamento Remoto, Goiânia, Brasil, 2005. Anais.

COHN, E.; DA SILVA, A.; GUO, J.; SIENKIEWICZ, M.; LAMICH, D. **Assessing the effects of data selection with the DAO Physical-space Statistical Analysis System.** *Monthly Weather Review*, 126:2913–2926, 1998.



CUCURULL, L.; NAVASCUES, B.; RUFFINI; G.; ELÓSEGUI, P.; RIUS, A; VILÀ, J.  
**The use of GPS to valuate NWP system: the HIRLAM model.** Journal  
Atmospheric and Oceanic Technology. V. 17, p. 773-787,2000.

DAVIS, J. L.; HERRING, T. A.; SHAPIRO I.; ROGERS, A. E.; ELGENED, G.  
**Geodesy by Interferometry: Effects of Atmospheric Modeling Errors on  
Estimates of Base Line Length.** Radio Sci., v. 20, p.1593-1607, 1985.

DUAN, J.; BEVIS, M.; FANG, P.; BOCK, Y.; CHISWELL, S.; BUSINGER, S.;  
ROCKEN, C.; SOLHEIM, F.; HOVE, T.; WARE, R.; MCCLUSK, S.; HERRING, T. A.;  
KING, R. W. **GPS meteorology: Direct Estimation of the absolute Value of  
Precipitable Water.** Journal of Applied Meteorology, v. 35, p.830-838, 1996.

EMARDSON, T. R. **Studies of atmospheric water vapor using the Global  
Positioning System.** School of Electrical and Computer Engineering Chalmers  
University of Technology, Göteborg, Sweden. Technical Report n. 339, 1998.

FJELDBO, G.; ESHLEMAN, V. R.; KLIORÉ, A. J. **The neutral atmosphere of  
Venus as studied with the Mariner V radio occultation experiments.**  
**Astronomical Journal**, V. 76, pag. 123- 140, March, 1971.

HOGG, D. C.; DECKER, M. T.; GUIRAUD, F. O.; GARNSHAW, K. B.; MERRITT, D.  
A.; MORAN, K. P.; SWEEZZY, W. B.; STRAUCH, R. G.; WESTWATER, E. R.;  
LITTLE, C. G. **Na automatic profile of the temperature, Wind and humidity in the  
troposphere** J. Climate Appl. Meteor., v.22, p.807-831,1983.

KALNAY, E., 2002. **Atmospheric Modeling, Data Assimilation, and Predictability.** Cambridge University Press, Cambridge. 364 p.

\_\_\_\_\_. E ; LIU, JJ 2008: **Estimating observation impact without adjoint model in an ensemble Kalman filter.** QUARTERLY JOURNAL OF THE ROYAL METEOROLOGICAL SOCIETY Volume: 134 Issue: 634 Pages: 1327-1335  
Published: 2008

KUO, Y. H., GUO, Y. R.; WESTWATER, E. R. **Assimilation of Precipitable Water Into Mesoscale Numerical Model.** Mon. Wea. Rev., v.121, p.1215-1238,1993

KURSINSKI, E. R.; HAJJ, G. A.; BERTIGER, W. I.; LEROY, S. S.; MEEHAN, T. K.; ROMANS, L. J.; SCHOFIELD, J. T.; McCLEESE, D. J.; MELBOURNE, W. G.; THORNTON, C. L.; YUNCK, T. P.; EYRE, J. R.; NAGATANI, R. N. **Initial results of radio occultation observations of Earth's atmosphere using the Global Positioning System,** Science , 271, 1107-1110, 1996.

\_\_\_\_\_. E.R.; HAJJ, G. A.; SCHOFIELD, J.T.; LINFIELD, R.P.; HARDY, K. R. **Observing Earth's Atmosphere With Radio Occultation Measurements Using the Global Positioning System.** Journal of Geophysical Research, 102, 23429-23465, 1997.

MAREL, H. **Exploitation of Ground Based GPS for Numerical Weather Prediction and climate applications in Europe.** Delf university of Tecnology, Department of Geodesy, Thijssesweg 11, 2629 JA Delf, The Netherlands, 2001.

MELFI, S. H.; WHITEMAN D.; FERRARE. **Observation of atmospheric fronts using Raman lidar moisture measurements.** J. Appl. Meteor.. V.28, p.789-806, 1989.

MENDES, V. B.; PRATES, G.; SANTOS, L.; LANGLEY, R. B. **An evaluation of the accuracy of models for the determination of the weighted mean temperature of the atmosphere.** Proceedings of In: IONGPS INTERNATIONAL TECHNICAL MEETING of Institute of Navigation. January 26-28, 2000, Pacific Hotel Disneyland, Anaheim, CA.

PACIONE, R.; SCIARRETTA, C.; FIONDA, E.; BORDONI, F. U.; FERRARA, R. **GPS and Ground-Based Microwave Radiometer PWV: A Case Study at Cagliari Astronomical Station.** In: IONGPS INTERNATIONAL TECHNICAL MEETING of Institute of Navigation., n. 14, 2001, Salk Lake City-Utah, 2001a.

REIGBER, C.; GENDT G.; TOMASSINI M. **Near real-time water vapor monitoring in a German GPS network and assimilation into weather forecast model.** In: IONGPS INTERNATIONAL TECHNICAL MEETING of Institute of Navigation, n 14 2001, Salk Lake City – Utah, 2001.

SAPUCCI, L. F; ARAVÉQUIA, J. A.; SOUZA, R. A. F.; FERREIRA, S. H. S.; HERDIES, D. L.; ANDREOLI, R. V.; MATTOS, J. G. F. **The inclusion of integrated water vapor estimates from AIRS/AMSU and SSM/I sensors into PSAS data assimilation system in the CPTEC.** In: The 15th International TOVS Study Conference, Maratea, 2006.

\_\_\_\_\_. L. F. **Estimativa do vapor d'água atmosférico e avaliação da modelagem do atraso zenital troposférico utilizando GPS.** 2001. 167 f. Dissertação (Mestrado em Ciências Cartográficas) - Faculdade de Ciências e Tecnologia, Universidade Estadual Paulista, Presidente Prudente.

\_\_\_\_\_. L. F. **Estimativas do IWV utilizando receptores GPS em bases terrestres no Brasil: Sinergia entre a Geodésia e a Meteorologia.** 2005. Tese (Doutorado em Ciências Cartográficas) - Faculdade de Ciências e Tecnologia, Universidade Estadual Paulista, Presidente Prudente.

SCHUELER, T.; HEIN G. W.; BIBERGER R, **A global analysis of the mean atmospheric temperature for GPS water vapor estimation.** In: IONGPS INTERNATIONAL THECHNICAL MEETING of Institute of Navigation. N 14 2001, Salk Lake City- Utah 2001

SMITH, W. L. **The retrieval of atmospheric profiles from VAS geostationary radiance observations.** , Atmos. Sci., v.40, p.2025-2035,1983.

TSUDA, T.; HEKI, K.; MIYAZAKI, S.; AONASHI, K.; HIRAHARA, K; NAKAMURA, H.; TOBITA, M.; KIMATA, F.; TABELI, T.; MATSUSHIMA, T.; KIMURA, F.; SATOMURA, M.; KATO, T.; NAITO, I. **GPS meteorology project of Japan - exploring frontiers of geodesy.** Research News Earth Planets Space, 50(10). 1998

WARE, R.; ALBER, C.; ROCKEN,C.; SOLHEIM, F. **Sensing integrated water vapor along GPS ray paths.** Geophys. Res. Lett., v.24, p.417-420, 1997.

YANG, X.; SASS, H.; ELGERED, G.; JOHNSON, J. M.; EMARDSON, T. R. **A comparison of precipitable water vapor estimates by a NWP simulation and GPS observations.** Journal of Applied Meteorology, v. 38,p. 941-956,1999.

## APÊNDICE A – Programa feito em Shell script para ajudar a converter arquivos do tipo NETCDF para ODS

Este Shell Script foi feito com o objetivo de ajudar a conversão dos dados NETCDF para ODS a ser realizada, o script tem a função de gerar um namelist contendo todos os nomes dos arquivos por range (00, 06, 12 e 18 horas). Cada range de hora tem uma quantidade diferente de arquivos referente a ele. Assim, o programa em fortran pode gerar os arquivos ODS para todos os ranges e com os conteúdos dos arquivos NETCDF dos quarto ranges.

