



Universidade do Estado do Rio de Janeiro
Centro de Tecnologia e Ciências
Faculdade de Engenharia

Ricardo Ramos Freire

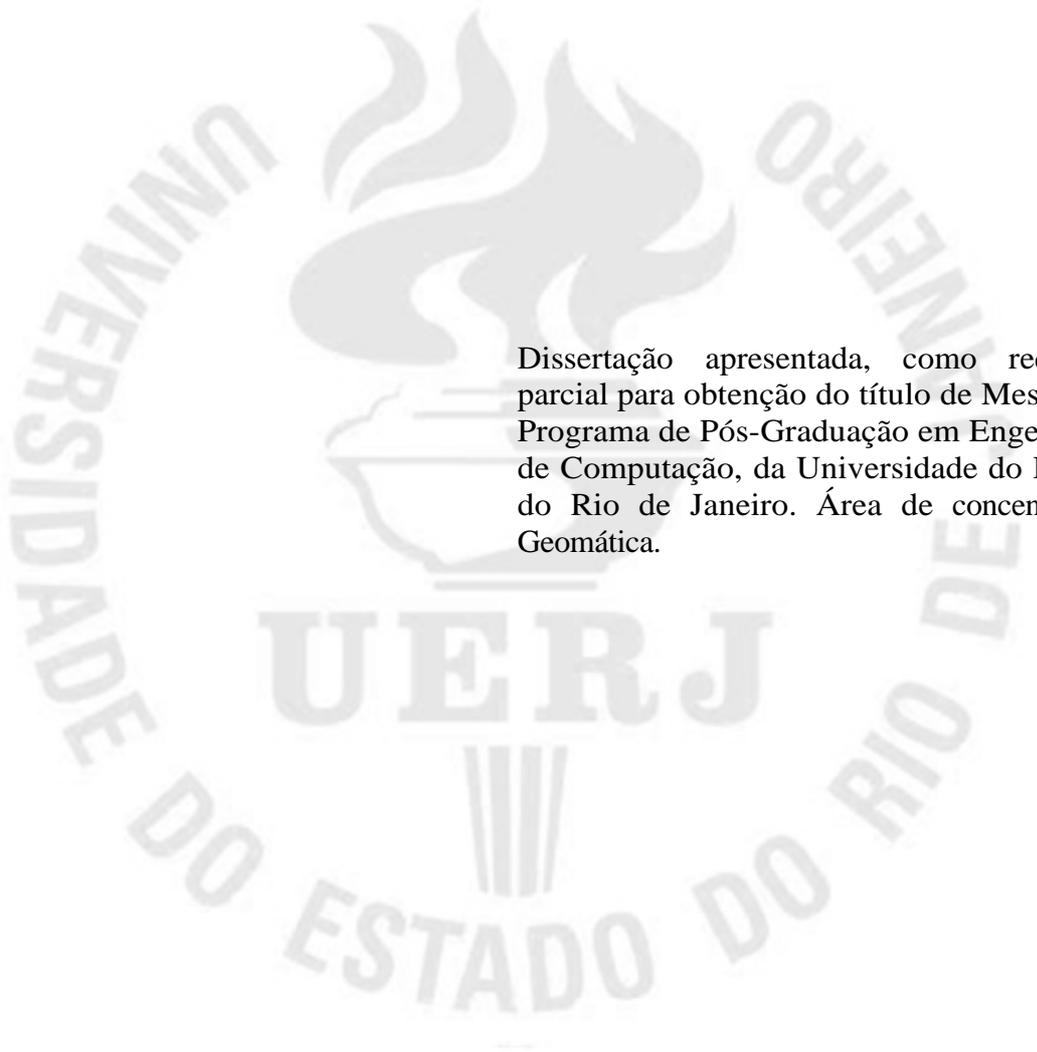
**Cálculo de área de poligonais geodésicas ou loxodrômicas
sobre o elipsóide do Sistema Geodésico WGS-84**

Rio de Janeiro

2009

Ricardo Ramos Freire

**Cálculo de área de poligonais geodésicas ou loxodrômicas
sobre o elipsóide do Sistema Geodésico WGS-84**



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia de Computação, da Universidade do Estado do Rio de Janeiro. Área de concentração: Geomática.

