



# ESCOLA NAVAL

talant de bi-faire



Departamento de Ciências do Mar

Gonçalo Miguel Rebelo Horta

*Ponto Astronómico pelo Método Marcq Saint-Hilaire*

*Desenvolvimento de uma Ferramenta de Cálculo do Ponto*

Dissertação para obtenção do grau de Mestre em Ciências Militares

Navais, na especialidade de Marinha



Alfeite

2020





# ESCOLA NAVAL

talant de bi-faire



Gonçalo Miguel Rebelo Horta

## *Ponto Astronómico pelo Método Marcq Saint-Hilaire*

*Desenvolvimento de uma Ferramenta de Cálculo do Ponto*

Dissertação para obtenção do grau de Mestre em Ciências Militares Navais,  
na especialidade de Marinha

**Orientação de:** CMG Costa Canas

**Coorientação de:** CFR Lourenço Gorricha

O aluno mestrando,

ASPOF Rebelo Horta

O orientador,

CMG Costa Canas

Alfrite

2020



*“Exploration is our nature. We began as wanderers, and we are wanderers still. We have lingered long enough on the shores of the cosmic ocean. We are ready at last to set sail for the stars.”*

Carl Sagan



## **Agradecimentos**

Não me alongando, quero agradecer ao meu Orientador, CMG António Costa Canas, por todo o seguimento e apoio constante que ao longo de todos estes meses de trabalho sempre mostrou para comigo.

Quero agradecer também ao meu Coorientador CFR Lourenço Gorricha pela disponibilidade.

Nesta sequência quero também agradecer ao CMG Franco Facada pela vertente de inspiração e matéria lecionada, bem como a inteira disponibilidade que mostrou para comigo quando o questioneei sobre o interesse em querer trabalhar o tema aqui apresentado.

Ao meu amigo e camarada Haroun Tlemçani, pela sincera amizade e por estar sempre pronto a discutir os temas aqui abordados tornando esta jornada numa caminhada conjunta.

Agradecer ao meu eterno camarada Fonseca Pereira, por todos os momentos vividos e por me fazer acreditar que tudo é possível com esforço e dedicação.

A toda a minha família e à minha namorada, que sempre me deram tudo.



## Resumo

A presente dissertação de mestrado teve como objetivo primordial construir uma ferramenta interativa que responda às necessidades de cálculo do ponto astronómico, com base em observações de diversos astros e seguindo o Método Marcq Saint-Hilaire.

Desta forma, recorreu-se ao programa MatLab para proporcionar a construção da mesma, assentando em bases teóricas, tendo sido possível programar as diversas fases de obtenção das componentes necessárias ao cálculo do ponto final. A esta ferramenta deu-se o nome de ASTROTH2020.

Concluída a construção do ASTROTH2020, para a qual se necessitou de efetuar diversas correções, realizou-se diversos testes de comparação com recurso a dois programas de referência para a navegação astronómica: NACPAC e NAV32. Estes testes tiveram o objetivo de analisar o rigor obtido pelo programa desenvolvido. Também foi possível a utilização de um programa de referência para a Astronomia: o Stellarium 0.20.2.

Considerou-se finalmente que todos os testes foram concluídos com sucesso e que a ferramenta desenvolvida proporciona um cálculo preciso e útil para a navegação astronómica, por vezes melhor do que as ferramentas já existentes.

Palavras-chave: MATLAB, ASTROTH2020, Navegação Astronómica, Marcq Saint-Hilaire, Astronomia



## Abstract

The main objective of this master's thesis was to build an interactive tool that responds to the needs of calculating the astronomical point, based on observations from different stars and following the Marcq Saint Hilaire Method.

In this way, the MatLab program was used to provide its construction, based on theoretical bases, and it was possible to program the various stages of obtaining the components necessary to calculate the final point. This tool was named ASTROTH2020.

Completed the construction of the ASTROTH2020, for which it was necessary to make several corrections, comparison tests were carried out using two important programs for astronomical navigation: NACPAC and NAV32. These tests were aimed to analyzing the rigor obtained by the developed program. It was also possible to use a reference program for Astronomy: the Stellarium 0.20.2.

Finally, it was considered that all tests were successfully completed and that the developed tool provides an accurate and useful calculation for astronomical navigation, sometimes better than the existing tools.

Keyword: MATLAB, ASTROTH2020, Astronomical Navigation, Marcq Saint-Hilaire, Astronomy



## Índice

Agradecimentos .....	V
Resumo.....	VIII
Abstract.....	IX
Lista de Abreviaturas e Acrónimos .....	XIII
Índice de Figuras .....	XV
Índice de Tabelas.....	XVII
Introdução .....	1
Enquadramento Geral.....	1
Metodologia.....	1
Pertinência do Tema .....	2
Estrutura da Dissertação .....	2
Capítulo 1. Enquadramento Teórico .....	4
1.1. A Navegação Astronómica .....	4
1.2. Coordenadas .....	4
1.2.1. Coordenadas Equatoriais .....	5
1.2.2. Coordenadas horárias .....	6
1.2.3. Coordenadas horizontais .....	9
1.2.4. Coordenadas Geográficas .....	10
1.3. Método Marcq Saint-Hilaire .....	11
1.4. O Triângulo de Posição.....	14
1.5. Curvas de Altura .....	16
1.6. Equações de Cálculo .....	18
1.6.1. Altura Corrigida.....	18
1.6.2. Altura estimada.....	24
1.6.3. Cálculo do Azimute .....	25
1.6.4. Cálculo do $\Delta$ .....	26
1.6.5. Cálculo da Posição Final.....	27
1.6.6. Iteração da Posição Final.....	28
1.7. Transporte de Retas de Altura.....	29
Capítulo 2. Algoritmos e estrutura de cálculo .....	30

2.1.	Algoritmo do Cálculo da Posição Estimada .....	30
2.2.	Algoritmo do Cálculo da altura estimada .....	32
2.3.	Algoritmo do Cálculo da altura verdadeira .....	33
2.4.	Cálculo do $\Delta a$ .....	35
2.5.	Algoritmo de Cálculo do azimute .....	36
2.6.	Algoritmo de cálculo da posição final .....	36
	Capítulo 3. Construção e Funcionalidade da Ferramenta de Cálculo .....	38
3.1.	Ponto Astronómico ASTROTH2020 .....	38
3.1.1.	Organização Estrutural da Ferramenta .....	39
	Capítulo 4. Discussão de Resultados .....	48
4.1.	Critério de Recolha e Escolha de Dados .....	48
4.1.1.	Dados sem dependência de programas .....	48
4.1.2.	Dados dependentes de programas .....	52
4.2.	Comparação dos Resultados Obtidos .....	60
	Conclusão .....	64
	Bibliografia .....	66
	Apêndices .....	68
	Apêndice A – Código do Programa .....	68

## Lista de Abreviaturas e Acrónimos

AM	Antemeridiana
$e_i$	Erro de Índice
GHA	<i>Greenwich Hour Angle</i> (Ângulo Horário em Greenwich)
HP	Paralaxe Horizontal
L	Longitude
LDP	Linha de Posição
LHA	Ângulo Horário no Lugar
M	Meridiana
P	Ângulo no Polo
PA	Posição Astral
PM	Pós-meridiana
R	Refração Atmosférica
SD	Semidiâmetro
Z	Azimute
$\gamma$	Ponto Vernal
$\delta$	Declinação
$\zeta$	Distância Zenital
$\varphi$	Latitude



## Índice de Figuras

Figura 1: Representação da declinação ( $\delta$ ) e do ângulo sideral (AS) do astro S (Adaptado de Gameiro, 1964).....	6
Figura 2: Representação do GHA .....	7
Figura 3: Representação do ângulo horário no lugar e ângulo no polo .....	8
Figura 4: Coordenadas Horizontais (Museu de Astronomia e Ciências Afins, s.d.)..	9
Figura 5 - Representação de uma altura não observada (1ª imagem) sobre o vertical do astro e o seu contrário (2ª imagem) (Marinha do Brasil: Diretoria de Portos e Costas, 1978, pp. 109).....	10
Figura 6: Latitude e Longitude (Gameiro, 1964).....	11
Figura 7: Comparação entre duas distâncias zenitais (Escola Naval, 1995) .....	12
Figura 8: $\Delta a > 0$ e $\Delta a < 0$ respetivamente (Mederos, s.d.).....	12
Figura 9: Representação gráfica do traçado de retas de posição. ....	13
Figura 10: Triângulo Esférico (Mederos, s.d.) .....	14
Figura 11: Triângulo de Posição (Gameiro, 1964, pp. 81) .....	15
Figura 12: Representação prática de uma curva de altura em projeção Mercator (Gameiro, 1964).....	16
Figura 13: Tangente de altura (Gameiro, 1964).....	17
Figura 14: Distância (milhas) máxima admissível entre o ponto determinante e o ponto observado (Ministério da Marinha & Instituto Hidrográfico, 1973) .....	18
Figura 15: Altitude de Observação (Umland, 2004).....	20
Figura 16: Erro de Refração (Umland, 2004) .....	20
Figura 17: Erro de Paralaxe (Umland, 2004) .....	22
Figura 18: Semidiâmetro (Umland, 2004) .....	23
Figura 19: Representação de um astro circumpolar visível e um invisível .....	25
Figura 20: Exemplo tabular da altura estimada e azimute calculado (National Imagery And Mapping Agency, 2001).....	25

Figura 21: Representação do $\Delta a$ (Gameiro, 1964) .....	26
Figura 22: Fluxograma de cálculo da posição estimada .....	32
Figura 23: Ferramenta Ponto Astronómico Escola Naval .....	39
Figura 24: Calculador da posição estimada .....	40
Figura 25: Posição estimada e correções .....	41
Figura 26: Variáveis e dados de saída .....	42
Figura 27: Escolha do astro .....	42
Figura 28: Escolha do limbo observado .....	43
Figura 29: Tabela de linhas de posição .....	43
Figura 30: Usar LDP .....	44
Figura 31: Opção de eliminação definitiva de LDP .....	44
Figura 32: Guardar LDP .....	45
Figura 33: Calcular posição final .....	45
Figura 34: Panorama Celeste do Stellarium 0.20.2 .....	52
Figura 35: Representação da altura aparente da estrela Deneb (Stellarium 0.20.2, 2020). .....	53
Figura 36: Posição e Elevação do observador (Stellarium 0.20.2, 2020). .....	53
Figura 37: Representação da Grelha Azimutal e "Zoom out" (Stellarium 0.20.2, 2020). .....	54
Figura 38: NAV32 - $\Delta a$ 's e azimutes .....	58
Figura 39: NAVPAC - $\Delta a$ 's e azimutes .....	58
Figura 40: ASTROTH2020 - $\Delta a$ 's e azimutes .....	58

## Índice de Tabelas

Tabela 1: A refração (em minutos de grau) de acordo com a altura observada .....	20
Tabela 2: Posições Geográficas.....	49
Tabela 3: Posições Estimadas.....	50
Tabela 4: Data para cada Posição .....	51
Tabela 5: Astros e alturas aparentes correspondentes.....	55
Tabela 6: Posições obtidas para cada programa de estudo .....	59
Tabela 7: Distâncias finais representadas em milhas náuticas .....	61



## Introdução

### Enquadramento Geral

O método de Marcq Saint-Hilaire, foi desenvolvido em finais do século XIX, por um oficial da Marinha Francesa, com o mesmo nome. Esse método foi usado, na condução de viagens oceânicas, praticamente em todo o mundo e ainda hoje é usado como método de recurso.

O princípio em que se baseia este método é a comparação direta entre os valores das alturas em relação aos astros observados, onde uma delas é calculada com base na resolução de um triângulo de posição e a outra com base na observação direta do astro. Isto permitirá que se consiga uma relação entre a distância a que o observador se encontra da posição astral, para um determinado astro. Uma vez obtidos vários destes parâmetros, efetuar-se-á o cruzamento das respetivas linhas de posição, o que culminará na obtenção da posição astronómica. Todos estes dados são relacionados com uma posição (posição estimada) que o observador irá considerar como sendo aquela onde presume encontrar-se.

Assim, com o seguimento dos cálculos obteremos à custa de diversos dados de entrada, uma nova e correta posição geográfica.

### Metodologia

A investigação científica requer um rigor elevado para dar resposta aos objetivos e ou problemas a que se propõem responder. Desta forma, é necessário escolher de forma concisa o método de investigação a utilizar (Fortin, 1999).

O objetivo do estudo integrado na presente dissertação é primeiramente a construção de um programa de navegação astronómica, recorrendo ao método Marcq Saint-Hilaire e de seguida a comparação dos resultados obtidos pelo novo programa, doravante designado como ASTROTH2020, com os resultados dos programas NAV32 e NAVPAC para as mesmas variáveis de entrada. Assim, optou-se por um estudo quantitativo, observacional e descritivo.

Para a construção do programa teve-se em consideração aspetos essenciais como o *layout*, o encadeamento dos dados e dos resultados, bem como a facilidade de uso do mesmo.

## **Pertinência do Tema**

Nos dias atuais, o recurso à navegação astronómica tende a ser praticado apenas durante a sua aprendizagem e mais tarde é somente utilizado durante uma navegação de emergência ou no abandono do navio para uma jangada. Este método de navegação, que evoluiu até ao que conhecemos hoje em dia, mantém-se como sendo o último método de navegação de recurso. Este trabalho vem na tentativa de tornar mais expedito e mais exato o recurso a este método.

Além de toda a componente histórica e de interesse que promovem a discussão dos resultados aqui abordados, existe ainda uma necessidade de possuir uma navegação de auxílio para uma situação que não seja de emergência.

*“(..). Batteries run out, systems get backed, and even advanced technology can be balky. In a pinch — or in a war — sailors need something to fall back on. And stars and sextants have been working pretty well for hundreds of years. So the Naval Academy started teaching its sailors how to navigate ships by looking to the heavens again (...)” (Peterson, 2016).<sup>1</sup>*

## **Estrutura da Dissertação**

Atendendo à complexidade da construção de uma ferramenta de cálculo do ponto astronómico, verificou-se a necessidade de separar o conteúdo da presente dissertação em quatro capítulos. Estes permitem ao leitor retirar detalhes que importam destacar relativamente ao funcionamento teórico da aplicação desenvolvida e do procedimento correto para a sua utilização.

No Capítulo 1 encontra-se o enquadramento teórico no qual se esclarece e se apresenta as mais diversas fórmulas e detalhes teóricos relativos ao cálculo das variáveis que

---

<sup>1</sup> **Tradução:** “As baterias acabam, os sistemas são invadidos e até a tecnologia avançada pode ser confusa. Num campo - ou numa guerra - os marinheiros precisam de algo de recurso. E estrelas e sextantes têm funcionado muito bem durante centenas de anos. Então, a Academia Naval começou a ensinar os seus alunos a navegar nos navios, olhando para o céu novamente.”

influenciam o resultado final. A construção sequencial deste capítulo permite um acompanhamento lógico da sequência de cálculos que culminam no resultado final apresentado pelo programa desenvolvido. Assim é apresentada uma revisão literária relativa à navegação astronómica, ao método Marcq Saint-Hilaire, ao triângulo de posição, às equações de cálculo necessárias ao cálculo da posição e às diversas coordenadas que importam salientar.

O segundo capítulo divide as fases de cálculo do programa ASTROTH2020, no qual é apresentada a sequência de raciocínio por trás da construção da rotina informática. Nele são também descritas as várias fases do cálculo do ponto astronómico.

No Capítulo 3 são descritas as várias componentes que constituem o programa ASTROTH2020. São explicadas as funcionalidades do mesmo e também o modo como o utilizador deve interagir com a janela gráfica. Serão também, embora indiretamente, relacionadas com essas componentes os algoritmos de cálculo que correspondem a cada parcela presente na janela gráfica de utilização.

Por fim, já no capítulo quatro, são apresentados cálculos efetuados pelo ASTROTH2020 e a comparação com programas de referência (NAVPAC e NAV32). Para tal foram usados os mesmos argumentos de entrada em todos os programas, de forma a tornar todo o teste de resultados o mais fidedigno possível.

## Capítulo 1. Enquadramento Teórico

### 1.1. A Navegação Astronómica

A Navegação teve ao longo de muitos anos vários desenvolvimentos, face ao desenvolvimento dos estudos na área. Estes visaram melhorar, até aos dias de hoje, o rigor em que os navegadores se baseiam e que assumem como correto para diversos fatores, como é o caso do posicionamento geográfico. Só desta forma, e com o continuar do desenvolvimento científico, foi possível chegar a métodos que têm como resultado um posicionamento geográfico com elevado rigor.

Foi a partir do conhecimento dos astros que os navegadores tiveram as suas primeiras interpretações de posição e de como calcular as suas coordenadas. Uma delas, a latitude, era interpretada diretamente pela medição do ângulo que um astro faz com a linha do horizonte.

Para isso, diversos instrumentos como o Astrolábio Grego ou a Balestilha foram sendo inventados. Estes instrumentos permitiam a medição da altura dos astros e com o passar do tempo vieram a culminar no que conhecemos hoje como sextante (Marinha do Brasil: Diretoria de Portos e Costas, 1978, pp. 83).

O sextante permite a medição de alturas dos astros com um rigor admissível para a navegação oceânica, no entanto a sua utilização não é exclusiva da navegação astronómica, visto que pode ser usado na recolha de qualquer ângulo que auxilie na prática da navegação, como é exemplo o cálculo da distância até terra em função da relação entre ângulos e altura de um ponto conspícuo, bem como a medição de ângulos horizontais para obtenção da respetiva linha de posição (segmento capaz).

### 1.2. Coordenadas

Têm-se como coordenadas, todos os sistemas referenciados com as mesmas, que são originados pelo cruzamento de duas linhas imaginárias, tais como a latitude e longitude que permitem obter uma posição geográfica.

Na vertente da navegação astronómica podemos retirar algumas coordenadas, como sendo fundamentais para o cálculo do ponto astronómico, nomeadamente:

- Coordenadas equatoriais;
- Coordenadas horárias;

- Coordenadas horizontais;
- Coordenadas geográficas.

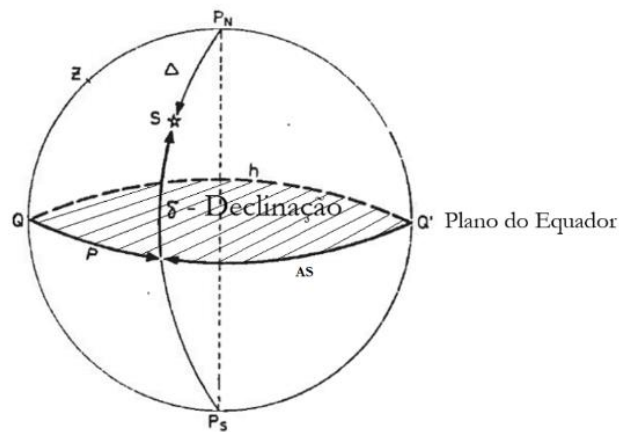
### 1.2.1. Coordenadas Equatoriais

As coordenadas equatoriais são valores angulares medidos na linha do equador ou ao longo deste e que têm como referência de início de contagem o ponto vernal. Por consequência, são valores que não variam consoante a localização do observador (Gameiro, 1964, pp. 7).

A **declinação ( $\delta$ ) (Figura 1)** corresponde ao ângulo, medido sobre o meridiano do astro, desde o plano do equador até ao astro (Gameiro, 1964, pp. 7). É apresentada num intervalo de  $0^\circ$  a  $90^\circ$ . Caso seja  $0^\circ$ , então o astro encontra-se sobre a linha do equador celeste, por outro lado, caso seja  $90^\circ$  este encontra-se no prolongamento do eixo da terra, ou seja, encontra-se num dos polos.