Nome do programa: nc2ods\_temp.ksh

```
#!/bin/ksh
#help#
#####
# Script....: nc2ods_temp.ksh #
# Shell.....: ksh Shell #
# Objetivo...: Esse script gera arquivos ods a partir de arquivos com #
# valores do IWV obtidos de arquivos tropIGS e estacoes met #
# #
# Detalhes...: Essa versao foi feita para pos processamento simulando uma #
# rodada operacional. Para a versao operacional alguns #
# ajustem devem ser feitos na selacao das secoes de dados #
# utilizadas em cada rodada, na tomada do tempo de referencia#
# no sistema. #
# #
# 2-Nessa versao as letras da secao estao como fora definidas #
# anteriormente no LGE, as quais foram modificadas em meados #
# 2009. Essa modificacao deve ser considerada na #
# operacionalizacao desse processo. #
# #
# Sintaxe...: iwv2ods.ksh YYYYMMDDHH #
# First: help #
# First: oper : Script ja preparado para rodar operacionalmente #
# First: LABELI : data inicial do processamento #
# Second : LABELF : data final do processamento #
# #
# LABELx : yyyyymmddhh #
# yyyy = four digit year #
# mm = two digit month #
# dd = two digit day #
# hh = two digit hour #
# #
# Autor : Luiz Sapucci (lsapucci@cptec.inpe.br) #
# Alterado por: Dérek Schubert (derek.schubert@cptec.inpe.com) #
# Flavio S. de Cerqueira (flavio.santos@cptec.inpe.br ) #
#####
```

```

#help#
if [ "${1}" = "help" ]
then
  cat < ${0} | sed -n '/^#help#/,/^#help#/p'
  exit 0
fi
if [ "${1}" = "oper" ]
then
  DATA=`date +%Y%m%d`
  HM=`date +%H%M`
  if [ $HM -ge 2100 ]; then
    HH=18
  else
    if [ $HM -ge 1500 ]; then
      HH=12
    else
      if [ $HM -ge 900 ]; then
        HH=06
      else
        HH=00
      fi
    fi
  fi
  DATA=${DATA}${HH}
  DATAF=${DATA}
else
  ##### Teste dos argumentos validos #####
  if [ ${1} -lt 1900010100 ]
  then
    echo "LABELI nao esta setado corretamente (yyyymmddhh)"
    exit 1
  fi
  if [ ${2} -lt 1900010100 ]
  then
    echo "LABELF nao esta setado corretamente (yyyymmddhh)"
    exit 1
  fi
  # Setando as variaveis de data
  DATA=$1
  DATAF=$2
fi
RUNTM=`date +%Y%m%d.%H.%M`
echo
"%%%%%%%%%%"
echo "Processamento para converter arquivos de dados Geop em ods"
echo "Data inicial: "${DATA}
echo "data final: "${DATAF}
echo
"%%%%%%%%%%"
echo "Data da rodada:"${RUNTM}
echo
"%%%%%%%%%%"
echo " "
##### Diretorios dos dados #####
DIRRUN=/home/flaviosc/gera_ods_ROGNSS/run/

```

```

DIRIN=/mpp/flaviosc/pathin/dados_jul09/
DIRINP=/home/lsapucci/gera_ods_GPS/dataIWV
DIROUT=/mpp/flaviosc/ods_out/
DIRBIN=/home/flaviosc/gera_ods_ROGNSS/source/
cd $DIRRUN
set imp=0
errfile=${DIRRUN}/err_file/nc2ods_${RUNTM}.err
##### Loop principal do período #####
while [ ${DATA} -le ${DATAF} ]
do
    echo
    logfile=${DIRRUN}/log_file/nc2ods_${DATA}_${RUNTM}.log
    echo
    "%%%%%%%%%%"> ${logfile}
    echo "%%%%%%%%%% Processando a data: "${DATA} "%%%%%%%%%%">> ${logfile}
    echo "%%%%%%%%%% Processando a data: "${DATA} "%%%%%%%%%%"
    echo
    "%%%%%%%%%%">> ${logfile}
    echo "Arquivo de log dessa rodada: " ${logfile}
    ##### Separando a Data #####
    YY=`echo ${DATA} | cut -c 1-4`
    AA=`echo ${DATA} | cut -c 3-4`
    MM=`echo ${DATA} | cut -c 5-6`
    DD=`echo ${DATA} | cut -c 7-8`
    HH=`echo ${DATA} | cut -c 9-10`
    cd ${DIRIN}
    ##### Gerando a separação dos arquivos por hora sinótica #####
    diajuliano=`/home/flaviosc/bin/converToJulian.ksh $YY $MM $DD`
    if [ $diajuliano -lt 100 ];then
        diajuliano=0$diajuliano
        if [ $diajuliano -lt 10 ]; then
            diajuliano=0$diajuliano
        fi
    fi
    if [ $HH == '00' ]; then
        dia=`expr $diajuliano - 1`
        if [ $dia -lt 100 ]
        then
            dia=0$dia
            if [ $dia -lt 10 ]
            then
                dia=0$dia
            fi
        fi
        if [ $dia -ne 0 ];then
            for h in $(seq 21 23); do
                for m in $(seq 0 1 59);do
                    if [ $m -lt 10 ]
                    then
                        m=0$m
                    fi
                    for s in $(seq 1 6); do
                        NI=`ls *C00${s}.${YY}.${dia}.${h}.${m}.G* 2> /dev/null | wc -l`
                        if [ $NI -gt 1 ];then
                            for g in $(seq 0 32); do
                                if [ $g -lt 10 ]

```



```

        then
            g=0$g
        fi
    if [ -e ${DIRIN}*C00${s}.${YY}.${dia}.${h}.${m}.G${g}* ]
    then
        arq=`ls *C00${s}.${YY}.${dia}.${h}.${m}.G${g}*`
        echo $arq
        echo $arq >> ${DIRIN}range.txt
    fi
done
else
    if [ -e ${DIRIN}*C00${s}.${YY}.${dia}.${h}.${m}.G* ]
    then
        arq=`ls *C00${s}.${YY}.${dia}.${h}.${m}.G*`
        echo $arq
        echo $arq >> ${DIRIN}range.txt
    fi
done
done
done
else
    dianterior=`~/bin/caldate.3.1.2 $DATA - 1d 'yyyymmddhh'`
    AYY=`echo ${dianterior} | cut -c 1-4`
    AMM=`echo ${dianterior} | cut -c 5-6`
    ADD=`echo ${dianterior} | cut -c 7-8`
    diajl=`/home/flaviosc/bin/converToJulian.ksh $AYY $AMM $ADD`
    for h in $(seq 21 23); do
        for m in $(seq 0 1 59);do
            if [ $m -lt 10 ]
            then
                m=0$m
            fi
            for s in $(seq 1 6); do
                NI=`ls *C00${s}.${AYY}.${diajl}.${h}.${m}.G* 2> /dev/null | wc -l`
                if [ $NI -gt 1 ];then
                    for g in $(seq 0 32); do
                        if [ $g -lt 10 ]
                        then
                            g=0$g
                        fi
                    if [ -e ${DIRIN}*C00${s}.${AYY}.${diajl}.${h}.${m}.G${g}* ]
                    then
                        arq=`ls *C00${s}.${AYY}.${diajl}.${h}.${m}.G${g}*`
                        echo $arq
                        echo $arq >> ${DIRIN}range.txt
                    fi
                done
            else
                if [ -e ${DIRIN}*C00${s}.${AYY}.${diajl}.${h}.${m}.G* ]
                then
                    arq=`ls *C00${s}.${AYY}.${diajl}.${h}.${m}.G*`
                    echo $arq
                    echo $arq >> ${DIRIN}range.txt
                fi
            fi
        done
    done
done
done

```

```

done
fi
for h in $(seq 0 2); do
    h=0$h
for m in $(seq 0 59); do
    if [ $m -lt 10 ];then
        m=0$m
    fi
    for s in $(seq 1 6); do
        NI=`ls *C00${s}.${YY}.${diajuliano}.${h}.${m}.G* 2> /dev/null | wc -l`
        if [ $NI -gt 1 ];then
            for g in $(seq 0 32); do
                if [ $g -lt 10 ]
                then
                    g=0$g
                fi
                if [ -e ${DIRIN}*C00${s}.${YY}.${diajuliano}.${h}.${m}.G${g}* ]
                then
                    arq=`ls *C00${s}.${YY}.${diajuliano}.${h}.${m}.G${g}*`
                    echo $arq
                    echo $arq >> ${DIRIN}range.txt
                fi
            done
        else
            if [ -e ${DIRIN}*C00${s}.${YY}.${diajuliano}.${h}.${m}.G* ]
            then
                arq=`ls *C00${s}.${YY}.${diajuliano}.${h}.${m}.G*`
                echo $arq
                echo $arq >> ${DIRIN}range.txt
            fi
        fi
    done
done
done
done
else
if [ $HH == '06' ]; then
for h in $(seq 3 8); do
    h=0$h
for m in $(seq 0 59);do
    if [ $m -lt 10 ];then
        m=0$m
    fi
    for s in $(seq 1 6); do
        NI=`ls *C00${s}.${YY}.${diajuliano}.${h}.${m}.G* 2> /dev/null | wc -l`
        if [ $NI -gt 1 ];then
            for g in $(seq 0 32); do
                if [ $g -lt 10 ]
                then
                    g=0$g
                fi
                if [ -e ${DIRIN}*C00${s}.${YY}.${diajuliano}.${h}.${m}.G${g}* ]
                then
                    arq=`ls *C00${s}.${YY}.${diajuliano}.${h}.${m}.G${g}*`
                    echo $arq
                    echo $arq >> ${DIRIN}range.txt
                fi
            done
        else
            fi
        fi
    done
done
else

