



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

**AVALIAÇÃO DA CORREÇÃO DO ERRO IONOSFÉRICO EM
RECEPTORES GNSS UTILIZANDO A SIMULAÇÃO
IONOSFÉRICA DESENVOLVIDA NO INPE E O SOFTWARE
RTKLIB**

Victor Dallagnol Bento

Relatório de Iniciação Científica do
programa PIBITI, orientado pelo Dr.
Adriano Petry.

INPE
Santa Maria
2020



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA, INOVAÇÕES E COMUNICAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

**AVALIAÇÃO DA CORREÇÃO DO ERRO IONOSFÉRICO EM
RECEPTORES GNSS UTILIZANDO A SIMULAÇÃO
IONOSFÉRICA DESENVOLVIDA NO INPE E O SOFTWARE
RTKLIB**

Victor Dallagnol Bento

Relatório de Iniciação Científica do
programa PIBITI, orientado pelo Dr.
Adriano Petry.

INPE
Santa Maria
2020

RESUMO

O projeto tem o objetivo de avaliar a eficiência de diferentes correções ionosféricas aplicadas em dispositivos GPS de frequência simples em comparação com a correção realizada em dispositivos de dupla frequência, uma vez que este consegue compensar a refração da ionosfera aplicando a combinação linear dos dados das duas frequências utilizadas. O erro nos dados de posicionamento em receptores de sistemas globais de navegação por satélite (GNSS) é causado por diversos fatores, mas o mais significativo é a refração ionosférica que depende diretamente do conteúdo total de elétrons (TEC) presente na camada ionosférica. O TEC tem sua variação dependente de diversas variáveis, como o ciclo solar, a época do ano e o campo magnético da Terra. Com uma correção ionosférica mais eficaz, serão obtidas melhores informações fornecidas pelos receptores GPS de frequência simples, resultando também em impactos na área de pesquisa acadêmica e no setor econômico. Os modelos de correções ionosféricas utilizados na comparação foram o modelo SUPIM-DAVS (Modelo de Plasma-Ionosfera da Universidade de Sheffield - Sistema de Visualização e Assimilação de Dados), utilizado pela simulação ionosférica do INPE (Instituto Nacional de Pesquisas Espaciais), o modelo GIM (Mapa Global Ionosférico) utilizado pelo IGS (Serviço Internacional de GNSS) e o modelo de Klobuchar. Para efetuar a avaliação, foram utilizadas 100 estações GNSS espalhadas pelo mundo, avaliadas durante um dia de cada mês do ano de 2018. O programa presente no RTKLib gerou os arquivos de pós-processamento para cada uma das correções ionosféricas, referentes a cada estação. Para avaliar estes arquivos de pós-processamento, foi desenvolvido um programa na linguagem Python, calculando a distância euclidiana entre a localização da estação (conhecida) e o posicionamento gerado pelo software presente no RTKLib. O modelo Klobuchar mostrou-se muito eficaz, em comparação com os demais modelos analisados neste projeto, na correção do erro ionosférico em receptores GNSS de frequência simples, justificando seu atual uso em diversos dispositivos GPS.

Palavras-chave: Receptores GNSS. Refração Ionosférica. Klobuchar. INPE. SUPIM-DAVS. IGS. GIM. Python, Avaliação.

LISTA DE FIGURAS

	<u>Pág.</u>
Figura 1: Camadas da Ionosfera	12
Figura 2: Relação recombinação, altitude, densidade	12
Figura 3: Caminho de propagação de um sinal de satélite para o receptor.	15
Figura 4: Exemplo de mapa gerado pelo modelo SUPIM-DAVS.....	18
Figura 5: Verificação do dia no arquivo sp3	23
Figura 6: Verificação do arquivo de antena.....	24
Figura 7: Configuração do RTKPost para os diferentes modelos	25
Figura 8: Fluxograma básico da mecânica do funcionamento do Software de Avaliação	30
Figura 9: Resultado numérico da avaliação feita pelo software criado	31
Figura 10: Exemplo de estações com amostragem alteradas	33
Figura 11: Representação da média do erro por estação com distribuição global	35
Figura 12: Representação do erro distribuído por estações em função dos meses.....	36

LISTA DE TABELAS

	<u>Pág.</u>
Tabela 1: Datas simuladas	22
Tabela 2: Erro médio de cada método de correção para os dias avaliados	31

LISTA DE SIGLAS E ABREVIATURAS

GNSS	Sistema Global de Navegação por Satélite
GPS	Sistema de Posicionamento Global
NAVSTAR-GPS	Satélite de Navegação com Tempo e Variação – Sistema de Posicionamento Global
EUA	Estados Unidos da América
SPS	Serviço de Posicionamento Padrão
PPS	Serviço de Posicionamento Preciso
TEC	Conteúdo Total de Elétrons
IGS	Serviço Internacional de GNSS
SUPIM	Modelo de Plasma-Ionosfera da Universidade de Sheffield
DAVS	Sistema de Visualização e Assimilação de Dados
INPE	Instituto Nacional de Pesquisas Espaciais
RTKLib	Biblioteca Cinemática em Tempo Real
HD	Alta Frequência
UV	Ultra Violeta
EUV	Extremo Ultra Violeta
GLONASS	Sistema de Navegação Global via Satélite Russo
Galileo	Sistema de Navegação Global via Satélite da União Européia
Beidou/Compass	Sistema de Navegação Global via Satélite Chinês
VTEC	Tec Vertical
STEC	TEC Inclinado
IONEX	Troca de Ionosfera
RINEX	Formato de Troca Independente do Receptor
GUI	Interface Gráfica de Usuário

CUI	Interface de Caracteres de Usuário
OBS	Arquivo RINEX de Observação
NAV	Arquivo RINEX de Observação
NAV	Arquivo RINEX de Navegação
ANTEX	Formato de Troca de Antena
CLK	Relógio
FTP	Protocolo de Transferência de Arquivo
HTTP	Protocolo de Transferência de Hipertexto
PPP	Posicionamento de Ponto Preciso
CDDIS	O sistema de Informação de Dados da Crustal Dynamics
SNR	Relação Sinal-Ruído
PCV	Variação do Centro da Fase da Antena
IPP	Ponto de Furo Ionosférico
CODE	Centro de Determinação de Órbitas na Europa
ESA	Agência Espacial Europeia
JPL	Laboratório Global de Propulsão a Jato
UPC	Catalunha-Barcelona Tech

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO	9
1.1 Objetivos Gerais.....	10
1.2 Objetivos Específicos.....	10
2 REFERENCIAL TEÓRICO	11
2.1 Ionosfera	11
2.2 GNSS e GPS	13
2.3 Correções Ionosfericas.....	14
2.3.1 Combinação Linear Livre da Ionosfera (ION-FREE).....	14
2.3.2 Modelos Ionosféricos	15
3 PÓS-PROCESSAMENTO	20
4 MATERIAIS E MÉTODOS	21
4.1 Instalação	21
4.2 Obtenção dos arquivos.....	21
4.3 Configuração.....	25
4.4 Execução	27
5 ANÁLISE E RESULTADOS	29
5.1 AVALIAÇÃO NUMÉRICA	30
5.2 AVALIAÇÃO GRÁFICA	33
6 CONCLUSÃO.....	37
APENDICE A – EXEMPLO DE ARQUIVO DE CONFIGURAÇÃO.....	41
APENDICE B – EXEMPLO DE SCRIPT DE AUTOMATIZAÇÃO PARA EXECUÇÃO DE PÓS-PROCESSAMENTO.....	45
APENDICE C – SOFTWARE AVALIADOR	47

1 INTRODUÇÃO

O GNSS (Sistema de Navegação Global por Satélite), é um conjunto de sistemas de navegação por satélite que possuem a capacidade de oferecer posicionamento em qualquer local da superfície terrestre. Dentre esses sistemas, encontra-se o mais popular e utilizado na atualidade, o GPS (Sistema Global de Posicionamento), originalmente conhecido por NAVSTAR-GPS (Satélite de navegação com sistema de posicionamento global de tempo e alcance). O GPS foi originalmente desenvolvido pelas forças armadas dos EUA e fornece serviços gratuitos de posicionamento e cronometragem em todo o mundo (TEUNISSEN; MONTENBRUCK, 2017).

O GPS fornece dois serviços de posicionamento diferentes: o Serviço de Posicionamento de Precisão (PPS, baseado no processamento de frequência dupla - L1 e L2), originalmente desenvolvido para aplicações militares, e o Serviço de Posicionamento Padrão (SPS, baseado em frequência simples - L1), para aplicações civis (TEUNISSEN; MONTENBRUCK, 2017).

Os receptores GPS de frequência simples são mais baratos que os de frequência dupla, o que os tornam mais populares. A refração ionosférica é a maior fonte de erro nos dispositivos GNSS (CAMARGO, 1999), e depende diretamente do conteúdo total de elétrons (TEC) na camada ionosférica. O TEC é influenciado pelo ciclo solar, as estações do ano, a hora local, a localização geográfica, o campo magnético da Terra, dentre outras variáveis (MATSUOKA; COLLISCHONN; CAMARGO, 2010). Através da combinação linear de suas frequências (L1 e L2) os receptores GPS de frequência dupla conseguem amenizar os efeitos da refração, compensando o TEC medido, enquanto receptores de frequência simples (L1) dependem de diferentes modelos que simulam a ionosfera e seu comportamento (HOLLWEG et al.). O objetivo principal deste trabalho é efetuar uma análise e comparação da correção do erro ionosférico em receptores de frequência simples utilizando modelos ionosféricos distintos, tendo como base a correção feita pelo receptor de dupla frequência.

1.1 Objetivos Gerais

O principal objetivo do projeto é a avaliação de diferentes métodos para correção do erro ionosférico em receptores GPS de frequência simples, utilizando diferentes estações espalhadas pelo mundo. Estas estações serão pós-processadas em um software presente no RTKLib. Estes arquivos de pós-processamento, para cada uma das estações referente a cada um dos modelos de correção, serão avaliados através de um software desenvolvido na linguagem Python.

1.2 Objetivos Específicos

Primeiramente será necessário instalar o software RTKLib, estudar os softwares de pós-processamento presentes nele (RTKpost e RNX2RTKP) e efetuar o download dos arquivos necessários para sua configuração e execução. Serão avaliadas correções baseadas na simulação ionosférica desenvolvida pelo INPE (modelo SUPIM), na correção desenvolvida pelo IGS (modelo GIM) e na correção padrão executada pelo software RTKLib (Klobuchar), em comparação com a correção utilizando dupla frequência. Para isso será necessário obter os mapas de TEC de cada uma das correções. Por fim, o RTKPost será configurado e executado, e o resultado desta execução será analisado e avaliado por um software desenvolvido.

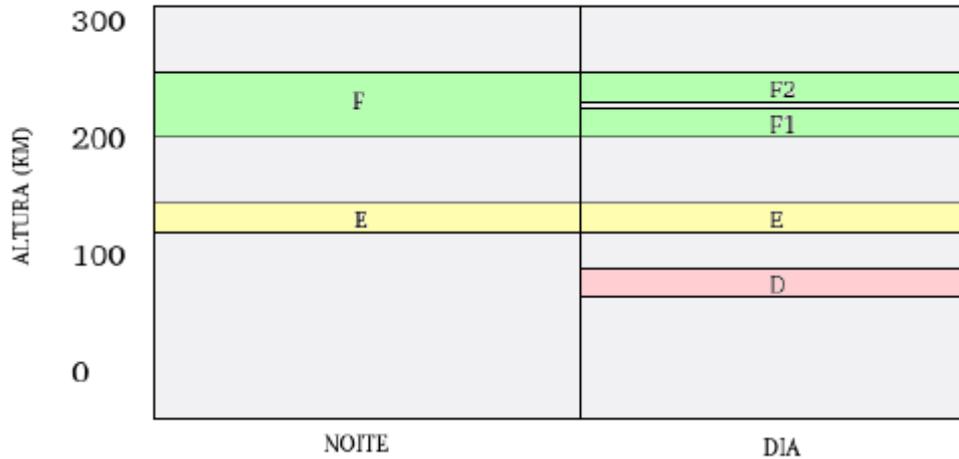
2 REFERENCIAL TEÓRICO

Neste capítulo serão apresentados os principais conceitos necessários para a avaliação das correções ionosféricas em dispositivos de frequência simples. Serão apresentados conceitos sobre a ionosfera, receptores GNSS (mais especificamente receptores GPS), o erro ionosférico, correção do erro ionosférico e sobre o software RTKLib.

2.1 Ionosfera

As regiões de maior interesse da Atmosfera para receptores GNSS envolvendo posicionamento geodésico são a troposfera e a ionosfera (LEICK; RAPOPORT; TATARNIKOV,2015). Localizada acima de 80 Km de altitude, encontra-se a ionosfera, tendo seu nome originado de que em elevadas altitudes os elétrons livres (criados pelo processo de ionização) sobrevivem tempo suficiente para se recombinarem com íons (átomos que vão perder ou ganhar elétrons). Ionização é o processo em que elétrons são removidos ou adicionados, produzindo átomos negativos ou positivos, sendo os íons positivos os mais comuns na camada ionosférica, ocasionados por partículas energéticas que precipitam na atmosfera e colidem com o gás circundante. O maior pico de elétrons livres encontra-se na região de aproximadamente 300 Km de altitude ou também pelos fótons solares de alta energia (Raio Ultravioleta e X) (MOLDWIN, 2008).