Note-se então que a declinação tomará um valor negativo ou positivo, consoante o hemisfério celeste onde o astro se encontrar, da mesma forma a latitude também deverá apresentar o sinal negativo se for de nome Sul e positivo se for de nome Norte. Assim, a declinação apresentará valor positivo se for do mesmo nome que a latitude e negativo se for de nome contrário.

O **ângulo sideral** (designado por “AS” na **Figura 1**) considera-se como sendo o ângulo medido desde o ponto vernal  $\gamma$  até ao meridiano do astro (S). Para um observador no polo Norte, este ângulo medir-se-á de  $0^\circ$  até  $360^\circ$  sob o plano do equador e no sentido dos ponteiros do relógio, ou seja, de Este para Oeste, consequentemente no polo Sul, o sentido de medição será o oposto (Gameiro, 1964, pp. 8).



**Figura 1:** Representação da declinação ( $\delta$ ) e do ângulo sideral (AS) do astro S (Adaptado de Gameiro, 1964)

### 1.2.2. Coordenadas horárias

As coordenadas horárias são medidas tendo como referência o plano do equador, são também medidas no sentido horário para um observador que se encontre a visualizar a Terra a partir do polo Norte. Além disto, caracterizam-se ainda pela variação do seu valor consoante a longitude do observador, e principalmente pela variação ao longo do dia, por isso se denominarem horárias.

O **ângulo horário Greenwich (GHA)** (**Figura 2**) é obtido pela medição da distância angular entre o meridiano de referência (Greenwich), até ao meridiano do astro observado. Este tomará um valor compreendido entre  $0^\circ$  e  $360^\circ$ .

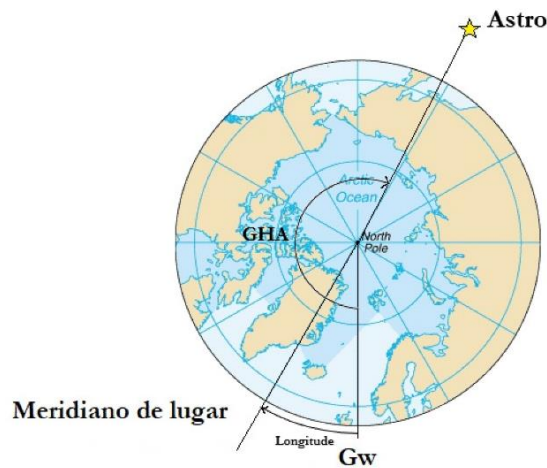


Figura 2: Representação do GHA<sup>2</sup>

Deste ângulo, resultam muitos outros que são calculados correlacionando-o com a longitude do observador, nomeadamente (**Figura 3**):

- Ângulo horário no Lugar (LHA);
- Ângulo no Polo (P);

O **ângulo horário no Lugar (LHA)** corresponde ao ângulo formado entre o meridiano de lugar do observador e o meridiano do astro observado, variando de 0° a 360°.

Podemos concluir desta forma que o ângulo horário no lugar é dado pela seguinte fórmula:

$$LHA = GHA + Longitude do Observador$$

Esta fórmula é comprometida diretamente pela seguinte condição:

- Caso o valor obtido por esta soma seja superior a 360° dever-se-á subtrair 360° ao mesmo.

---

<sup>2</sup> Esta representação foi adaptada pelo autor da dissertação para elucidar a descrição realizada;

O **ângulo no Polo (P)** é um ângulo medido em relação ao meridiano de lugar para Este ou Oeste podendo tomar valor máximo de 180 graus (Gameiro, 1964, pp. 11). Este ângulo corresponde à medida angular desde o meridiano de lugar até ao Astro.

Na prática, através da seguinte relação podemos obter o ângulo no Polo (Gameiro, 1964, pp. 11):

$$\text{Astro a } \begin{cases} W \\ E \end{cases} \text{ do meridiano de lugar } \begin{cases} P_W = LHA \\ P_E = 360 - LHA \end{cases}$$

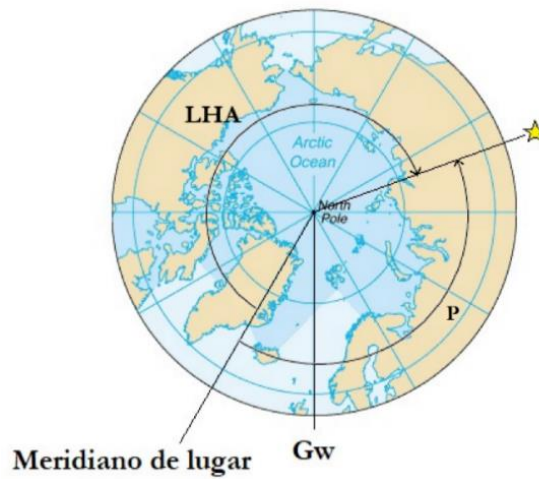


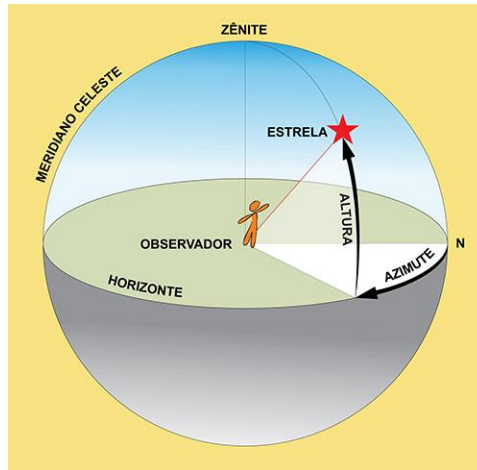
Figura 3: Representação do ângulo horário no lugar e ângulo no polo<sup>3</sup>

---

<sup>3</sup> Esta representação foi elaborada pelo autor da dissertação para elucidar a descrição realizada;

### 1.2.3. Coordenadas horizontais

Algumas coordenadas são diretamente observadas e medidas pelo observador, das quais fazem parte o azimute e altura de um astro (**Figura 4**), que são medidas relativamente ao plano horizontal, ou seja, o horizonte.



**Figura 4:** Coordenadas Horizontais (Museu de Astronomia e Ciências Afins, s.d.s)

A **distância zenital** é medida desde o astro até ao zênite do observador. Não podendo ser medida no mar, devido à impossibilidade de materializar o zênite do observador, não é praticada a sua leitura.

A **altura, ou altura verdadeira**, é o ângulo medido no vertical do astro, desde a linha de horizonte até à posição do astro no céu visível, podendo e devendo ser garantida pela ação de bascular o sextante<sup>4</sup>. O ângulo varia de 0° até 90°, sendo que neste último caso o mesmo ter-se-á que encontrar sob o zênite do observador (Gameiro, 1964, pp. 10).

---

<sup>4</sup> Considera-se “bascular o sextante” à ação de oscilar o sextante, com a finalidade de garantir que o objeto celeste descreve uma trajetória semicircular e que tangencie o horizonte no ponto central dessa trajetória. Garante-se assim a altura registada foi medida sobre o vertical do astro (Marinha do Brasil: Diretoria de Portos e Costas, 1978, pp. 109).

No caso do **azimute**, este é medido desde o ponto cardinal definido como origem e o vertical do astro observado. Sendo que se for medido desde o ponto cardinal Norte para Este, Sul e Oeste, então o mesmo tomará valores de  $0^\circ$  a  $360^\circ$  (Gameiro, 1964, pp. 10).

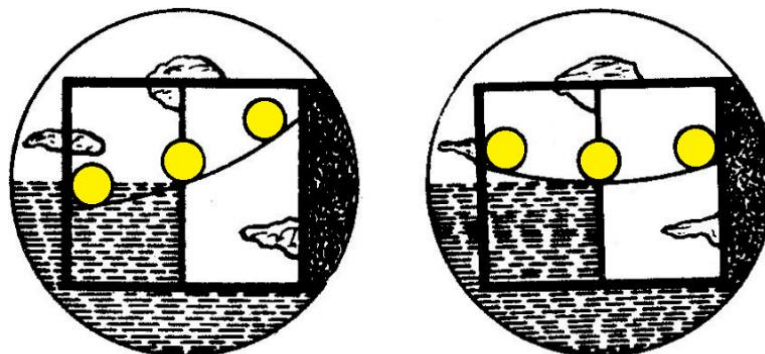


Figura 5 - Representação de uma altura não observada (1ª imagem) sobre o vertical do astro e o seu contrário (2ª imagem) (Marinha do Brasil: Diretoria de Portos e Costas, 1978, pp. 109).

#### 1.2.4. Coordenadas Geográficas

Para o início do cálculo do ponto astronómico o observador necessita de uma primeira posição geográfica (posição estimada do observador), cujas componentes que dela fazem parte são a latitude e a longitude celestes.

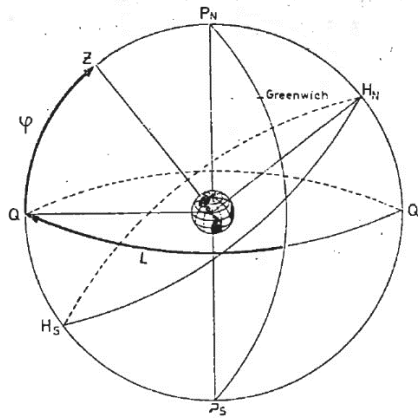
A latitude e a longitude (**Figura 6**) são ambas medidas em graus e apresentadas em navegação astronómica com um rigor até à décima do grau.

A **latitude** é medida sobre o meridiano de lugar do observador e conta-se desde a linha do Equador até ao zénite do observador. Esta pode variar de  $0^\circ$  (observador no equador) até  $90^\circ$  (observador num dos polos) e o seu nome (Norte ou Sul) varia consoante o hemisfério de localização do observador (Gameiro, 1964, pp. 12).

A **longitude** é medida a partir do meridiano de referência, Greenwich. Esta varia de  $0^\circ$  até  $180^\circ$ , tomando o nome de Este ou Oeste.

Note-se que na prática estas coordenadas, quando transportadas para o estudo astronómico, são descritas em relação à esfera celeste. Por este motivo e de forma a

representar a localização terrestre do observador na esfera celeste, o ponto de localização é projetado na mesma, tomando o nome de Zénite.



**Figura 6:** Latitude e Longitude (Gameiro, 1964)

### 1.3. Método Marcq Saint-Hilaire

O método da altura ou método Marcq Saint-Hilaire consiste num procedimento que utiliza como ponto determinante da linha de posição (reta de altura) um ponto marcado na direção do astro (azimute ( $Z$ )), traçado a partir de uma posição estimada na carta onde a distância entre esses pontos irá ser dada pela diferença entre a altura calculada (para a posição estimada) e a altura observada do astro em questão (Escola Naval, 1995).

Na **Figura 7** podemos observar que adotando uma posição  $Z$  e efetuando a resolução do triângulo de posição  $[PZS]$ , obtemos os resultados da altura estimada e do azimute. Seguindo este raciocínio obtemos, através da diferença do parágrafo anterior, o  $\Delta a$  onde a partir do qual será marcada a reta de posição perpendicular ao azimute nesse ponto. Observamos assim que, o principal conceito do método em estudo é equivalente à comparação direta entre duas distâncias zenitais (Escola Naval, 1995), como observado na figura abaixo.

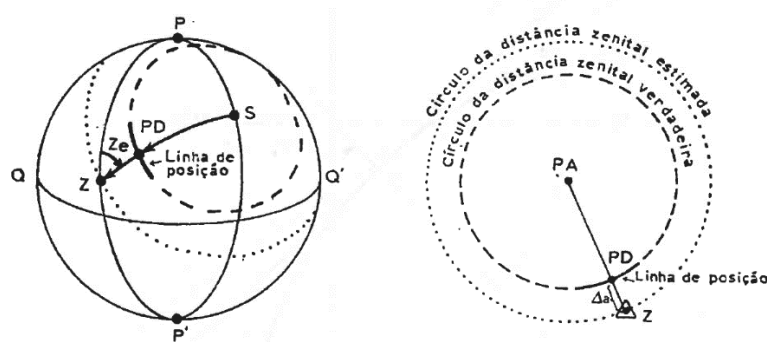


Figura 7: Comparação entre duas distâncias zenitais (Escola Naval, 1995)

Após obtido o  $\Delta a$  conseguimos retirar algumas conclusões relativas à nossa posição. Estas conclusões resumem-se a duas situações que se relacionam diretamente com o sinal aritmético do valor do  $\Delta a$  obtido:

Caso 1: Se a altura verdadeira for maior do que a altura estimada ( $\Delta a > 0$ ), o observador estará mais próximo da posição astral.

Caso 2: Se a altura verdadeira for menor do que a altura estimada ( $\Delta a < 0$ ), o observador estará mais afastado da posição astral.

Têm-se como posição astral (PA) a projeção do astro na superfície terrestre.

Na **Figura 8** pode observar-se o aspeto gráfico para as duas situações descritas.

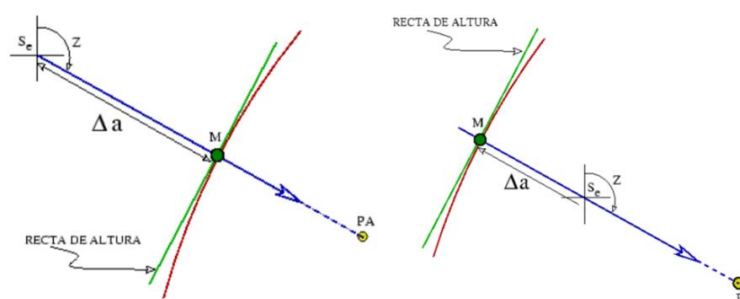
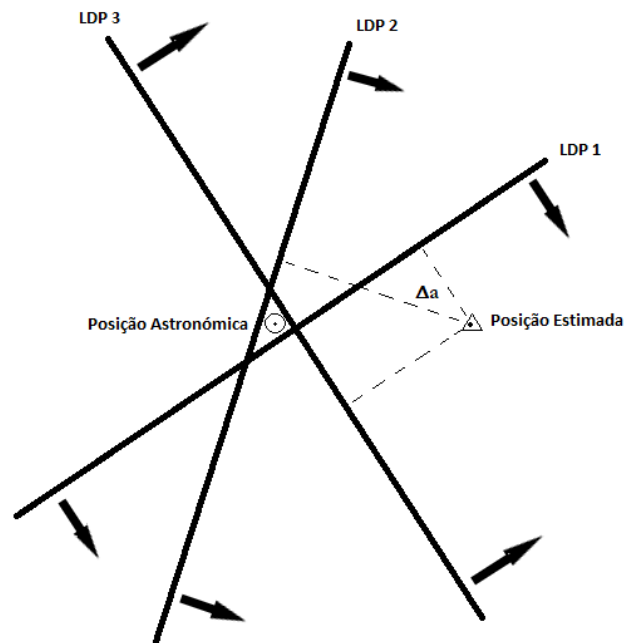


Figura 8:  $\Delta a > 0$  e  $\Delta a < 0$  respetivamente (Mederos, s.d.)

Note-se que durante todo este processo, e pela impossibilidade de traçar uma curva da circunferência de altura do astro, recorre-se ao traçado de uma reta tangente a um segmento dessa mesma circunferência, cuja distância é dada pelo  $\Delta a$ , e o traçado é feito perpendicularmente à linha de direção do astro.

Após a observação de diversos astros e do traçado das suas correspondentes retas de altura numa carta, podemos obter, na prática, uma figura geométrica que em tudo se assemelha, na navegação costeira, a uma situação de cruzamento de linhas de posição traçadas a partir de pontos conspícuos. É possível verificar na **Figura 9** a demonstração de uma representação gráfica do cruzamento das retas de posição.



**Figura 9:** Representação gráfica do traçado de retas de posição.<sup>5</sup>

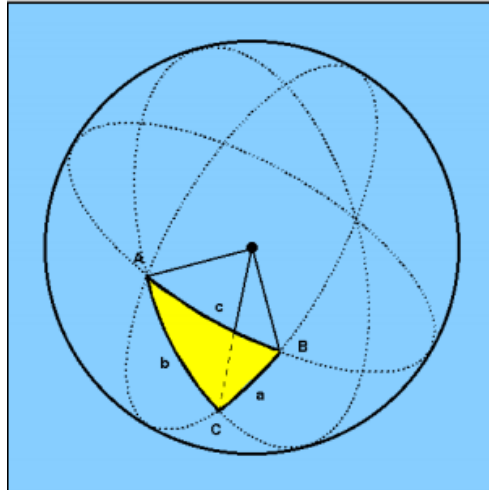
Assim, o resultado obtido pelo cálculo da posição final cairá sobre a interpretação visual e análise do observador, isto para o caso de um traçado manual. No caso de softwares informáticos, existem modelos matemáticos que permitem ao computador encontrar a melhor solução possível para os dados introduzidos pelo observador.

---

<sup>5</sup> Esta representação foi elaborada pelo autor da dissertação para elucidar a descrição realizada;

#### 1.4. O Triângulo de Posição

Como representado na **Figura 10**, considera-se por triângulo esférico a figura geométrica obtida (na superfície da esfera) pela interceção de três círculos máximos<sup>6</sup> que se cruzem entre si mesmos (Mederos, s.d.).



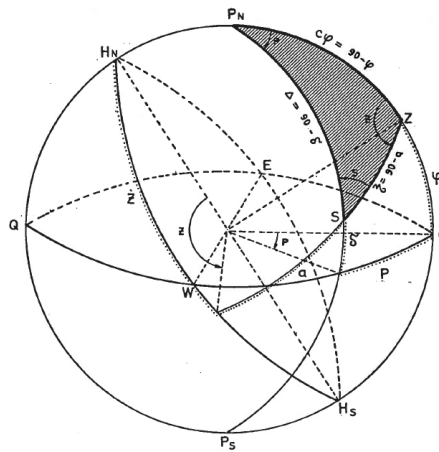
**Figura 10:** Triângulo Esférico (Mederos, s.d.)

O triângulo de posição, a partir do qual é estudada e calculada a posição do observador, é um triângulo esférico conseguido pela interceção de círculos máximos conhecidos pelo observador. Estas interceções originam os seguintes vértices (**Figura 11**):

- Pólo Elevado (P);
- Zénite do Observador (Z);
- Astro (S).

---

<sup>6</sup>Um círculo máximo consiste numa secção circular da esfera que possui um diâmetro igual ao da esfera onde está inscrito (Kern e Bland, 1948, p. 87, cit in. Weisstein, 2003).



**Figura 11:** Triângulo de Posição (Gameiro, 1964, pp. 81)

Após a obtenção do triângulo, podemos constatar a presença dos seus três lados, sendo eles:

- $PZ = \text{Co-Latidude } (C\varphi)$ ;
- $ZS = \text{Distância Zenital } (\zeta)$ ;
- $PS = \text{Distância Polar } (\Delta)$ .

Salienta-se que os lados conhecidos pelo observador recaem sobre cálculos necessários para os poder obter. Sendo que as seguintes fórmulas são utilizadas para esse efeito (Silva Gameiro, 1964, pp. 82):

- $C\varphi = 90^\circ - \varphi$ ;
- $\zeta = 90^\circ - \text{altura verdadeira}$ ;
- $\Delta = 90^\circ - \text{declinação } (\delta)^7$ .

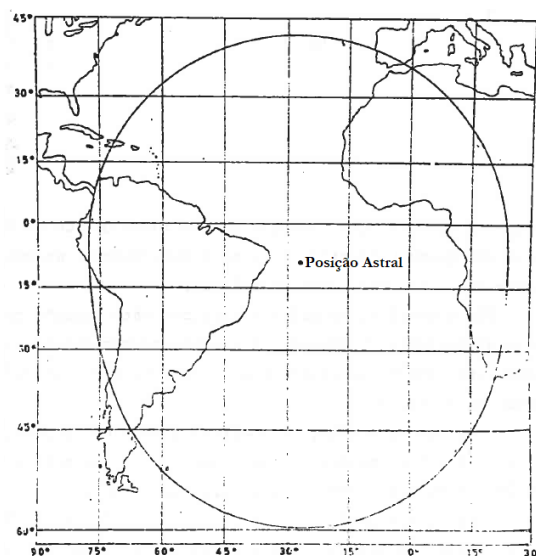
A resolução deste triângulo de posição é a base de partida para a obtenção dos valores de altura estimada e azimute, bem como para a resolução de todos os problemas de navegação astronómica. Assim, o ângulo no Polo relaciona-se com o Ângulo Horário no

<sup>7</sup> A declinação deverá ser positiva se for do mesmo nome que a Latidude e negativa no caso contrário.