```

```

        if [ -e ${DIRIN}*C00${s}.${YY}.${diajuliano}.${h}.${m}.G* ]
        then
            arq=`ls *C00${s}.${YY}.${diajuliano}.${h}.${m}.G*`
            echo $arq
            echo $arq >> ${DIRIN}range.txt
        fi
    fi
done
done
else
    if [ $HH == '12' ]; then
        for h in $(seq 9 14); do
            if [ $h -lt 10 ];then
                h=0$h
            fi
            for m in $(seq 0 59);do
                if [ $m -lt 10 ];then
                    m=0$m
                fi
                for s in $(seq 1 6); do
                    NI=`ls *C00${s}.${YY}.${diajuliano}.${h}.${m}.G* 2> /dev/null | wc -l`
                    if [ $NI -gt 1 ];then
                        for g in $(seq 0 32); do
                            if [ $g -lt 10 ]
                            then
                                g=0$g
                            fi
                            if [ -e ${DIRIN}*C00${s}.${YY}.${diajuliano}.${h}.${m}.G${g}* ]
                            then
                                arq=`ls *C00${s}.${YY}.${diajuliano}.${h}.${m}.G${g}*`
                                echo $arq
                                echo $arq >> ${DIRIN}range.txt
                            fi
                        done
                    else
                        if [ -e ${DIRIN}*C00${s}.${YY}.${diajuliano}.${h}.${m}.G* ]
                        then
                            arq=`ls *C00${s}.${YY}.${diajuliano}.${h}.${m}.G*`
                            echo $arq
                            echo $arq >> ${DIRIN}range.txt
                        fi
                    fi
                done
            done
        done
    else
        if [ $HH == '18' ]; then
            for h in $(seq 15 20); do
                for m in $(seq 0 1 59);do
                    if [ $m -lt 10 ];then
                        m=0$m
                    fi
                    for s in $(seq 1 6); do
                        NI=`ls *C00${s}.${YY}.${diajuliano}.${h}.${m}.G* 2> /dev/null | wc -l`
                        if [ $NI -gt 1 ];then
                            for g in $(seq 0 32); do
                                if [ $g -lt 10 ]

```



## APÊNDICE B – Programa feito em Fortran que serve para fazer a conversão dos valores dos arquivos do tipo NETCDF para o tipo ODS

Este programa em Fortran foi feito para poder auxiliar na conversão dos dados do tipo NETCDF para o tipo ODS, assim, o modelo utilizado no CPTEC pode assimilar este novo dado. Neste caso o Fortran utilizará o namelist gerado pelo shell que se encontra no apêndice A para extrair as informações das variáveis que se encontram dentro dos arquivos NETCDF. Para isso foi usada uma biblioteca do Fortran que dá acesso aos arquivos NETCDF, depois de atribuir os valores para as variáveis que o Fortran irá utilizar o Fortran faz uma conversão de hora da observação em minutos. Depois de mais algumas correções em algumas variáveis, o programa faz a conversão de todos os dados que se encontravam dentro das variáveis para o formato do arquivo CDL. Que posteriormente será convertido em ODS.

**Nome do programa: nc2cdl\_temp.f90**

```

! Last change:
!CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
!   PROGRAMA PARA CONVERTER VALORES DO ROGNSS.nc EM CDL      C
!   OBJETIVANDO CRIAR ARQUIVOS ODS                          C
!   VERSAO 0.0                                              C
!   DETALHES DESSA VERSAO:                                  C
! 1- Os arquivos para cada horário são definidos por um script que C
!   gera uma lista dos arquivos gegeDDDx.09iww presentes e devem C
!   ser utilizados para aplicações operacionais.            C
! 2- Nessa versão as letras da seção estão como foram definidas C
!   anteriormente no LGE, as quais foram modificadas em meados de C
!   2009. Essa modificação deve ser considerada na          C
!   operacionalização desse processo.                        C
!CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
!   Flavio S. de Cerqueira                                  C
!   Dérek Schubert                                         C
!   Adaptado de Luiz Fernando Sapucci                      C
!   22/07/2010                                             C
!CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
program NC2CDL
USE M_CD
use netcdf
implicit none
!#####Declaração de variáveis utilizadas para ler os dados no arquivo NetCDF #####
integer :: ncdl,status,MSLVarId,TempVarId,LatVarId,LonVarId,PvVarId,tempo2
integer :: PresVarId, RefVarId, Ref_obsVarId,kt,ktq,nOBS,n,kx
integer, dimension(400) :: dimIDs
integer, dimension(19) :: NIVEL
real, dimension(400) :: MSL,Temp,Lat,Lon

```

```

real, dimension(400) :: Pv,Pres,Ref,Ref_obs,GEOP,q
real, dimension(50000) :: VGEOp,Vq,VLat,VLon,VPres,Vks,T
real :: RT, GAMA,GLat,Glou
character*2 :: horatime,minutotime
character*6 :: tempo
!##### Declaracao de variaveis utilizadas no m_ods #####
type(ods_vect) :: Obs ! Estrutura que contem as observacoes convertidas
integer :: nOBS_max ! Numero maximo de dados na estrutura O
integer :: erro
integer :: II,NOBSER
real*8 :: date1,date2,dj,mdate,dateobs
character(len=8)::odsdate
character(len=2)::odstime
! ##### VARIABEIS DECLARADAS PARA A UTILIZACAO Na ROTINA #####
INTEGER
MIN,I,J,K,L,Nin,Nout,aqui,ANO,MES,DIA,HORA,difmin,horacentral,odstimeInt,horaObs,minutoObs,m
CHARACTER*12 YMDHM
CHARACTER*10 DATAref, anomesdia
CHARACTER*150 PATHMET,PATHIN,PATHOUT,NAMEPATHOUT,NAMEOUT
CHARACTER*150 NAMEPATHIN,NAMEIN,TEXTO,PATHEST
! ##### ABRINDO ARQUIVO DE GERENCIA (NAMELIST) #####
OPEN(2,FILE='/home/flaviosc/gera_ods_ROGNSS/run/nc2cdl.nml',STATUS='OLD')
! ##### LENDO A REFERENCIA DE TEMPO NO ARQUIVO DE GERENCIA #####
9 READ(2,'(A)')DATAref
IF(DATAref(1:1).EQ. '#')GOTO 9
odsdate=DATAref(1:8)
odstime=DATAref(9:10)
! ##### LENDO NIVEIS PADROES #####
62 READ(2,*,END=770)(NIVEL(N),N=1,19)
63 FORMAT(18(I4))
PRINT*,(NIVEL(N),N=1,8)
! ##### LENDO PATH DOS ARQUIVOS DE SAIDA CDL #####
64 READ(2,'(A)',END=770)PATHOUT
IF(PATHOUT(1:1).EQ. '#')GOTO 64
Nout=0
DO I=1,150
IF (PATHOUT(I:I).EQ. '/') Nout=I
END DO
! ##### LENDO PATH DOS ARQUIVOS DE ENTRADA NET-CDF #####
67 READ(2,'(A)',END=770)PATHIN
IF(PATHIN(1:1).EQ. '#')GOTO 67
Nin=0
DO J=1,150
IF (PATHIN(J:J).EQ. '/') Nin=J
END DO
WRITE(*,*) '
WRITE(*,*)#####'
WRITE(*,*)# PROGRAMA PARA CONVERTER NET-CDF EM CDL #
WRITE(*,*)#####'
PRINT*, 'Endereco dos arquivo de saida'
WRITE(*,*)PATHOut: ',PATHOut(1:Nout)
PRINT*, 'Endereco dos arquivo de dados net-cdf disponivel'
WRITE(*,*)PATHIn: ',PATHIn(1:Nin)
WRITE(*,*)
!##### Loop principal com a leitura do arquivo NET-CDF e conversao para o CDL e armazenamento no
arquivo de saida #####
NOBSER=0 ! Zerando o contador no numero de observacoes por range
n=0 !zerando o contador do kx(quantidade de arquivos por range)