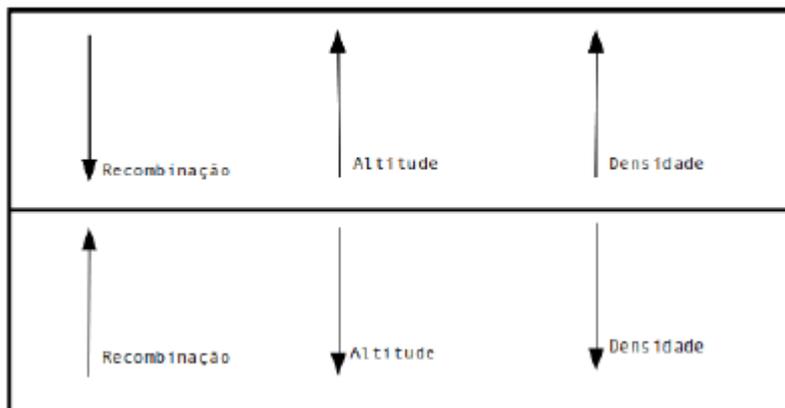
Figura 1: Camadas da Ionosfera



Fonte: Autor

A ionosfera é mais densa no período diurno pois possui um maior número de elétrons livres (fotoionização), em contraponto, no período noturno, acontece a recombinação que é o tempo gasto para que um elétron e um íon se juntem para formar um neutro. Em baixas altitudes a recombinação, assim como a densidade, é mais alta, como pode ser visto na Figura 2: Relação recombinação, altitude, densidade. A quantidade de ionização é determinada pela produção dos íons (por fotoionização) com a perda dos íons (por recombinação) (MOLDWIN, 2008).

Figura 2: Relação recombinação, altitude, densidade



Fonte: Autor

A ionosfera é dividida em regiões (Figura 1: Camadas da Ionosfera) com base nas densidades de íons. A região D é a camada mais baixa da ionosfera, responsável por absorver as ondas de comunicação por rádio de alta frequência (HF) causando degradação quando a

comunicação é de longa distância. A região E, que é formada por raios X de baixa energia e ionização por radiação solar Ultravioleta (UV) do oxigênio (O_2), é responsável pela luz emitida (auroras) nos hemisférios norte e sul. A última, e a mais densa, região da ionosfera é a região F, formada pela radiação solar UV, e também diminui a noite, mas diferentemente das regiões D e E, a região F possui uma taxa de recombinação baixa. Devido a fotoionização nas grandes altitudes, a região F é subdividida em outras duas regiões durante o dia (F1 e F2). Em relação às duas camadas formadas, a camada F2 é mais densa que a camada F1 (MOLDWIN, 2008).

Esses elétrons livres na camada ionosférica são mensurados em termos de conteúdo total de elétrons (TEC), que significa o número de elétrons livres, considerando um cilindro de área $1m^2$ ao longo do caminho entre emissor (satélite) e receptor (estação GNSS). Esses elétrons livres ocasionam uma interferência na portadora do sinal de transmissão, fazendo a ionosfera se comportar como um meio dispersivo.

2.2 GNSS e GPS

O posicionamento global requer a estimação das distâncias entre o local que será feita a análise e um conjunto de pelo menos quatro satélites GNSS com coordenadas conhecidas. As coordenadas dos satélites são determinadas pelo receptor usando sinais e dados de navegação transmitidos pelos satélites. O GNSS é formado por constelações de satélites que garantem que cada usuário tenha em qualquer ponto da terra e a qualquer momento, pelo menos quatro satélites à vista simultaneamente (SUBIRANA; ZORNOZA; HERNÁNDEZ-PAJARES, 2013).

Um GNSS consiste em três segmentos: o segmento espacial (satélites), segmento de controle ou terrestre (funcionamento do sistema) e o segmento de usuários (posicionamento, velocidade e tempo preciso dos sistemas GNSS). GPS, Glonass, Galileo e Beidou são alguns dos sistemas de navegação por satélite mais importantes, sendo capazes de fornecer posicionamento contínuo por todo mundo (SUBIRANA; ZORNOZA; HERNÁNDEZ-PAJARES, 2013).

O GPS é operado pelo governo dos EUA e é capaz de fornecer o posicionamento em qualquer lugar do mundo, sob qualquer condição climática, durante as 24 horas do dia, oferecendo dois tipos de serviços inicialmente: um serviço destinado ao uso militar (PPS)

e o serviço de posicionamento padrão (SPS) disponível para qualquer pessoa (LEICK; RAPOPORT; TATARNIKOV, 2015). Em 1995 o GPS foi considerado totalmente operacional, com 24 satélites GPS em órbita, separados em grupos de quatro, organizados em seis planos orbitais igualmente espaçados ao redor da Terra. (TEUNISSEN; MONTENBRUCK, 2017; SUBIRANA; ZORNOZA; HERNÁNDEZ-PAJARES, 2013; SEEBER, 2003).

2.3 Correções Ionosféricas

Atualmente, a maior fonte de erro sistemático no posicionamento com receptores GPS de frequência simples é devido à refração ionosférica. O erro associado à refração ionosférica depende do conteúdo total de elétrons (TEC) na camada ionosférica, que por sua vez é influenciado por diversos fatores, tais como ciclo solar, época do ano, hora local, localização geográfica e atividade geomagnética, e é difícil de ser corrigido (CAMARGO, 1999). Os sinais do GPS provenientes do satélite, propagam-se nas diferentes regiões da ionosfera até chegar ao receptor. Cada uma dessas regiões possui suas características, apresentando no geral um comportamento dispersivo, afetando a modulação e a fase da portadora, ocasionando um retardo e um avanço respectivamente (LEICK; RAPOPORT; TATARNIKOV, 2015). Esses efeitos na portadora são os causadores dos atrasos ionosféricos que aumentam o comprimento do caminho percorrido pelo sinal.

A alternativa para minimizar o erro devido à ionosfera nas observáveis GPS de receptores de dupla frequência é a combinação linear (ion-free) de suas frequências L1 e L2. Já nas observáveis GPS de receptores de uma única frequência, é necessário utilizar algum modelo ionosférico (MATSUOKA; COLLISCHONN; CAMARGO, 2010).

2.3.1 Combinação Linear Livre da Ionosfera (ION-FREE)

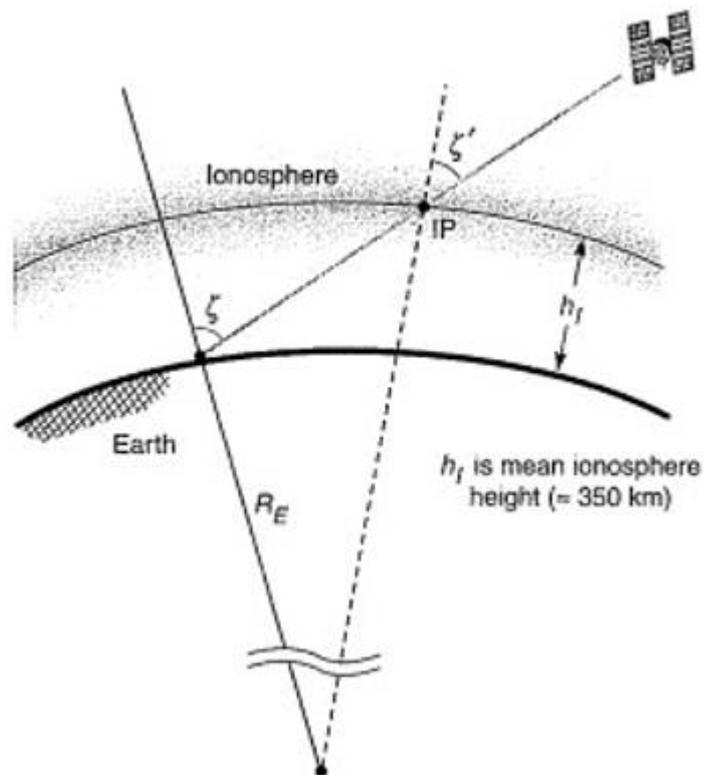
Os receptores de dupla frequência conseguem estimar o TEC compensando em até 99,9% a distorção do caminho de propagação utilizando a combinação linear livre da atmosfera (ion-free) das suas portadoras (L1 e L2) (SUBIRANA; ZORNOZA; HERNÁNDEZ-PAJARES, 2013). Esses efeitos ionosféricos podem ser de primeira, segunda e terceira ordem, sendo que o de primeira ordem corresponde a 99% dos efeitos ionosféricos e

está relacionado apenas ao TEC na camada ionosférica (TEUNISSEN; MONTENBRUCK, 2017). Os efeitos de segunda ordem da ionosfera estão relacionados com a densidade máxima de elétrons nas diferentes regiões da ionosfera, enquanto que os efeitos de terceira ordem são iguais ao de primeira ordem e segunda ordem, mas em função do STEC (MARQUES et al., 2014).

2.3.2 Modelos Ionosféricos

Como a correção dos efeitos ionosféricos de primeira ordem dependem apenas do TEC e o termo de correção depende apenas da integral da densidade de elétrons ao longo do caminho do raio (STEC), um modelo 2D do VTEC pode ser suficiente para compensar essa distorção em muitas aplicações. Para isso, o VTEC deve ser usado para se obter uma estimativa para o STEC (TEUNISSEN; MONTENBRUCK, 2017). Essa estimativa acontece no IPP (ponto onde o sinal fura a ionosfera, considerada concentrada em uma casca esférica na faixa de 300-400 km de altitude) como indicado na Figura 3: Caminho de propagação de um sinal de satélite para o receptor. (MISRA; ENGE, 2006).

Figura 3: Caminho de propagação de um sinal de satélite para o receptor.



Fonte: (MISRA; ENGE, 2006).

O TEC estimado pode ser calculado pela equação:

$$VTEC = TEC(\delta) \times \cos(\delta')$$

onde δ representa o ângulo zenital (elevação do satélite) e $\cos(\delta')$ representa a função de mapeamento. A função é expressa pela equação:

$$\cos(\delta') = [1 - (\frac{re \times \cos(\delta)}{re + hi})^2]^{-1/2}$$

onde re é o raio da terra, δ é o angulo zenital e hi é a altitude de perfuração ionosférica.

Ao utilizar modelos ionosféricos para correções ionosféricas em receptores de frequência simples, a qualidade da correção depende da função de mapeamento aplicada pelo modelo. Muitos usuários de dispositivos GPS de frequência simples utilizam modelos de TEC, como por exemplo o modelo de Klobuchar, o modelo NeQuick, utilizado por usuários de dispositivos Galileo, dentre outros modelos (TEUNISSEN; MONTENBRUCK, 2017).

2.3.2.1 Klobuchar

O Modelo de Klobuchar consiste em um algoritmo empírico com equações capazes de reduzir o erro ionosférico em sistemas de posicionamento global de frequência simples (L1). O objetivo do algoritmo é incluir os principais elementos para descrever com uma maior veracidade o comportamento complexo da ionosfera. Visando um mínimo de coeficientes possíveis e um menor tempo computacional para o usuário, o algoritmo faz uso de oito coeficientes que são transmitidos nas efemérides dos satélites GPS como parte da mensagem para efetuar a correção (KLOBUCHAR, 1987).

Também chamado de modelo de Broadcast, o algoritmo consegue produzir uma correção de pelo menos 50% (KLOBUCHAR, 1987). Por isso, faz-se necessário dispor de uma estratégia mais efetiva de eliminar os efeitos da ionosfera, a qual tem sido investigada a partir do uso de modelos regionais para a ionosfera, que representam com maior eficácia o comportamento da ionosfera da região de estudo. As mensagens de navegação fornecem a latitude e longitude do usuário, o azimute, o ângulo de elevação e os coeficientes de Klobuchar e necessários para efetuar os cálculos que podem ser encontrados em (SUBIRANA; ZORNOZA; HERNÁNDE-PAJARES, 2013).