Lugar (*LHA*), onde este último irá desempenhar um papel fundamental nas fórmulas necessárias para o cálculo do azimute e cálculo da altura estimada.

### 1.5. Curvas de Altura

Tem-se como circunferência de altura de um astro uma circunferência centrada na posição astral, projetada na superfície da Terra, e com raio igual ao valor da distância zenital. Desta maneira, para uma carta com projeção de Mercator, tornar-se-ia impossível o seu traçado na mesma, pelo que se adota um procedimento que visa aproximar uma reta dessa mesma circunferência como se irá explicar de seguida.



**Figura 12:** Representação prática de uma curva de altura em projeção Mercator (Gameiro, 1964)

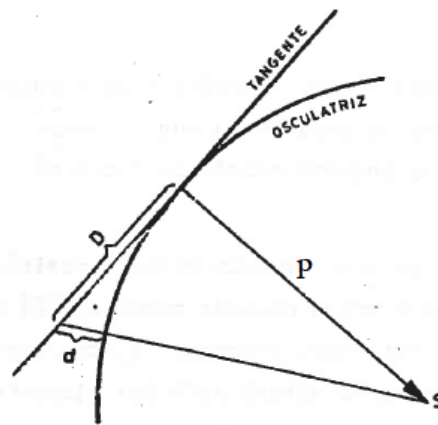
A necessidade de desenhar, na respetiva carta, uma linha de posição (representante de apenas um segmento da curva) torna-se impossível devido à sua grande dimensão. Desta forma adota-se a substituição dessa curva de altura por uma circunferência oscultriz a ela mesma. Isto permitirá uma aproximação simples da posição provável do navio, mantendo o rigor necessário na prática da navegação astronómica (Escola Naval, 1995).

“Com efeito, é possível confundir, sem cometer grande erro e em consideráveis extensões, a curva de altura pela sua circunferência oscultriz” (Escola Naval, 1995).

No entanto, para o traçado final da reta de altura do astro, existe a necessidade de se saber verdadeiramente a distância entre a circunferência oscultriz e a reta desenhada. Isto de forma a cometer um erro máximo tolerável de uma milha.

Na **Figura 13**, pode observar-se três variáveis:

- $d$  – Distância de um ponto da reta de altura até um ponto da circunferência oscultriz;
- $D$  – Distância do ponto observado e o ponto determinante;
- $P$  – Raio da circunferência oscultriz.



**Figura 13:** Tangente de altura (Gameiro,1964)

Tomando todas estas variáveis, e após a devida interpretação matemática, é possível construir uma tabela que representativa da distância máxima admissível entre o ponto determinante e o ponto observado (**Figura 14**).

LATIT.	ALTURA DO ASTRO												
	0°	10°	20°	30°	40°	50°	60°	70°	80°	82°	84°	86°	88°
0°	—	197	137	109	90	76	63	50	35	31	27	22	15
10	197	140	113	95	82	71	60	48	34	31	27	22	15
20	137	113	97	85	76	66	57	47	34	30	26	22	15
30	109	95	85	77	70	62	55	45	33	30	26	21	15
40	90	82	76	70	64	58	52	44	32	28	25	21	15
50	76	71	66	62	58	54	48	42	32	29	25	21	15
60	63	60	57	55	52	48	45	39	30	28	25	21	15
70	50	48	47	46	44	42	39	35	29	26	24	20	15

Figura 14: Distância (milhas) máxima admissível entre o ponto determinante e o ponto observado (Ministério da Marinha & Instituto Hidrográfico, 1973)

Na realidade, com o aumentar do valor de altura de um astro, estaremos a diminuir cada vez mais o comprimento admissível da reta de altura. Por sua vez, para um valor de  $a$  superior a  $88^\circ$  considerar-se-á mais exato o desenho da circunferência de altura do astro na carta, em vez do desenho da reta correspondente. Para esta situação específica o observador encontra-se na presença de circum-zenitais (Escola Naval, 1995).

## 1.6. Equações de Cálculo

Com a finalidade de obter um resultado final rigoroso, e dispendo de cálculo computacional com elevada precisão, é necessário recorrer a diversas fórmulas para o cálculo de todas as variáveis que são utilizadas na sequência de cálculo do ponto astronómico. Sendo elas a altura verdadeira, a altura estimada, o azimute, o  $\Delta a$  e o cálculo da posição final.

### 1.6.1. Altura Corrigida

Todas as alturas obtidas com um sextante são afetadas por fenómenos físicos. Assim, as correções de altitude têm como principal objetivo eliminar ou reduzir esses erros. A altura verdadeira é obtida a partir da adição e/ou subtração de correções.

As correções são aplicadas diretamente no valor da altura instrumental, sendo elas: o erro de índice, a depressão, a refração atmosférica, a paralaxe e o semidiâmetro.

Todas estas correções são fundamentais para o rigor das observações efetuadas, e serão em seguida explicadas.

Qualquer sextante que seja utilizado com alguma regularidade pode apresentar os seus espelhos de reflexão desajustados. Cabe ao utilizador registar o valor, em minutos de grau, do erro de índice do sextante antes de proceder à observação e registo de alturas. O erro considera-se positivo quando o valor apresentado é maior do que o valor real, e negativo no caso contrário. Se este for igual, considera-se que não existe Erro de Índice (EI) (Umland, 2004).

A fórmula que permite aplicar esta correção é aplicada em primeiro lugar na altura observada. Assim o novo valor de altura verdadeira será dado por:

$$1^{\text{a}} \text{ Correção}^8: H_1 = Hs + EI$$

A altitude de observação tem influência direta na observação e registo de alturas observadas, isto porque a terra apresenta uma superfície esférica e atmosfera. Estes dois fatores influenciam a refração da luz e por fim transmitem uma sensação errada de horizonte geométrico. Desta forma é necessário calcular a correção para a **depressão**, que é obtida correlacionando a altitude de observação com a diferença entre o horizonte geométrico e o horizonte visível, como é possível observar na **Figura 15**.

É neste sentido que o observador terá que registar a altura ou elevação (metros) a que está a observar os astros, de forma a que possa anular os efeitos combinados que a curvatura da terra e a refração atmosférica trazem para as observações.

Desta forma, a diferença do horizonte geométrico e o horizonte visível (depressão ou em inglês *dip*), onde *HE* é a altitude de observação, é dada pela seguinte fórmula<sup>9</sup>.

$$DIP \approx 1.76 \times \sqrt{HE(m)}$$

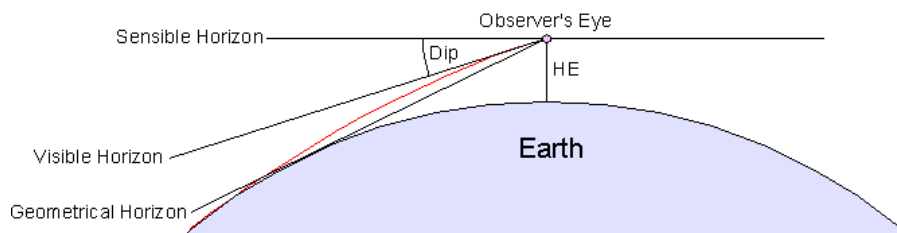
A fórmula que permite aplicar esta correção na altura observada é dada por:

$$2^{\text{a}} \text{ Correção}^{10}: H_2 = H_1 - DIP$$

---

<sup>8</sup> Retirado de Umland (2004).

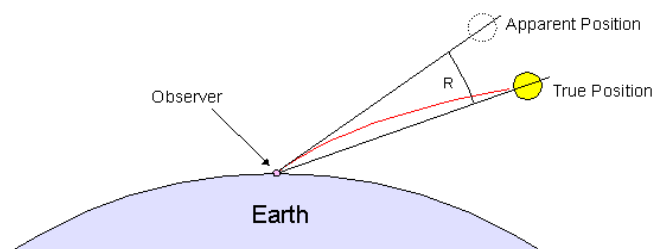
<sup>9,10</sup> Retirado de Umland (2004)



**Figura 15:** Altitude de Observação (Umland, 2004)

A **refração atmosférica**, como vimos anteriormente, influencia não só o cálculo do horizonte, mas também o verdadeiro posicionamento dos astros no céu aquando a sua observação.

Ao atravessarem a espessa camada atmosférica, os raios luminosos sofrem desvios. Estes desvios são tanto maiores quanto mais próximo o astro se encontrar da linha de horizonte (**Figura 16**). Isto acontece, pois, a camada de atmosfera que a luz terá de percorrer é maior longitudinalmente do que verticalmente, em relação ao observador. Desta forma, a correção para a refração é máxima perto da linha de horizonte e nula no zénite do observador, como demonstrado na **Tabela 1** (adaptada de Umland (2004)).



**Figura 16:** Erro de Refração (Umland, 2004)

**Tabela 1:** A refração (em minutos de grau) de acordo com a altura observada

Altura observada	0°	1°	2°	5°	10°	20°	30°	40°	50°	60°	70°	80°	90°
Refração	~34'	~24'	~18'	9.9'	5.3'	2.6'	1.7'	1.2'	0.8'	0.6'	0.4'	0.2'	0.0'

No entanto, a aplicação direta da tabela apresentada não seria vantajosa nem rigorosa para um método computacional. Desta forma, para o cálculo exato dos valores intrínsecos a esta correção recorre-se a duas fórmulas que têm como valores de entrada a pressão e a temperatura ambientes. São estes valores que irão afetar o trajeto da luz na camada atmosférica.

$$f^{11} = \frac{p [m\ bar]}{1010} \times \frac{283}{273 + T [^{\circ}C]}$$

$$R_0^{12} = \frac{1}{\tan\left(H_2[^{\circ}] + \frac{7.31}{H_2[^{\circ}] + 4.4}\right)} - 0.06 \times \sin\left(14.7 \times \frac{1}{\tan\left(H_2[^{\circ}] + \frac{7.31}{H_2[^{\circ}] + 4.4}\right)} + 13\right)$$

Finalmente, basta aplicar a expressão seguinte e obter assim o valor da altura verdadeira para a correção da refração.

$$3^a \text{ Correção}^{13}: H_3 = H_2 - f \times R_0$$

A correção da **paralaxe** encontra-se diretamente relacionada com a distância do observador ao centro da Terra e com a distância do astro em relação ao planeta Terra.

Todas as alturas obtidas através da leitura do sextante fazem referência ao horizonte celeste. Desta forma, para todos os cálculos efetuados, assume-se que o observador se encontra no centro da Terra e não na sua superfície. Por este motivo, torna-se necessário corrigir esta diferença, utilizando para isso o valor de paralaxe horizontal que se encontra vertido no Almanaque Náutico. Note-se, porém, que a correção da paralaxe é tanto maior

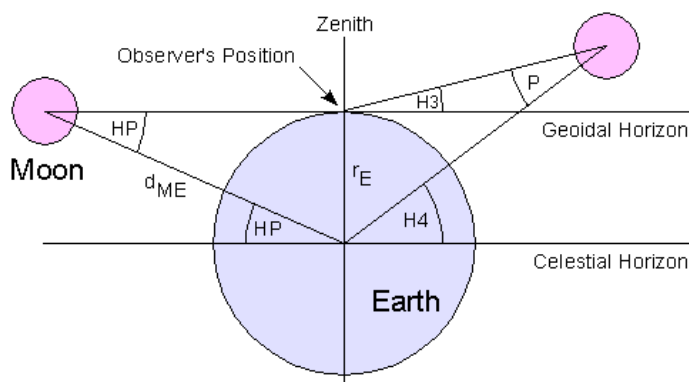
---

<sup>11</sup> Retirado de Umland (2004)

<sup>13</sup> Adaptado de Umland (2004);

quanto mais próximo o astro se encontra do nosso planeta. Para o caso das estrelas e dos planetas Júpiter e Saturno este valor é desprezável, visto que estes encontram-se bastante distantes e por este motivo a diferença entre os ângulos relativos ao horizonte geodésico e o horizonte celeste é praticamente nula. Porém a Lua, por ser o astro com maior proximidade ao planeta Terra, apresentará os maiores valores de paralaxe horizontal.

Através da **Figura 17** pretende-se exemplificar, representando a Lua, que o ângulo formado entre o astro e o horizonte geodésico é menor que o ângulo entre o astro e o horizonte celeste. No entanto, a paralaxe toma o valor nulo quando o astro se encontra no zénite do observador e máxima quando se encontra na linha do horizonte. Isto permite que, após a obtenção do valor da HP, se aplique o mesmo numa fórmula que calcula o efeito da paralaxe consoante a altura do astro observado.



**Figura 17:** Erro de Paralaxe (Umland, 2004)

Para o cálculo da correção da paralaxe, neste caso da Lua, existe a necessidade de recorrer ao Almanaque Náutico ou a outro documento de onde se possa extrair o valor de paralaxe horizontal. Após o registo deste valor, o valor da paralaxe é calculado pela seguinte fórmula (Umland, 2004):

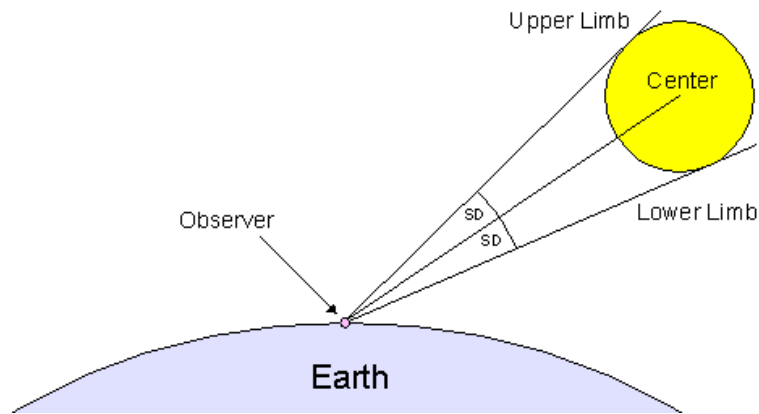
$$P = HP \times \cos(H_3) + \frac{1}{298.257} \times HP \times [\sin(2 \times \varphi) \times \cos Z_l \times \sin H_3 - \sin^2(\varphi) \times \cos H_3]$$

Após a obtenção do valor de paralaxe, onde  $\varphi$  significa a latitude de observação e  $Z_l$  o azimute calculado da lua, aplica-se o mesmo no último valor de altura verdadeira, com recurso à seguinte fórmula:

$$4^a \text{ Correção: } H_4 = H_3 + P$$

O **semidiâmetro** consiste numa correção provocada pela incapacidade de medir a altura de um astro a partir do seu centro geométrico (Sol e Lua). O observador recorre ao limbo inferior ou superior para a medição da altura do astro e, por consequência a diferença da distância que irá desde o limbo observado até ao centro geométrico da imagem do astro, terá de ser incrementada na correção da altura observada.

Na **Figura 18** pretende-se exemplificar a diferença descrita no parágrafo anterior.



**Figura 18:** Semidiâmetro (Umland, 2004)

O incremento para a última correção a fazer nas observações é calculado através do valor da paralaxe horizontal para esse mesmo astro. No caso da Lua, além da paralaxe horizontal, também se considera uma constante ( $k$ ) que corresponde ao rácio entre o raio da Lua e o raio da Terra.

$$k_{lua} = 0.2725$$

$$SD_{geocêntrico} = \sin^{-1}(k \times \sin HP) \approx k \times HP$$

Considera-se ainda que, no caso dos planetas Marte e Vénus, é necessário proceder à presente correção visto que estes planetas apresentam variação do valor de paralaxe horizontal ao longo do ano. No caso das estrelas, devido às reduzidas dimensões das suas imagens, o semidiâmetro é considerado nulo (Umland, 2004).

Resta assim, e de acordo com o limbo observado, efetuar a última correção na altura observada através da seguinte fórmula (Umland, 2004), em que limbo inferior: +SD e limbo superior: -SD:

$$5^{\text{a}} \text{ Correção: } H_5 = H_4 \pm SD_{geocêntrico}$$

Finalmente, conclui-se que  $H_5$  é o valor correspondente à altura verdadeira e que será utilizado para o cálculo final do  $\Delta a$ .

### 1.6.2. Altura estimada

Para a obtenção do valor correspondente à altura estimada é necessária uma interpretação do triângulo de posição. Na prática os valores que irão originar as variáveis de entrada na fórmula do cálculo da altura são a declinação do astro, a latitude e longitude do observador e a hora média de Greenwich. Estes dados, através da interpretação direta do almanaque náutico ou de um programa destinado para o efeito, permitem obter a declinação e o  $LHA$ .

$$a = \sin^{-1}(\sin(\varphi) \times \sin(\delta) + \cos(\varphi) \times \cos(\delta) \times \cos(LHA))$$

Nesta rotina de cálculo, caso se proceda à sua utilização manual ou através de meios tabulares, ter-se-á que garantir que:

$$\delta \times \varphi > 0$$

e

$$\varphi - \delta < 90^\circ$$

ou:

$$\delta \times \varphi < 0 \wedge \varphi - \delta < 90^\circ$$

Caso esta última não se verifique então:

$$\varphi - \delta > 90^\circ$$

Desta forma estaríamos na presença de um astro circumpolar invisível e impossível de ser observado, como demonstrado na **Figura 19**.

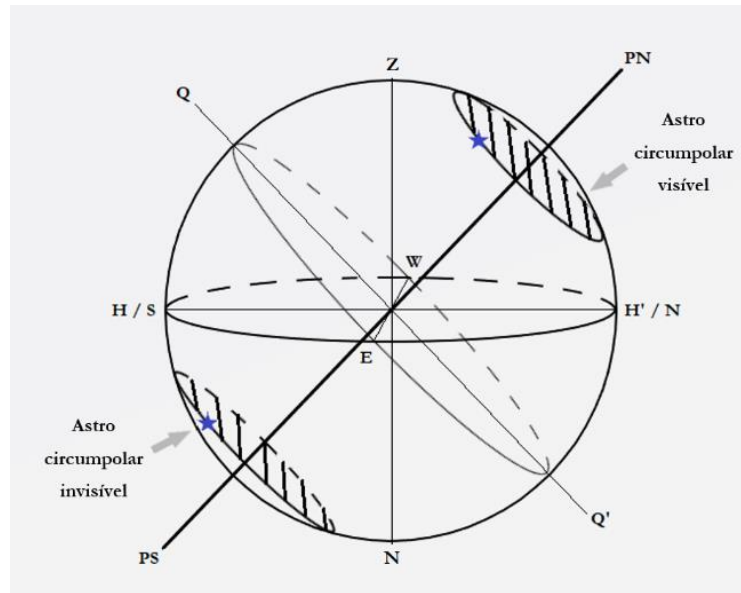


Figura 19: Representação de um astro circumpolar visível e um invisível

### 1.6.3. Cálculo do Azimute

Com a mesma lógica de cálculo da altura estimada, e já tendo o seu valor, podemos obter o valor do azimute através da seguinte fórmula:

$$Z = \cos^{-1}((\sin \delta - (\sin(\varphi) \times \sin(a)) \div (\cos(\varphi) \times \cos(a)))$$

Contudo a utilização desta fórmula pode, em certos casos, originar valores com demasiadas casas decimais. Note-se que para a representação na carta de um valor de azimute, o rigor necessário para a sua marcação é admissível até 1/2 do grau (Silva Gameiro, 1964).