```

```

! ##### LENDO O ARQUIVO DE DADOS A SER PROCESSADO #####
70  READ(2,'(A)',END=700)NAMEIN
    IF(NAMEIN(1:1).EQ.#)GOTO 70
    NAMEPATHIN=PATHIN(1:Nin)//NAMEIN(1:43)
!##### Lendo as variaveis no arqvo entrada nc e pondo-as em vetores #####
status = nf90_open(NAMEPATHIN, nf90_NoWrite, ncid)
    status = nf90_inq_varid(ncid, "MSL_alt", MSLVarId)
    status = nf90_inq_varid(ncid, "Temp", TempVarId)
    status = nf90_inq_varid(ncid, "Lat", LatVarId)
    status = nf90_inq_varid(ncid, "Lon", LonVarId)
    status = nf90_inq_varid(ncid, "Vp", PvVarId)
    status = nf90_inq_varid(ncid, "Pres", PresVarId)
    status = nf90_inq_varid(ncid, "Ref", RefVarId)
    status = nf90_inq_varid(ncid, "Ref_obs", Ref_obsVarId)
    status = nf90_get_var(ncid, MSLVarId, MSL)
    status = nf90_get_var(ncid, TempVarId, Temp)
    status = nf90_get_var(ncid, LatVarId, Lat)
    status = nf90_get_var(ncid, LonVarId, Lon)
    status = nf90_get_var(ncid, PvVarId, Pv)
    status = nf90_get_var(ncid, PresVarId, Pres)
    status = nf90_get_var(ncid, RefVarId, Ref)
    status = nf90_get_var(ncid, Ref_obsVarId, Ref_obs)
    status = nf90_get_att(ncid, nf90_global, "lat", GLat)
    status = nf90_get_att(ncid, nf90_global, "lon", GLon)
!##### Transformação de Altura em Geop e de Pressão de Vapor em q #####
    n=n+1
    kx= 177
    kt=6
    ktq=7
    anomesdia=DATAref(1:8)
    horatime=NAMEIN(22:23)
    minutotime=NAMEIN(25:26)
    tempo=horatime//minutotime//'00'
    read (tempo,'(I6)') tempo2
    read (horatime,'(I2)') horaObs
    read (minutotime,'(I2)') minutoObs
    read (odstime,'(I2)') odstimeInt
!!! CONVERSÃO DA HORA DA OBS EM MINUTOS POSITIVOS OU NEGATIVOS EM TORNO DA HORA
SINOTICA,PARA SER UTILIZADO NO CDL
    horacentral=odstimeInt
    if(horacentral .eq. 0 .and. horaObs.ge.21)then
        horaObs=horaObs-24
        difmin = ((horaObs-horacentral)*60)+(minutoObs)
    elseif(horacentral .eq. 0 .and. horaObs.le.2)then
        difmin = ((horaObs-horacentral)*60)+(minutoObs)
    else
        difmin = ((horaObs)*60)+(minutoObs)
    end if
    DO nOBS=1,400
!! SUBSTITUINDO VALORES AUSENTES DE LAT(-999) POR VALORES GLOBAIS DE REFERENCIA (GLat)
        IF (Lat(nOBS).LE.-91) THEN
            Lat(nOBS)= GLat
        END IF
        IF (Lon(nOBS).LE.-181) THEN
            Lon(nOBS)= GLon
        END IF
!! CONVERSÃO DA ALTURA GEOMETRICA AMSL EM ALTURA GEOPOTENCIAL E DA PRESSÃO DE VAPOR

```

```

EM RAZAO DE MISTURA
  RT= 6378137/(1.006803-(0.006706*SIND(Lat(nOBS))**2))
  GAMA=(9.7803253359*(1+(0.001932*SIND(Lat(nOBS))**2)))/((1-(0.006694*SIND(Lat(nOBS))**2))**0.5)

  GEOP(nOBS)= (GAMA*(MSL(nOBS)*1000)*RT)/(9.80665*(RT+(MSL(nOBS)*1000)))
  if(Pres(nOBS).gt.0)then
    q(nOBS)= (622*Pv(nOBS))/(Pres(nOBS)-Pv(nOBS))
  end if
END DO
!CONTROLE DE QUALIDADE QC1 - RETIRA DADOS ABAIXO DE NÍVEIS ONDE DN/DZ < -50 , FAZENDO P=-
999
  DO m= 1, 44
    if (Ref_obs(11)-Ref_obs(1).lt.-50.and.m.le.11) then
      Pres(m)= -99
    end if

    if (Ref_obs(22)-Ref_obs(12).lt.-50.and.m.le.22) then
      Pres(m)= -99
    end if

    if (Ref_obs(33)-Ref_obs(23).lt.-50.and.m.le.33) then
      Pres(m)= -99
    end if
    if (Ref_obs(44)-Ref_obs(34).lt.-50.and.m.le.44) then
      Pres(m)= -99
    end if
  END DO
!!!!#### Interpolação dos níveis observados para níveis padrões (925,850,700,500,etc..) ####
  DO L= 1, 19
    DO I= 1, 400
      IF (Pres(I).NE.NIVEL(L).AND.Pres(I+1).NE.NIVEL(L).AND.(I+1).ne.401) then
        IF (Pres(I).GT.NIVEL(L).AND.Pres(I+1).LT.NIVEL(L)) THEN
          NOBSER=NOBSER+1
          VGEOP(NOBSER) = GEOP(I+1) - ((Pres(I+1)- NIVEL(L))*(GEOP(I+1)-
GEOP(I))/(Pres(I+1)-Pres(I)))
          Vq(NOBSER) = q(I+1) - ((Pres(I+1)- NIVEL(L))*(q(I+1)-q(I))/(Pres(I+1)-Pres(I)))
          VLat(NOBSER) = (Lat(I)+ Lat(I+1))/2
          VLon(NOBSER)= (Lon(I)+ Lon(I+1))/2
          VPres(NOBSER)= NIVEL(L)
          Vks(NOBSER)= n
          T(NOBSER)= difmin
          CONTINUE
        END IF
        ELSEIF (Pres(I).EQ.NIVEL(L)) THEN
          NOBSER=NOBSER+1
          VGEOP(NOBSER)= GEOP(I)
          Vq(NOBSER) = q(I)
          VLat(NOBSER)= Lat(I)
          VLon(NOBSER)= Lon(I)
          VPres(NOBSER)= NIVEL(L)
          Vks(NOBSER)= n
          T(NOBSER)= difmin
          CONTINUE
        ELSEIF (I.NE.400) THEN
          NOBSER=NOBSER+1
          VGEOP(NOBSER)= GEOP(I+1)
          Vq(NOBSER) = q(I+1)
          VLat(NOBSER)= Lat(I+1)

```



```

                VLon(NOBSER)= Lon(I+1)
                VPres(NOBSER)= NIVEL(L)
                Vks(NOBSER)= n
                T(NOBSER)= difmin
                CONTINUE
            END IF
        END DO
    END DO
!ccccccC LENDO O PROXIMO DADO DE ENTRADA
!##### FECHANDO OS ARQUIVOS #####
        CLOSE (4)
        CLOSE (5)
        GOTO 70
700 CLOSE(2)
! ##### GERANDO CDL PARA O GEOP #####
!##### Iniciando variaveis para gravacao do ods #####
        nobs_max=500000
        local_odstables="/home/flaviosc/gera_ods_ROGNSS/source/odstables/"
        call OdS_init(Obs,nObs_max,erro) ! Aloca e inicializa a estrutura de dados O (Observacoes em ODS)
        print*, 'erro', erro
!#####(Por via das duvidas, zerando estas parte da estrutura #####
        OBS%data%kt(:)=0
        OBS%data%qcexcl(:)=0
        OBS%data%qchist(:)=0
        OBS%data%xm(:)=obs_missing
        II=0
        DO J=1,NOBSER
            II=II+1
            OBS%DATA%lev(II)=VPres(J)
            OBS%DATA%lat(II)=VLat(J)
            OBS%DATA%lon(II)=VLon(J)
            OBS%DATA%kt(II)=kt
            OBS%DATA%kx(II)=kx
            OBS%DATA%ks(II)=Vks(J)
            OBS%DATA%obs(II)=VGEOPI(J)
            OBS%DATA%time(II)=T(J)
        END DO
! ##### GERANDO O ARQUIVO CDL #####
        nOBS_max=II
        print*, '#####'
        print*, 'Numero de observacao: ', II
        print*, '#####'
        print*, ''
        NAMEPATHOUT=PATHout(1:Nout)//'ROGNSS_GEOPI_ods.t'
        NAMEPATHOUT=NAMEPATHOUT(1:Nout+17)//'DATAref'
        IF (II .NE. 0)THEN
            call put_CDPI(15,NAMEPATHOUT,OBS,nOBS_max,odsdate,odstime)
        else
            print*, 'Observacoes ausentes p/ esse horario: cdl nao gerado'
        end if
! ##### FINAL PARA O DADO GEOP #####

! ##### GERANDO CDL PARA A UMIDADE ESPECIFICA(Q) #####
! Iniciando variaveis para gravacao do ods
        nobs_max=500000
        call OdS_init(Obs,nObs_max,erro) ! Aloca e inicializa a estrutura de dados O (Observacoes em ODS)
        print*, 'erro', erro
!#####(Por via das duvidas, zerando estas parte da estrutura #####

```

```

OBS%data%kt(:)=0
OBS%data%qcexcl(:)=0
OBS%data%qchist(:)=0
OBS%data%xm(:)=obs_missing
!}
  II=0
DO K=1,NOBSER
  II=II+1
  OBS%DATA%lev(II)=VPres(K)
  OBS%DATA%lat(II)=VLat(K)
  OBS%DATA%lon(II)=VLon(K)
  OBS%DATA%kt(II)=ktq
  OBS%DATA%kx(II)=kx
  OBS%DATA%ks(II)=Vks(K)
  OBS%DATA%obs(II)=Vq(K)
  OBS%DATA%time(II)=T(K)
END DO
!#####GERANDO O ARQUIVO CDL#####
  nOBS_max=II
  print*,'#####'
  print*,'Numero de observacao: ',II
  print*,'#####'
  print*,''
  NAMEPATHOUT=PATHout(1:Nout)//'ROGNSS_q_ods.t'
  NAMEPATHOUT=NAMEPATHOUT(1:Nout+14)//'DATAref'
  IF (II .NE. 0)THEN
    call put_CD_L(15,NAMEPATHOUT,OBS,nOBS_max,odsdate,odstime)
  else
    print*,'Observacoes ausentes p/ esse horario: cdL nao gerado'
  end if
! ##### FINAL PARA A UMIDADE Q #####
  GOTO 800
! PROCEDIMENTOS FINAIS
770 PRINT*,'Erro na leitura das informacoes no arquivo nml'
  STOP
780 WRITE(*,*)'Erro na leitura da tabela das estacoes GPS'
  STOP
800 STOP
END

```

## APÊNDICE C – Rotina feita em Shell Script em para criar o namelist e também para executar o Fortran.

O programa em Shell Script foi feito para criar um namelist contendo o caminho da saída dos arquivos e o nome de todos os arquivos .TXT que foram criados, assim, quando o programa em Fortran for processar os dados para fazer os cálculos de Vies, RMS e de desvio padrão ele precisará de um arquivo que indique a ele onde esses dados estão e qual é seu nome. Além de gerar o namelist o Shell também converte os arquivos que se encontram no formato ODS para o formato TXT, isso é necessário porque não se consegue visualizar os dados que estão dentro do ODS.