Orientador: Prof. Dr. José Carlos Penna de Vasconcellos

Rio de Janeiro

2009

CATALOGAÇÃO NA FONTE

UERJ/REDE SIRIUS/CTC/B

F866	<p>Freire, Ricardo Ramos. Cálculo de área de poligonais geodésicas ou loxodrômicas sobre o elipsóide do Sistema Geodésico WGS-84 / Ricardo Ramos Freire. – 2009. 121 f.: Il.</p> <p>Orientador: José Carlos Penna de Vasconcellos. Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro. Faculdade de Engenharia.</p> <p>1. Elipsóide – Cálculo - Teses. 2. Geodésia - Teses. 3. Engenharia de Computação. I. Vasconcellos, José Carlos Penna de. II. Universidade do Estado do Rio de Janeiro. III. Título.</p> <p>CDU 528.232.2</p>
------	--

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta dissertação, desde que citada a fonte.

Assinatura

Data

Ricardo Ramos Freire

**Cálculo de área de poligonais geodésicas ou loxodrômicas
sobre o elipsóide do Sistema Geodésico WGS-84**

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia da Computação, da Universidade do Estado do Rio de Janeiro. Área de concentração: Geomática.

Aprovado em: 05 de outubro de 2009.

Banca Examinadora

Prof. Dr. José Carlos Penna de Vasconcellos (Orientador)
Faculdade de Engenharia - UERJ

Prof. Dr. Amauri Ribeiro Destri
Faculdade de Engenharia - UERJ

Prof. Dr. Luiz Felipe Coutinho Ferreira da Silva
Instituto Militar de Engenharia

Rio de Janeiro

2009

DEDICATÓRIA

À minha amada esposa, Adriane Thelma Ribeiro de Oliveira Freire, pelas palavras, atitudes de incentivo e amor incondicional, nunca me permitindo esmorecer durante a elaboração deste trabalho. À minha filha, Anabella Angélica Ribeiro de Oliveira Freire, ainda por nascer, pela alegria de sua constante presença.

AGRADECIMENTOS

À Deus, pela saúde física e mental durante o processo de elaboração deste trabalho.

À Marinha do Brasil e à UERJ pela oportunidade a mim concedida de aprimoramento acadêmico-profissional.

À minha mãe, Marcia Ramos Freire, pela criação e educação, sempre me incentivando a estudar e me aprimorar intelectualmente. Ao meu pai Ricardo Augusto Freire, por seus exemplos de perseverança, nunca me permitindo desistir até o fim da missão.

Aos meus sogros, Milton Teixeira de Oliveira e Angélica Ribeiro de Oliveira, meus segundos pai e mãe, sempre dispostos a ouvir e prestar apoio, em qualquer tempo ou situação, mostrando-me o norte a ser seguido.

A todos os meus familiares pelas orações constantes em prol da conclusão deste trabalho.

Ao professor José Carlos Penna de Vasconcellos pelo voto de confiança depositado em minha pessoa, mesmo sabendo das minhas dificuldades, oriundas de minhas responsabilidades profissionais, em comparecer à UERJ.

Aos meus chefes e colegas do Centro de Hidrografia da Marinha, na pessoa do Capitão-de-Mar-e-Guerra Luiz Carlos Torres, por suas atitudes acolhedoras e constantes incentivos, mesmo agora na reserva remunerada; do Capitão-de-Fragata Francisco Rogério Ribeiro da Silva, Encarregado da Divisão de Cartografia, por sua imprescindível colaboração ao revisar este trabalho, em curtíssimo intervalo de tempo; do Capitão-de-Corveta Sebastião Simões de Oliveira, Encarregado da Seção de Cartas Eletrônicas, por seu apoio e orientação; da Capitão-de-Corveta Flávia Mandarino, minha encarregada direta, por sua colaboração profissional.