Desta maneira é possível obter tabelas onde sejam apresentados os diferentes valores que o azimute e alturas estimadas podem tomar, tendo estas tabelas o ângulo horário no lugar, a latitude e a declinação do astro como variáveis de entrada (Figura 20).

LATITUDE CONTRARY NAME TO DECLINATION													L.H.A. 34°, 326°												
Dec.	0°			1°			2°			3°			4°			5°			6°			7°			Dec.
	Hc	d	Z	Hc	d	Z	Hc	d	Z	Hc	d	Z	Hc	d	Z	Hc	d	Z	Hc	d	Z	Hc	d	Z	
0	56 00.0	-0.8	90.0	55 59.2	-2.6	91.5	55 56.9	-4.5	93.0	55 53.0	-6.3	94.4	55 47.6	-8.2	95.9	55 40.7	-10.0	97.4	55 32.2	-11.8	98.8	55 22.3	-13.6	100.2	0
1	55 59.2	-2.3	91.8	55 56.6	-4.2	93.3	55 52.4	-6.1	94.7	55 46.7	-7.9	96.2	55 39.4	-9.7	97.7	55 30.7	-11.5	99.1	55 20.4	-13.2	100.5	55 08.7	-15.0	102.0	1
2	55 56.9	-3.9	93.6	55 52.4	-5.7	95.0	55 46.3	-7.5	96.5	55 38.8	-9.4	98.0	55 29.7	-11.2	99.4	55 19.2	-13.0	100.8	55 07.2	-14.7	102.2	54 53.7	-16.4	103.6	2
3	55 53.0	-5.4	95.4	55 46.7	-7.3	96.8	55 38.8	-9.1	98.3	55 29.4	-10.9	99.7	55 18.5	-12.6	101.1	55 06.2	-14.3	102.5	54 52.5	-16.1	103.9	54 37.3	-17.7	105.3	3
4	55 47.6	-6.9	97.1	55 39.4	-8.7	98.6	55 29.7	-10.5	100.0	55 18.5	-12.3	101.4	55 05.9	-14.0	102.9	54 51.9	-15.8	104.2	54 36.4	-17.4	105.6	54 19.6	-19.1	107.0	4
5	55 40.7	-8.5	98.9	55 30.7	-10.3	100.3	55 19.2	-12.0	101.8	55 06.2	-13.7	103.2	54 51.9	-15.5	104.5	54 36.1	-17.1	105.9	54 19.0	-18.8	107.3	54 00.5	-20.3	108.6	5
6	55 32.2	-9.9	100.6	55 20.4	-11.7	102.1	55 07.2	-13.5	103.5	54 52.5	-15.2	104.9	54 36.4	-16.8	106.2	54 19.0	-18.5	107.6	54 00.2	-20.0	108.9	53 40.2	-21.6	110.2	6
7	55 22.3	-11.4	102.4	55 08.7	-13.1	103.8	54 53.7	-14.8	105.2	54 37.3	-16.5	106.5	54 19.6	-18.2	107.9	54 00.5	-19.7	109.2	53 40.2	-21.3	110.5	53 18.6	-22.9	111.7	7
8	55 10.9	-12.8	104.1	54 55.6	-14.6	105.5	54 38.9	-16.2	106.9	54 20.8	-17.8	108.2	54 01.4	-19.4	109.5	53 40.8	-21.0	110.8	53 18.9	-22.6	112.0	52 55.7	-24.0	113.3	8
9	54 58.1	-14.3	105.8	54 41.0	-15.9	107.2	54 22.7	-17.6	108.5	54 03.0	-19.2	109.8	53 42.0	-20.7	111.1	53 19.8	-22.3	112.4	52 56.9	-23.7	113.5	52 31.7	-25.1	114.8	9
10	54 43.8	-15.6	107.5	54 25.1	-17.3	108.8	54 05.1	-18.9	110.1	53 43.8	-20.5	111.4	53 21.3	-22.0	112.7	52 57.5	-23.4	113.9	52 32.6	-24.8	115.1	52 06.6	-26.2	116.3	10
11	54 28.2	-17.0	109.2	54 07.8	-18.6	110.5	53 46.2	-20.2	111.8	53 23.3	-21.6	113.0	52 59.3	-23.1	114.2	52 34.1	-24.6	115.4	52 07.8	-26.0	116.6	51 40.4	-27.3	117.7	11
12	54 11.2	-18.4	110.8	53 49.2	-19.9	112.1	53 26.0	-21.4	113.3	53 01.7	-22.9	114.6	52 36.2	-24.3	115.8	52 09.5	-25.7	116.9	51 41.8	-27.0	118.1	51 13.1	-28.3	119.2	12

Figura 20: Exemplo tabular da altura estimada e azimute calculado (National Imagery And Mapping Agency, 2001)

A tabela apresentada, oferece valores de azimute que correspondem diretamente ao azimute calculado. No entanto, após a utilização direta da fórmula apresentada, notamos que o valor de azimute apresentado é correspondente ao valor de azimute astronómico, o qual é medido desde o polo elevado e no sentido do ponto cardinal Este ou Oeste.

Porém, para a representação final na tabela, este valor de azimute astronómico ( $Z$ ) terá que ser convertido no valor de azimute ( $Z_n$ ) útil para a navegação, através das seguintes condições:

$$\text{Latitude Norte} = \begin{cases} LHA > 180^\circ \rightarrow Zn = Z \\ LHA < 180^\circ \rightarrow Zn = 360^\circ - Z \end{cases}$$

$$\text{Latitude Sul} = \begin{cases} LHA > 180^\circ \rightarrow Zn = 180^\circ - Z \\ LHA < 180^\circ \rightarrow Zn = 180^\circ + Z \end{cases}$$

#### 1.6.4. Cálculo do $\Delta a$

O cálculo do  $\Delta a$  é dado pela diferença direta entre a altura estimada e a altura verdadeira. Este valor tem elevada importância para a navegação astronómica, visto ser a partir do mesmo e do azimute que se efetuam os traçados das linhas de posição na carta.

$$\Delta a = \text{altura verdadeira} - \text{altura estimada}$$



Figura 21: Representação do  $\Delta a$  (Gameiro, 1964)

### 1.6.5. Cálculo da Posição Final

Na conclusão do cálculo final, existe a necessidade de comparar a posição estimada do observador com os dados obtidos das observações ( $\Delta a$  e azimute). Este cálculo é conseguido utilizando o método dos mínimos quadrados, onde as componentes que serão utilizadas na fórmula final são calculadas através dos seguintes somatórios:

$$A = \sum \cos^2(Z_n)$$

$$B = \sum [\cos(Z_n) \times \sin(Z_n)]$$

$$C = \sum \sin^2(Z_n)$$

$$D = \sum [\Delta a_n \times \cos(Z_n)]$$

$$E = \sum [\Delta a_n \times \sin(Z_n)]$$

$$G = A \times C - B^2$$

Nestas componentes, o número de variáveis correspondentes a  $n$  é correspondente ao número de observações efetuadas. Todas elas entrarão no cálculo da posição final pelas fórmulas descritas abaixo.

$$\varphi = B_f + (C \times D - B \times E)/G$$

$$L = L_f + (A \times E - B \times D)/(G \times \cos B_f)$$

Onde:

$L_f = \text{Longitude estimada}$

$B_f = \text{Latitude estimada}$

$B = \text{Nova latitude}$

$L = \text{Nova Longitude}$

Este processo de cálculo da nova posição é obtido através do ajuste do resultado para quando a soma das componentes é mínima (Van Der Grinten, 1975).

#### 1.6.6. Iteração da Posição Final

Na conclusão do cálculo da posição final, pode obter-se valores diretos de latitude e longitude de acordo com as observações efetuadas. No entanto pode proceder-se à iteração desses valores de forma a ajustá-los e obter uma posição mais próxima da real, tanto quanto seja possível.

Usando como ponto de partida a posição obtida através do método dos mínimos quadrados e comparando-a com a posição estimada do observador, pode-se deduzir uma nova posição. Esta posição deverá ser recalculada através das fórmulas do subcapítulo anterior, mas deve ter em atenção a seguinte condição (British Admiralty, 2015, pp. 282):

$$d = 60 \times \sqrt{\left( (L - L_f)^2 \times \cos^2(B_f) + (B - B_f)^2 \right)}$$

O resultado da fórmula anterior calculará a distância entre a nova posição e a posição estimada. Na situação do valor  $d$  exceder 20 milhas, dever-se-á então adotar o seguinte:

$$L_f = L$$

e

$$B_f = B$$

Após estabelecidas estas igualdades, calcular-se-á novamente a posição final até que o valor  $d$  seja inferior a 20 milhas. Salienta-se que, sempre que este valor exceder as 20 milhas, dever-se-á recalcular novamente com base nos últimos valores obtidos, e assim sucessivamente.

### 1.7. Transporte de Retas de Altura

Após serem calculadas e traçadas na carta as retas de altura para diversos astros, pode ser necessário o seu transporte para um dado instante. Usualmente, considera-se necessário efetuar transporte de retas de altura aquando das observações do Sol. Isto deve-se ao facto de o tempo decorrido entre as observações do Sol (AM, M e PM) ser considerável. Este fator aliado à velocidade do navio, poderá traduzir-se numa grande distância percorrida.

Assim, torna-se necessário transportar as retas de altura de cada um dos momentos de observação para um dado instante onde o navio se encontrou. Convenientemente, transporta-se a antemeridiana e pós-meridiana para o momento da meridiana do Sol. Isto porque se consegue calcular com alguma exatidão a latitude do observador no momento da passagem meridiana do Sol (ponto ao meio dia).

Na prática, ao efetuar o transporte das retas de altura iremos obter um triângulo de erro. Todavia, este transporte é feito à custa da velocidade, rumo e tempo decorridos entre as observações.

A reta obtida através do transporte é sempre paralela à reta a partir do qual foi desenhada. Isto acontece, pois, é a partir da segunda que é definido um ponto onde irá ser aplicado um vetor com um determinado rumo e distância (calculada com base no tempo decorrido e velocidade).

Através de uma análise gráfica, podemos observar que transportar a reta de altura para uma determinada posição corresponderá ao que foi explicado anteriormente. No entanto, para um processo analítico, considera-se o transporte da posição estimada de uma determinada observação para uma outra posição estimada, onde será com esta última que se irá aplicar todos os cálculos efetuados e calcular a partir dela a posição final.

## Capítulo 2. Algoritmos e estrutura de cálculo

Durante o cálculo do ponto astronómico, e seguindo o raciocínio do método estudado neste trabalho, existe uma lógica sequencial de aplicação dos algoritmos de cálculo. Estes, por sua vez, culminam na obtenção de uma posição final.

Para a estruturação e seguimento do cálculo do ponto astronómico analisaram-se diversas publicações, nomeadamente o Almanaque Náutico. Dele foram extraídas as fórmulas que dão desenvolvimento ao referido cálculo (British Admiralty, 2015).

Pretende-se neste capítulo mostrar a estratificação e organização do cálculo do ponto astronómico tendo como base de partida dados retirados do Almanaque Náutico e dados calculados pelo observador.

### 2.1. Algoritmo do Cálculo da Posição Estimada

O primeiro passo para dar início a este processo, é a obtenção de uma posição geográfica estimada do observador.

Para isso é necessário que o mesmo faça uma recolha da posição de partida do navio, hora, rumo e velocidade.

*STEP 1* – Obtenção dos dados de entrada nas fórmulas

- Horas – Formato (hora / minutos / segundos)
- Rumo – Formato ( $0^\circ - 360^\circ$ , rigor de  $1^\circ$ )
- Velocidade – Formato (nós, rigor de  $\frac{1}{2}$  nó)
- Posição geográfica inicial – Latitude ( $\varphi$ ) e Longitude ( $\lambda$ ) em graus e décimas de grau. Se a latitude for sul deverá tomar valor negativo, bem como se a longitude for oeste.

*STEP 2* – Calcular o valor  $t$

Neste passo dever-se-á escolher a hora para a qual se quer calcular a nossa posição. Seguidamente efetua-se o cálculo da variável  $t$ .

- $t = \text{hora} - \text{hora inicial}$

*STEP 3* – Cálculo da posição estimada

- A posição estimada será dada por (British Admiralty, 2015):

$$\text{Latitude estimada} = \varphi \text{ inicial} + t \times \left( \frac{\text{velocidade}}{60} \right) \times \cos(\text{rumo})$$

$$\text{Longitude estimada} = \text{longitude inicial} + \left[ \frac{t \times \left( \frac{\text{velocidade}}{60} \right) \times \sin(\text{rumo})}{\cos(\varphi \text{ inicial})} \right]$$

É possível observar na figura seguinte o fluxograma dos passos a tomar, já mencionados, de uma forma esquemática.

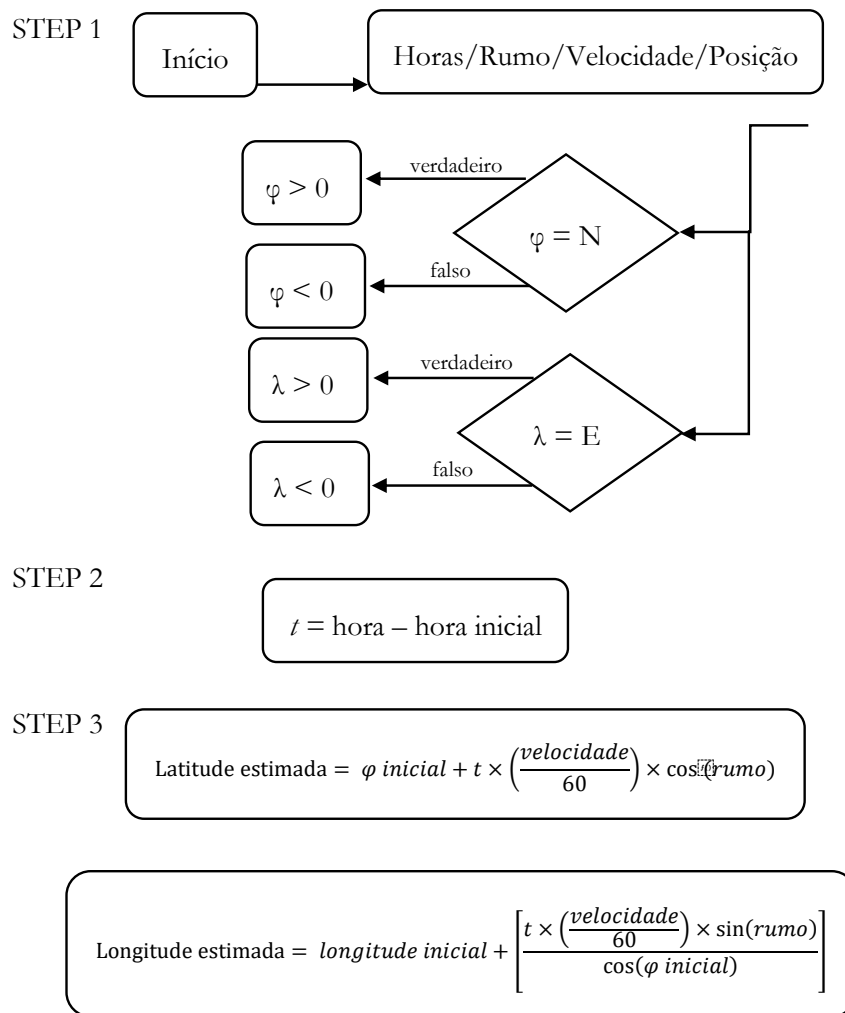


Figura 22: Fluxograma de cálculo da posição estimada<sup>14</sup>

## 2.2. Algoritmo do Cálculo da altura estimada

Para o cálculo da altura estimada são necessários valores extraídos do almanaque náutico e a posição estimada do observador. Os valores a recolher do almanaque náutico são

<sup>14</sup> Este fluxograma foi elaborado pelo autor da dissertação para efeitos de explicitação;

o GHA ou ângulo sideral, e a declinação do astro em estudo. Para o caso das estrelas recolhe-se também o valor do GHA relativo ao ponto vernal.

*STEP 1* – Transformação do GHA em LHA

- Retirar do almanaque o valor de GHA.

$$LHA = GHA + Longitude\ do\ Observador$$

- A Longitude deverá apresentar sinal aritmético negativo se for de nome Oeste e positivo se for de nome Este.
- Por forma a ser aplicável esse valor em graus, dever-se-á apresentar o valor do LHA como apresentado no subcapítulo 1.2.2.

*STEP 2* – Cálculo da altura estimada

- Aplicar a latitude estimada do observador e a declinação e LHA do astro, no cálculo da altura.

$$a = \sin^{-1}(\sin(\varphi) \times \sin(\delta) + \cos(\varphi) \times \cos(\delta) \times \cos(LHA))$$

### 2.3. Algoritmo do Cálculo da altura verdadeira

Neste campo de cálculo o observador terá que introduzir a altura observada registada com o sextante. Além deste dado, o observador deverá registar a temperatura ambiente, a pressão atmosférica, a altitude de observação e o erro de índice do sextante, isto por forma a que as seguintes correções sejam aplicáveis.

- Temperatura – Formato (graus Celcius)
- Altitude de observação – Formato (metros)
- Erro de índice – Formato (minutos de grau)
- Pressão atmosférica – Formato (milibares)

- Todas as restantes variáveis serão recolhidas do almanaque náutico e apresentadas até à décima de minuto (paralaxe e semidiâmetro)

De seguida, é apresentada a lógica de cálculo enunciada no subcapítulo 1.6.1 pela seguinte sequência:

### 1ª correção

$$\text{Erro de índice} \quad H_1 = H_s + \text{Erro de índice}$$

### 2ª Correção

$$\text{Depressão:} \quad H_2 = H_1 - 1.76 \times \sqrt{HE(m)}$$

### 3ª Correção

$$\begin{aligned} \text{Temperatura de Pressão:} \quad H_3 = H_2 - \frac{p [m \text{ bar}]}{1010} \times \frac{283}{273+T [^\circ\text{C}]} \times \\ \frac{1}{\tan\left(H_2 [^\circ] + \frac{7.31}{H_2 [^\circ] + 4.4}\right)} - 0.06 \times \sin\left(14.7 \times \frac{1}{\tan\left(H_2 [^\circ] + \frac{7.31}{H_2 [^\circ] + 4.4}\right)} + 13\right) \end{aligned}$$

### 4ª Correção

O cálculo da **paralaxe** é aplicável apenas aos astros mais próximos da Terra sendo eles: a Lua, o Sol, Vénus e Marte.

Para o cálculo da correção da **paralaxe da Lua**, esta pode assumir valores bastantes significativos (cerca de 1° em algumas circunstâncias). Deste modo, a fórmula que proporciona a sua correção é a seguinte:

$$\begin{aligned} H_4 = H_3 + HP \times \cos(H_3) \\ + \frac{1}{298.257} \times HP \times [\sin(2 \times \varphi) \times \cos Z_l \times \sin H_3 - \sin^2(\varphi) \times \cos H_3] \end{aligned}$$

A correção da **paralaxe para o Sol** é calculada por:

$$H_4 = H_3 + 0.15 \times \cos(H_3)$$

Por último no caso de **Vénus e Marte** a fórmula aplicável é:

$$H_4 = H_3 + HP \times \cos(H_3)$$

#### 5ª correção

A **correção do semidiâmetro** é aplicável para os astros aos quais podemos diferenciar o limbo de observação, tais como a Lua e o Sol, e que se podem calcular da seguinte forma:

- Lua

$$H_5 = H_4 \pm k \times HP$$

onde,

$$k = 0.2725 \quad \text{e} \quad k \times HP = \textit{Semidiâmetro da Lua}$$

- Sol

$$H_5 = H_4 \pm SD$$

Em ambas as expressões o sinal aritmético positivo ou negativo, representará correspondentemente o limbo inferior ou superior do astro.

Finalmente, após as correções supracitadas, obtém-se a altura verdadeira do astro observado ( $H_5$ ).