**Nome do Programa: omf\_txt.ksh**

```
#!/bin/ksh

DIRIN=/home/promet/flaviosc/Transf/
DIROUT=/home/promet/flaviosc/anaOMF_txt/
PATH=/rede/assimdados/psas/rita/AIRSreana/gera_ods_airs/src_f/
cd $DIRIN

datai=2009010100
dataf=2009013118

rm -rf /home/flaviosc/namelist_omf.txt

cat $DIROUT > /home/flaviosc/namelist_omf.txt

while [ $datai -le $dataf ]; do
    if [ -e ${DIRIN}GANLCPT${datai}S.obs.T213L42.ods ]
    then
        echo 'GANLCPT${datai}S.obs.T213L42.ods'
        ${path}readods.x -i GANLCPT${datai}S.obs.T213L42.ods > ${DIROUT}${datai}.txt
        cat ${datai}.txt >> /home/flaviosc/namelist_omf.txt
    fi
    datai=`/home/derek/bin/inctime $datai + 6 hour 'yyyymmddhh'`
done
# rodando o fortran para calcular as estatísticas.
./omf.exe
```

## APÊNDICE D – Namelist que será lido pelo programa em Fortran na atividade

### 3.2

Este apêndice contém um exemplo do namelist que será lido e usado pelo programa em Fortran para obter os nomes dos arquivos que ele irá usar para calcular as estatísticas.

**Nome do Namelist:** namelist\_omf.txt

```
/home/promet/flaviosc/anaOMF_txt/
```

```
2009010100.txt
```

```
2009010500.txt
```

```
2009010900.txt
```

```
2009011300.txt
```

```
2009011700.txt
```

```
2009012106.txt
```

```
2009012506.txt
```

```
2009012912.txt
```

```
2009010106.txt
```

```
2009010506.txt
```

```
2009010906.txt
```

```
2009011306.txt
```

```
2009011706.txt
```

```
2009012112.txt
```

```
2009012512.txt
```

```
2009012918.txt
```

```
2009010112.txt
```

```
2009010512.txt
```

```
2009010912.txt
```

```
2009011312.txt
```

```
2009011712.txt
```

```
2009012118.txt
```

```
2009012518.txt
```

```
2009013000.txt
```

```
2009010118.txt
```

```
2009010518.txt
```

```
2009010918.txt
```

```
2009011318.txt
```

```
2009011718.txt
```

```
.
```

```
.
```

```
.
```

```
.
```

## APÊNDICE E – Código em Fortran para fazer o cálculo das estatísticas

Este apêndice contém a rotina escrita em Fortran que serve para calcular as estatísticas dos dados ROGNSS. Inicialmente o programa lê os dados que foram obtidos através da conversão dos arquivos ODS para TXT feita anteriormente, depois são feitos os cálculos de Vies, RMS e Desvio padrão. Após os dados terem sido calculados eles foram impressos em arquivos, cada variável com o seu arquivo, para mostrar os resultados ao usuário.

**Nome do programa: omf.f90**

```
program erroOMF
implicit none
! PROGRAMA PARA LER OMF E CALCULAR VALOR MÉDIO PARA CADA NÍVEL
integer :: i,j,k,n,m,o,p,q,x,y,z,w,nivel,KX,KT,QCex,QCh,LEVEL,n20,n60,n90,t20,t60,t90
integer, dimension(925) :: COUNT1,COUNT2
real, dimension(925) ::
GEOPSOMA,qSOMA,OMFGEOP,OMFq,OMFGEOPQ,GEOPSOMAQ,OMFQ_q,qSOMAQ
real, dimension(925) :: OMFGEOPQ20,GEOPSOMAQ20,GEOPSOMA20,OMFGEOP20
real, dimension(925) :: OMFGEOPQ60,GEOPSOMAQ60,GEOPSOMA60,OMFGEOP60
real, dimension(925) :: OMFGEOPQ90,GEOPSOMAQ90,GEOPSOMA90,OMFGEOP90
real, dimension(925) :: SIGMAGEOP,DESVIOGEOP,SIGMAGEOP20,DESVIOGEOP20
real, dimension(925) :: SIGMAGEOP60,DESVIOGEOP60,SIGMAGEOP90,DESVIOGEOP90
real, dimension(925) :: OMFq_Q20,qSOMAQ20,qSOMA20,OMFq20
real, dimension(925) :: OMFq_Q60,qSOMAQ60,qSOMA60,OMFq60
real, dimension(925) :: OMFq_Q90,qSOMAQ90,qSOMA90,OMFq90
real, dimension(925) :: SIGMAq,DESVIOq,SIGMAq20,DESVIOq20
real, dimension(925) :: SIGMAq60,DESVIOq60,SIGMAq90,DESVIOq90

real :: LAT,OMF
real, dimension(100000,965) :: MGEOP,Mq
real, dimension(50000,925) :: MGEOP20,MGEOP60,MGEOP90,Mq20,Mq60,Mq90
character*60 :: ARQUIVO, PATHARQUIVO, PATH
DO I=1,925
COUNT1(I)=1
COUNT2(I)=1
END DO
! ABRINDO ARQUIVO DE GERENCIA
OPEN(1,FILE='/home/flaviosc/namelist_omf.txt',STATUS='OLD')
! LENDO O NOME DO ARQUIVO TXT A SER ABERTO
READ(1,'(A33)')PATH
10 READ(1,'(A14)',END=100)ARQUIVO
PATHARQUIVO=PATH(1:33)//ARQUIVO(1:14)
! ABRINDO O ARQUIVO TXT E LENDO AS LINHAS
OPEN(2,FILE=PATHARQUIVO,STATUS='UNKNOWN')
20 READ(2,30,ERR=20,END=40)LAT,LEVEL,KX,KT,OMF,QCex,QCh
30 FORMAT (7X,F10.2,10X,I7,19X,I8,I8,F10.2,10X,I8,I8)
! MONTANDO MATRIZES GEOP E UMIDADE PARA CADA NÍVEL, CONTENDO OMF, LAT, QCX E QCH
```

```

IF(KX.EQ.177.AND.KT.EQ.6)THEN
  n=COUNT1(LEVEL-1)
  MGEOP(n,LEVEL)=OMF
  MGEOP(n,LEVEL+4)=LAT
  MGEOP(n,LEVEL+13)=QCex
  MGEOP(n,LEVEL+39)=QCh
  COUNT1(LEVEL-1)=COUNT1(LEVEL-1)+1
! print *, LEVEL,OMF,LAT,QCex,QCh,
ELSEIF(KX.EQ.177.AND.KT.EQ.7)THEN
  n=COUNT2(LEVEL-1)
  Mq(n,LEVEL)=OMF
  Mq(n,LEVEL+4)=LAT
  Mq(n,LEVEL+13)=QCex
  Mq(n,LEVEL+39)=QCh
  COUNT2(LEVEL-1)=COUNT2(LEVEL-1)+1
END IF
GOTO 20
40 CLOSE (2)
GOTO 10
100 CLOSE (1)
!CALCULANDO OS VALORES MÉDIOS E OS DESVIOS DE OMF
!CRIANDO OS ARQUIVOS DE SAÍDA DAS ESTATÍSTICAS
OPEN(4,FILE='ErroOMF_Geop.txt',STATUS='unknown')
OPEN(5,FILE='ErroOMF_q.txt',STATUS='unknown')
!CRIANDO CABEÇALHO DOS ARQUIVOS DE SAÍDA
WRITE(4,122)'Nivel','S','S(0-20)','S(20-50)','S(50-
90)','RMS','RMS20','RMS50','RMS90','VIÉS','Viés20','Viés50','Viés90','N20','N50','N90','Nobs'
122
FORMAT(1X,A5,4X,A1,4X,A7,1X,A8,1X,A8,2X,A3,3X,A5,2X,A5,1X,A5,6X,A4,2X,A6,2X,A6,2X,A6,2X,A3,2
X,A3,2X,A3,4X,A4)
WRITE(5,132)'Nivel','S','S(0-20)','S(20-50)','S(50-
90)','RMS','RMS20','RMS50','RMS90','VIÉS','Viés20','Viés50','Viés90','N20','N50','N90','Nobs'
132
FORMAT(1X,A5,4X,A1,4X,A7,1X,A8,1X,A8,2X,A3,3X,A5,2X,A5,1X,A5,6X,A4,2X,A6,2X,A6,2X,A6,2X,A3,2
X,A3,2X,A3,4X,A4)
open(3,file='/home/flaviosc/Nnivel.txt',status='old')
DO k=1, 18
  read(3,200)nivel
200 format((I3))
!ZERANDO VARIÁVEIS UTILIZADAS PARA CADA NÍVEL ATRIBUÍDO PELO DO K=1,18
  n20=0
  n60=0
  n90=0
  t20=0
  t60=0
  t90=0
  !!!!!!!!!!!!!CÁLCULO ERROS GEOPOTENCIAL!!!!!!!!!!!!!!!!!!!!
  DO i=1, COUNT1(nivel-1)
!SOMANDO OS VALORES DE OMF PARA CALCULAR VIES RMS E DESVIO PADRÃO
  GEOPSOMA(nivel)=GEOPSOMA(nivel)+MGEOP(i,nivel)
  GEOPSOMASQ(nivel)=GEOPSOMASQ(nivel)+MGEOP(i,nivel)**2
  if(abs(MGEOP(i,nivel+4)).LE.20)then
    GEOPSOMA20(nivel)=GEOPSOMA20(nivel)+MGEOP(i,nivel)
    GEOPSOMASQ20(nivel)=GEOPSOMASQ20(nivel)+MGEOP(i,nivel)**2

    n20=n20+1
    MGEOP20(n20,nivel)=MGEOP(i,nivel)
  end if