A todos os professores, funcionários e colegas do Curso de Geomática.

Apresento aqui também meus agradecimentos a todos os professores do Departamento de Engenharia Cartográfica, que contribuíram para minha formação como Engenheiro Cartógrafo. Como não poderia faltar, agradeço ao Professor Ivan de Araújo Medina, por todas as horas extras dedicadas à minha formação, inclusive nos finais de semana, sempre me abastecendo de materiais e de seus incomparáveis conhecimentos.

RESUMO

FREIRE, Ricardo Ramos. *Cálculo de área de poligonais geodésicas ou loxodrômicas sobre o elipsóide do Sistema Geodésico WGS-84*. 2009. 121 f. Dissertação (Mestrado em Engenharia de Computação) - Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2009.

O cálculo da área de poligonais geodésicas é um desafio matemático instigante. Como calcular a área de uma poligonal sobre o elipsóide, se seus lados não possuem parametrização conhecida? Alguns trabalhos já foram desenvolvidos no intuito de solucionar este problema, empregando, em sua maioria, sistemas projetivos equivalentes ou aproximações sobre esferas autálicas. Tais métodos aproximam a superfície de referência elipsoidal por outras de mais fácil tratamento matemático, porém apresentam limitação de emprego, pois uma única superfície não poderia ser empregada para todo o planeta, sem comprometer os cálculos realizados sobre ela. No Código de Processo Civil, Livro IV, Título I, Capítulo VIII, Seção III artigo 971 diz, em seu parágrafo único, que “não havendo impugnação, o juiz determinará a divisão geodésica do imóvel”. Além deste, existe ainda a Lei 10.267/2001, que regula a obrigatoriedade, para efetivação de registro, dos vértices definidores dos limites dos imóveis rurais terem suas coordenadas georreferenciadas ao Sistema Geodésico Brasileiro (SGB), sendo que áreas de imóveis menores que quatro módulos fiscais terão garantida isenção de custos financeiros. Este trabalho visa fornecer uma metodologia de cálculo de áreas para poligonais geodésicas, ou loxodrômicas, diretamente sobre o elipsóide, bem como fornecer um programa que execute as rotinas elaboradas nesta dissertação. Como a maioria dos levantamentos geodésicos é realizada usando rastreadores GPS, a carga dos dados é pautada em coordenadas (X, Y, Z), empregando o Sistema Geodésico WGS-84, fornecendo a área geodésica sem a necessidade de um produto tipo SIG. Para alcançar o objetivo deste trabalho, foi desenvolvida parametrização diferente da abordagem clássica da Geodésia geométrica, para transformar as coordenadas (X, Y, Z) em geodésicas.

Palavras-chave: Cálculo de áreas; Linhas geodésicas; Linhas loxodrômicas; Parametrização geodésica.

ABSTRACT

The area calculation of geodetic polygonal is a compelling mathematical challenge. How could one calculate the area of a polygon over the ellipsoid, if the sides do not have known parameterization? Some works have already been developed in order to solve this problem, employing mostly equivalent projective systems or authalic spheres approaches. Such methods near the ellipsoidal reference surface by other of easier mathematical treatment, but have limited employment, for a single surface cannot be used for the entire planet, without compromising the calculations over it. In the Code of Civil Procedure, Book IV, Title I, Chapter VIII, Section III, Article 971 says, in its sole paragraph, that "if there is no objection, the judge shall determine the division of the geodesic property". Besides this, there is the Law 10.267/2001, which regulates the requirement for effective registration, that the vertices defining the boundaries of the farms should have their geo-referenced coordinates to Brazilian Geodetic System (BGS), and areas of buildings less than four modules have guaranteed tax-free financial costs. This paper aims to provide a methodology of area calculation for traverses delimited by geodetic lines, or rhumb lines, directly on the ellipsoid, and provide a program that executes routines developed on this work. Since most geodetic surveys are developed using GPS equipment, the data input is based on (X, Y, Z) coordinates, using WGS-84 datum, providing the geodetic area without needing a GIS product. In order to achieve the paper objective, it was developed a different parameterization from the classical geometric Geodesy approach, to transform (X, Y, Z) coordinates into geodetic ones.