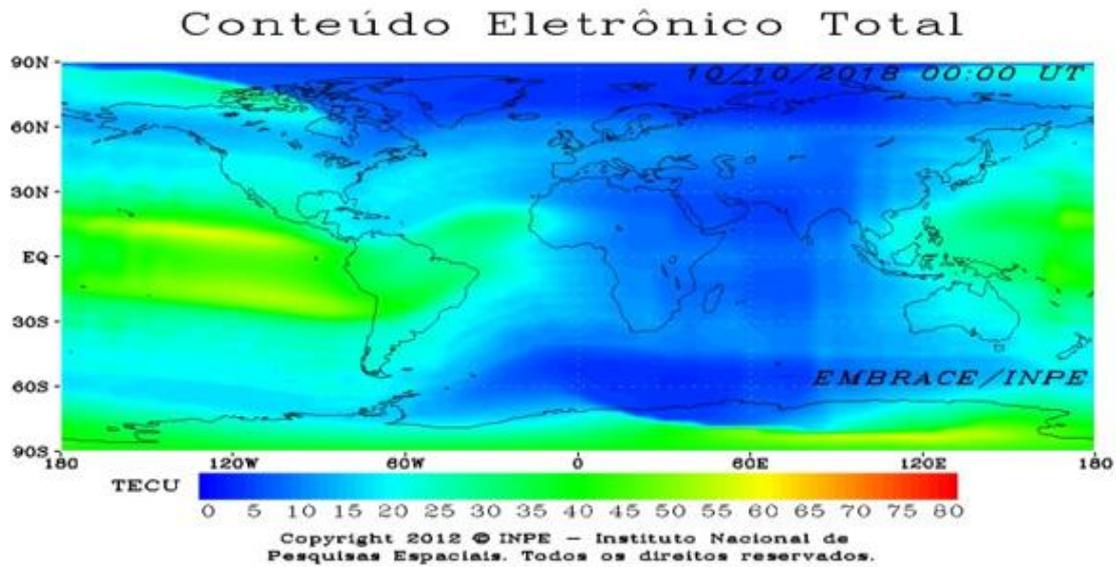
2.3.2.2 SUPIM-DAVS

SUPIM-DAVS é um modelo físico da ionosfera e plasmosfera da Terra que tem sido desenvolvido nas últimas três décadas (PETRY JONAS RODRIGUES DE SOUZA, 2014; PETRY; SOUZA; VELHO,; BAILEY; SELLEK, 1990; BAILEY; BALAN, ; BAILEY; BALAN;SU, 1997). No modelo, as equações da continuidade, movimento e do equilíbrio de energia, dependentes do tempo e são resolvidas ao longo das linhas de campo magnético fechadas para calcular os valores das densidades, fluxos e temperaturas dos elétrons e dos íons O⁺, H⁺, He⁺, N₂⁺⁺, O₂⁺ e NO⁺. O SUPIM-DAVS inclui uma representação do campo magnético da Terra que é um dipolo excêntrico (BAILEY; SELLEK; RIPPETH, 1993), porém a excentricidade faz o sistema de coordenada magnética ser diferente do geográfico.

Os principais processos físicos e químicos do modelo são: produção de íons devido à radiação solar EUV, produções e perdas dos íons devido às reações químicas entre os constituintes ionizados e neutros, difusões ambi-polar e térmica, colisões íon-íon e íon-partícula neutra, ventos termosféricos, deriva eletromagnética (ExB), condução térmica, aquecimento fotoelétrico, aquecimento friccional e um grande número de aquecimentos locais e mecanismos de esfriamento (BAILEY; BALAN; SU, 1997).

O modelo ionosférico SUPIM-DAVS, utilizado pelo INPE, descreve a distribuição da ionização na terra para períodos geomagneticamente calmos. As saídas do modelo são dadas em um plano bidimensional alinhado às linhas de campo magnético da Terra, com coordenadas magnéticas. Durante a execução do sistema, são feitas múltiplas simulações em diferentes longitudes equidistantes. Utiliza-se uma técnica de busca de vizinhos para obter os valores de concentração eletrônica em uma grade regularmente espaçada, usando interpolação ponderada pelo inverso da distância (IDW). Uma grade tridimensional contendo as concentrações eletrônicas é gerada para cada hora do dia simulado e a partir dessa grade o VTEC é calculado pela integração em altitude, sendo mostrado em um mapa bidimensional. Esse processo é executado todos os dias e prevê os valores de VTEC para a região da América do Sul com quase 24 horas de antecedência (PETRY et al., 2014).

Figura 4: Exemplo de mapa gerado pelo modelo SUPIM-DAVS



Fonte: Instituto Nacional de Pesquisas Espaciais

2.3.2.3 IGS

Outro modelo ionosférico é o GIM (Global Ionospheric Maps Mapas Globais da Ionosfera) fornecido pelo IGS (International GNSS Service). O modelo GIM começou a ser produzido em 1998, contendo valores de VTEC obtidos pela sua rede de receptores (MATSUOKA; COLLISCHONN; CAMARGO, 2010), que posteriormente são convertidos para valores TEC na direção satélite/receptor utilizando-se uma função de mapeamento (CAMARGO, 1999). Os mapas fornecem valores de VTEC em um grid com resolução espacial de $5^\circ \times 2.5^\circ$ em longitude e latitude, respectivamente, e resolução temporal de 2 horas (SCHAER; GÉODÉSISQUE, 1999; MATSUOKA; COLLISCHONN; CAMARGO, 2010).

Os mapas TEC do IGS são gerados, todos os dias, por quatro centros de análise:

- CODE - Centro de Determinação de Órbitas na Europa,
- ESA - Agência Espacial Europeia,
- JPL - Laboratório Global de Propulsão a Jato,
- UPC - Catalunha-Barcelona Tech

Todos os centros são diferentes entre si em relação às estratégias utilizadas, porém, todos os centros de análise utilizam os dados das estações GNSS ativas do IGS, originando

assim quatro arquivos IONEX (Ionosphere Exchange) que no final são combinados para gerar apenas um arquivo IONEX (mapa final). Todos os mapas (IONEX) são disponibilizados pelo IGS, com acesso livre, no endereço: <ftp://cddis.gsfc.nasa.gov/gnss/products/ionex> (SCHAER; GÉODÉSIQUE, 1999; MATSUOKA; COLLISCHONN; CAMARGO, 2010).

3 PÓS-PROCESSAMENTO

O pós-processamento é um método de posicionamento que coleta dados dos receptores simultaneamente. A principal diferença entre o pós-processamento e o processamento em tempo real é que os dados de posicionamento não são fornecidos em tempo real, sendo necessário o uso de um arquivo RINEX (Receiver Independent Exchange Format) para processá-los. O formato RINEX é um formato de troca de dados para dados brutos de estações de posicionamento de satélite que permite ao usuário pós-processar os dados recebidos para produzir um resultado mais preciso.

Para uma análise de pós-processamento, foi utilizado o software RTKPost, que está disponível no RTKLib. Que por sua vez é um pacote de software de código aberto para posicionamento padrão e preciso com GNSS (Global Satellite Navigation System).

4 MATERIAIS E MÉTODOS

4.1 Instalação

A instalação ocorre em um sistema Linux Kernel, o Elementary OS 5.0, baseado no Ubuntu 18.04.2LTS. Para a análise foi utilizado o RTKLib na versão 2.43 b33 (agosto de 2019). No RTKLib temos a possibilidade de baixar duas opções de arquivos, os arquivos que contêm as aplicações GUI (sistemas windows ou wine no linux) e outro arquivo com os códigos binários de todas as aplicações. O seguinte comando foi usado para a instalação do RTKLib no Elementary OS 5.0:

```
sudo apt install rtklib
```

4.2 Obtenção dos arquivos

RKLib tem dois aplicativos semelhantes para análise de pós-processamento, RTKPost e RNX2RTKP. O RTKpost é um aplicativo gráfico (GUI) e foca nos sistemas Windows e o RNX2RTKP é semelhante ao RTKPost mas só funciona no console (linhas de comando) com algumas vantagens em relação ao RTKPost. Ambas as aplicações necessitam de alguns arquivos, como RINEX (observação e organização), e definições do sistema de posicionamento que será analisado (GPS, GLONASS, Galileo e BeiDou).

Para a busca dos arquivos no site do IGS, foi necessário obter dia juliano e a semana GPS, dos dias escolhidos para a simulação (LABSat). Os arquivos necessários para executar o RTKPost / RNX2RTKP estão disponíveis no site do IGS com um link direto para o CDDIS da NASA (The Crustal Dynamics Data Information System) e são arquivos como: Observação RINEX, Navegação RINEX, Antena, IONEX (quando necessário), Correções de Relógio e Órbita de Satélite. Os dias simulados podem ser vistos na Tabela 1: Datas simuladas

Tabela 1: Datas simuladas

Data	Juliano	Semana GPS
10.01.02	010	1983
10.02.02	041	1987
10.03.02	069	1991
10.04.02	100	1996
10.05.02	130	2000
10.06.02	161	2005
10.07.02	191	2009
10.08.02	222	2013
10.09.02	253	2018
10.10.02	283	2022
10.11.02	314	2026
10.12.02	344	2031

Fonte: Autor

- **Orbita dos Satélites (.sp3):** esses arquivos são importantes pois o movimento do satélite girando ao redor da Terra tem uma força associada a ele que o empurra para longe da Terra. Para cada órbita, há uma velocidade para a qual a gravidade e a força centrífuga se equilibram e o satélite permanece em uma órbita estável. É necessário verificar cada arquivo para ver se o arquivo corresponde ao dia que será simulado, como indica a Figura 5

Figura 5: Verificação do dia no arquivo sp3

```

1 #cP2018 1 10 0 0 0.00000000 96 ORBIT IGS14 HLM IGS
2 ## 1983 259200.00000000 900.00000000 58128 0.00000000000000
3 + 32 G01G02G03G04G05G06G07G08G09G10G11G12G13G14G15G16G17
4 + G18G19G20G21G22G23G24G25G26G27G28G29G30G31G32 0 0
5 + 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 + 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7 + 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 ++ 2 2 2 0 2 2 2 2 2 2 2 2 2 2 2 2
9 ++ 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0
10 ++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
11 ++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
12 ++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
13 %c G cc GPS ccc cccc cccc cccc cccc cccc cccc cccc cccc|
14 %c cc cc ccc ccc cccc cccc cccc cccc cccc cccc cccc cccc
15 %f 1.2500000 1.025000000 0.00000000000 0.0000000000000000
16 %f 0.0000000 0.000000000 0.00000000000 0.0000000000000000
17 %i 0 0 0 0 0 0 0 0 0
18 %i 0 0 0 0 0 0 0 0 0
19 /* RAPID ORBIT COMBINATION FROM WEIGHTED AVERAGE OF:
20 /* cod emr esa gfz jpl ngs sio usn whu
21 /* REFERENCED TO IGS TIME (IGST) AND TO WEIGHTED MEAN POLE:
22 /* PCV:IGS14_1977 OL/AL:FES2004 NONE Y ORB:CMB CLK:CMB
23 * 2018 1 10 0 0 0.00000000
24 PG01 -1848.580725 -15106.245948 -21883.113101 -22.783715 9 6 5 137
25 PG02 -19665.298798 14877.974231 10016.905412 232.493484 10 6 6 132

```

Fonte: Autor

- Correções de Clock (.clk): após efetuar o download dos arquivos das Órbitas, efetua-se no mesmo site o download do arquivo de clock, semelhante ao arquivo de Órbitas mudando apenas a extensão de .sp3 para .clk. Os arquivos de correções de clock contém soluções para erros de sincronismo de clock, referentes à escala de tempo do GNSS (GPS, GLONASS, Galileo e outros). Uma lista com nomes de estações espalhadas pelo mundo que utilizam a Correção de Clock, está contida no arquivo .clk (CDDIS, 2019). As estações foram filtradas com base na informação contida no arquivos de clock, restando apenas as estações disponíveis com os arquivos.
- Rinex de Observação (.YYo): os arquivos de todas as estações encontradas nos arquivos de Correção de Clock (.clk) foram baixados (100 estações selecionadas). O formato RINEX consiste de arquivo-texto para os dados de observação, mensagem de navegação e dados meteorológicos, referindo-se a um determinado dia e estação receptora, contendo um cabeçalho para informações gerais e uma seção de dados. A seção de dados conta com as pseudo-distâncias de código (metros), fase (ciclos inteiros) e instante de observação (hora GPS) registrada de acordo com o relógio do receptor. O Link para download dos RINEX de Observações é <ftp://cddis.gsfc.nasa.gov/gnss/data/daily/YYYY/DDD/YYo/> onde YYYY e YY são representações do ano e DDD do dia juliano (CDDIS, 2019).
- RINEX de Navegação (.YYn) - efemérides de transmissão (broadcast): os arquivos RINEX de Navegação ou Broadcast Ephemerides (Efemérides de Transmissão) contém informações de posição, velocidade e relógio para todos os satélites da constelação de GPS para cada dia. Os RINEX contém todas as

mensagens de transmissão necessárias para o pós-processamento. As efemérides de transmissão podem ser acessadas através do link: <ftp://cddis.gsfc.nasa.gov/gnss/data/daily/YYYY/DDD/YYn/brdcDDD0>. YYn.Z (CDDIS, 2019).

- Antena (.atx): os arquivos para calibração de antena para o sistema global de navegação por satélite (GNSS) estão no formato ANTEX (.atx) (IGS, 2019) tendo como objetivo correções absolutas do centro de fase (PCVs). O centro da fase é definido como a fonte aparente de radiação dependendo do ângulo de elevação, azimute e da frequência do sinal. O nome do arquivo de Correção de Antena utilizado para análise de pós-processamento de um dia específico pode ser encontrado no arquivo de clock (.clk) e esses arquivos de antena podem ser encontrados em: ftp://ftp.igs.org/pub/station/general/pcv_archive/.