```

```

if(abs(MGEOP(i,nivel+4)).GT.20.AND.abs(MGEOP(i,nivel+4)).LT.50)then
  GEOPSOMA60(nivel)=GEOPSOMA60(nivel)+MGEOP(i,nivel)
  GEOPSOMQA60(nivel)=GEOPSOMQA60(nivel)+MGEOP(i,nivel)**2

  n60=n60+1
  MGEOP60(n60,nivel)=MGEOP(i,nivel)
end if
if(abs(MGEOP(i,nivel+4)).GE.50)then
  GEOPSOMA90(nivel)=GEOPSOMA90(nivel)+MGEOP(i,nivel)
  GEOPSOMQA90(nivel)=GEOPSOMQA90(nivel)+MGEOP(i,nivel)**2

  n90=n90+1
  MGEOP90(n90,nivel)=MGEOP(i,nivel)
end if
END DO
!erro médio do omf-geop
OMFGEOP(nivel)=GEOPSOMA(nivel)/(COUNT1(nivel-1)-1)
!erro mádio quadrático do omf-Geop
OMFGEOPQ(nivel)=(GEOPSOMQA(nivel)/(COUNT1(nivel-1)-1))**0.5
!erro médio e RMS omf-geop entre as latitudes de (-20 a +20)
OMFGEOP20(nivel)=GEOPSOMA20(nivel)/n20
OMFGEOPQ20(nivel)=(GEOPSOMQA20(nivel)/n20)**0.5
!erro médio e RMS omf-geop entre as latitudes de (30 a 60)
OMFGEOP60(nivel)=GEOPSOMA60(nivel)/n60
OMFGEOPQ60(nivel)=(GEOPSOMQA60(nivel)/n60)**0.5
!erro médio e RMS omf-geop entre as latitudes de (60 a 90)
OMFGEOP90(nivel)=GEOPSOMA90(nivel)/n90
OMFGEOPQ90(nivel)=(GEOPSOMQA90(nivel)/n90)**0.5
!!!!CÁLCULO DO DESVIO PADRÃO PARA AS MEDIDAS DE OMF GEOPOTENCIAL!!!!!!!
DO m=1, COUNT1(nivel-1)
  SIGMAGEOP(nivel)=SIGMAGEOP(nivel)+((MGEOP(m,nivel)-OMFGEOP(nivel))**2)
END DO
DESVIOGEOP(nivel)=(SIGMAGEOP(nivel)/(COUNT1(nivel-1)-2))**0.5

!!CONDIÇÃO PARA EVITAR A DIVISÃO POR ZERO,VISTO QUE N COMEÇA COM 1 E N=2
QUANDO HÁ UM DADO NO NIVEL.
if(n20.GT.2.and.n60.GT.2.and.n90.GT.2)then
  DO o=1, n20
    SIGMAGEOP20(nivel)=SIGMAGEOP20(nivel)+((MGEOP20(o,nivel)-
OMFGEOP20(nivel))**2)
  end do
  DESVIOGEOP20(nivel)=(SIGMAGEOP20(nivel)/(n20-1))**0.5
  DO p=1, n60
    SIGMAGEOP60(nivel)=SIGMAGEOP60(nivel)+((MGEOP60(p,nivel)-
OMFGEOP60(nivel))**2)
  end do
  DESVIOGEOP60(nivel)=(SIGMAGEOP60(nivel)/(n60-1))**0.5
  DO q=1, n90
    SIGMAGEOP90(nivel)=SIGMAGEOP90(nivel)+((MGEOP90(q,nivel)-
OMFGEOP90(nivel))**2)
  end do
  DESVIOGEOP90(nivel)=(SIGMAGEOP90(nivel)/(n90-1))**0.5
end if
!!!!!!!CÁLCULO ERROS UMIDADE!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
if(nivel.GE.400)then
  DO j=1, COUNT2(nivel-1)
    qSOMA(nivel)=qSOMA(nivel)+Mq(j,nivel)

```

```

qSOMAQ(nivel)=qSOMAQ(nivel)+Mq(j,nivel)**2
if(abs(Mq(j,nivel+4)).LE.20)then
  qSOMA20(nivel)=qSOMA20(nivel)+Mq(j,nivel)
  qSOMAQ20(nivel)=qSOMAQ20(nivel)+Mq(j,nivel)**2
  t20=t20+1
  Mq20(t20,nivel)=Mq(j,nivel)
end if
if(abs(Mq(j,nivel+4)).GT.20.AND.abs(Mq(j,nivel+4)).LT.50)then
  qSOMA60(nivel)=qSOMA60(nivel)+Mq(j,nivel)
  qSOMAQ60(nivel)=qSOMAQ60(nivel)+Mq(j,nivel)**2
  t60=t60+1
  Mq60(t60,nivel)=Mq(j,nivel)
end if
if(abs(Mq(j,nivel+4)).GE.50)then
  qSOMA90(nivel)=qSOMA90(nivel)+Mq(j,nivel)
  qSOMAQ90(nivel)=qSOMAQ90(nivel)+Mq(j,nivel)**2
  t90=t90+1
  Mq90(t90,nivel)=Mq(j,nivel)
end if
END DO
!if(COUNT2(nivel-1).NE.1)then
  OMFq(nivel)= qSOMA(nivel)/(COUNT2(nivel-1)-1)
  OMFQ_q(nivel)= (qSOMAQ(nivel)/(COUNT2(nivel-1)-1))**0.5
!end if

!erro médio e RMS omf-umidade entre as latitudes de (-20 a +20)
OMFq20(nivel)=qSOMA20(nivel)/t20
OMFq_Q20(nivel)=(qSOMAQ20(nivel)/t20)**0.5

!erro médio e RMS omf-q entre as latitudes de (20 a 60)
OMFq60(nivel)=qSOMA60(nivel)/t60
OMFq_Q60(nivel)=(qSOMAQ60(nivel)/t60)**0.5

!erro médio e RMS omf-geop entre as latitudes de (60 a 90)
OMFq90(nivel)=qSOMA90(nivel)/t90
OMFq_Q90(nivel)=(qSOMAQ90(nivel)/t90)**0.5

!!!!CÁLCULO DO DESVIO PADRÃO PARA AS MEDIDAS DE OMF UMIDADE!!!!!!!

DO x=1, COUNT1(nivel-1)
  SIGMAq(nivel)=SIGMAq(nivel)+((Mq(x,nivel)-OMFq(nivel))**2)
END DO
!if(COUNT2(nivel-1).GT.2)then
  DESVIOq(nivel)=(SIGMAq(nivel)/(COUNT2(nivel-1)-2))**0.5
!end if

DO y=1, t20
  SIGMAq20(nivel)=SIGMAq20(nivel)+((Mq20(y,nivel)-OMFq20(nivel))**2)
end do
!if(t20.GT.1)then
  DESVIOq20(nivel)=(SIGMAq20(nivel)/(t20-1))**0.5
!end if

DO z=1, t60
  SIGMAq60(nivel)=SIGMAq60(nivel)+((Mq60(z,nivel)-OMFq60(nivel))**2)
end do
DESVIOq60(nivel)=(SIGMAq60(nivel)/(t60-1))**0.5