Keywords: Area calculation; Geodetic lines; Rhumb lines; Geodetic parameterization.

LISTA DE ILUSTRAÇÕES

Figura 1 - Vetor Gradiente ∇f em um Ponto Qualquer (x,y,z) é Normal ao Plano Tangente à Superfície de Nível no Mesmo Ponto.....	19
Figura 2 - Comprimento do Arco de Curva.....	22
Figura 3 - A Superfície T é a Projeção de S no Plano uv.....	23
Figura 4 - Interpretação Geométrica do Método de Newton-Raphson.....	24
Figura 5 - Cálculo da Área S de uma Poligonal Plana de n Lados	30
Figura 6 - Parametrização Elipsoidal Geodésica e Geocêntrica... ..	38
Figura 7 - Elipsóide Gerado a Partir da Rotação da Elipse Meridiana	41
Figura 8 - Aplicativo AreaCalc Desenvolvido na Dissertação	67

LISTA DE TABELAS

Tabela 1 - Coeficientes Obtidos pelo Polinômio de Maclaurin para Transformar a (4.27) em um Polinômio	47
Tabela 2 - Erros Calculados pela (4.34), de Acordo com o Grau do Polinômio.....	48
Tabela 3 - Coeficientes Obtidos pelo Ajustamento Paramétrico para Complementar o Polinômio de Maclaurin	48
Tabela 4 - Coordenadas dos Pontos que Compõem a Poligonal 1	61
Tabela 5 - Coordenadas dos Pontos que Compõem a Poligonal 2	62
Tabela 6 - Coordenadas dos Pontos que Compõem a Poligonal 3	62
Tabela 7 - Coordenadas dos Pontos que Compõem a Poligonal 4	62
Tabela 8 - Coordenadas dos Pontos que Compõem a Poligonal 5	62
Tabela 9 - Coordenadas dos Pontos que Compõem a Poligonal 6	63
Tabela 10 - Coordenadas dos Pontos que Compõem a Poligonal 7	63
Tabela 11 - Cálculo das Áreas das Poligonais e Respektivas Precisoões	64
Tabela 12 - Resultados Obtidos para Poligonal 7.....	65
Tabela 13 - Coordenadas dos Pontos que Compõem a Poligonal 8	65
Tabela 14 - Coordenadas dos Pontos da Poligonal 8 Projetados	65

SUMÁRIO

	INTRODUÇÃO	12
1	FUNDAMENTAÇÃO MATEMÁTICA E GEODÉSICA	17
1.1	Polinômio de Taylor	17
1.2	Vetor Gradiente	18
1.3	Matriz Jacobiana	19
1.4	Multiplicadores de Lagrange	20
1.5	Comprimento de arco de curva	21
1.6	Integrais de superfície	22
1.7	Método de Newton-Raphson	23
1.8	Ajustamento paramétrico	25
1.9	Solução Direta e Inversa de Vincenty	27
1.10	Equação das áreas dos trapézios de Gauss	30
1.11	Integração Numérica pelo Método de Simpson	32
2	MODELAGEM DA GEOMETRIA ELIPSOIDAL PARA O TRATAMENTO MATEMÁTICO DAS COORDENADAS RASTREADAS NO SISTEMA GPS 34	
2.1	Redução das coordenadas XGPS, YGPS e ZGPS ao elipsóide	34
2.2	Parametrização do elipsóide por coordenadas curvilíneas	37
3	METODOLOGIA DE APOIO PARA O CÁLCULO DAS ÁREAS DE TRIÂNGULOS ELIPSOIDAIIS POLARES	41
3.1	Cálculo da área total do elipsóide	41
3.2	Área de triângulos elipsoidais polares	45
3.3	Simplificação do cálculo das áreas dos triângulos elipsoidais polares	46
4	METODOLOGIA	51
4.1	Preparação dos dados	51
4.1.1	<u>Coordenadas de entrada (XGPS, YGPS, ZGPS)</u>	51
4.1.2	<u>Coordenadas de entrada (u,v)</u>	52
4.2	Determinação das coordenadas dos pontos pertencentes a uma linha geodésica 54	
4.3	Cálculo da área da poligonal geodésica	55
4.4	Adaptação do método para linhas loxodrômicas	57
5	RESULTADOS OBTIDOS	61
6	CONCLUSÕES E RECOMENDAÇÕES	67