Figura 6: Verificação do arquivo de antena

```

163 ZAMB 34601M001      5415352950 2917210152 -168588863450LN ST
164 ZECK 12351M001      3451174412 3060335619 439195576450LN STA
165   31                # OF SOLN SATS
166 G01 G02 G03 G05 G06 G07 G08 G09 G10 G11 G12 G13 G14 G15 G16 PRN LI
167 G17 G18 G19 G20 G21 G22 G23 G24 G25 G26 G27 G28 G29 G30 G31 PRN LI
168 G32                PRN LIST
169 G      igs14_1977.atx      SYS / PCVS APPLIED
170                END OF HEADER
171 AR GPST 2018 01 10 00 00 0.000000 2 -6.600484788507e-09 0.000000
172 AR ABMF 2018 01 10 00 00 0.000000 2  2.554838786609e-08 1.11379
173 AR ADIS 2018 01 10 00 00 0.000000 2 -2.933009095708e-04 6.62179
174 AR ALBH 2018 01 10 00 00 0.000000 2 -2.275178436348e-04 1.19827
175 AR ALGO 2018 01 10 00 00 0.000000 2 -7.740806889439e-06 2.20700
176 AR ALIC 2018 01 10 00 00 0.000000 2  2.681524390549e-08 2.946144
177 AR ALRT 2018 01 10 00 00 0.000000 2 -4.422341597811e-08 4.31533
178 AR ANKR 2018 01 10 00 00 0.000000 2 -1.118517217107e-09 6.91564
179 AR AREQ 2018 01 10 00 00 0.000000 2 -2.004850059198e-04 2.53967
180 AR ARTU 2018 01 10 00 00 0.000000 2 -7.938205521491e-08 1.76503
181 AR BJFS 2018 01 10 00 00 0.000000 2 -3.209282046171e-08 3.872337
182 AR BOGT 2018 01 10 00 00 0.000000 2  3.161204241133e-09 1.53704
183 AR BOB1 2018 01 10 00 00 0.000000 2 -3.601958801751e-08 7.14261

```

Fonte: Autor

- IONEX - Mapas de TEC (.YYi): Arquivos IONEX contém mapas de conteúdo total de elétrons vertical da ionosfera (VTEC) e valores diários de polarização do código diferencial de satélite GNSS (DCB) derivados dos dados do GNSS de dupla frequência. Esses mapas são gerados diariamente e serão utilizados arquivos IONEX gerados pela simulação ionosférica do INPE que é baseada no modelo SUPIM-DAVS e pelo IGS que é baseado na combinação de modelos (GIM) e pode ser acessado em: <ftp://cddis.gsfc.nasa.gov/gnss/products/ionex/YYYY/DDD/igsgDDD0.YYi.Z> (CDDIS, 2019).

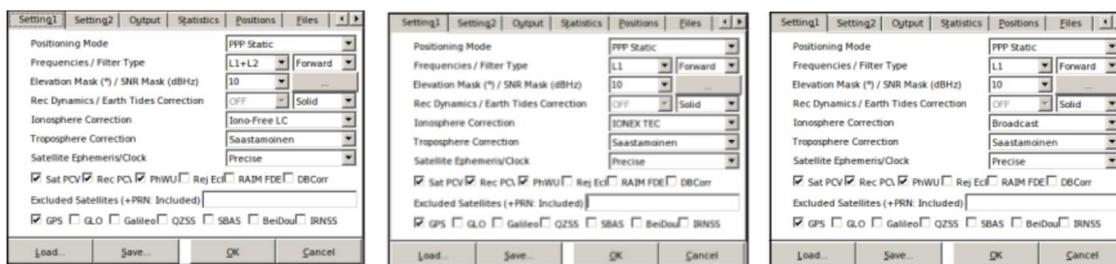
Após efetuar o download e organização de todos os arquivos necessários para a análise de pós-processamento dos dias escolhidos, o próximo passo é a configuração da aplicação presente no RTKLib.

4.3 Configuração

O RTKPost e RNX2RTKP usam os mesmos arquivos da etapa anterior para fazer suas configurações. Por possuírem muitas opções e diversas combinações para análise de pós-processamento, inicialmente optou-se por fazer as configurações no aplicativo RTKPost, pois ele possui uma interface gráfica (GUI) com maior intuitividade, favorecendo o entendimento. Três configurações foram analisadas para avaliar as correções ionosféricas em receptores GPS de frequência única em comparação com as correções ionosféricas em receptores de dupla frequência. Os conjuntos de configuração foram usados para correções isentas de íons (dupla frequência), Broadcast (modelo Klobuchar) e IONEX TEC (modelos SUPIM-DAVS e GIM).

As configurações do RTK-Post são separadas por guias, cada uma com sua função específica. A figura abaixo mostra as configurações usadas para os três métodos de correção de frequência dupla (esquerda), arquivo IONEX (centro) e Klobuchar (direita).

Figura 7: Configuração do RTKPost para os diferentes modelos



Fonte: Autor

A guia Setting1 é uma das configurações mais importantes, podendo definir o número de frequências, o modo de posicionamento, a máscara de elevação, SNR (Signal-to-Noise Ratio), correções ionosféricas, troposféricas e terrestres de marés, o GNSS sistema de navegação, entre outras configurações. As configurações usadas para avaliar as correções foram:

- **Positioning Mode (Modo de Posicionamento):** Os receptores GNSS possuem diferentes modos de posicionamento e de navegação, onde cada modo possui seu

grau de complexidade, precisão e exatidão. O PPP (Posicionamento de Ponto Preciso) é uma evolução do modo de posicionamento de um único ponto (SPP), tem como característica estimar órbitas e clocks de satélites através de uma rede de estações GNSS de referência distribuídas em todo o globo, porém apenas uma única estação é processada por vez (TEUNISSEN; MONTENBRUCK, 2017; SEEBER, 2003). A principal diferença entre o modo estático (static) e o cinemático (kinematic) é que no primeiro as coordenadas são ajustadas para todo o intervalo de observação, enquanto no segundo o ajuste é por época.

- Frequencies/Filter Type (Frequências e tipo de filtro): Frequência simples (L1) são utilizadas para sistemas GPS de apenas uma frequência e frequência dupla (L1+L2) para sistemas GPS de frequência dupla. O filtro no modo forward, os dados de observação são processados começando no início dos dados e continuando até o fim.
- Elevation Mask/SNR Mask (Máscara de Elevação/Máscara Relação Sinal Ruído): Os satélites serão excluídos se estiverem abaixo de um determinado ângulo de elevação ou com baixa relação sinal ruído (SNR).
- Rec Dynamics/Earth Tides Correction (Rec Dinâmico/Correção das Marés da Terra): O Rec Dinâmico só é ligado quando há a utilização de Rover (veículo de exploração espacial), e no caso esse posicionamento dinâmico foi desligado. A maré da Terra é um fenômeno da deformação do corpo elástico da Terra causado pela força de atração gravitacional da Lua e do Sol (MELCHIOR, 1983). Os efeitos causados pelas marés terrestres não são cruciais para o posicionamento relativo, porém, utilizando o modo de posicionamento PPP todas as fontes de erros devem ser consideradas pois o método de posicionamento tenta determinar a posição absoluta de um ponto com um único receptor GPS (HU et al., 2005). O modelo padrão das marés terrestres sólidas é o IERS92 (YIN et al., 2010).
- Ionosphere Correction (Correção Ionosférica): Os softwares para análise de pós-processamento do RTKLib disponibilizam muitas opções de correção ionosférica. As utilizadas neste trabalho são a correção Iono-Free LC, para receptores de dupla frequência (ion-free); Broadcast, para receptores de frequência simples utilizando o modelo de Klobuchar; IONEX TEC, também para receptores de frequência simples mas utilizando arquivos IONEX (mapas de VTEC).
- Troposphere Correction (Correção Troposférica): O atraso troposférico depende do caminho do sinal através da atmosfera neutra e, portanto, pode ser modelado em função do ângulo de elevação do satélite. Devido às diferenças entre os perfis atmosféricos dos gases secos e vapor de água, é melhor usar diferentes mapeamentos para os componentes secos e úmidos. O modelo Saastamoinen (SAASTAMOINEN, 2006) é uma opção em muitos pacotes de software GPS e é considerado o modelo hidrostático mais preciso. A atmosfera seca pode ser modelada a partir da pressão e temperatura da superfície usando as leis dos gases ideais. O componente úmido é mais imprevisível e difícil de modelar; portanto, para o posicionamento de alta precisão, esse atraso é estimado juntamente com as coordenadas (SUBIRANA; ZORNOZA; HERNÁNDEZ-PAJARES, 2013).
- Satellite Ephemeris/Clock (Efemérides de Satélite/Relógio): A utilização do modo de posicionamento preciso por ponto (PPP) nos faz altamente dependentes da utilização de efemérides precisas (opção Precise) e dos produtos de clock

usados pelos satélites (MOHAMMED et al., 2018). O modo Preciso utilizado nas efemérides e clock tem como função efetuar um cálculo mais preciso das órbitas de satélite reais, obtidas usando dados coletados em estações de rastreamento espalhadas pelo mundo.

- Sat PCV (Sistema Autenticador e Transmissor da Variação do Centro da Fase da Antena): Definida para poder utilizar arquivo de Antena.
- Rec PCV (Gravação da Variação do Centro da Fase da Antena): Definida para poder utilizar arquivo de Antena.
- PhWindup (Conclusão de Fase): Definida se a correção de conclusão de fase para os modos PPP é aplicada ou não.
- Navigation System (Sistema de Navegação): Seção para definir o sistema de navegação GNSS. O sistema de análise utilizado neste trabalho foi o GPS.

Na guia Setting2, a opção Integer Ambiguity Resolution (GPS) foi desligada, desabilitando estratégias ambíguas de resolução inteira para GPS. O posicionamento do GPS no nível de precisão centimétrico é baseado na medição da fase da onda portadora, o que torna o observável ambíguo de todo um número de ciclos entre as antenas do satélite e receptor no momento da primeira observação, denominado ambigüidade do GPS. A guia Output contém as configurações de saída para o arquivo gerado a partir do pós-processamento (.pos). Foi definido que os arquivos de saída teriam as coordenadas no formato XYZ-ECEF (sistema "convencional terrestre" ou centrado na terra) com hora no formato UTC de três casas decimais para possibilitar a estimativa do erro com base no cálculo da distância euclidiana aos locais (conhecidos) para as estações GNSS usadas.

Na aba Positions, a opção de usar um arquivo de antena, Antenna Type, foi selecionada na opção *. Outra configuração muito importante é a opção Arquivos, onde os arquivos da antena (.atx) e o arquivo IONEX são carregados (quando o modo de correção ionosférica IONEX TEC é selecionado). As outras guias e configurações não foram alteradas.

4.4 Execução

Para rodar o software de pós-processamento com correção ionosférica baseado no IONEX TEC precisamos dos arquivos no formato IONEX gerados pela simulação ionosférica do INPE, com base no modelo SUPIM-DAVS, e pela simulação ionosférica do IGS, baseada no Modelo GIM. Os arquivos IGS IONEX podem ser baixados do site IGS descrito na Etapa 2: Obtenção dos arquivos. O IONEX da simulação ionosférica do INPE precisava

ser gerado por meio de simulação que ocorre por meio de linhas de comando usando um cluster de computadores. Obtidos todos os arquivos necessários para o pós-processamento dos dias escolhidos e realizada a configuração do RTKPost, optou-se por utilizar o software RNX2RTKP.

Embora o RTKPost tenha uma interface gráfica e seja mais intuitivo para o usuário, sua execução acabaria sendo dificultada pela quantidade de estações e arquivos a serem manipulados. RNX2RTKP é uma versão do RTKPost sem uma interface gráfica e, embora também execute um arquivo de observação por vez, é mais fácil automatizar as operações porque a linha de comando é usada. Além disso, o RNX2RTKP é capaz de executar arquivos de configuração exportados do RTKPost usando o botão Salvar. Após configurar o RTKPost na Etapa 3 - Configuração do software para as três formas de correção (ion-free, Klobuchar e IONEX), as configurações foram salvas em arquivos de texto com a extensão ".conf". Como o RNX2RTKP pode usar arquivos RTKPost pré-configurados em sua execução, foram criados quatro scripts, um para cada método de correção, a fim de otimizar a execução.

As entradas do script são o dia juliano, os arquivos de configuração e outros arquivos necessários mencionados acima. Foi criada uma função de mapeamento de diretórios, para que o programa mapeie o diretório que contém os arquivos de observação e armazene seus nomes em uma lista. Cada linha da lista contém o nome de um arquivo de observação para cada estação, que é usado em um loop para percorrer toda a lista e invoca o programa RNX2RTKP variando seus parâmetros. Nota. O arquivo de saída recebe o mesmo nome do arquivo de observação, mudando seu tipo para ".pos". A cada script é gerado um diretório, sendo que cada diretório contém 100 arquivos de pós-processamento, referentes ao arquivo de observação de cada estação. No total, usando os quatro métodos de correção (frequência dupla, Klobuchar, IONEX SUPIM-DAVS, IONEX GIM), haverá quatro arquivos, com correções diferentes, referentes a uma única estação selecionada.

5 ANÁLISE E RESULTADOS

Os arquivos gerados do pós-processamento possuem a extensão ".pos" e contém cabeçalho e dados de informações. O cabeçalho contém informações sobre a versão da aplicação, os arquivos utilizados, data e hora do início e o fim da execução e as configurações utilizadas. A parte dos dados contida no arquivo está no formato configurado conforme indicado na etapa de configuração, com as coordenadas x, y e z estimadas de trinta em trinta segundos, partindo da data e hora de início até a data e hora final.