```



```

DO w=1, t90
    SIGMAq90(nivel)=SIGMAq90(nivel)+((Mq90(w,nivel)-OMFq90(nivel))**2)
end do
DESVIOq90(nivel)=(SIGMAq90(nivel)/(t90-1))**0.5

print *, nivel, 'EM_q =',OMFq(nivel), 'Nobs=', COUNT2(nivel-1)-1
print *, nivel, 'EM_q20=',OMFq20(nivel), 'Nobs=', t20
print *, nivel, 'EM_q60=',OMFq60(nivel), 'Nobs=', t60
print *, nivel, 'EM_q90=',OMFq90(nivel), 'Nobs=', t90
print *, ' '
print *, nivel, 'EMQ_q =',OMFQ_q(nivel), 'Nobs=', COUNT2(nivel-1)-1
print *, nivel, 'EMQ_q20=',OMFQ_q20(nivel), 'Nobs=', t20
print *, nivel, 'EMQ_q60=',OMFQ_q60(nivel), 'Nobs=', t60
print *, nivel, 'EMQ_q90=',OMFQ_q90(nivel), 'Nobs=', t90
print *, ' '
print *, nivel, ' DESVIOq =',DESVIOq(nivel), 'Nobs=', COUNT2(nivel-1)-2
print *, nivel, ' DESVIOq20=',DESVIOq20(nivel), 'Nobs=', t20
print *, nivel, ' DESVIOq60=',DESVIOq60(nivel), 'Nobs=', t60
print *, nivel, ' DESVIOq90=',DESVIOq90(nivel), 'Nobs=', t90

end if

! ESCRIBENDO AS ESTATISTICAS JÁ CALCULADAS

WRITE(4,134)nivel,DESVIOGEOP(nivel),DESVIOGEOP20(nivel),DESVIOGEOP60(nivel),DESVIOGEOP90(nivel
),OMFGEOPQ(nivel),OMFGEOPQ20(nivel),OMFGEOPQ60(nivel),OMFGEOPQ90(nivel),OMFGEOP(nivel),OMF
GEOP20(nivel),OMFGEOP60(nivel),OMFGEOP90(nivel),n20,n60,n90,COUNT1(nivel-1)-1
134
FORMAT(2X,I3,2X,F6.2,1X,F6.2,3X,F6.2,3X,F6.2,2X,F6.2,1X,F6.2,1X,F6.2,1X,F6.2,3X,F6.2,1X,F7.2,1X,F7.2,1X,
F7.2,1X,I5,1X,I5,1X,I5,1X,I5)

WRITE(5,135)nivel,DESVIOq(nivel),DESVIOq20(nivel),DESVIOq60(nivel),DESVIOq90(nivel),OMFQ_q(nivel),OM
Fq_Q20(nivel),OMFq_Q60(nivel),OMFq_Q90(nivel),OMFq(nivel),OMFq20(nivel),OMFq60(nivel),OMFq90(nivel),t2
0,t60,t90,COUNT2(nivel-1)-1
135
FORMAT(2X,I3,2X,F6.2,1X,F6.2,3X,F6.2,3X,F6.2,2X,F6.2,1X,F6.2,1X,F6.2,1X,F6.2,3X,F6.2,1X,F7.2,1X,F7.2,1X,
F7.2,1X,I5,1X,I5,1X,I5,1X,I5)

print *, ' '
print *, nivel, ' EM_Geop=',OMFGEOP(nivel), 'Nobs=',COUNT1(nivel-1)-1
print *, nivel, 'EM_Geop20=',OMFGEOP20(nivel), 'Nobs=',n20
print *, nivel, 'EM_Geop60=',OMFGEOP60(nivel), 'Nobs=',n60
print *, nivel, 'EM_Geop90=',OMFGEOP90(nivel), 'Nobs=',n90
print *, ' '
print *, nivel, ' EMQ_Geop=',OMFGEOPQ(nivel), 'Nobs=',COUNT1(nivel-1)-1
print *, nivel, 'EMQ_Geop20=',OMFGEOPQ20(nivel), 'Nobs=',n20
print *, nivel, 'EMQ_Geop60=',OMFGEOPQ60(nivel), 'Nobs=',n60
print *, nivel, 'EMQ_Geop90=',OMFGEOPQ90(nivel), 'Nobs=',n90
print *, ' '
print *, nivel, ' DESVIOGeop=',DESVIOGEOP(nivel), 'Nobs=', COUNT1(nivel-1)-2
print *, nivel, 'DESVIOGeop20=',DESVIOGEOP20(nivel), 'Nobs=', n20
print *, nivel, 'DESVIOGeop60=',DESVIOGEOP60(nivel), 'Nobs=', n60
print *, nivel, 'DESVIOGeop90=',DESVIOGEOP90(nivel), 'Nobs=', n90
print *, ' '
END DO
END PROGRAM

```

## APÊNDICE F – Rotina que foi feita em Shell Script que confere os arquivos ODS presentes em um determinado diretório.

Este apêndice contém a rotina que faz uma verificação dos arquivos do tipo ODS que estão presentes na pasta que o programa foi setado para rodar. Com isso dá para se saber qual arquivo ODS esta faltando. E com esta informação será convertido outro arquivos ODS para a data que esta faltando.

**Nome do programa: confere\_ods.ksh**

```
#!/bin/ksh
# ##### Programa para testar quais são os arquivos de .ods que estão faltando
##### Criado por Dérek Schubert 29/06/2010
##### Insira a data
ano=2007
mes=03
diaf=31
path=/rede/nas/gdad_reanalise/reanalisis/ods_files/'${ano}'/'${mes}
datai=${ano}${mes}0100
dataf=${ano}${mes}${diaf}18
ct=0
#####Criando arquivo de log#####
logfile='ods_faltando.log'
echo '##### Comencando a Conferencia #####' >
${logfile}
echo >> ${logfile}
echo 'ARQUIVOS QUE ESTAO FALTANDO' >> ${logfile}
echo >> ${logfile}

##### Inicio do loop#####
while [ ${datai} -le ${dataf} ]; do
    echo 'Verificando para data: ${datai}'
    for i in AIRS_tpw.ods.t AIRS_GEOP.ods.t
    do
        if [ $i == 'AIRS_tpw.ods.t' ]; then
            ARQUI=${i}${datai}
            if [ -s ${path}/${ARQUI} ]; then
                ct=1
            else
                echo ${ARQUI} >> ${logfile}
            fi
        else
            if [ $i == 'AIRS_GEOP.ods.t' ];then
                ARQUI=${i}${datai}
                if [ -s ${path}/${ARQUI} ]; then
                    ct=1
                else
                    echo ${ARQUI} >> ${logfile}
                fi
            fi
        fi
    fi
done
```

```
done
if [ ${ct} -eq 1 ]; then
    ct=0
else
    echo >> ${logfile}
fi
# incrementando o horário
datai=`~/bin/inctime ${datai} +6 hour yyyymmddhh`
# Final do loop
done
echo '##### Conferencia Finalizada #####' >> ${logfile}
cat ods_faltando.log
```