REFERÊNCIAS	69
APÊNDICE A – Função Saída2.....	71
APÊNDICE B – Função Saída1	74
APÊNDICE C – Vetor Lat	77
APÊNDICE D – Vetor Função.....	78
APÊNDICE E – Matriz A	79
APÊNDICE F – Função Resultado	80
APÊNDICE G – Matriz Geodésicos2	81
APÊNDICE H – Matriz Loxodrômicos	83
APÊNDICE I – Matriz Geocêntricos	85
APÊNDICE J – Matriz Geocêntricos_loxo.....	86
APÊNDICE K – Código Fonte do Aplicativo AreaCalc	87

INTRODUÇÃO

Caracterização do Problema

O cálculo da área de poligonais geodésicas sobre um elipsóide de referência representa um desafio matemático instigante. Algumas soluções já foram apresentadas, sendo em sua maioria calcadas em aproximações da superfície elipsoidal em superfícies desenvolvíveis ao plano, de forma a simplificar o cálculo da área de interesse. Uma metodologia para o cálculo das áreas de poligonais geodésicas que fosse desenvolvida diretamente sobre o elipsóide, sem aproximações planas, esferas autálicas – de mesma área que a superfície do elipsóide de revolução (Galo, Monico, Oliveira, 2003) – ou quaisquer outras simplificações da superfície de referência proporcionaria a determinação inequívoca da área de determinada porção do planeta.

Algumas soluções comerciais apresentam metodologias distintas para resolver o problema das áreas, conforme abaixo:

- a) o pacote matemático MATLAB 2009b possui a função *AREAINT* que permite o cálculo de poligonais no elipsóide por meio da utilização da esfera autálica e da integral de linha baseada no Teorema de Green. Não há referências quanto à precisão do modelo. As informações encontram-se disponíveis no módulo de ajuda da *Mapping Toolbox* do próprio MATLAB, ou na página da empresa The Mathworkstm;
- b) o Sistema Gerenciador de Banco de Dados (SGBD) ORACLE SPATIAL 11g, ao calcular a área de metade do elipsóide pela função *SDO_AREA*, apresenta um erro de 0,1% (Kothuri, Godfrind, Beinat, 2007). Áreas menores apresentam precisão maior. Não há referência acerca do método empregado para o cálculo das áreas;
- c) a empresa Blue Marble Geographics, produtora do renomado Geographic Calculator, desenvolveu o *GeoCalc.NET 6.3*, que permite o cálculo de poligonais geodésicas por meio do método *PolygonArea*. Não há referências quanto à metodologia nem quanto à precisão da mesma. As informações encontram-se na página de internet da empresa.
- d) a empresa CARIS-UNIVERSAL SYSTEMS LTD criou a solução CARIS HPD (*Hydrographic Production Database*) que está começando a ser empregada no Centro de Hidrografia da Marinha, fora do ambiente de produção, para armazenamento e atualização das cartas náuticas. Essa solução possui ferramentas geodésicas que permitem calcular as áreas de poligonais geodésicas ou loxodrômicas em tempo “real”