## 2.4. Cálculo do $\Delta a$

Reunindo os dados dos dois últimos subcapítulos efetuamos uma diferença direta entre eles e obtemos o  $\Delta a$ .

$$\Delta a = H_5 - \textit{altura estimada}$$

O  $\Delta a$  desempenhará um papel fundamental no algoritmo de cálculo da posição final.

## 2.5. Algoritmo de Cálculo do azimute

Para o cálculo do azimute, através da fórmula apresentada em seguida, o observador terá de reunir os seguintes dados de entrada na equação:

*STEP 1* – Calcular o Azimute

- Declinação do astro ( $\delta$ ) – Formato (graus e décimas de grau, Norte ou Sul)
- Latitude estimada ( $\varphi$ ) – Formato (graus e décimas de grau, Norte ou Sul)
- Altura estimada ( $a$ ) – Formato (graus e décimas de grau)

$$Z = \cos^{-1}((\sin \delta - (\sin(\varphi) \times \sin(a)) \div (\cos(\varphi) \times \cos(a)))$$

*STEP 2* – Correlacionar  $Z$  com o  $LHA$  e obter o azimute

Para este passo será necessária uma condição, utilizando o valor do LHA do astro observado. Será depois desta que obteremos o valor de azimute a considerar para o cálculo da posição final.

Se  $LHA > 180$

Então  $Z_{final} = Z$

Mas se  $LHA < 180$

Então,  $Z_{final} = 360 - Z$

## 2.6. Algoritmo de cálculo da posição final

Consoante o número de observações efetuadas, ir-se-á aplicar os  $\Delta a$  obtidos como também os azimutes e a posição estimada.

- $\Delta a$  – Formato (rigor até à décima de minuto)
- Azimutes – Formato ( $0^\circ - 360^\circ$ , em número inteiro)

- Posição estimada – Formato (grau e décimas de grau, negativas ou positivas consoante o nome)

Utilizando estas variáveis nos somatórios e fórmulas enunciados no subcapítulo 1.6.5, aplicamos as fórmulas de cálculo para obtenção da posição final.

### Capítulo 3. Construção e Funcionalidade da Ferramenta de Cálculo

O cálculo do ponto astronómico pelo método Marcq Saint-Hilaire torna-se menos preciso e cómodo aquando a sua utilização com recurso a métodos tabulares.

A ferramenta desenvolvida vem, de certa forma, facilitar o processo de cálculo do ponto astronómico. Um dos objetivos da mesma é proporcionar ao observador, uma maneira simples de calcular um ponto astronómico com recurso a um processo simples e sequencial.

Toda a construção da ferramenta foi efetuada a partir do software de programação MatLab 2015a, sendo necessária a construção de uma *Graphical User Interface* (GUI) para a existência de interatividade com o utilizador.

De seguida será apresentada a estrutura da ferramenta, bem como a sua sequência de utilização e alguns procedimentos lógicos que possibilitaram algumas das suas capacidades interativas.

#### 3.1. Ponto Astronómico ASTROTH2020

O principal objetivo da ferramenta desenvolvida é o cálculo de uma posição rigorosa, com a vantagem de ser possível ao utilizador optar pelos dados a inserir na mesma. Obteve-se esta vantagem através da formulação e construção de uma tabela que irá ser explicada ainda neste capítulo.

Na **Figura 23** é possível observar a representação da ferramenta desenvolvida e posteriormente a explicação do funcionamento de cada um dos respetivos campos de utilização.

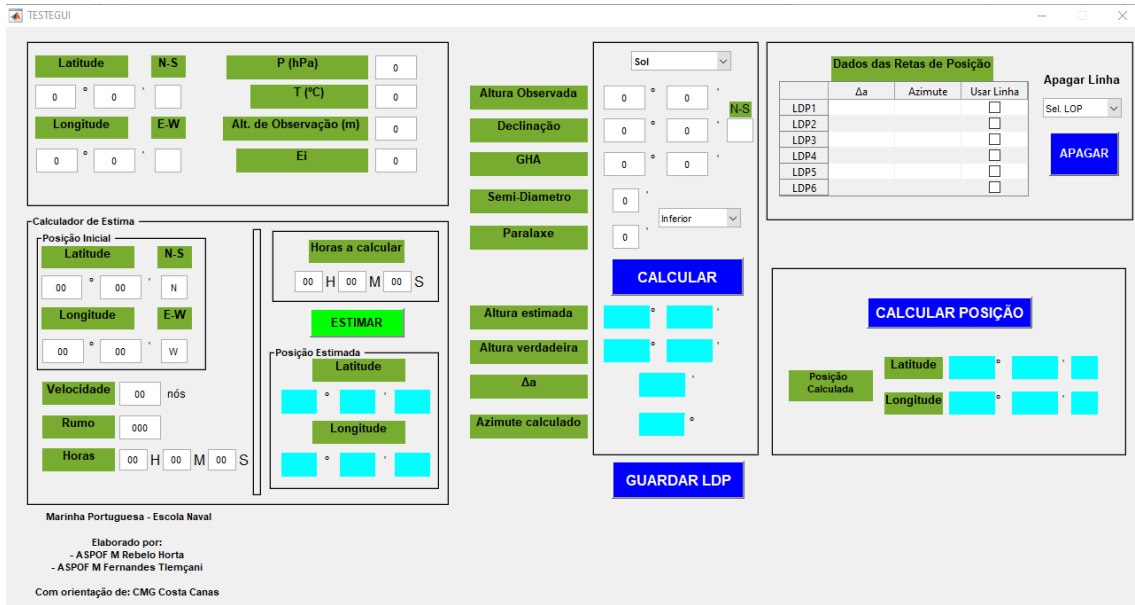


Figura 23: Ferramenta Ponto Astronómico

### 3.1.1. Organização Estrutural da Ferramenta

#### Calculador de Estima

Na parte inferior esquerda da ferramenta pode observar-se um calculador da posição estimada (**Figura 24**) para uma determinada hora. Este calculador permite que o observador possa variar a sua posição para o cálculo dos parâmetros de cada astro.

Esta ferramenta pode ter uma enorme influência no rigor dos dados apresentados, caso a velocidade do navio seja considerável, que implique a eventual necessidade de transporte de retas de posição.

É de realçar, no entanto, que este auxiliar de estima não considera a corrente nem o vento como fatores de cálculo.

O utilizador deve introduzir os dados no lado esquerdo do calculador e a hora para a qual pretende obter uma posição. Deve de seguida pressionar o botão “ESTIMAR” para a obtenção da posição desejada.

Figura 24: Calculador da posição estimada

### Transporte de retas

Ainda no âmbito da estima, existe uma particularidade na ferramenta desenvolvida que se pode considerar vantajosa.

Esta particularidade centra-se no facto de o utilizador poder calcular  $\Delta$ 's e azimutes com base em determinadas posições e estes serem guardados na tabela para esse efeito. No caso do Sol, o utilizador terá que calcular os  $\Delta$ 's e azimutes correspondentes às posições de estima referentes aos três momentos de observação desse astro (AM, M e PM). Note-se então que para aplicar o transporte a que seriam sujeitas as retas, bastará ao utilizador inserir no campo da posição estimada (**Figura 25**), a posição estimada final para o qual querará efetuar o transporte e por consequência obter a posição calculada. Assim, os dados que se encontravam guardados na tabela (referentes a três posições estimadas diferentes) serão automaticamente transportados para a posição estimada à escolha do observador, que normalmente, no caso do Sol, coincide com a posição correspondente à hora da passagem meridiana.

### Posição estimada para o cálculo e dados de correção

No campo da posição de cálculo e de dados de correção (**Figura 25**) são introduzidos os valores da posição estimada a utilizar, bem como os dados referentes às correções que se irão realizar no cálculo da altura verdadeira.

É de referir que os valores de referência para condições ambientais normais é de 1010 hPa para a pressão atmosférica e de 10°C para a temperatura do ar. Estes valores, juntamente com uma elevação de 5 metros e um Ei de 0 minutos, vêm por *default* representados nos respetivos campos ao iniciar a ferramenta.

Posição Estimada		Correções	
<b>Latitude</b>	<b>N-S</b>	<b>P (hPa)</b>	<input type="text" value="0"/>
<input type="text" value="0"/> ° <input type="text" value="0"/>	<input type="text"/>	<b>T (°C)</b>	<input type="text" value="0"/>
<b>Longitude</b>	<b>E-W</b>	<b>Alt. de Observação (m)</b>	<input type="text" value="0"/>
<input type="text" value="0"/> ° <input type="text" value="0"/>	<input type="text"/>	<b>Ei</b>	<input type="text" value="0"/>

**Figura 25:** Posição estimada e correções

### Executador de cálculo do azimute e $\Delta a$

Por sua vez, no campo do executador de cálculo do azimute e  $\Delta a$ , através do botão “CALCULAR” apresentado na **Figura 26**, existe a possibilidade de escolher o astro para o qual se pretende efetuar os cálculos finais. Sendo que, no caso do Sol e da Lua, também se encontra disponível a opção de escolha do limbo observado (**Figura 28**).

Todavia, é de realçar que no campo destinado a “Estrelas/Planetas” (**Figura 27**), o cálculo corresponde a todas as estrelas representadas no almanaque náutico e a todos os planetas, exceto Marte e Vénus que possuem a sua própria opção de cálculo. Isto deve-se ao facto de existirem correções especiais a introduzir para estes astros. No caso das estrelas o campo de “Paralaxe” e “Semidiâmetro” são desprezáveis.

The interface features a central panel with input fields and a 'CALCULAR' button. To the left, there are green buttons for 'Altura Observada', 'Declinação', 'GHA', 'Semi-Diametro', and 'Paralaxe'. Below the central panel, there are green buttons for 'Altura estimada', 'Altura verdadeira', ' $\Delta a$ ', and 'Azimute calculado'. At the bottom is a 'GUARDAR LDP' button. Callouts point to the input fields (labeled 'Variáveis do Almanaque e altura observada') and the output fields (labeled 'Dados de saída').

Figura 26: Variáveis e dados de saída

The dropdown menu is open, showing the following options: Sol, Lua, Estrela / Planeta, Vénus, and Marte. A callout points to the 'Estrela / Planeta' option with the label 'Todas as Estrelas'. The background shows the 'Altura Observada', 'Declinação', and 'GHA' input fields.

Figura 27: Escolha do astro

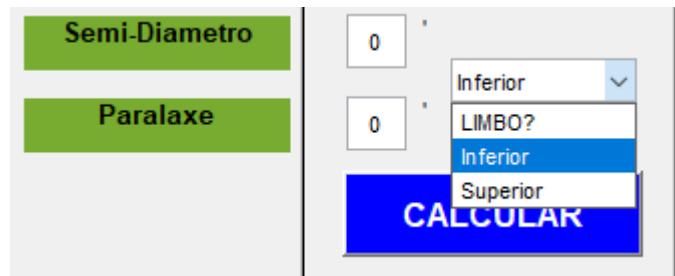


Figura 28: Escolha do limbo observado

### Tabela de armazenamento de dados

Para que seja possível o cálculo da posição final à custa de diversos astros, tornou-se necessário a criação de uma tabela (Figura 29) onde fosse possível o armazenamento dos diversos  $\Delta a$  obtidos e correspondentes azimutes.

Dados das Retas de Posição			
	$\Delta a$	Azimute	Usar Linha
LDP1			<input type="checkbox"/>
LDP2			<input type="checkbox"/>
LDP3			<input type="checkbox"/>
LDP4			<input type="checkbox"/>
LDP5			<input type="checkbox"/>
LDP6			<input type="checkbox"/>

Figura 29: Tabela de linhas de posição

Porém, houve a necessidade de limitar o número de dados relativos às linhas de posição até um máximo de 6, uma vez que se considerou suficiente para o rigor de cálculo final.

A tabela apresentada, tem duas funcionalidades de elevada importância, nomeadamente:

- Capacidade de escolha do observador relativamente às linhas de posição que este querera adotar para o cálculo da posição final (Figura 30). Note-se que

esta possibilidade é permitida mesmo mantendo os respetivos dados, isto é, não é obrigatório a eliminação de uma certa linha de posição para efetuar o cálculo final. Contudo, esta possibilidade oferece a vantagem de testar o cálculo final com diferentes linhas de posição e conseqüentemente diferentes posições finais.



Figura 30: Usar LDP

- Capacidade de eliminação de uma certa linha de posição à escolha do utilizador (Figura 31).

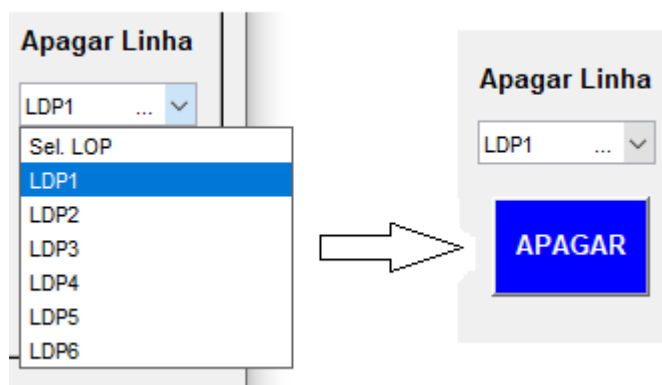


Figura 31: Opção de eliminação definitiva de LDP

O armazenamento de dados na tabela é feito pressionando o botão “GUARDAR LDP” (Figura 32). Este mecanismo está inteiramente relacionado com o segundo ponto descrito anteriormente, visto que, consoante exista uma vaga disponível na tabela, o botão

funcionará de acordo com o previsto, efetuando o armazenamento dos dados na tabela. Caso o utilizador tenha eliminado uma certa linha e queira guardar novos dados nesta, poderá fazê-lo de igual forma.

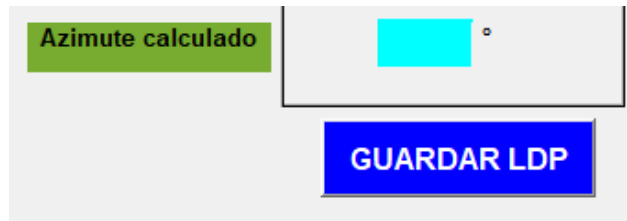


Figura 32: Guardar LDP

### Calculador de Posição Final

O último campo de funcionamento interativo com o utilizador é o de cálculo da posição final (**Figura 33**). Para efetuar esta operação, o utilizador deverá pressionar o botão “CALCULAR POSIÇÃO”, depois de escolhidas quais as linhas de posição a utilizar.

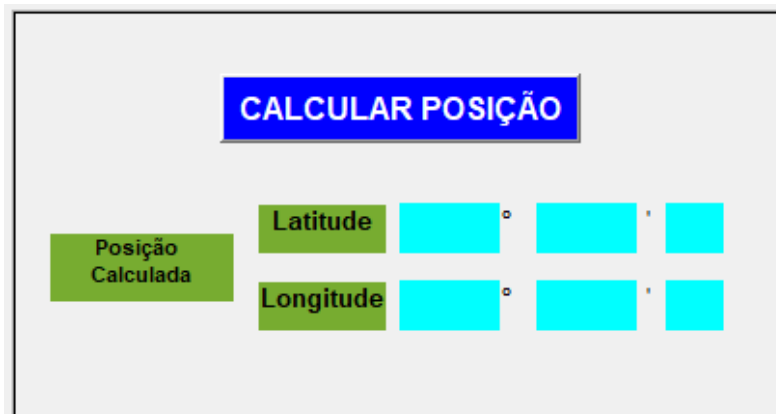


Figura 33: Calcular posição final

Contudo, na sequência de cálculos através das fórmulas descritas no subcapítulo 2.6, a posição estimada a utilizar neste processo será sempre a que se encontrar inserida no campo predestinado para esse efeito no início da GUI.

Este campo de cálculo permite a execução do mesmo vezes sem conta, não sendo necessário reiniciar o programa ou a utilização de um botão de *reset*.



## Capítulo 4. Discussão de Resultados

De forma a perceber a fiabilidade do programa construído, procedeu-se à comparação dos valores correspondentes à posição final entre o programa ASTROTH2020 e programas que têm como produto final o mesmo resultado. Após a obtenção dos valores por parte destes programas e do ASTROTH2020, efetuou-se a comparação direta dos resultados obtidos.

Existiu assim uma necessidade de definir uma sequência de valores de entrada que serviram para efetuar o cálculo nos diversos programas de referência, sendo eles:

- NAV32;
- NAVPAC.

Ao longo do presente capítulo irá ser apresentado a sequência de raciocínio e toda a estrutura de cálculo, bem como algumas referências de utilização dos programas que neste estudo foram envolvidos.

### 4.1. Critério de Recolha e Escolha de Dados

#### 4.1.1. Dados sem dependência de programas

No presente estudo, definiu-se inicialmente diversas posições geográficas. Para tal recorreu-se a um gerador aleatório on-line (Random.org), de posições na superfície terrestre. Foram recolhidas ao todo 20 posições geográficas, que naturalmente culminaram na obtenção de 20 dados de amostra.

As posições obtidas são apresentadas na **Tabela 2**, com as suas respetivas Latitude e Longitude. Todos os dados presentes na tabela encontram-se representados com o valor do grau e arredondados até ao seu milionésimo. Nos mesmo o sinal aritmético negativo representa valores Sul para as Latitudes, no caso das Longitudes este representa valores Oeste.

**Tabela 2: Posições Geográficas**

Posição	Latitude	Longitude
1	-3,03	-132,546667
2	2,0166667	-173,905
3	76,666667	-160,11
4	-34,99833	-141,231667
5	-33,12167	156,285
6	60,743333	-20,385
7	-44,925	-148,178333
8	43,981667	143,698333
9	-56,95167	-138,128333
10	0,185	-3,96166667
11	3,4533333	123,703333
12	4,1466667	-79,66
13	54,883333	-100,08
14	13,871667	19,6216667
15	-20,77333	-109,925
16	-27,65333	-34,98
17	38,315	-119,198333
18	24,473333	-4,13666667
19	-39,47833	-147,868333
20	-0,186667	-123,823333

Após a obtenção destes dados, foi necessário (para efeitos de simulação) criar uma posição aproximada de cada uma destas posições geográficas. Este valor obtido representa, na teoria, uma posição estimada onde o observador se possa encontrar. Será a partir deste valor que serão aplicados todos os cálculos das retas de posição, e que no final serão comparados com as posições da **Tabela 2**.

As posições estimadas representadas na **Tabela 3** foram obtidas através do seguinte modo:

- Para obtenção das posições desde a posição 1 até à 15, foram calculadas com base numa distância de 10 milhas náuticas em relação à posição original

(**Tabela 2**). Este valor foi obtido através da aplicação de um vetor de deslocação, com um rumo aleatório, a um navio fictício a navegar a 10 nós e durante 60 minutos. A ferramenta utilizada para este cálculo, foi o calculador de estima presente no ASTROTH2020.

- Para obtenção das posições desde a posição 16 até à 20, o critério foi exatamente o mesmo, mas neste caso a velocidade aplicada no navio foi de 20 nós. Desta forma estas posições encontram-se 20 milhas afastadas da original.

**Tabela 3: Posições Estimadas**

Posição Estimada	Latitude	Longitude
1	-2,916666667	-132,43333333
2	1,866666667	-173,88333333
3	76,66666667	-159,4
4	-35,133333333	-141,36666667
5	-33,06666667	156,5
6	60,833333333	-20,1
7	-44,833333333	-148,38333333
8	43,86666667	143,8666667
9	-57	-138,4166667
10	0,033333333	-4,016666667
11	3,55	123,5666667
12	4,166666667	-79,833333333
13	54,733333333	-100,1166667
14	13,71666667	19,583333333
15	-20,733333333	-109,75
16	-27,55	-34,61666667
17	38,633333333	-119,05
18	24,36666667	-4,483333333
19	-39,76666667	-147,65
20	-0,083333333	-123,5

Segue-se a necessidade de escolher uma hora e dia de cálculo para cada uma das posições. Para a hora de cálculo adotou-se sempre as 12 horas GMT+00. Desta forma apenas se variou a data de cada uma das posições, como representado na **Tabela 4**.