Considerando a resolução temporal de 2 horas para os mapas de VTEC do IGS, e de 1 hora para a simulação ionosférica do INPE, baseada no modelo SUPIM-DAVS, decidiu-se optar por uma comparação a cada duas horas, iniciando 00h00 até 22h00. Assim, ainda que os resultados do pós-processamento sejam apresentados a cada 30 segundos, apenas os posicionamentos coincidentes com os horários estabelecidos foram utilizados, ou seja, 12 resultados por dia foram avaliados.

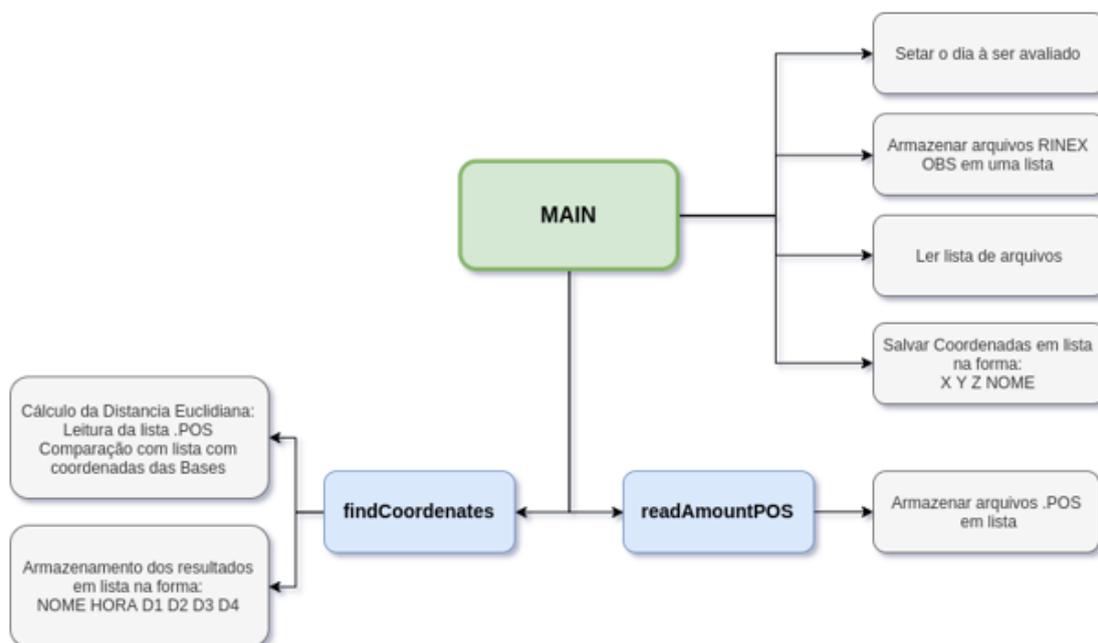
Como os arquivos de pós-processamento contém as coordenadas x, y, e z de cada amostra, optou-se por escolher o método de cálculo da Distância Euclidiana entre o posicionamento realizado e as localidades conhecidas de cada estação GNSS. Os arquivos RINEX de Observação de cada estação contém as coordenadas (x, y e z). Assim, cada método de correção teria uma certa distância (ou erro) estimado em metros em relação a estação base. Também, notou-se que algumas estações, em determinadas horas, não conseguiam enviar sua amostra de pós-processamento. Por conta disso adicionou-se uma tolerância de 10 minutos para cada hora, totalizando assim uma avaliação com base em 00h00 (até 00h10) até 22h00 (até 22h10).

A Distância Euclidiana é a menor distância entre dois pontos num plano, que utiliza do teorema de Pitágoras para cálculo. Inicialmente para a análise foi desenvolvido um programa em linguagem C. Posteriormente, o algoritmo foi migrado para linguagem de programação Python, principalmente pelo fato de ela trabalhar mais facilmente com listas. O funcionamento básico do software está demonstrado no Fluxograma da Figura 3.10 e para maior entendimento de seu funcionamento, o código fonte encontra-se no Apêndice C.

A única entrada que software recebe é o dia Juliano, e com isso, ele é capaz de acessar o diretório onde encontram-se os arquivos RINEX de Observação de cada estação. Estes

arquivos contém as informações sobre a base, incluindo as coordenadas (x, y e z) que são lidas e armazenadas em uma lista. Cada método de correção possui sua pasta com seus arquivos de pós-processamento gerados pelo RNX2RTKP. É feita então a leitura destes arquivos e também ocorre seu armazenamento em listas. Essas listas são lidas e busca-se em cada arquivo de pós-processamento as coordenadas a cada duas horas (00h00 até 22h00) com tolerância de 10 minutos. Também, ocorre uma busca na lista de estações GNSS utilizadas para encontrar as respectivas coordenadas. Por fim, é efetuado o cálculo da distancia euclidiana entre as coordenadas da estação e as coordenadas do pós-processamento. Tem-se então uma estimativa de erro a cada duas horas para cada uma das estações. Esses dados são armazenados em lista de listas, como pode ser visto na Figura 9

Figura 8: Fluxograma básico da mecânica do funcionamento do Software de Avaliação



Fonte: Autor.

5.1 AVALIAÇÃO NUMÉRICA

O resultado do software implementado em Python pode ser visto (parcialmente) pela Figura 9. Como previsto, o resultado nos mostra a distância (ou erro), em metros de cada uma das quatro diferentes formas de correção para cada estação GNSS começando

na hora 00h00 e indo até a hora 22h00 com resolução temporal de 2 horas. A Tabela 2: Erro médio de cada método de correção para os dias avaliados informa a média das distâncias acumuladas, ou erro médio, de cada um dos métodos de correção para os dias avaliados.

Tabela 2: Erro médio de cada método de correção para os dias avaliados

Data	Dual Frequency (m)	Klobuchar (m)	SUPIM-DAVS (m)	GIM (m)
10.01.2018	0,91962	3,72282	7,50705	4,09374
10.02.2018	0,94719	3,84452	7,61696	4,14246
10.03.2018	1,02883	4,02970	8,35225	4,44542
10.04.2018	0,93333	3,84254	9,12165	4,59838
10.05.2018	1,00341	3,41075	9,66887	4,63577
10.06.2018	0,88518	3,38392	10,99009	4,04849
10.07.2018	0,86637	3,10855	8,86025	3,15241
10.08.2018	0,92836	2,91003	7,43534	3,08802
10.09.2018	0,93162	3,52176	9,08037	4,14424
10.10.2018	0,85419	3,84419	8,69792	4,04793
10.11.2018	1,00656	4,01972	4,34822	4,51752
10.12.2018	0,99414	3,70765	3,82055	3,43482

Fonte: Autor.

Figura 9: Resultado numérico da avaliação feita pelo software criado

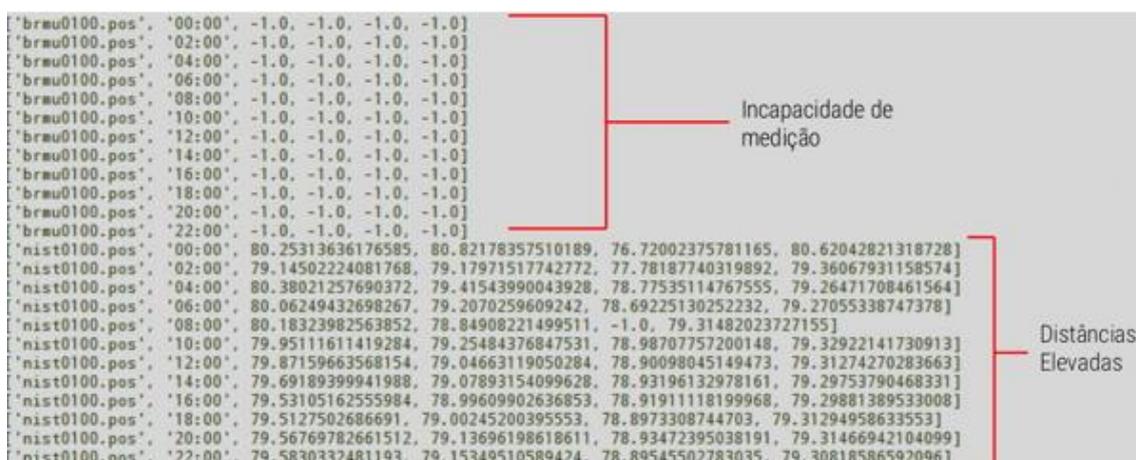
Estação	Hora	GIM	KLOBUCHAR	DUAL FREQUENCY
mac10100.pos	'00:00'	4.591805147888944,	3.126340208601937,	4.1770752553914985, 1.4262751303806587]
mac10100.pos	'02:00'	3.774549713596879,	1.4836819366952763,	-1.0, 0.8679095916993244]
mac10100.pos	'04:00'	4.089466329690383,	1.2676574105222218,	3.3929567544414376, 0.8731746099489709]
mac10100.pos	'06:00'	3.5147004784757407,	1.2130525919303967,	2.9868870963948932, 0.85675449231177]
mac10100.pos	'08:00'	2.265681633021873,	1.033329555261261,	3.1495402263164047, 0.8773986090080432]
mac10100.pos	'10:00'	2.4328210582038103,	1.2811445427864472,	2.984736142220127, 0.8956197234077452]
mac10100.pos	'12:00'	2.094964918899922,	1.274518846209181,	2.9437807493362715, 0.9112981339523653]
mac10100.pos	'14:00'	2.140054174696854,	1.0482697647151182,	2.780167789619694, 0.9091313764944559]
mac10100.pos	'16:00'	2.0669844390985945,	0.6762221602303109,	2.6813255038540325, 0.9044121292164476]
mac10100.pos	'18:00'	1.999928824318885,	0.7347835390945662,	2.624136326134108, 0.8987622935424999]
mac10100.pos	'20:00'	1.9246427048051322,	0.6616418738773517,	-1.0, 0.8853492751597675]
mac10100.pos	'22:00'	1.8178714452271296,	0.7527618480332756,	2.44385956610737, 0.880054822149028]
naw10100.pos	'00:00'	8.692930754073362,	12.758445902009607,	16.92535575096127, 2.300716068626401]
naw10100.pos	'02:00'	5.046293679430192,	5.819626677974605,	14.947253329947939, 0.973637248272797]
naw10100.pos	'04:00'	3.9207813226147,	4.941115410719517,	14.19578406622556, 0.9040614476942026]
naw10100.pos	'06:00'	3.9228835557373634,	6.316663382551703,	13.608106669553361, 0.8735322376093546]
naw10100.pos	'08:00'	3.452585687431494,	5.2450000754644295,	13.707938836656863, 0.8472199129204276]
naw10100.pos	'10:00'	3.274767571541271,	5.516880373995425,	-1.0, 0.8547439914621244]
naw10100.pos	'12:00'	2.9030190165146483,	5.178905152742042,	13.612543428286898, 0.8653172539846418]
naw10100.pos	'14:00'	2.343212747190608,	4.75878201317404,	0.8404873175837299, 0.8541405099833838]
naw10100.pos	'16:00'	2.2574799156574845,	4.748742302837258,	13.4425064706614, 0.8724936966172141]
naw10100.pos	'18:00'	2.0195140155280433,	4.506862204753726,	13.331053366215194, 0.8768456705256054]
naw10100.pos	'20:00'	1.7843803320118536,	4.248632427432368,	13.273413872072407, 0.8739117750704393]
naw10100.pos	'22:00'	1.6601364942642884,	3.885365537496697,	13.009154336808876, 0.8666649005248794]
flin0100.pos	'00:00'	3.4747786940445757,	8.298819945771253,	6.299891993044341, 0.8858647074236576]

Fonte: Autor.

Numericamente é possível perceber que o modelo SUPIM-DAVS apresentou uma pior correção do erro ionosférico, apresentando uma maior distância, em metros, para cada estação, seguido pelo modelo GIM utilizado pelo IGS e tendo o Klobuchar como a melhor correção dentre as alternativas analisadas para receptores GPS de frequência simples. As correções para o erro ionosférico em receptores GPS de frequência simples foram analisadas e comparadas com a correção utilizando frequência dupla.

O resultado numérico também nos forneceu informações sobre estações que apresentaram algum problema no pós-processamento (Figura 10). Por convenção, optou-se por desconsiderar distâncias que apresentaram problemas na hora do seu cálculo. O problema de incapacidade de medição da distância (erro em metros), pode ser ocasionado pela inexistência de amostras para a hora avaliada, consequência de cintilação ou outros problemas ionosféricos, como ocorrência de tempestades geomagnéticas ou bolhas ionosféricas. A Tempestade Geomagnética ou Tempestade Solar, é uma perturbação no campo magnético da terra ocasionada por uma nuvem magnética, podendo ou não ser acompanhada de uma onda de choque do vento solar. Bolhas ionosféricas ocorrem na região do equador magnético, sendo uma diminuição da densidade de plasma ionosférico. Outro ponto interessante a ser analisado na Figura 10 são as elevadas distâncias, em metros, que foram calculadas. Essas, por sua vez, podem ser ocasionadas por fatores como erro nas coordenadas da estação contidas no RINEX de Observação (coordenadas inexistentes), resultando em uma elevada distância calculada pela fórmula da distância euclidiana ou as estações localizadas de forma desfavorável, prejudicando sua análise de pós-processamento e conseqüentemente o cálculo da distância.