## APÊNDICE G – Template que ajuda a gerar um matlab para cada estação

### Template

Este apêndice contém o template usado para gerar os gráficos para cada estação. Caso deseje utiliza-la é necessário que altere os campos onde se esta escrito 'NAMEest' e 'NAMEfig'. No lugar de NAMEest coloque o nome do arquivos onde estão os seus dados e no lugar de NAMEfig escreva o nome que deseja para as figuras que serão geradas.

```
clear
%
% Rotina para calcular o VIES, RMS E O DESVIO PADRAO para os dados de temperatura
% para as estacoes de radiossonda
%
% Criado por Derek Schubert 24/09/2010
% Load dos arquivos para as estacoes
load 'NAMEest.txt'
% Atribuindo valores as variaveis
TMR=NAMEest(:,13);
TMmend=NAMEest(:,16);
TMsch1b=NAMEest(:,17);
TMsch1f=NAMEest(:,18);
TMsch2=NAMEest(:,19);
TMsch3b=NAMEest(:,20);
TMsch3f=NAMEest(:,21);
TMsapu=NAMEest(:,22);
TMregi=NAMEest(:,23);
% ##### Calculando o VIES #####
somadifTMmend=0;
somadifTMsch1b=0;
somadifTMsch1f=0;
somadifTMsch2=0;
somadifTMsch3b=0;
somadifTMsch3f=0;
somadifTMsapu=0;
somadifTMregi=0;

viesTMmend=0;
viesTMsch1b=0;
viesTMsch1f=0;
viesTMsch2=0;
viesTMsch3b=0;
viesTMsch3f=0;
viesTMsapu=0;
viesTMregi=0;
tamVet=length(TMR);
for l=1:tamVet;
    somadifTMmend=somadifTMmend + (TMmend(l)-TMR(l));
    somadifTMsch1b=somadifTMsch1b + (TMsch1b(l)-TMR(l));
    somadifTMsch1f=somadifTMsch1f + (TMsch1f(l)-TMR(l));
    somadifTMsch2=somadifTMsch2 + (TMsch2(l)-TMR(l));
```

```

somadifTMsch3b=somadifTMsch3b + (TMsch3b(l)-TMR(l));
somadifTMsch3f=somadifTMsch3f + (TMsch3f(l)-TMR(l));
somadifTMsapu=somadifTMsapu + (TMsapu(l)-TMR(l));
somadifTMregi=somadifTMregi + (TMregi(l)-TMR(l));
end
viesTMmend=somadifTMmend/tamVet;
viesTMsch1b=somadifTMsch1b/tamVet;
viesTMsch1f=somadifTMsch1f/tamVet;
viesTMsch2=somadifTMsch2/tamVet;
viesTMsch3b=somadifTMsch3b/tamVet;
viesTMsch3f=somadifTMsch3f/tamVet;
viesTMsapu=somadifTMsapu/tamVet;
viesTMregi=somadifTMregi/tamVet;
% ##### Calculando o RMS #####
somadifRTMmend=0;
somadifRTMsch1b=0;
somadifRTMsch1f=0;
somadifRTMsch2=0;
somadifRTMsch3b=0;
somadifRTMsch3f=0;
somadifRTMsapu=0;
somadifRTMregi=0;
rmsTMmend=0;
rmsTMsch1b=0;
rmsTMsch1f=0;
rmsTMsch2=0;
rmsTMsch3b=0;
rmsTMsch3f=0;
rmsTMsapu=0;
rmsTMregi=0;
for J=1:tamVet;
    somadifRTMmend=somadifRTMmend + (TMmend(J)-TMR(J))^2;
    somadifRTMsch1b=somadifRTMsch1b + (TMsch1b(J)-TMR(J))^2;
    somadifRTMsch1f=somadifRTMsch1f + (TMsch1f(J)-TMR(J))^2;
    somadifRTMsch2=somadifRTMsch2 + (TMsch2(J)-TMR(J))^2;
    somadifRTMsch3b=somadifRTMsch3b + (TMsch3b(J)-TMR(J))^2;
    somadifRTMsch3f=somadifRTMsch3f + (TMsch3f(J)-TMR(J))^2;
    somadifRTMsapu=somadifRTMsapu + (TMsapu(J)-TMR(J))^2;
    somadifRTMregi=somadifRTMregi + (TMregi(J)-TMR(J))^2;
end
rmsTMmend=sqrt(somadifRTMmend/tamVet);
rmsTMsch1b=sqrt(somadifRTMsch1b/tamVet);
rmsTMsch1f=sqrt(somadifRTMsch1f/tamVet);
rmsTMsch2=sqrt(somadifRTMsch2/tamVet);
rmsTMsch3b=sqrt(somadifRTMsch3b/tamVet);
rmsTMsch3f=sqrt(somadifRTMsch3f/tamVet);
rmsTMsapu=sqrt(somadifRTMsapu/tamVet);
rmsTMregi=sqrt(somadifRTMregi/tamVet);
% ##### Calculando o Desvio Padrao #####
desvioTMmend=0;
desvioTMsch1b=0;
desvioTMsch1f=0;
desvioTMsch2=0;
desvioTMsch3b=0;
desvioTMsch3f=0;
desvioTMsapu=0;
desvioTMregi=0;

```

```

somadifDPTMmend=0;
somadifDPTMsch1b=0;
somadifDPTMsch1f=0;
somadifDPTMsch2=0;
somadifDPTMsch3b=0;
somadifDPTMsch3f=0;
somadifDPTMsapu=0;
somadifDPTMregi=0;
for K=1:tamVet;
    somadifDPTMmend=somadifDPTMmend + ((TMmend(K)-TMR(K))-viesTMmend)^2;
    somadifDPTMsch1b=somadifDPTMsch1b + ((TMsch1b(K)-TMR(K))-viesTMsch1b)^2;
    somadifDPTMsch1f=somadifDPTMsch1f + ((TMsch1f(K)-TMR(K))-viesTMsch1f)^2;
    somadifDPTMsch2=somadifDPTMsch2 + ((TMsch2(K)-TMR(K))-viesTMsch2)^2;
    somadifDPTMsch3b=somadifDPTMsch3b + ((TMsch3b(K)-TMR(K))-viesTMsch3b)^2;
    somadifDPTMsch3f=somadifDPTMsch3f + ((TMsch3f(K)-TMR(K))-viesTMsch3f)^2;
    somadifDPTMsapu=somadifDPTMsapu + ((TMsapu(K)-TMR(K))-viesTMsapu)^2;
    somadifDPTMregi=somadifDPTMregi + ((TMregi(K)-TMR(K))-viesTMregi)^2;
end
desvioTMmend=sqrt(somadifDPTMmend/(tamVet-1));
desvioTMsch1b=sqrt(somadifDPTMsch1b/(tamVet-1));
desvioTMsch1f=sqrt(somadifDPTMsch1f/(tamVet-1));
desvioTMsch2=sqrt(somadifDPTMsch2/(tamVet-1));
desvioTMsch3b=sqrt(somadifDPTMsch3b/(tamVet-1));
desvioTMsch3f=sqrt(somadifDPTMsch3f/(tamVet-1));
desvioTMsapu=sqrt(somadifDPTMsapu/(tamVet-1));
desvioTMregi=sqrt(somadifDPTMregi/(tamVet-1));
for o=200:350;
    vetor(o)=o;
end
% ##### Plotando grafico de comparacao de TEMPERATURA #####
figure(1);

plot(TMR, TMmend, 'bo', 'LineWidth', 2, ...
     'MarkerEdgeColor', 'b', ...
     'MarkerFaceColor', 'b', ...
     'MarkerSize', 3)
hold on
plot(TMR, TMsch1b, 'r*')
hold on
plot(TMR, TMsch1f, 'g+')
hold on
plot(TMR, TMsch2, 'k.')
hold on
plot(TMR, TMsch3b, 'cx')
hold on
plot(TMR, TMsch3f, 'm*')
hold on
plot(TMR, TMsapu, 'ko')
hold on
plot(TMR, TMregi, 'yx')
hold on
plot(vetor, vetor, 'k', 'LineWidth', 2, ...
     'MarkerEdgeColor', 'k', ...
     'MarkerFaceColor', 'k', ...
     'MarkerSize', 6)
hold on
title('Comparacao temperatura NAMEfig', 'FontSize', 15)
xlabel('Temperatura Media(K) Radiossonda');

```

```

set(gca,'YLIM',[min(TMR-5),max(TMR)+5]);
set(gca,'XLIM',[min(TMR)-5,max(TMR)+5]);
ylabel('Temperatura Media(K) do Modelo');
legend('Tmmend ','Tmsch1b','Tmsch1f','Tmsch2','Tmsch3b','Tmsch3f','Tmsapu','TMregi',2);
print('-djpeg100','figura_NAMEest.jpg');
close(1)
% ##### Atribuindo valores para o grafico de estatistica #####
calcVIES(8)=0;
calcDP(8)=0;
calcRMS(8)=0;

calcVIES(1)=viesTmmend;
calcVIES(2)=viesTmsch1b;
calcVIES(3)=viesTmsch1f;
calcVIES(4)=viesTmsch2;
calcVIES(5)=viesTmsch3b;
calcVIES(6)=viesTmsch3f;
calcVIES(7)=viesTmsapu;
calcVIES(8)=viesTMregi;

calcDP(1)=desvioTmmend;
calcDP(2)=desvioTmsch1b;
calcDP(3)=desvioTmsch1f;
calcDP(4)=desvioTmsch2;
calcDP(5)=desvioTmsch3b;
calcDP(6)=desvioTmsch3f;
calcDP(7)=desvioTmsapu;
calcDP(8)=desvioTMregi;

calcRMS(1)=rmsTmmend;
calcRMS(2)=rmsTmsch1b;
calcRMS(3)=rmsTmsch1f;
calcRMS(4)=rmsTmsch2;
calcRMS(5)=rmsTmsch3b;
calcRMS(6)=rmsTmsch3f;
calcRMS(7)=rmsTmsapu;
calcRMS(8)=rmsTMregi;
Tvet=num2str(tamVet);
numeroOBS=strcat('Numero de Observacoes: ',Tvet);
% ##### Plotando grafico de comparacao de VIES DP RMS #####
figure(2);
plot(calcVIES,'-bo','LineWidth',2,...
      'MarkerEdgeColor','b',...
      'MarkerFaceColor','w',...
      'MarkerSize',8)
hold on
plot(calcDP,'-rx','LineWidth',2,...
      'MarkerEdgeColor','r',...
      'MarkerFaceColor','r',...
      'MarkerSize',10)
hold on
plot(calcRMS,'-g+', 'LineWidth',2,...
      'MarkerEdgeColor','g',...
      'MarkerFaceColor','g',...
      'MarkerSize',10)
hold on
plot([0,1,2,3,4,5,6,7,8,9],[0,0,0,0,0,0,0,0,0,0],'k','LineWidth',2,...
      'MarkerEdgeColor','k',...

```

```

        'MarkerFaceColor','k',...
        'MarkerSize',6)
hold on
title ('Estadísticas de NAMEfig','FontSize',15)
set(gca,'XTick',[1,2,3,4,5,6,7,8]);
set(gca,'YTick',[-13:1:13]);
set(gca,'XTickLabel',{'Tmmend','Tmsch1b','Tmsch1f','Tmsch2','Tmsch3b','Tmsch3f','Tmsapu','TMregi'},'FontSize',10);
set(gca,'XLIM',[0,9]);
set(gca,'YLIM',[-13,13]);
ylabel('Valor');
text(5.7,-9,numeroOBS,'FontSize',12)
legend('VIES','DP','RMS',4);
grid
print('-djpeg100','figura_VIES_DP_RMS_NAMEest.jpg');
close(2)

```