**Tabela 4: Data para cada Posição**

<b>POSIÇÃO</b>	<b>DATA</b>
1	10/01/2020
2	20/01/2020
3	10/02/2020
4	20/02/2020
5	10/03/2020
6	20/03/2020
7	15/04/2020
8	15/05/2020
9	15/06/2020
10	20/06/2020
11	10/07/2020
12	20/07/2020
13	10/08/2020
14	20/08/2020
15	10/09/2020
16	20/09/2020
17	10/10/2020
18	20/10/2020
19	15/11/2020
20	20/12/2020

Tentou-se assim, como representado na **Tabela 4**, percorrer os diversos meses do ano, pela necessidade de procurar astros que apenas são visíveis em determinadas alturas do ano (alguns planetas) para certas posições.

#### 4.1.2. Dados dependentes de programas

Após o registo das posições geográficas de coordenadas conhecidas, procurou-se usar as mesmas de modo a obter o panorama celeste dessas localizações.

O objetivo principal será procurar um programa informático que exemplifique, o mais realista quanto possível, a disposição dos astros no céu numa determinada posição.

Assim, reunindo os dados de uma certa posição exata e usando um programa com elevado rigor de cálculo e de representação virtual, foi possível escolher os astros que melhores características forneceram para o presente estudo.

O programa utilizado para este estudo foi o **Stellarium 0.20.2**. Este programa, além do rigor de cálculo que possui, permite que seja possível a observação quase realista do céu envolvente para uma determinada posição (**figura 34**).



**Figura 34:** Panorama Celeste do Stellarium 0.20.2

Na **Figura 35** pode-se observar o campo destinado neste programa à apresentação da altura aparente. É de realçar também a necessidade de possuir a opção “Atmosfera [A]” selecionada, visto ser desta forma que o programa ignorará as correções necessárias nas alturas. Por sua vez, todos os valores de altura apresentados para os astros selecionado serão tão rigorosos quanto uma situação real de localização por parte do observador.

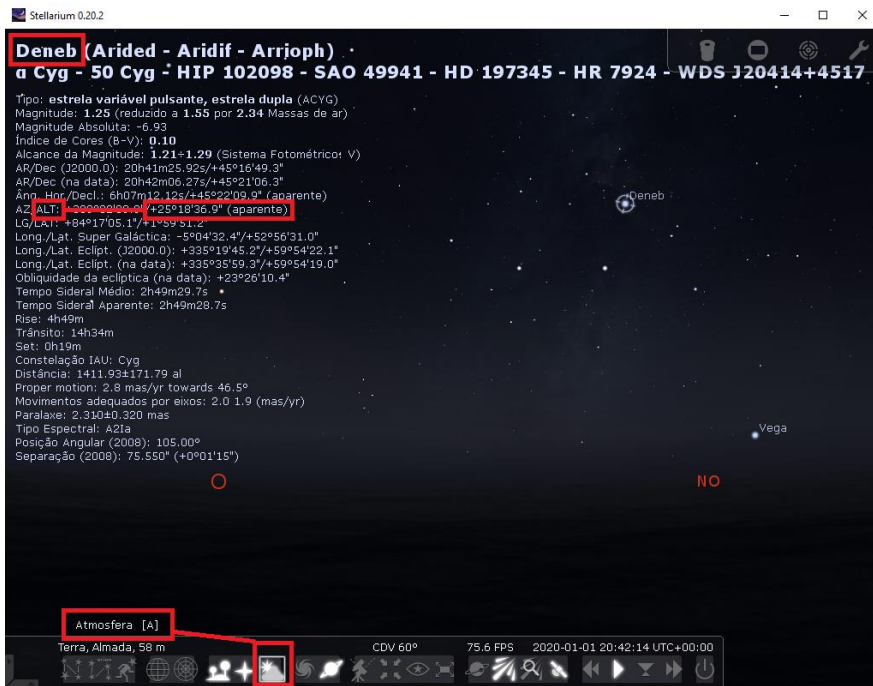


Figura 35: Representação da altura aparente da estrela Deneb (Stellarium 0.20.2, 2020).

Deste programa foram retirados os valores de altura aparente para diversos astros, onde os mesmos foram escolhidos pela interpretação direta da sua direção (azimute). Também foi necessária a colocação do fator de elevação do observador, que se adotou ser de 5 metros (Figura 36).

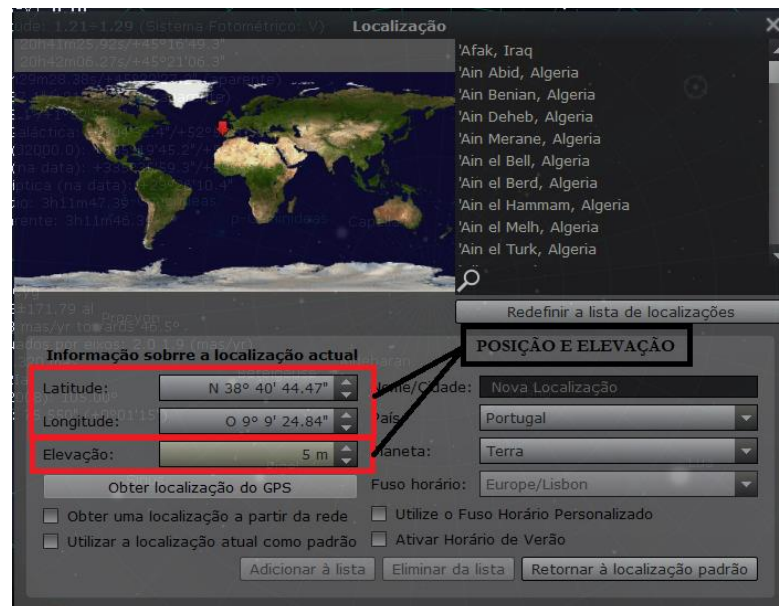
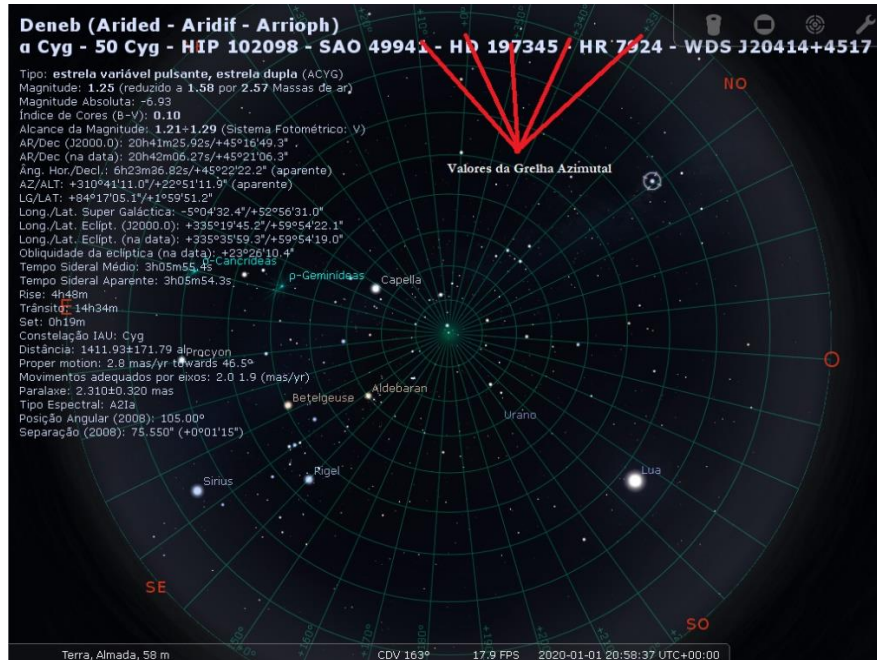


Figura 36: Posição e Elevação do observador (Stellarium 0.20.2, 2020).

O *Stellarium* proporciona ainda, diversas ferramentas que auxiliam na escolha dos astros a utilizar. Na **Figura 37** representa-se a opção “Grelha Azimutal” selecionada, bem como uma observação de todo o céu em formato circular. Estas foram algumas das ferramentas mais importantes utilizadas para escolher os astros observados.



**Figura 37:** Representação da Grelha Azimutal e "Zoom out" (Stellarium 0.20.2, 2020).

Assim, os valores de altura proporcionaram um cruzamento de retas de posição bastante eficiente, o que por sua vez veio a permitir o cálculo de posições com os mesmos dados de entrada em todos os programas estudados.

Na **Tabela 5**, pode-se observar os astros escolhidos e as respetivas alturas aparentes para as 20 posições geográficas.

Tabela 5: Astros e alturas aparentes correspondentes

POSIÇÃO	DIA / HORAS	NOME DO ASTRO	ALTURAS Stellarium	
			GRAUS (°)	MINUTOS E DÉCIMAS (')
1	10/01/2020 12:00:00 GMT+00	Spica	45	2,3
		Hadar	19	28,8
		Canopus	19	43,9
		Sirius	33	55,7
2	20/01/2020 12:00:00 GMT+00	Capella	30	34,9
		Rigel	42	29,1
		Suhail	43	16,9
		Denebola	37	8,4
3	10/02/2020 12:00:00 GMT+00	Lua (limbo inferior)	25	3,5
		Elnath	30	33,8
		Schedar	44	43,6
		Eltanin	45	33,8
4	20/02/2020 12:00:00 GMT+00	Canopus	25	45,6
		Júpiter	4	40,1
		Alphard	40	7,8
		Arcturus	30	47,1
5	10/03/2020 12:00:00 GMT+00	Regulus	44	28,1
		Sirius	47	45,5
		Mekent	35	59,3
		Miaplacidus	53	9,2
6	20/03/2020 12:00:00 GMT+00	Dubhe	32	34,3
		Deneb	67	41,9
		Hamal	37	45,8
7	15/04/2020 12:00:00 GMT+00	Peacock	45	51,7
		Acrux	57	7,6
		Alphecca	18	27,1
8	15/05/2020 12:00:00 GMT+00	Arcturus	61	25,2
		Regulus	39	51,6
		Capella	15	30,5
9	15/06/2020 12:00:00 GMT+00	Júpiter	53	47
		Marte	26	34,6
		Miaplacidus	37	4,9
10	20/06/2020	Mirfak	32	33,1
		Regulus	22	24

POSIÇÃO	DIA / HORAS	NOME DO ASTRO	ALTURAS Stellarium	
			GRAUS (°)	MINUTOS E DÉCIMAS (')
	12:00:00 GMT+00	Acamar	35	24,4
11	10/07/2020 12:00:00 GMT+00	Eltanin	33	5,1
		Altair	24	53,4
		Nunki	32	1,3
		Menkent	45	42,4
		Spica	55	49,9
		Denebola	34	41,2
12	20/07/2020 12:00:00 GMT+00	Shedar	32	49,1
		Capella	35	23,6
		Sirius	24	56
		Acamar	45	20,3
		Diphda	54	45,4
		Markab	37	6,7
13	10/08/2020 12:00:00 GMT+00	Lua	43	28,2
		Capella	63	20,8
		Kochab	38	59,8
		Deneb	36	12,2
14	20/08/2020 12:00:00 GMT+00	Alkaid	42	51,9
		Spica	49	0,9
		Menkent	25	22
		Alphard	55	7,8
15	10/09/2020 12:00:00 GMT+00	Hamal	37	59,1
		Achernar	45	21,6
		Adhara	48	47,6
16	20/09/2020 12:00:00 GMT+00	Regulus	49	49,8
		Suhail	72	58,8
		Gacrux	47	43
		Sirius	48	39,3
17	10/10/2020 12:00:00 GMT+00	Dubhe	34	56
		Regulus	21	25,5
		Betelgeuse	58	6,9
		Hamal	46	36,8
18	20/10/2020 12:00:00 GMT+00	Kochab	39	34,1
		Rosalhague	33	29,7
		Menkent	28	45,6
		Denebola	62	4,2

POSIÇÃO	DIA / HORAS	NOME DO ASTRO	ALTURAS Stellarium	
			GRAUS (°)	MINUTOS E DÉCIMAS (')
19	15/11/2020 12:00:00 GMT+00	Marte	9	49,1
		Achernar	46	39,8
		Menkar	32	9,6
		Procyon	38	28,4
20	20/12/2020 12:00:00 GMT+00	Alioth	21	55,5
		Menkent	19	2,8
		Suhail	46	1,4
		Elnath	22	44,3

Pela análise da **Tabela 5**, pode notar-se que existem posições com diferentes números de observações. Este pormenor foi escolhido com o objetivo futuro de averiguar a capacidade de cálculo do ponto astronómico para os diferentes programas com diferentes números de observações.

No seguimento da obtenção de dados importantes ao cálculo do  $\Delta a$  e do azimute dos diversos astros observados, tornou-se necessário a obtenção de valores de GHA, declinação, semidiâmetro e paralaxe dos quais o ASTROTH2020 é dependente.

Para a obtenção destes valores considerou-se os valores oferecidos pelo programa NAVPAC. É de realçar que os valores apresentados pelo NAVPAC e NAV32 têm variações, em alguns casos chegam a atingir 0.5' para as variáveis referidas no paragrafo anterior.

Assim, escolheu-se o NAVPAC para o registo desses dados no ASTROTH2020 e iniciou-se o cálculo das variáveis das retas de altura nos três programas informáticos.

Nas seguintes figuras (**Figura 38, 39, 40**) pretende-se exemplificar, utilizando todos os dados de entrada da primeira posição, os resultados do  $\Delta a$  e azimute obtidos para cada um dos programas informáticos utilizados.

Line of Position		Astronomical Position		Visible Sky	Other Calculations		
#	Lon	Lat	Delta	Body	Time	Hi	Az
1	132°26.0'W	2°55.0'S	-9.0	Spica	12:00:00	45°02.3'	103°
2	132°26.0'W	2°55.0'S	-0.8	Hadar	12:00:00	19°28.8'	155°
3	132°26.0'W	2°55.0'S	5.2	Canopus	12:00:00	19°43.9'	214°
4	132°26.0'W	2°55.0'S	4.0	Sirius	12:00:00	33°55.7'	252°

Figura 38: NAV32 -  $\Delta a$ 's e azimutes

Date	UT	No.	Body	Use	p	Z	Rhumb Line Track (WGS84)	
	h m s			kn ?	nm	o	Lat	Long
							o ' o '	o ' o '
2020 Jan 10	12:00:00		Fix				S 02 55.0 W 132 26.0	
2020 Jan 10	12:00:00		Leg 1	000.0 0.0			S 02 55.0 W 132 26.0	
2020 Jan 10	12:00:00	1	Spica	Y	- 9.0	103.1	S 02 55.0 W 132 26.0	
2020 Jan 10	12:00:00	2	Hadar	Y	- 0.9	154.9	S 02 55.0 W 132 26.0	
2020 Jan 10	12:00:00	3	Canopus	Y	+ 5.2	214.2	S 02 55.0 W 132 26.0	
2020 Jan 10	12:00:00	4	Sirius	Y	+ 4.3	251.8	S 02 55.0 W 132 26.0	

Figura 39: NAVPAC -  $\Delta a$ 's e azimutes

Dados das Retas de Posição			
	$\Delta a$	Azimute	Usar Linha
LDP1	-8.9898	103.0590	<input checked="" type="checkbox"/>
LDP2	-0.9373	154.9010	<input checked="" type="checkbox"/>
LDP3	5.1487	214.1910	<input checked="" type="checkbox"/>
LDP4	3.9607	251.7710	<input checked="" type="checkbox"/>
LDP5			<input type="checkbox"/>
LDP6			<input type="checkbox"/>

Figura 40: ASTROTH2020 -  $\Delta a$ 's e azimutes

Na continuação lógica deste processo de cálculo, segue-se a obtenção da posição final com base nos  $\Delta a$ 's e azimutes obtidos. Todos os programas, ao executarem o algoritmo de cálculo final, oferecem diferentes resultados de posição final. Na **Tabela 5** apresentam-se os resultados obtidos para cada programa, em cada uma das posições estimadas utilizadas no cálculo final.

**Tabela 6: Posições obtidas para cada programa de estudo**

Posição	NAVPAC		NAV32		ASTROTH2020	
	LAT	LONG	LAT	LONG	LAT	LONG
1	02°56.0' S	132°32.8' W	02°55.8' S	132°32.6' W	02°55.9' S	132°32.7' W
2	02°01.2' N	173°53.4' W	02°00.0' N	173°52.0' W	02°01.3' N	173°53.3' W
3	76°37.0 N	159°59.9' W	76°36.7' N	159°58.0' W	76°37.7' N	160°00.1' W
4	35°00.1' S	141°13.2' W	34°58.5' S	141°10.7' W	34°59.5' S	141°13.0' W
5	33°07.0' S	156°17.3' E	33°05.5' S	156°16.0' E	33°07.0' S	156°17.4' E
6	60°44.1' N	020°23.5' W	60°44.5' N	020°23.7' W	60°44.2' N	020°23.7' W
7	44°54.6' S	148°10.9' W	44°55.5' S	148°10.4' W	44°54.7' S	148°10.8' W
8	44°01.6' N	143°46.7' E	44°04.0' N	143°50.6' E	44°01.8' N	143°46.6' E
9	56°57.6' S	138°12.0' W	56°57.0' S	138°07.8' W	56°57.5' S	138°12.5' W
10	00°10.1' N	003°57.5' W	00°11.1' N	003°57.8' W	00°10.1' N	003°57.5' W
11	03°28.4' N	123°42.2' E	03°29.0' N	123°41.8' E	03°28.4' N	123°42.2' E
12	04°08.7' N	079°39.0' W	04°09.2' N	079°38.1' W	04°08.7' N	079°39.0' W
13	54°52.0' N	100°04.9' W	54°50.8' N	100°02.9' W	54°52.0' N	100°05.1' W
14	13°54.6' N	019°34.7' E	13°53.7' N	019°35.7' E	13°54.5' N	019°34.8' E
15	20°45.2' S	109°55.0' W	20°46.3' S	109°55.5' W	20°45.2' S	109°55.0' W
16	27°37.9' S	034°57.1' W	27°38.1' S	034°56.1' W	27°37.9' S	034°57.2' W
17	38°19.8' N	119°13.8' W	38°21.1' N	119°13.9' W	38°19.9' N	119°13.8' W
18	24°29.1' N	004°09.1' W	24°30.4' N	004°10.7' W	24°29.2' N	004°09.1' W
19	39°31.5' S	147°47.5' W	39°32.0' S	147°46.0' W	39°31.5' S	147°47.6' W
20	00°11.0' N	123°49.9' W	00°09.1' N	123°50.1' W	00°11.0' N	123°49.9' W

A partir da análise da Tabela 6 pode-se notar que cada rotina informática alvo de estudo, oferece diferentes posições. O rigor de cálculo de certos parâmetros intermédios, pode ser um dos fatores consideráveis neste aspeto.

O NAVPAC ao utilizar como algoritmo final de cálculo o mesmo algoritmo usado pelo ASTROTH2020, faz com que o resultado obtido seja muito semelhante. No entanto, o NAV32 oferece alguns valores bastante díspares.

#### 4.2. Comparação dos Resultados Obtidos

Após a obtenção das posições finais por parte das três rotinas informáticas utilizadas, foi possível comparar as mesmas em relação à posição geográfica inicial.

Existindo à partida dificuldade ao observar a posição exata e a posição astronómica obtida e perceber a diferença entre elas, verificou-se a necessidade de recorrer ao cálculo da distância entre as duas. Pode-se assim apresentar, de uma maneira mais direta e intuitiva, a verdadeira diferença entre ambas as posições.

Assim pode-se analisar o rigor de cálculo de cada um dos programas utilizados, bastando verificar quais foram as posições calculadas que mais se aproximaram da correspondente posição exata.