Figura 10: Exemplo de estações com amostragem alteradas



Fonte: Autor.

5.2 AVALIAÇÃO GRÁFICA

Para solucionar o problema das estações alteradas, descrito anteriormente, optou-se por ter como referência as distâncias providas da correção em receptores GPS de dupla frequência que possuem uma melhor exatidão na correção. Decidiu-se então optar por remover as estações que foram impossibilitadas de obter um posicionamento. Além disso, definiu-se como 5 metros o erro máximo a ser tolerado para a correção ionosférica de dupla frequência, visto que erros acima desse valor evidenciam outros tipos de problema provavelmente não relacionados à ionosfera.

A Figura 11 apresenta o erro médio de cada uma das estações espalhadas pelo mundo, para cada um dos métodos de correção utilizados. O tamanho dos círculos preenchidos é proporcional ao erro ionosférico (maior distância).

Observa-se que dentre as correções utilizadas em dispositivos de frequência simples, o modelo SUPIM-DAVS, utilizado pelo INPE, e a combinação de modelos GIM, utilizado pelo IGS, apresentaram piores correções ionosféricas se comparados com o modelo Klobuchar. Para ter uma melhor avaliação das correções para cada estação em relação ao tempo, foi gerada a Figura 12.

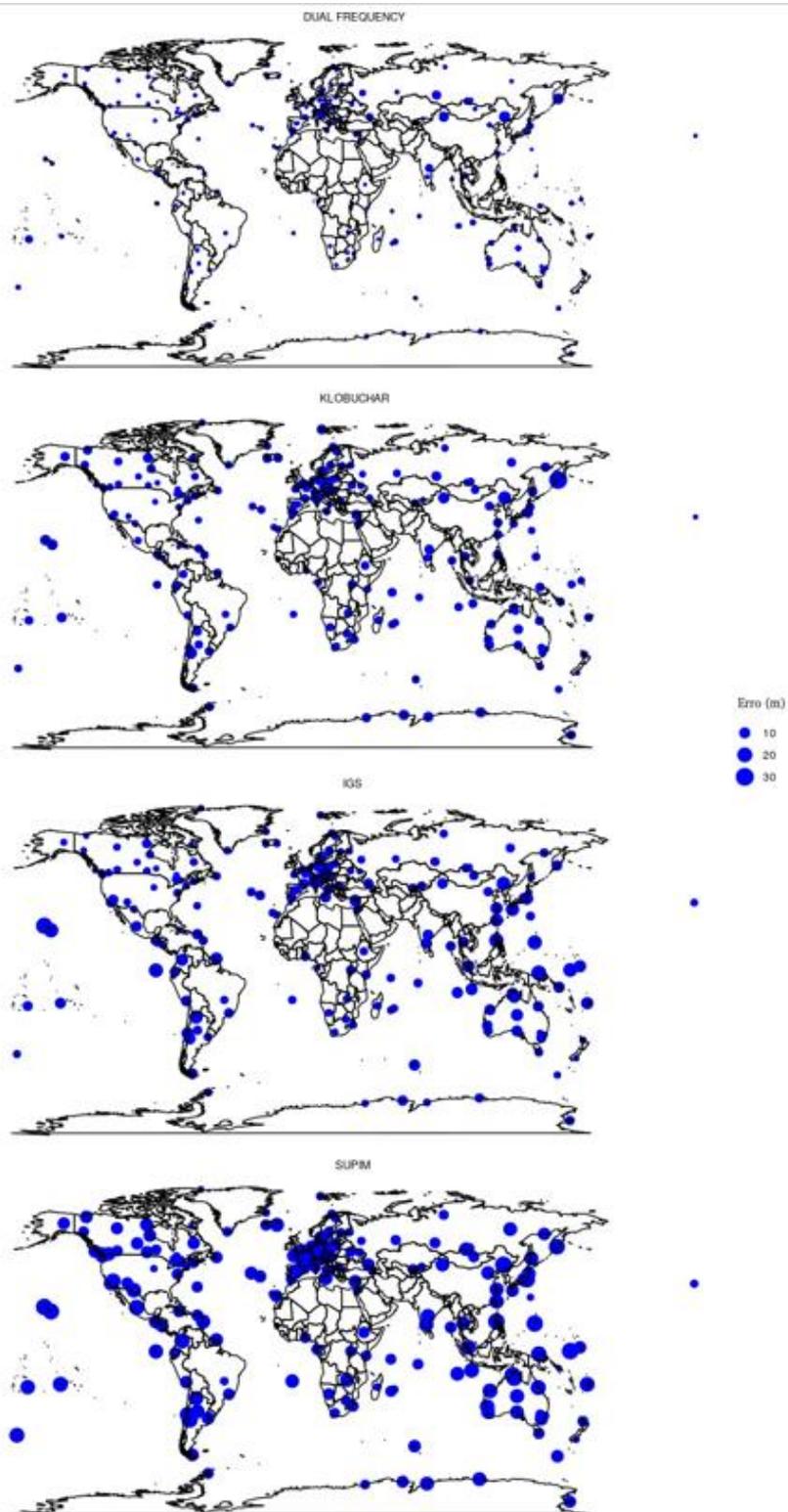
A Figura 12 apresenta o erro médio de cada estação, com referência mensal, para cada uma das correções ionosféricas avaliadas, apresentando a possibilidade de analisar

eventuais anomalias sazonais. Os meses apresentados são para o ano de 2018, onde a transição de cores (do azul para o vermelho) representa os erros de cada estação. As lacunas em branco indicam que há ausência de arquivos de observações de algumas estações para determinado dia.

Através da Figura 12 percebe-se que algumas estações continuam apresentando valores elevados de distância, independentemente da correção ionosférica aplicada, como por exemplo as estações *textitieng*, *mdvj*, *urum*, *wtzt*, entre outras. Por apresentarem uma similaridade em todos os meses, para todas as correções, possivelmente foram estações que apresentaram uma distância menor que 5 metros para a correção de dupla frequência mas que apresentaram distâncias elevadas nas correções utilizadas em receptores de frequência simples. Além disso, nota-se uma variação constante em alguns meses, mais notável no meses de maio e junho. O SUPIM-DAVS foi o mais afetado pela anomalia sazonal, que por estar presente em todas as correções, possivelmente oriunda de uma tempestade geomagnética que aconteceu no dia avaliado.

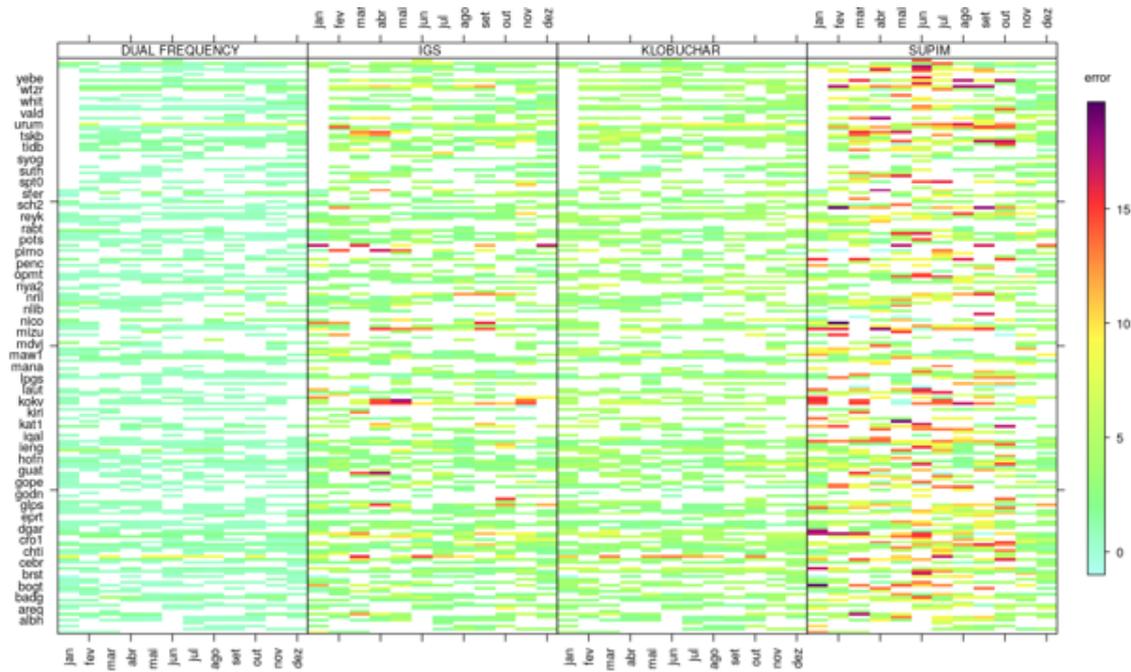
Comparando as Figura 11 e Figura 12 nota-se que o modelo Klobuchar apresentou ótimos resultados, mostrando-se mais eficiente que os modelos ionosféricos SUPIM-DAVS e GIM, e justificando seu uso atual em dispositivos utilizados no cotidiano. Outra possível explicação para os piores resultados para correção ionosférica através de arquivos IONEX seria algum problema no software RTKLib que prejudique os posicionamentos quando esses arquivos são utilizados.

Figura 11: Representação da média do erro por estação com distribuição global



Fonte: Autor

Figura 12: Representação do erro distribuído por estações em função dos meses



Fonte: Autor.

6 CONCLUSÃO

Na atualidade, a maior fonte de erro em receptores GNSS é a refração ionosférica, que depende do conteúdo total de elétrons (TEC - Total Electron Content) na camada ionosférica, que por sua vez é influenciado pelo ciclo solar, as estações do ano, a hora local, a localização geográfica, o campo magnético da Terra, e outras variáveis. Receptores GPS de frequência dupla conseguem corrigir este erro utilizando uma combinação linear (ion-free) de suas frequências, enquanto que receptores de frequência simples necessitam do uso de modelos ionosféricos para minimizar os efeitos da refração ionosférica. No presente trabalho foi efetuada a avaliação de três modelos ionosféricos: o modelo SUPIM-DAVS, utilizado na simulação ionosférica do INPE, modelo GIM, utilizado pelo IGS e o modelo de Klobuchar. Os três modelos tiveram como base de comparação a correção ionosférica utilizada por receptores GPS de frequência dupla que conseguem ter uma eficácia de até 99% na correção dos efeitos da refração ionosférica.

A avaliação dos modelos ionosféricos presentes neste trabalho pode possibilitar pesquisas para melhorar os modelos, deixando-os mais eficazes e precisos para o posicionamento utilizado no dia a dia e em áreas e serviços que necessitam de um posicionamento mais exato em equipamentos mais simples. Para executar a avaliação, precisou-se obter os arquivos gerados pelo software de pós-processamento presente no RTKLib. Criou-se um programa em Python capaz de avaliar, pela distância euclidiana, os arquivos gerados pelo pós-processamento. O modelo ionosférico SUPIM-DAVS, utilizado pela simulação ionosférica do INPE nunca havia sido avaliado em termos de eficácia para correção do erro ionosférico.

Conhecida a eficácia de cada modelo pela avaliação do software desenvolvido, levantase a possibilidade de um estudo mais abrangente, ampliando a avaliação para uma quantidade maior de datas, podendo também analisar sua eficácia através análises comparativas, para poder identificar com clareza possíveis modificações que podem ser efetuadas no modelo, em particular no modelo desenvolvido e operado pelo INPE, a fim de melhorar sua acurácia no posicionamento. Ademais, os modelos ionosféricos são essenciais para o uso de receptores GNSS de frequência simples, visto que por serem economicamente mais viáveis que receptores GNSS de frequência dupla, são popularmente utilizados em diversos dispositivos do cotidiano.

REFERÊNCIAS BIBLIOGRÁFICAS

- BENTO, D. V. Avaliação de Correções do Erro Ionosférico em Receptores GNSS de Frequência SJJKKimples https://github.com/bentodvictor/RTKLib-Post_Processing_Guide/blob/dev/reference/TCC%20-%20Victor%20Dallagnol%20Bento.pdf
- BAILEY, G.; BALAN, N. A low-latitude ionosphere-plasmasphere model, step handbook on ionospheric models. **RW Schunk**, v. 173.
- BAILEY, G.; BALAN, N.; SU, Y. The sheffield university plasmasphere ionosphere modela review. **Journal of Atmospheric and Solar-Terrestrial Physics**, Elsevier, v. 59, n. 13, p. 1541–1552, 1997.
- BAILEY, G.; SELLEK, R. A mathematical model of the earth's plasmasphere and its application in a study of he (+) at $l=3$. In: **Annales Geophysicae**. [S.l.: s.n.], 1990. v. 8, p. 171–189.
- BAILEY, G.; SELLEK, R.; RIPPETH, Y. A modelling study of the equatorial topside ionosphere. In: **Annales geophysicae**. [S.l.: s.n.], 1993. v. 11, p. 263–272.
- BEIDOU, I. **BeiDou navigation satellite system signal in space interface control document open service signal B1I (Version 1.0)**. [S.l.]: China Satellite Navigation Office, 2012.
- CAMARGO, P. d. O. Modelo regional da ionosfera para uso em posicionamento com receptores gps de uma freüência. 1999.
- CDDIS. **CDDIS - The Crustal Dynamics Data Information System**. 2019. Disponível em: <<https://cddis.nasa.gov>>.
- HOLLWEG, G. V. et al. Preliminary evaluation of accuracy improvement in arduino-based single-frequency gps receiver using ionosphere simulation data.
- HU, C.-W. et al. Data processing for gps precise point positioning. **Transactions of Nanjing University of Aeronautics & Astronautics**, Editorial Dept. of Journal of NUA.A., 2005.
- IGS. **ANTEX: The Antenna Exchange Format**. 2019. Disponível em: <<https://kb.igs.org/hc/en-us/articles/216104678-ANTEX-format-description>>.
- KLOBUCHAR, J. A. Ionospheric time-delay algorithm for single-frequency gps users. **IEEE Transactions on aerospace and electronic systems**, IEEE, n. 3, p. 325–331, 1987.
- LEICK, A.; RAPOPORT, L.; TATARNIKOV, D. **GPS satellite surveying**. [S.l.]: John Wiley & Sons, 2015.
- MARQUES, H. A. et al. Análise dos efeitos ionosféricos de ordem superior no ciclo solar 24 e influência no posicionamento gnss absoluto. **Boletim de Ciências Geodésicas**, Universidade Federal do Paraná, v. 20, n. 4, p. 879–901, 2014.
- MATSUOKA, M. T.; CAMARGO, P. D. O. Correção ionosférica utilizando o modelo de klobuchar e o modelo regional da ionosfera: Avaliação da performance no posicionamento por ponto. 2002.

MATSUOKA, M. T.; COLLISCHONN, C.; CAMARGO, P. D. O. Análise do desempenho do modelo global da ionosfera do igs no posicionamento por ponto durante períodos de tempestades geomagnéticas: Estudo de caso para 29-30 de outubro de 2003 na região sul do brasil. 2010.

MELCHIOR, P. The tides of the planet earth. **Organic Photonics and Photovoltaics**, 1983.

MISRA, P.; ENGE, P. Global positioning system: signals, measurements and performance second edition. **Global Positioning System: Signals, Measurements And Performance Second Editions**, 2006.

MOHAMMED, J. et al. An assessment of the precise products on static precise point positioning using multi-constellation gnss. In: IEEE. **2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)**. [S.l.], 2018. p. 634–641.

MOLDWIN, M. **An introduction to space weather**. [S.l.]: Cambridge University Press, 2008.

PETRY, A.; SOUZA, J. R. de; VELHO, H. F. de C. Sistema para previsao operacional da dinamica da ionosfera baseado no modelo supim.

PETRY, A. et al. First results of operational ionospheric dynamics prediction for the brazilian space weather program. **Advances in Space Research**, Elsevier, v. 54, n. 1, p. 22–36, 2014.

PETRY JONAS RODRIGUES DE SOUZA, H. F. d. C. V. A. G. P. G. J. B. A. First results of operational ionospheric dynamics prediction for the brazilian space weather program. v. 54, p. 22–36, 2014.

RINEX-WIKIPEDIA. **RINEX**. 2019. Disponível em: <<https://en.wikipedia.org/wiki/RINEX>>.

ROLLOVERGPS-WIKIPEDIA. **GPS Week Number Rollover**. 2013. Disponível em: <https://en.wikipedia.org/wiki/GPS_Week_Number_Rollover>.

SAASTAMOINEN, J. Simplified model for calculation of devolatilization in fluidized beds. **Fuel**, Elsevier, v. 85, n. 17-18, p. 2388–2395, 2006.

SCHAER, S.; GÉODÉSIQUE, S. helvétique des sciences naturelles. C. **Mapping and predicting the Earth's ionosphere using the Global Positioning System**. [S.l.]: Institut für Geodäsie und Photogrammetrie, Eidg. Technische Hochschule Zürich, 1999. v. 59.

SEEBER, G. Satellite geodesy, 2nd completely revised and extended edition. **Walter de Gruyter GmbH & Co. KG**, v. 10785, p. 303–304, 2003.

SUBIRANA, J. S.; ZORNOZA, J. J.; HERNÁNDEZ-PAJARES, M. Gnss data processing, volume i: Fundamentals and algorithms. **ESA Communications, ESTEC, Noordwijk, Netherlands**, p. 145–161, 2013.

TAKASU, T. **RTKLib: an open source program package for GNSS positioning**. 2007. Disponível em: <www.rtklib.com>.

____. **RTKLIB ver. 2.4.2 Manual**. 2013. Disponível em: <http://rtklib.com/prog/manual_2.4.2.pdf>.

TEUNISSEN, P.; MONTENBRUCK, O. **Springer handbook of global navigation satellite systems**. [S.l.]: Springer, 2017.

WEBSTER, I. R. **A regional model for the prediction of ionospheric delay for single frequency users of the Global Positioning System**. [S.l.]: University of New Brunswick, 1994.

YIN, H. et al. Effect of solid-earth-tide on gps time series. **Geodesy and Geodynamics**, Elsevier, v. 1, n. 1, p. 64–69, 2010.

APENDICE A – EXEMPLO DE ARQUIVO DE CONFIGURAÇÃO

1. # rtkpost options (2019/09/29 22:32:30, v.2.4.3 b33)
2. pos1-posmode =ppp-static #
(0:single,1:dgps,2:kinematic,3:static,4:movingbase,5:fixed,6:ppp-kine,7:ppp-static,8:ppp-fixed)
3. pos1-frequency =11 # (1:11,2:11+12,3:11+12+15,4:11+15)
4. pos1-soltype =forward # (0:forward,1:backward,2:combined)
5. pos1-elmask =10 # (deg)
6. pos1-snrmask_r =off # (0:off,1:on)
7. pos1-snrmask_b =off # (0:off,1:on)
8. pos1-snrmask_L1 =0,0,0,0,0,0,0,0
9. pos1-snrmask_L2 =0,0,0,0,0,0,0,0
10. pos1-snrmask_L5 =0,0,0,0,0,0,0,0
11. pos1-dynamics =off # (0:off,1:on)
12. pos1-tidecorr =on # (0:off,1:on,2:otl)
13. pos1-ionoopt =ionex-tec # (0:off,1:brdc,2:sbas,3:dual-freq,4:est-stec,5:ionex-tec,6:qzs-brdc,7:qzs-lex,8:stec)
14. pos1-tropopt =saas # (0:off,1:saas,2:sbas,3:est-ztd,4:est-ztdgrad,5:ztd)
15. pos1-sateph =precise # (0:brdc,1:precise,2:brdc+sbas,3:brdc+ssrapc,4:brdc+ssrcom)
16. pos1-posopt1 =on # (0:off,1:on)
17. pos1-posopt2 =on # (0:off,1:on)
18. pos1-posopt3 =on # (0:off,1:on,2:precise)
19. pos1-posopt4 =off # (0:off,1:on)
20. pos1-posopt5 =off # (0:off,1:on)
21. pos1-posopt6 =off # (0:off,1:on)
22. pos1-exclsats = # (prn ...)
23. pos1-navsys =1 # (1:gps+2:sbas+4:glo+8:gal+16:qzs+32:comp)
24. pos2-armode =off # (0:off,1:continuous,2:instantaneous,3:fix-and-hold)
25. pos2-gloarmode =on # (0:off,1:on,2:autocal)
26. pos2-bdsarmode =on # (0:off,1:on)
27. pos2-arthres =3
28. pos2-arthres1 =0.9999
29. pos2-arthres2 =0.25
30. pos2-arthres3 =0.1
31. pos2-arthres4 =0.05

32. pos2-arlockcnt =0
33. pos2-arelmask =0 # (deg)
34. pos2-arminfix =10
35. pos2-armaxiter =1
36. pos2-elmaskhold =0 # (deg)
37. pos2-aroutcnt =5
38. pos2-maxage =30 # (s)
39. pos2-syncsol =off # (0:off,1:on)
40. pos2-slipthres =0.05 # (m)
41. pos2-rejionno =30 # (m)
42. pos2-rejgdop =30
43. pos2-niter =1
44. pos2-baselen =0 # (m)
45. pos2-basesig =0 # (m)
46. out-solformat =xyz # (0:llh,1:xyz,2:enu,3:nmea)
47. out-outhead =on # (0:off,1:on)
48. out-outopt =on # (0:off,1:on)
49. out-outvel =off # (0:off,1:on)
50. out-timesys =utc # (0:gpst,1:utc,2:jst)
51. out-timeform =hms # (0:tow,1:hms)
52. out-timendec =3
53. out-degform =deg # (0:deg,1:dms)
54. out-fieldsep =
55. out-outsingl =off # (0:off,1:on)
56. out-maxsolstd =0 # (m)
57. out-height =ellipsoidal # (0:ellipsoidal,1:geodetic)
58. out-geoid =internal # (0:internal,1:egm96,2:egm08_2.5,3:egm08_1,4:gsi2000)
59. out-solstatic =all # (0:all,1:single)
60. out-nmeaintv1 =0 # (s)
61. out-nmeaintv2 =0 # (s)
62. out-outstat =off # (0:off,1:state,2:residual)
63. stats-eratio1 =100
64. stats-eratio2 =100
65. stats-errphase =0.003 # (m)
66. stats-errphaseel =0.003 # (m)

67. stats-errphasebl =0 # (m/10km)
68. stats-errdoppler =10 # (Hz)
69. stats-stdbias =30 # (m)
70. stats-stdiono =0.03 # (m)
71. stats-stdtrop =0.3 # (m)
72. stats-prnaccelh =10 # (m/s^2)
73. stats-prnaccelv =10 # (m/s^2)
74. stats-prnbias =0.0001 # (m)
75. stats-prniono =0.001 # (m)
76. stats-prntrop =0.0001 # (m)
77. stats-prnpos =0 # (m)
78. stats-clkstab =5e-12 # (s/s)
79. ant1-postype =llh # (0:llh,1:xyz,2:single,3:posfile,4:rinexhead,5:rtcm,6:raw)
80. ant1-pos1 =90 # (deg|m)
81. ant1-pos2 =0 # (deg|m)
82. ant1-pos3 =-6335367.6285 # (m|m)
83. ant1-anttype =*
84. ant1-antdele =0 # (m)
85. ant1-antdeln =0 # (m)
86. ant1-antdelu =0 # (m)
87. ant2-postype =llh # (0:llh,1:xyz,2:single,3:posfile,4:rinexhead,5:rtcm,6:raw)
88. ant2-pos1 =90 # (deg|m)
89. ant2-pos2 =0 # (deg|m)
90. ant2-pos3 =-6335367.6285 # (m|m)
91. ant2-anttype =
92. ant2-antdele =0 # (m)
93. ant2-antdeln =0 # (m)
94. ant2-antdelu =0 # (m)
95. ant2-maxaveep =0
96. ant2-initrst =off # (0:off,1:on)
97. misc-timeinterp =off # (0:off,1:on)
98. misc-sbasatsel =0 # (0:all)
99. misc-rnxopt1 =
100. misc-rnxopt2 =
101. misc-pppopt =

- 102. file-satantfile =igs14_2030.atx
- 103. file-rcvantfile =igs14_2030.atx
- 104. file-staposfile =
- 105. file-geoidfile =
- 106. file-ionofile =I2D3440.18I
- 107. file-dcbfile =
- 108. file-eopfile =
- 109. file-blqfile =
- 110. file-tempdir =
- 111. file-geexefile =
- 112. file-solstatfile =
- 113. file-tracefile =

APENDICE B – EXEMPLO DE SCRIPT DE AUTOMATIZAÇÃO PARA EXECUÇÃO DE PÓS-PROCESSAMENTO

```
1. #-----#
2. ##### Universidade Federal de Santa Maria #####
3. ##### Instituto Nacional de Pesquisas Espaciais #####
4. ##### Centro de Tecnologias #####
5. ##### Engenharia de Computação #####
6. ##### Victor Dallagnol Bento #####
7. #-----#

8. # O Script deve estar no mesmo diretório dos "Arquivos Fixos"

9. # Diretório de saída
10. SimOut='$SUPIM'

11. #---- Campo para alterações dependendo do dia que será simulado
12. # Diretório da Simulação Simulation data
13. SimData='$344'
14. # Arquivos Fixos
15. RINEX='brdc3440.18n'
16. CLK='igr20311.clk'
17. ORBIT='igr20311.sp3'
18. #-----

19. # Contador para indicar número de arquivos simulados
20. COUNT=1

21. # Lista arquivos RINEX de Observação (RINEX OBS) de cada estação
22. cd /home/bent0vict0r/Downloads/SIMULATIONS/$SimData/RINEX
23. ls > /home/bent0vict0r/Downloads/SIMULATIONS/$SimData/RinexList_SUPIM.txt
24. cd /home/bent0vict0r/Downloads/SIMULATIONS/$SimData/

25. # Leitura da lista contendo o nome dos RINEX OBS
26. while read LINHA; do
27. RINEX_OBS=$(echo $LINHA | awk '{print $1}')
```

```

28. echo $COUNT
29. COUNT=$((COUNT+1))

30. # Atribui o nome do arquivo RINEX OBS para o arquivo de saída mudando apenas sua extensão
31. OUT=${RINEX_OBS/.18o/.pos}

32. # PÓS-PROCESSAMENTO
33. rnx2rtkp -k
    /home/bent0vict0r/Downloads/SIMULATIONS/$SimData/rtkpost_ionex_supim.conf -o
    /home/bent0vict0r/Downloads/SIMULATIONS/$SimData/$SimOut/$OUT
    /home/bent0vict0r/Downloads/SIMULATIONS/$SimData/RINEX/$RINEX_OBS
    /home/bent0vict0r/Downloads/SIMULATIONS/$SimData/$RINEX
    /home/bent0vict0r/Downloads/SIMULATIONS/$SimData/$CLK
    /home/bent0vict0r/Downloads/SIMULATIONS/$SimData/$ORBIT
34. done < RinexList_SUPIM.txt

35. # Remoção da Lista criada anteriormente com o nome dos RINEX OBS
36. rm RinexList_SUPIM.txt

37. echo "DONE!"