Para este estudo foi necessário recorrer à seguinte fórmula de cálculo, obtida de Mapanet (2019):

$$D = 6378.137 \times \cos^{-1}(\cos(\varphi_1) \times \cos(\varphi_2) \times \cos(\lambda_2 - \lambda_1) + \sin(\varphi_1) \times \sin(\varphi_2))$$

onde,

$D$  = Distância (km)

$\varphi_1$  = Latitude da posição exata

$\lambda_1$  = Longitude da posição exata

$\varphi_2$  = Latitude da posição obtida

$\lambda_2 =$  Longitude da posição exata

6378.137 = Diâmetro da Terra em Km para o elipsoide de referência WGS84

Após obtenção da distância por meio da fórmula anterior foi necessário a conversão do resultado obtido para um valor em milhas náuticas (NM).

Para tal foi considerado o seguinte cálculo:

$$D (nm) = D (Km) \times 0,539957$$

onde,

$$1 \text{ Km} = 0,539957 \text{ NM}$$

Por fim reuniram-se todas as distâncias calculadas em relação a todos os programas utilizados. Os valores das distâncias obtidos variaram bastante, revelando maior rigor os obtidos pelo NAVPAC e ASTROTH2020.

No entanto, existe uma particularidade em relação ao cálculo da posição com base em três observações. O programa NAV32 apresenta resultados mais rigorosos para esta situação. Isto porque, o modelo matemático utilizado no funcionamento do NAV32 não é o método dos mínimos quadrados, o que pode influenciar o cálculo da posição final com base em 3 observações.

Assim o resultado obtido para 3 observações, pelo NAV32, tem de facto maior rigor no cálculo da posição final.

Na **Tabela 7** apresentam-se os resultados, em milhas náuticas, relativos às distâncias obtidas para cada posição calculada. Os valores mais próximos da posição exata encontram-se simbolizados a verde, a amarelo os intermédios e a vermelho os mais distantes.

**Tabela 7: Distâncias finais representadas em milhas náuticas**

Posições	NAVPAC	NAV32	ASTROTH2020
1	5,81	6,01	5,91

Posições	NAVPAC	NAV32	ASTROTH2020
2	0,92	2,51	1,05
3	3,38	3,85	2,75
4	0,6	2,97	0,84
5	0,34	2,02	0,39
6	0,53	0,31	0,49
7	0,91	0,21	0,8
8	4,39	8,09	4,46
9	2,4	0,11	2,65
10	1,02	0,1	1,02
11	1,2	1,84	1,21
12	0,62	1,55	0,6
13	1	2,46	1,01
14	3,42	2,09	3,28
15	1,29	0,1	1,29
16	1,99	2,63	1,93
17	1,74	2,7	1,79
18	1,07	3,03	1,14
19	4,52	5,75	4,46
20	0,53	2,21	0,52



Valor mais correto



Valor intermédio



Valor menos correto



## Conclusão

Atualmente na Marinha Portuguesa a prática da navegação astronómica, bem como o seu treino e aprendizagem, recai bastante na utilização de métodos tabulares e de consulta. Com o avanço tecnológico esta prática começa a cair em desuso, sendo que no presente trabalho propôs-se uma formulação e estruturação de uma componente gráfica e interativa que visa melhorar o aspeto menos prático descrito anteriormente.

A existência de uma ferramenta computacional ao dispor da navegação é uma mais valia para a navegação astronómica, podendo assim serem dispensáveis certas publicações que noutro caso se utilizariam.

Assim, o objetivo do estudo descrito na presente dissertação de mestrado foi a criação de um programa computacional, com elevado grau de rigor de cálculo, ao qual se designou ASTROTH2020.

Foi também possível efetuar uma comparação de dados com outros programas de cálculo, o que por si só se considera uma vantagem enorme, pois foi desta maneira que se conseguiu apurar até que ponto a construção da ferramenta ASTROTH2020 foi bem-sucedida.

Numa primeira análise dos resultados obtidos por comparação com o programa NAVPAC, pode-se concluir que o rigor apresentado pelo NAVPAC é muito superior ao dos outros programas estudados. Tal característica dever-se-á ao facto deste programa ter na sua constituição algorítmica a possibilidade de efetuar iterações de cálculo, algo que não é possível nos outros programas utilizados.

No entanto, dada a impossibilidade de automatizar a recolha de dados, a quantidade dos mesmos tornou-se bastante reduzida. Futuramente, deve ser considerada a aplicação de um estudo com maior amostra de resultados, para que desta forma se desmistifique, com maior certeza, o rigor da aplicação desenvolvida em comparação com outras já existentes.

Como referido anteriormente no subcapítulo 4.2, o NAV32 ao não possuir o mesmo método de cálculo que os outros dois programas, conseguirá obter melhores resultados para o cálculo de uma posição com base em três observações, quando em comparação com o NAVPAC e ASTROTH2020. Tal condição, leva à discussão de ser ou não possível incluir futuramente no ASTROTH2020 a opção de cálculo que o NAV32 usa, apenas para quando

o utilizador considerar três observações para uma dada posição. Deste modo, o ASTROTH2020, passaria a ser uma ferramenta ainda mais vantajosa e exata para o uso durante a navegação astronómica.

Após a conclusão do programa considerou-se que, para uma ação mais expedita de análise de dados, existe uma necessidade futura de apresentar ao utilizador a representação gráfica do ponto obtido juntamente com uma elipse que definisse o erro associado a essa posição, como é apresentado pelo NAVPAC.

Concluiu-se ainda que o ASTROTH2020 efetua o cálculo das posições astronómicas com um rigor intermédio quando comparado com os restantes programas referenciados, pois apresenta valores de distância à posição exata que se encontram compreendidos entre os obtidos pelo NAVPAC e NAV32, para a maioria dos casos observados.

Por fim, seria importante numa atualização do ASTROTH2020 incrementar critérios de cálculo das iterações finais, e realizar uma melhoria a nível de interatividade com a possibilidade de ligação com um outro programa desenvolvido, cujo objetivo fosse a obtenção dos dados que são necessários ao cálculo da posição final.

## Bibliografia

- British Admiralty (2015). *Nautical Almanac 2016*. United Kingdom: UK Hydrographic Office.
- Curbera, M. (1982). *Astronomia Náutica Y Navegacion*. Vigo: Hijos de E. Minuesa, S. L., ISBN: 84 400 6516 7
- Escola Naval & Diretoria de Portos e Costas (1978). *Navegação Astronómica*. (3º edição) Brasil: Diretoria de Portos e Costas.
- Escola Naval (1995). *Navegação Astronómica*. Lisboa: Serviço de Publicações Escolares.
- Fortin, M. (1999). *O Processo de Investigação: da concepção à realização*. (2º edição). Brasil: Décarie Editeur
- Gameiro, E. S. (1964). *Astronomia Náutica*. Lisboa: Oficinas Gráficas da Papelaria Fernandes
- Hohenkerk, C. Y. & Yallop, B. D. (2006). *NavPac and Compact Data*. Londres: TSO. ISBN: 0 11 887331 8
- Mapanet (2019). *Distância entre dois pontos*. Disponível em: <https://www.mapanet.eu/PT/resources/Script-Distance.htm>
- Mederos, L. (s.d.). *Introducción a la Navegación Astronómica*. Disponível para consulta em: <https://docplayer.es/25578366-Como-utilizar-este-documento.html>
- Ministério da Marinha & Instituto Hidrográfico. (1973). *Tábuas Náuticas*. (1º Edição). Lisboa: Instituto Hidrográfico.
- MOHPC: The Museum of HP Calculators. (2020). *(35S) Most probable position of a vessel*. Disponível em: <https://www.hpmuseum.org/forum/thread-8241.html>
- Morais, V. & Vieira, C. (2013). *MATLAB: Curso completo*. (1º edição). Lisboa: FCA-Editora de Informática, Lda.
- National Imagery And Mapping Agency. (2001) *Sight Reduction Tables For Marina Navigation: Latitudes 0° - 15°, Inclusive*. Pub. No. 229. Vol. 1. Celestair, INC.
- Peterson, A. (2016). Why Naval Academy students are learning to sail by the stars for the first time in a decade. *The Washington Post*. Disponível em: <https://www.washingtonpost.com/news/the-switch/wp/2016/02/17/why-naval-academy-students-are-learning-to-sail-by-the-stars-for-the-first-time-in-a-decade/>

Van Der Griten, H. F. (1975). The Accuracy of Celestial Fixes as Compared to NAVSAT Positions by Least Squares Adjustment. *Navigation: Journal of The Institute of Navigation*. Vol. 22. No. 2. Pp.128-134.

Weisstein, E., W. (2003). *Crc Concise Encyclopedia of Mathematics*. (2º Edição). Estados Unidos da América: CRC Press Company

Zotti, G. & Wolf, A. (2020) *Stellarium 0.20.2 User Guide*. Disponível em: <https://stellarium.org/>

## Apêndices

### Apêndice A – Código do Programa

```

function varargout = TESTEGUI(varargin)
% TESTEGUI MATLAB code for TESTEGUI.fig
% TESTEGUI, by itself, creates a new TESTEGUI or raises the existing
% singleton*.
%
% H = TESTEGUI returns the handle to a new TESTEGUI or the handle to
% the existing singleton*.
%
% TESTEGUI('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in TESTEGUI.M with the given input arguments.
%
% TESTEGUI('Property','Value',...) creates a new TESTEGUI or raises the
%existing singleton*. Starting from the left, property value pairs are
%applied to the GUI before TESTEGUI_OpeningFcn gets called. An
%unrecognized property name or invalid value makes property application
%stop. All inputs are passed to TESTEGUI_OpeningFcn via varargin.
%
%*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help TESTEGUI
% Last Modified by GUIDE v2.5 02-Jun-2020 23:03:21
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @TESTEGUI_OpeningFcn, ...
'gui_OutputFcn',  @TESTEGUI_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);
if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before TESTEGUI is made visible.
function TESTEGUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% varargin    command line arguments to TESTEGUI (see VARARGIN)
% Choose default command line output for TESTEGUI
dados=get(handles.tabela,'Data');
dados(:,:)=[];
set(handles.tabela,'Data',dados);
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes TESTEGUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = TESTEGUI_OutputFcn(hObject, eventdata, handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function LAT_Callback(hObject, eventdata, handles)
% hObject     handle to LAT (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of LAT as text
% str2double(get(hObject,'String')) returns contents of LAT as a double

% --- Executes during object creation, after setting all properties.
function LAT_CreateFcn(hObject, eventdata, handles)
% hObject     handle to LAT (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function LAT1_Callback(hObject, eventdata, handles)
% hObject     handle to LAT1 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of LAT1 as text

```

```

%      str2double(get(hObject,'String')) returns contents of LAT1 as a
double

% --- Executes during object creation, after setting all properties.
function LAT1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to LAT1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function LONG_Callback(hObject, eventdata, handles)
% hObject    handle to LONG (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of LONG as text
%      str2double(get(hObject,'String')) returns contents of LONG as a
double

% --- Executes during object creation, after setting all properties.
function LONG_CreateFcn(hObject, eventdata, handles)
% hObject    handle to LONG (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function LONG1_Callback(hObject, eventdata, handles)
% hObject    handle to LONG1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of LONG1 as text
%      str2double(get(hObject,'String')) returns contents of LONG1 as a
double

% --- Executes during object creation, after setting all properties.
function LONG1_CreateFcn(hObject, eventdata, handles)

```

```

% hObject    handle to LONG1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function PA_Callback(hObject, eventdata, handles)
% hObject    handle to PA (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of PA as text
%         str2double(get(hObject,'String')) returns contents of PA as a double

% --- Executes during object creation, after setting all properties.
function PA_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PA (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function TEMP_Callback(hObject, eventdata, handles)
% hObject    handle to TEMP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TEMP as text
%         str2double(get(hObject,'String')) returns contents of TEMP as a
double

% --- Executes during object creation, after setting all properties.
function TEMP_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TEMP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function AOBS11_Callback(hObject, eventdata, handles)
% hObject    handle to AOBS11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of AOBS11 as text
%        str2double(get(hObject,'String')) returns contents of AOBS11 as a
double

% --- Executes during object creation, after setting all properties.
function AOBS11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to AOBS11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function AOBS12_Callback(hObject, eventdata, handles)
% hObject    handle to AOBS12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of AOBS12 as text
%        str2double(get(hObject,'String')) returns contents of AOBS12 as a
double

% --- Executes during object creation, after setting all properties.
function AOBS12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to AOBS12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

```

```

function Dec1_Callback(hObject, eventdata, handles)
% hObject      handle to Dec1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Dec1 as text
%         str2double(get(hObject,'String')) returns contents of Dec1 as a
double

% --- Executes during object creation, after setting all properties.
function Dec1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Dec1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function Dec11_Callback(hObject, eventdata, handles)
% hObject      handle to Dec11 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Dec11 as text
%         str2double(get(hObject,'String')) returns contents of Dec11 as a
double

% --- Executes during object creation, after setting all properties.
function Dec11_CreateFcn(hObject, eventdata, handles)
% hObject      handle to Dec11 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function GHAl_Callback(hObject, eventdata, handles)
% hObject      handle to GHAl (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of GHAl as text
%         str2double(get(hObject,'String')) returns contents of GHAl as a
double

% --- Executes during object creation, after setting all properties.
function GHAl_CreateFcn(hObject, eventdata, handles)
% hObject      handle to GHAl (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function GHAl1_Callback(hObject, eventdata, handles)
% hObject      handle to GHAl1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of GHAl1 as text
%         str2double(get(hObject,'String')) returns contents of GHAl1 as a
double

% --- Executes during object creation, after setting all properties.
function GHAl1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to GHAl1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function DIP_Callback(hObject, eventdata, handles)
% hObject      handle to DIP (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of DIP as text
%         str2double(get(hObject,'String')) returns contents of DIP as a double

```

```

% --- Executes during object creation, after setting all properties.
function DIP_CreateFcn(hObject, eventdata, handles)
% hObject    handle to DIP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function EI_Callback(hObject, eventdata, handles)
% hObject    handle to EI (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of EI as text
%        str2double(get(hObject,'String')) returns contents of EI as a double

% --- Executes during object creation, after setting all properties.
function EI_CreateFcn(hObject, eventdata, handles)
% hObject    handle to EI (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%TEMPERATURA
TEMP=str2double(get(handles.TEMP,'string'));
%PRESSÃO ATMOFERICA
PA=str2double(get(handles.PA,'string'));
%ERRO INDICE
EI=str2double(get(handles.EI,'string'));
EIf=EI/60;
%DIP
DIP=str2double(get(handles.DIP,'string'));
DIPc=1.76*sqrt(DIP);
DIPcf=DIPc/60;
%LATITUDE ESTIMADA

```

```

latcorr=(upper(get(handles.latcorr,'String')));

if latcorr=='N'
latcorr1=1;
elseif latcorr=='S'
latcorr1=(-1);
end

LAT=(str2double(get(handles.LAT,'string')))*latcorr1;
LAT1=(str2double(get(handles.LAT1,'string')))*latcorr1;
LATfg=(LAT+LAT1/60);

LATfm=( (LAT*60)+LAT1);

%LONGITUDE ESTIMADA
longcorr=(upper(get(handles.longcorr,'String')));

if longcorr=='E'
longcorr1=1;
elseif longcorr=='W'
longcorr1=(-1);
end

LONG=(str2double(get(handles.LONG,'string')))*longcorr1;
LONG1=(str2double(get(handles.LONG1,'string')))*longcorr1;
LONGfg=LONG+LONG1/60;
LONGfm=(LONG*60)+LONG1;

%ALTURA OBSERVADA 1
AOBS11=str2double(get(handles.AOBS11,'string'));
AOBS12=str2double(get(handles.AOBS12,'string'));
AOBS13=AOBS11+AOBS12/60;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PRIMEIRA
ALTURA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%ERRO DE INDICE
AOBS13f1=AOBS13-EIf;
%ERRO DIP
AOBS13f2=AOBS13f1-DIPcf;
%ERRO REFRAÇÃO
RE1=((PA/1010)*(283/(273+TEMP)))*((1/(tan(degtorad(AOBS13f2+(7.31/(AOBS13f2+4.4))))))-
(0.06*sin((1/(tan(degtorad(AOBS13f2+(7.31/(AOBS13f2+4.4))))))+13)))));
AOBS13f3=AOBS13f2-(RE1/60);
%1ª RETA
Deccorr=(upper(get(handles.Deccorr,'String')));

if Deccorr=='N'
Deccorr1=1;
elseif Deccorr=='S'
Deccorr1=(-1);
end

Dec1=(str2double(get(handles.Dec1,'string')))*Deccorr1;
Dec11=(str2double(get(handles.Dec11,'string')))*Deccorr1;
Dec1f=Dec1+Dec11/60;

```

```

%ANGULO NO POLO ASTRO 1
GHA1=str2double(get(handles.GHA1,'string'));
GHA11=str2double(get(handles.GHA11,'string'));
GHA1fg=GHA1+GHA11/60;

%LHA -> GRAUS
LHA=GHA1fg+LONGfg;
LHA1=wrapTo360(LHA);
%PRIMEIRO ASTRO

hc=asind(sind(LATfg)*sind(Declf)+cosd(LATfg)*cosd(Declf)*cosd(LHA1));

alturalf=hc;

%PASSAGEM DAS ALTURAS PARA GRAUS E MINUTOS
alturalfg=fix(alturalf);

alturalfm=((alturalf-fix(alturalf))*60);
%AZIMUTES
Z = acosd((sind(Declf) - sind(LATfg) * sind(alturalf)) / (cosd(LATfg) *
cosd(alturalf)));
if LHA1>180
Zn=Z;
else
Zn=360-Z;
end
%1° ASTRO

%%%%%ERRO DE INDICE
AOBS13f1=AOBS13-EIf;
%%%%%ERRO DIP
AOBS13f2=AOBS13f1-DIPcf;
%%%%%ERRO REFRAÇÃO
RE=((PA/1010)*(283/(273+TEMP)))*(1/(tan(degtorad(AOBS13f2+(7.31/(AOBS13f2+4.4))))))-(0.06*sin(1/(tan(degtorad(AOBS13f2+(7.31/(AOBS13f2+4.4))))))+13));
AOBS13f3=AOBS13f2-(RE/60);

%%%%%LUA/SOL
items=get(handles.popupmenu1,'String');
a=get(handles.popupmenu1,'Value');
b=get(handles.popupmenu2,'Value');
set(handles.popupmenu1,'String',items);

switch b
case 1
limbo=0;
case 2
limbo=1;
case 3
limbo=-1;
end

SD=(str2double(get(handles.SD,'String')))*limbo;
HPL=str2double(get(handles.HPL,'String'));

```

```

switch a
case 1
AOBS13f5=VAMOSTENTARFUNCAOSOL (TEMP, PA, DIP, EI, AOBS11, AOBS12, SD) ;
case 2
AOBS13f5=VAMOSTENTARFUNCAOLUA (LATfg, Zn, TEMP, PA, DIP, EI, AOBS11, AOBS12, HPL, limbo)
;
case 3
AOBS13f5=AOBS13f3;
case 4
AOBS13f5=VAMOSTENTARFUNCAOVENUS (TEMP, PA, DIP, EI, AOBS11, AOBS12, HPL) ;
case 5
AOBS13f5=VAMOSTENTARFUNCAOMARTE (TEMP, PA, DIP, EI, AOBS11, AOBS12, HPL) ;
end

%PASSAGEM DAS ALTURAS PARA GRAUS E MINUTOS
AOBS13f5g=fix(AOBS13f5);

AOBS13f5m=((AOBS13f5-fix(AOBS13f5))*60);
%ALTURAS OBSERVADA - ALTURAS CALCULADA

deltaA1=AOBS13f5-alturalf;
deltaA1f=deltaA1*60;

%DISPLAY DAS ALTURAS CALCULADAS
set(handles.text42, 'string', alturalfg)
set(handles.text44, 'string', alturalfm)
set(handles.text158, 'String', round((alturalfm),1))
%DISPLAY DAS ALTURAS CORRIGIDAS
set(handles.text46, 'string', AOBS13f5g)
set(handles.text48, 'string', AOBS13f5m)
set(handles.text159, 'String', round((AOBS13f5m),1))
%DISPLAY AZIMUTES
set(handles.text70, 'String', Zn)

if round(Zn)<=100 && round(Zn)>=10
Znf=strcat('0', num2str(round(Zn)));
elseif round(Zn)<10
Znf=strcat('00', num2str(round(Zn)));
elseif round(Zn)>100
Znf=round(Zn);
end

set(handles.text157, 'String', Znf)

%DISPLAY DELTA.A's
set(handles.text76, 'String', deltaA1f)
set(handles.text161, 'String', round((deltaA1f),1))
function longcorr_Callback(hObject, eventdata, handles)
% hObject    handle to longcorr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of longcorr as text

```

```

%         str2double(get(hObject,'String')) returns contents of longcorr as a
double

% --- Executes during object creation, after setting all properties.
function longcorr_CreateFcn(hObject, eventdata, handles)
% hObject    handle to longcorr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function latcorr_Callback(hObject, eventdata, handles)
% hObject    handle to latcorr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of latcorr as text
%         str2double(get(hObject,'String')) returns contents of latcorr as a
double

% --- Executes during object creation, after setting all properties.
function latcorr_CreateFcn(hObject, eventdata, handles)
% hObject    handle to latcorr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function Deccorr_Callback(hObject, eventdata, handles)
% hObject    handle to Deccorr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Deccorr as text
%         str2double(get(hObject,'String')) returns contents of Deccorr as a
double

% --- Executes during object creation, after setting all properties.

```

```

function Deccorr_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Deccorr (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents
as cell array
%         contents{get(hObject,'Value')} returns selected item from popupmenu1

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function SD_Callback(hObject, eventdata, handles)
% hObject    handle to SD (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of SD as text
%         str2double(get(hObject,'String')) returns contents of SD as a double

% --- Executes during object creation, after setting all properties.
function SD_CreateFcn(hObject, eventdata, handles)
% hObject    handle to SD (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

```

```

%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function HPL_Callback(hObject, eventdata, handles)
% hObject    handle to HPL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of HPL as text
%       str2double(get(hObject,'String')) returns contents of HPL as a double

% --- Executes during object creation, after setting all properties.
function HPL_CreateFcn(hObject, eventdata, handles)
% hObject    handle to HPL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
D=str2double(get(handles.text76,'String'));
A=str2double(get(handles.text70,'String'));

dados=get(handles.tabela,'Data');

if isempty(dados)
dados{1,1}=D;
dados{1,2}=A;
dados{1,3}=true;
else
if size(dados) < 6
dados{end+1,1}=D;
dados{end,2}=A;
dados{end,3}=true;
else
limite = 6;
for indice = 1:limite
if dados{indice,3} == false
if isempty(dados{indice,1})

```

```

dados{indice,1}=D;
dados{indice,2}=A;
dados{indice,3}=true;
break;
end
end
end
end
end

set(handles.tabela,'Data',dados)

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%LATITUDE ESTIMADA
latcorr=(upper(get(handles.latcorr,'String')));

if latcorr=='N'
latcorr1=1;
elseif latcorr=='S'
latcorr1=(-1);
end

LAT=(str2double(get(handles.LAT,'string')))*latcorr1;
LAT1=(str2double(get(handles.LAT1,'string')))*latcorr1;
LATfg=(LAT+LAT1/60);

%LONGITUDE ESTIMADA
longcorr=(upper(get(handles.longcorr,'String')));

if longcorr=='E'
longcorr1=1;
elseif longcorr=='W'
longcorr1=(-1);
end

LONG=(str2double(get(handles.LONG,'string')))*longcorr1;
LONG1=(str2double(get(handles.LONG1,'string')))*longcorr1;
LONGfg=LONG+LONG1/60;
dados=get(handles.tabela,'Data');
%A1=(cos(degtorad(az1))*cos(degtorad(az1)))+(cos(degtorad(az2))*cos(degtorad(az2)))+(cos(degtorad(az3))*cos(degtorad(az3)))+(cos(degtorad(az4))*cos(degtorad(az4)))+(cos(degtorad(az5))*cos(degtorad(az5)))+(cos(degtorad(az6))*cos(degtorad(az6)));
%B1=(cos(degtorad(az1))*sin(degtorad(az1)))+(cos(degtorad(az2))*sin(degtorad(az2)))+(cos(degtorad(az3))*sin(degtorad(az3)))+(cos(degtorad(az4))*sin(degtorad(az4)))+(cos(degtorad(az5))*sin(degtorad(az5)))+(cos(degtorad(az6))*sin(degtorad(az6)));
%C1=(sin(degtorad(az1))*sin(degtorad(az1)))+(sin(degtorad(az2))*sin(degtorad(az2)))+(sin(degtorad(az3))*sin(degtorad(az3)))+(sin(degtorad(az4))*sin(degtorad(az4)))+(sin(degtorad(az5))*sin(degtorad(az5)))+(sin(degtorad(az6))*sin(degtorad(az6)));

```

```

    (az4)))+(sin(degtorad(az5))*sin(degtorad(az5)))+(sin(degtorad(az6))*sin(degtor
    ad(az6)));
    %D1=(da1*cos(degtorad(az1)))+(da2*cos(degtorad(az2)))+(da3*cos(degtorad(az3))
    +(da4*cos(degtorad(az4)))+(da5*cos(degtorad(az5)))+(da6*cos(degtorad(az6)));
    %E1=(da1*sin(degtorad(az1)))+(da2*sin(degtorad(az2)))+(da3*sin(degtorad(az3))
    +(da4*sin(degtorad(az4)))+(da5*sin(degtorad(az5)))+(da6*sin(degtorad(az6)));
    %G1=(A1*C1)-(B1*B1);

    limite = size(dados);
    A1=0;
    B1=0;
    C1=0;
    D1=0;
    E1=0;
    for indice = 1:limite
    if dados{indice,3} == true
    azim = dados{indice,2};
    deltaA = dados{indice,1};
    if ~isempty(deltaA)
    A1 = A1 + calculaA(azim);
    B1 = B1 + calculaB(azim);
    C1 = C1 + calculaC(azim);
    D1 = D1 + calculaD(azim,deltaA/60);
    E1 = E1 + calculaE(azim,deltaA/60);
    end;
    end
    end

    G1=(A1*C1)- round((B1.^2),10);
    LI=(( (A1*E1) - (B1*D1) ) / (G1*cos(degtorad(LATfg))));

    BI=(( (C1*D1) - (B1*E1) ) / G1);

    LIif=LONGfg+LI;

    BIf=LATfg+BI;

    %%DADOS DA POSIÇÃO CALCULADA
    set(handles.text90, 'String', fix(abs(BIf)))
    set(handles.text92, 'String', abs((BIf-fix(BIf))*60))
    set(handles.text162, 'String', round((abs((BIf-fix(BIf))*60)),1))

    if BIf<=0
    slat='S';
    elseif BIf>0
    slat='N';
    end

    set(handles.text105, 'String', slat)

    set(handles.text101, 'String', fix(abs(LIif)))

```

```

set(handles.text103,'String',abs((LIf-fix(LIf))*60))
set(handles.text163,'String',round((abs((LIf-fix(LIf))*60)),1))

if LIf<=0
slon='W';
elseif LIf>0
slon='E';
end
set(handles.text107,'String',slon)

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu2 contents
as cell array
%         contents{get(hObject,'Value')} returns selected item from popupmenu2

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes when selected cell(s) is changed in tabela.
function tabela_CellSelectionCallback(hObject, eventdata, handles)
% hObject    handle to tabela (see GCBO)
% eventdata  structure with the following fields (see MATLAB.UI.CONTROL.TABLE)
%         Indices: row and column indices of the cell(s) currently selecteds
% handles    structure with handles and user data (see GUIDATA)

function valorA = calculaA(az)
valorA = cos(degtorad(az))*cos(degtorad(az));

function valorB = calculaB(az)
valorB = cos(degtorad(az))*sin(degtorad(az));

function valorC = calculaC(az)
valorC = sin(degtorad(az))*sin(degtorad(az));

function valorD = calculaD(az, da)

```

```

valorD = da*cos(degtorad(az));

function valorE = calculaE(az, da)
valorE = da*sin(degtorad(az));

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu3 contents
as cell array
%         contents{get(hObject,'Value')} returns selected item from popupmenu3

% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
apagarlinha = get(handles.popupmenu3,'Value');
dados=get(handles.tabela,'Data');
if apagarlinha>1
dados{apagarlinha-1,1}='';
dados{apagarlinha-1,2}='';
dados{apagarlinha-1,3}=false;
set(handles.tabela,'Data',dados);
end

function edit38_Callback(hObject, eventdata, handles)
% hObject    handle to edit38 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit38 as text
%         str2double(get(hObject,'String')) returns contents of edit38 as a
double

```

```

% --- Executes during object creation, after setting all properties.
function edit38_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit38 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit39_Callback(hObject, ~, handles)
% hObject    handle to edit39 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit39 as text
%         str2double(get(hObject,'String')) returns contents of edit39 as a
double

% --- Executes during object creation, after setting all properties.
function edit39_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit39 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit40_Callback(hObject, eventdata, handles)
% hObject    handle to edit40 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit40 as text
%         str2double(get(hObject,'String')) returns contents of edit40 as a
double

% --- Executes during object creation, after setting all properties.
function edit40_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit40 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit41_Callback(hObject, eventdata, handles)
% hObject      handle to edit41 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit41 as text
%      str2double(get(hObject,'String')) returns contents of edit41 as a
double

% --- Executes during object creation, after setting all properties.
function edit41_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit41 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit42_Callback(hObject, eventdata, handles)
% hObject      handle to edit42 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit42 as text
%      str2double(get(hObject,'String')) returns contents of edit42 as a
double

% --- Executes during object creation, after setting all properties.
function edit42_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit42 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit43_Callback(hObject, eventdata, handles)
% hObject    handle to edit43 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit43 as text
%        str2double(get(hObject,'String')) returns contents of edit43 as a
double

% --- Executes during object creation, after setting all properties.
function edit43_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit43 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit44_Callback(hObject, eventdata, handles)
% hObject    handle to edit44 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit44 as text
%        str2double(get(hObject,'String')) returns contents of edit44 as a
double

% --- Executes during object creation, after setting all properties.
function edit44_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit44 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

```

```

function edit47_Callback(hObject, eventdata, handles)
% hObject      handle to edit47 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit47 as text
%        str2double(get(hObject,'String')) returns contents of edit47 as a
double

% --- Executes during object creation, after setting all properties.
function edit47_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit47 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit49_Callback(hObject, eventdata, handles)
% hObject      handle to edit49 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit49 as text
%        str2double(get(hObject,'String')) returns contents of edit49 as a
double

% --- Executes during object creation, after setting all properties.
function edit49_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit49 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit50_Callback(hObject, eventdata, handles)
% hObject      handle to edit50 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit50 as text
%        str2double(get(hObject,'String')) returns contents of edit50 as a
double

% --- Executes during object creation, after setting all properties.
function edit50_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit50 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit51_Callback(hObject, eventdata, handles)
% hObject      handle to edit51 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit51 as text
%        str2double(get(hObject,'String')) returns contents of edit51 as a
double

% --- Executes during object creation, after setting all properties.
function edit51_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit51 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%          See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit52_Callback(hObject, eventdata, handles)
% hObject      handle to edit52 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit52 as text
%        str2double(get(hObject,'String')) returns contents of edit52 as a
double

```

```

% --- Executes during object creation, after setting all properties.
function edit52_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit52 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit53_Callback(hObject, eventdata, handles)
% hObject    handle to edit53 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit53 as text
%       str2double(get(hObject,'String')) returns contents of edit53 as a
double

% --- Executes during object creation, after setting all properties.
function edit53_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit53 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function edit54_Callback(hObject, eventdata, handles)
% hObject    handle to edit54 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit54 as text
%       str2double(get(hObject,'String')) returns contents of edit54 as a
double

% --- Executes during object creation, after setting all properties.
function edit54_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit54 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%LATITUDE INICIAL

latcorr=(upper(get(handles.edit43,'String')));

if latcorr=='N'
latcorr1=1;
elseif latcorr=='S'
latcorr1=(-1);
end

LAT=(str2double(get(handles.edit38,'string')))*latcorr1;
LAT1=(str2double(get(handles.edit39,'string')))*latcorr1;
Latantiga=(LAT+LAT1/60);

%LONGITUDE INICIAL

longcorr=(upper(get(handles.edit42,'String')));

if longcorr=='E'
longcorr1=1;
elseif longcorr=='W'
longcorr1=(-1);
end

LONG=(str2double(get(handles.edit40,'string')))*longcorr1;
LONG1=(str2double(get(handles.edit41,'string')))*longcorr1;
Longantiga=LONG+LONG1/60;
%HORA INTRODUZIDA INICIAL

hh1=str2double(get(handles.edit44,'String'));
hm1=str2double(get(handles.edit47,'String'));
hs1=str2double(get(handles.edit49,'String'));
hh1f=(hh1)+(hm1/60)+(hs1/3600);

%HORA A CALCULAR

hh2=str2double(get(handles.edit52,'String'));
hm2=str2double(get(handles.edit53,'String'));
hs2=str2double(get(handles.edit54,'String'));

```

```

hh2f=hh2+(hm2/60)+(hs2/3600);

%CALCULO DA DIFERENÇA DO TEMPO EM HORAS
t=hh2f-hh1f;

%VELOCIDADE EM NÓS

V=str2double(get(handles.edit51,'String'));

%RUMO

T=str2double(get(handles.edit50,'String'));

%CALCULO DA NOVA POSIÇÃO

Longnova=Longantiga+(t*(V/60))*sind(T)/cosd(Latantiga);
Latnova=Latantiga+t*(V/60)*cosd(T);

%PASSAGEM PARA GRAUS E MINUTOS

set(handles.text147,'String',fix(abs(Latnova)))
set(handles.text148,'String',abs((Latnova-fix(Latnova))*60))
set(handles.text154,'String',round(abs((Latnova-fix(Latnova))*60),1))

if Latnova<=0
slat1='S';
elseif Latnova>0
slat1='N';
end

set(handles.text150,'String',fix(abs(Longnova)))
set(handles.text151,'String',abs((Longnova-fix(Longnova))*60))
set(handles.text153,'String',round(abs((Longnova-fix(Longnova))*60),1))

if Longnova<=0
slong1='W';
elseif Longnova>0
slong1='E';
end

set(handles.text152,'String',slong1)
set(handles.text149,'String',slat1)
function [AOBS13f5] =
VAMOSTENTARFUNCAOLUA (LATfg,Zn,TEMP,PA,DIP,EI,AOBS11,AOBS12,HPL,limbo)
%INSERIR O DIP
DIPc=1.76*sqrt(DIP);
DIPcf=DIPc/60;
%INSERIR O ERRO DE INDICE
EIf=EI/60;
%ALTURA OBSERVADA
AOBS13=AOBS11+AOBS12/60;
%CORREÇÕES

```

```

AOBS13f1=AOBS13-EIf;
%%%%%ERRO DIP
AOBS13f2=AOBS13f1-DIPcf;
%%%%%ERRO REFRAÇÃO
RE1=((PA/1010)*(283/(273+TEMP)))*((1/(tan(degtorad(AOBS13f2+(7.31/(AOBS13f2+4.4))))))-0.06*sin((1/(tan(degtorad(AOBS13f2+(7.31/(AOBS13f2+4.4))))))+13)));
AOBS13f3=AOBS13f2-(RE1/60);
%%%%%ERRO PARALAXE
PAL1=HPL*cos(degtorad(AOBS13f3));
f=(1/298.257);
DPL=f*HPL*((sin(2*degtorad(LATfg))*cos(degtorad(Zn))*sin(degtorad(AOBS13f3)))-((sin(degtorad(LATfg))*sin(degtorad(LATfg)))*cos(degtorad(AOBS13f3))));
PAL=(PAL1+DPL)/60;
AOBS13f4=AOBS13f3+PAL;
%%%%%ERRO SEMI DIAMETRO
SDL=(0.2725*HPL)/60;
AOBS13f5=AOBS13f4+(SDL*limbo);
end

```

```

function [AOBS13f4] = VAMOSTENTARFUNCAOMARTE (TEMP, PA, DIP, EI, AOBS11, AOBS12, HPL)
%INSERIR O DIP
DIPc=1.76*sqrt(DIP);
DIPcf=DIPc/60;
%INSERIR O ERRO DE INDICE
EIf=EI/60;
%ALTURA OBSERVADA
AOBS13=AOBS11+AOBS12/60;
%CORREÇÕES
AOBS13f1=AOBS13-EIf;
%%%%%ERRO DIP
AOBS13f2=AOBS13f1-DIPcf;
%%%%%ERRO REFRAÇÃO
RE1=((PA/1010)*(283/(273+TEMP)))*((1/(tan(degtorad(AOBS13f2+(7.31/(AOBS13f2+4.4))))))-0.06*sin((1/(tan(degtorad(AOBS13f2+(7.31/(AOBS13f2+4.4))))))+13)));
AOBS13f3=AOBS13f2-(RE1/60);
%%%%%ERRO PARALAXE
AOBS13f4=AOBS13f3+HPL/60;
end

```

```

function [AOBS13f5] = VAMOSTENTARFUNCAOSOL (TEMP, PA, DIP, EI, AOBS11, AOBS12, SD)
%INSERIR O DIP
DIPc=1.76*sqrt(DIP);
DIPcf=DIPc/60;
%INSERIR O ERRO DE INDICE
EIf=EI/60;
%ALTURA OBSERVADA
AOBS13=AOBS11+AOBS12/60;
%CORREÇÕES
AOBS13f1=AOBS13-EIf;
%%%%%ERRO DIP
AOBS13f2=AOBS13f1-DIPcf;
%%%%%ERRO REFRAÇÃO

```

```

RE1=((PA/1010)*(283/(273+TEMP)))*((1/(tan(degtorad(AOBS13f2+(7.31/(AOBS13f2+4.4))))))-(0.06*sin((1/(tan(degtorad(AOBS13f2+(7.31/(AOBS13f2+4.4))))))+13)));
AOBS13f3=AOBS13f2-(RE1/60);
%%%%%ERRO PARALAXE
PAL1=(0.15*cos(degtorad(AOBS13f3)))/60;
AOBS13f4=AOBS13f3+PAL1;
%%%%%ERRO SEMI DIAMETRO
AOBS13f5=AOBS13f4+SD/60;
end

function [AOBS13f4] = VAMOSTENTARFUNCAOVENUS (TEMP, PA, DIP, EI, AOBS11, AOBS12, HPL)
%INSERIR O DIP
DIPc=1.76*sqrt(DIP);
DIPcf=DIPc/60;
%INSERIR O ERRO DE INDICE
EIf=EI/60;
%ALTURA OBSERVADA
AOBS13=AOBS11+AOBS12/60;
%CORREÇÕES
AOBS13f1=AOBS13-EIf;
%%%%%ERRO DIP
AOBS13f2=AOBS13f1-DIPcf;
%%%%%ERRO REFRAÇÃO
RE1=((PA/1010)*(283/(273+TEMP)))*((1/(tan(degtorad(AOBS13f2+(7.31/(AOBS13f2+4.4))))))-(0.06*sin((1/(tan(degtorad(AOBS13f2+(7.31/(AOBS13f2+4.4))))))+13)));
AOBS13f3=AOBS13f2-(RE1/60);
%%%%%ERRO PARALAXE
AOBS13f4=AOBS13f3+HPL/60;
end

```