```

APENDICE C – SOFTWARE AVALIADOR

```
1. # -*- coding: utf-8 -*-
2. #-----#
3. ##### Universidade Federal de Santa Maria #####
4. ##### Instituto Nacional de Pesquisas Espaciais #####
5. ##### Centro de Tecnologias #####
6. ##### Engenharia de Computação #####
7. ##### Victor Dallagnol Bento #####
8. #-----#
9. import os, sys, math

10. #..... VARIÁVEIS GLOBAIS
11. listRinex = [] # Lista para arquivos RINEX
12. countRinex = 0 # Contador para quantidade de arquivos RINEX
13. listBases = [] # Lista para armazenar dados das bases
14. #JDAY = "010"
15. #JDAY = "041"
16. #JDAY = "069"
17. #JDAY = "100"
18. #JDAY = "130"
19. #JDAY = "161"
20. #JDAY = "191"
21. #JDAY = "222"
22. #JDAY = "253"
23. #JDAY = "283"
24. #JDAY = "314"
25. JDAY = "344"

26. # Caminho para abrir arquivos RINEX
27. pathRinex = "/YOUR_PATH/SIMULATIONS/"+ JDAY + "/RINEX/"
28. dirsRinex = os.listdir(pathRinex)
```

```

29. NumberOfSamples = 12    # Numero de amostras (2h em 2h = 12 amostras no dia)

30. listPos = []    # Lista para arquivos .POS de cada pasta (x, y, z, nome)

31. save = 0    # Para salvar nome e hora somente uma vez na lista final

32. # Caminho para abrir arquivos POS de cada correção
33. pathPos1 = "/YOUR_PATH/SIMULATIONS/"+ JDAY + "/IGS/"
34. dirsPos1 = os.listdir(pathPos1)
35. pathPos2 = "/YOUR_PATH/SIMULATIONS/"+ JDAY + "/Klobuchar/"
36. dirsPos2 = os.listdir(pathPos2)
37. pathPos3 = "/YOUR_PATH/SIMULATIONS/"+ JDAY + "/SUPIM/"
38. dirsPos3 = os.listdir(pathPos3)
39. pathPos4 = "/YOUR_PATH/SIMULATIONS/"+ JDAY + "/DualFreq/"
40. dirsPos4 = os.listdir(pathPos4)
41. # listas para armazenar informações de cada posicionamento de cada estação de 2h-2h
42. listCorrections = []
43. #.....

44. #.....          FUNÇÃO PRINCIPAL
45. def main():
46. global listRinex, countRinex, listBases, pathRinex, dirsRinex, listPos, pathPos1, pathPos2,
    pathPos3, pathPos4, pathPos5, dirsPos1, dirsPos2, dirsPos3, dirsPos4, dirsPos5, save    # Para
    utilizar e editar variáveis globais

47. # Leitura de todos os arquivos do diretório PATHDIR e armazenamento em Lista
48. for file in dirsRinex:
49. listRinex.insert(countRinex, file)
50. countRinex += 1

51. # For para leitura de todas as distancias dos arquivos RINEX
52. iRinex = 0
53. while iRinex < countRinex:

```

```

54. readECEF = pathRinex + listRinex[iRinex] # Caminho para leitura do arquivo RINEX
55. line_num = 0 # Leitura do arquivo RINEX e busca por linhas que contém XYZ
56. f = open(readECEF)
57. for line in f:
58. if "APPROX POSITION XYZ" in line:
59. xyzBase = line # Armazena linha que contém XYZ na variavel xyzBase
60. break
61. else:
62. line_num += 1

63. # Separa string por espaço (" ") e grava em lista listXYZ
64. listXYZ = []
65. listXYZ = list(xyzBase.split(" ")) # Separa elementos por " " e transforma em lista

66. auxBase = []
67. for index in listXYZ: # Laço para percorrer Lista:
68. if index == "": # Se valor do índice da lista for " (vazio), lê o próximo
69. pass
70. else:
71. auxBase.append(index) # Se valor for diferente, adicionar valor (x, y ou z) a lista auxiliar
    auxBase

72. # Remove parte da lista que não é X, Y e Z
73. auxBase.remove("APPROX")
74. auxBase.remove("POSITION")
75. auxBase.remove("XYZ\n")

76. auxBase.append(listRinex[iRinex]) # Adiciona nome da base ao ultimo índice da lista
77. listBases.append(auxBase) # Adiciona lista com dados da base (x, y, z e nome) para
    Lista de Bases (lista de listas)
78. #print auxBase
79. iRinex += 1

80. listPos = readAmountPOS(pathPos1, dirsPos1)

```

```

81. # Esquema para rodar todas as correções, entrando em todas os diretórios
82. countCorrection = 0
83. if countCorrection == 0:      # IGS
84. findCoordenates(pathPos1)
85. countCorrection += 1
86. # Já salvou na lista final o nome de cada .POS e as horas das amostras
87. save = 1
88. if countCorrection == 1:     # KLOBUCHAR
89. findCoordenates(pathPos2)
90. countCorrection += 1
91. if countCorrection == 2:     # SUPIM
92. findCoordenates(pathPos3)
93. countCorrection += 1
94. if countCorrection == 3:     # DUAL FREQUENCY
95. findCoordenates(pathPos4)
96. countCorrection += 1

97. print " Estação " + "      Hora " + "\t GIM" + "\t\tKLOBUCHAR" + "\t SUPIM-DAVS" +
      "\t DUAL FREQUENCY"
98. for idx in listCorrections:
99. if idx[5] > 5.0 or idx[5] == -1.0:
100. del(idx[5])
101. del(idx[4])
102. del(idx[3])
103. del(idx[2])
104. del(idx[1])
105. del(idx[0])

106. cdf = 0
107. tdf = 0
108. for idx in listCorrections:
109. if idx != []:
110. print idx

```

```

111. if idx[2] != -1.0:
112.     tdf += idx[2]
113.     cdf += 1

114. #print tdf/cdf
115. #del(listCorrections[0])

116. #.....

117. #..... LISTAGEM DOS ARQUIVOS .POS
118. def readAmountPOS(pathPos_, dirsPos_):
119.     countPos = 0 # Contador para quantidade de arquivos .POS de cada pasta
120.     listPos_ = []

121. # Leitura de todos os arquivos do diretório PATHDIR e armazenamento em Lista
122. for file in dirsPos_:
123.     listPos_.insert(countPos, file)
124.     countPos += 1
125. return listPos_
126. #.....

127. #..... LEITURA DOS ARQUIVOS .POS
128. def findCoordinates(pathPos):
129.     global listCorrections, save

130.     counterPos = 0 # Contador para arquivos .POS
131.     eptStr = "0.0000 0.0000 0.0000 0.0000"

132. # fazer for para ler cada .pos da determinada correção
133. while counterPos < countRinex:
134.     # Auxiliares

```

```

135. saveLine = [] # Lista para salvar as linhas específicas das horas analisadas (de hora em
    hora, ou 2h em 2h, por ex)
136. aux = []
137. listAux = []
138. readPOS = pathPos + listPos[counterPos] # Caminho do arquivo .POS

139. saveLine.insert(0, eptStr)
140. saveLine.insert(1, eptStr)
141. saveLine.insert(2, eptStr)
142. saveLine.insert(3, eptStr)
143. saveLine.insert(4, eptStr)
144. saveLine.insert(5, eptStr)
145. saveLine.insert(6, eptStr)
146. saveLine.insert(7, eptStr)
147. saveLine.insert(8, eptStr)
148. saveLine.insert(9, eptStr)
149. saveLine.insert(10, eptStr)
150. saveLine.insert(11, eptStr)

151. # for para leitura e armazenamento das linhas de 2h em 2h
152. f = open(readPOS)
153. for line in f: # Tolerancia de 10 minutos entre as amostras, sempre salva a última
    sobreescrevendo as anteriores
154. if " 00:0" in line:
155. saveLine.pop(0)
156. saveLine.insert(0, line)

157. if " 02:0" in line:
158. saveLine.pop(1)
159. saveLine.insert(1, line)

160. if " 04:0" in line:
161. saveLine.pop(2)
162. saveLine.insert(2, line)

```

163. if " 06:0" in line:
164. saveLine.pop(3)
165. saveLine.insert(3, line)

166. if " 08:0" in line:
167. saveLine.pop(4)
168. saveLine.insert(4, line)

169. if " 10:0" in line:
170. saveLine.pop(5)
171. saveLine.insert(5, line)

172. if " 12:0" in line:
173. saveLine.pop(6)
174. saveLine.insert(6, line)

175. if " 14:0" in line:
176. saveLine.pop(7)
177. saveLine.insert(7, line)

178. if " 16:0" in line:
179. saveLine.pop(8)
180. saveLine.insert(8, line)

181. if " 18:0" in line:
182. saveLine.pop(9)
183. saveLine.insert(9, line)

184. if " 20:0" in line:
185. saveLine.pop(10)
186. saveLine.insert(10, line)

187. if " 22:0" in line:
188. saveLine.pop(11)
189. saveLine.insert(11, line)

```

190. counterLine = 0 # Contador para cada linha específica
191. while counterLine < NumberOfSamples: # Como serão analisadas amostras de 2h em 2h
    teremos 12 amostras
192. aux = [x.strip() for x in saveLine[counterLine].split(" ")] # separa por espaços duplos ("
    ") string e salva em lista aux
193. for i_ in aux:
194. if i_ == ": # Elimina posições inválidas da lista
195. aux.remove("")

196. aux[0] = listPos[counterPos] # Sobreescreve primeiro elemento da lista (dia e hora no
    formato .POS - desnecessários) pelo nome da estação e salva em listAux
197. listAux.append(aux[0])
198. if counterLine == 0: # Salva hora na listAux e atualia índice padrão de escrita
    saveInList
199. listAux.append("00:00")
200. saveInList = (12*counterPos)
201. if counterLine == 1:
202. listAux.append("02:00")
203. saveInList = (12*counterPos)+1
204. if counterLine == 2:
205. listAux.append("04:00")
206. saveInList = (12*counterPos)+2
207. if counterLine == 3:
208. listAux.append("06:00")
209. saveInList = (12*counterPos)+3
210. if counterLine == 4:
211. listAux.append("08:00")
212. saveInList = (12*counterPos)+4
213. if counterLine == 5:
214. listAux.append("10:00")
215. saveInList = (12*counterPos)+5
216. if counterLine == 6:
217. listAux.append("12:00")
218. saveInList = (12*counterPos)+6

```

```

219. if counterLine == 7:
220.     listAux.append("14:00")
221.     saveInList = (12*counterPos)+7
222.     if counterLine == 8:
223.         listAux.append("16:00")
224.         saveInList = (12*counterPos)+8
225.         if counterLine == 9:
226.             listAux.append("18:00")
227.             saveInList = (12*counterPos)+9
228.             if counterLine == 10:
229.                 listAux.append("20:00")
230.                 saveInList = (12*counterPos)+10
231.                 if counterLine == 11:
232.                     listAux.append("22:00")
233.                     saveInList = (12*counterPos)+11

234. # Calculo da distancia euclidiana para determinada hora
235. for idx in listBases:
236.     if idx[3][:4] == listPos[counterPos][:4]: # Buca estação referente ao arquivo .POS para
        pegar coordenadas e fazer o cálculo
237.         if aux[1] == "0.0000" and aux[2] == "0.0000" and aux[3] == "0.0000":
238.             d = -1.0
239.             listAux.append(d)
240.         else:
241.             d_1 = (float(aux[1])-float(idx[0]))**2
242.             d_2 = (float(aux[2])-float(idx[1]))**2
243.             d_3 = (float(aux[3])-float(idx[2]))**2
244.             d = math.sqrt(d_1 + d_2 + d_3)
245.             listAux.append(d) # Salva na lista ListAux
246.         if save == 1: # Depois que efetua a primeira correção, os nomes das estações e as horas já
            estão salvos na lista final (listCorrections), basta apenas adicionar a distância da nova correção á
            listCorrections.
247.             listCorrections[saveInList].append(d)
248.         else:
249.             pass

```

```

250. if save == 0:    # Se ainda verificou nenhuma correção, salva nomes e horas das estações na
    lista final listCorrections
251. listCorrections.append(listAux)
252. # Limpa vetores e atualiza contadores
253. aux = []
254. listAux = []
255. counterLine += 1

256. counterPos += 1
257. #.....

258. #.....          CHAMADA DA FUNÇÃO PRINCIPAL
259. if __name__ == '__main__':
260.     main() # chamada da função main
261. #.....

```