(Lévesque, Cockburn, McLeay, 2008). Apesar de não haver referência acerca do método utilizado, acredita-se que as ferramentas geodésicas da CARIS sejam de grande precisão, principalmente em função da grande difusão do CARIS LOTS (*Law of the Sea*). A fim de atender as necessidades geodésicas impostas pela *United Nations Convention on the Law of the Sea* (UNCLOS), esta solução foi especialmente desenvolvida para adequar um Sistema de Informações Geográficas (SIG) à ferramentas geodésicas eficazes, tendo sido empregada por diversos Serviços Hidrográficos desde 1999 (Levésque, 2008). Apesar de não haver documentação comprobatória, acredita-se que os módulos geodésicos do CARIS LOTS estejam presentes no HPD, uma vez que os ângulos e distâncias geodésicos avaliados em ambas as soluções apresentam mesma precisão. Assim, esta ferramenta foi escolhida para confrontar os resultados das áreas geodésicas calculadas nesta dissertação, de forma a estabelecer uma precisão interna entre os métodos. Para avaliar externamente a precisão de ambas as soluções, foram empregadas poligonais passíveis de cálculo algébrico, indicando assim a precisão de cada uma das soluções.

Como é possível constatar, quando há referência acerca do modelo matemático empregado, verifica-se que há uma simplificação dos cálculos para a esfera autálica. Também é possível encontrar estudos baseados em projeções equivalentes (STANQUE, 2007).

Justificativa e Motivação

O Código de Processo Civil, Livro IV, Título I, Capítulo VIII, Seção III artigo 971 diz, em seu parágrafo único, que “... não havendo impugnação, o juiz determinará a divisão geodésica do imóvel; se houver, proferirá no prazo de dez (10) dias, decisão sobre os pedidos e os títulos que devem ser atendidos na formação dos quinhões.” Além deste, existe ainda a Lei 10.267/2001, que regula a obrigatoriedade, para efetivação de registro, dos vértices definidores dos limites dos imóveis rurais terem suas coordenadas georreferenciadas ao Sistema Geodésico Brasileiro (SGB), sendo que áreas de imóveis menores que quatro módulos fiscais terão garantida isenção de custos financeiros.

Tendo em vista que até 2014 deverá estar concluída a mudança para o Sistema de Referência Geocêntrico para as Américas (SIRGAS) como novo SGB, que os novos mapeamentos deverão ser feitos já neste sistema e que não há parâmetros de conversão entre ele e o Sistema Geodésico WGS-84 por serem praticamente idênticos, conforme página de perguntas mais frequentes do IBGE (itens 15 e 16), este estudo foi desenvolvido utilizando-se como referência o WGS-84 (G1150) com os parâmetros referidos a 2000,4. Como todos os

vértices possuirão coordenadas geodésicas geocêntricas, vislumbrou-se criar um modelo matemático que calculasse as áreas geodésicas cujos vértices fossem levantados utilizando rastreadores GPS, que fosse válido para, praticamente, áreas de quaisquer dimensões na superfície do planeta. As coordenadas de entrada poderiam ser as geocêntricas nativas do sistema GPS (X, Y, Z) ou representadas já em latitudes e longitudes geodésicas.

Objetivo Geral

Estabelecer um modelo matemático para o cálculo das áreas de poligonais geodésicas a partir de coordenadas X,Y,Z geocêntricas dos seus vértices, referidas ao WGS-84, e poligonais cujos lados sejam linhas loxodrômicas.

Objetivos Específicos

- a) desenvolver um modelo matemático para reduzir as coordenadas geocêntricas X,Y,Z dos vértices ao elipsóide adotado no Sistema Geodésico WGS-84;
- b) elaborar uma parametrização geodésica, e outra geocêntrica, para as coordenadas curvilíneas sobre o elipsóide – diferente da parametrização geodésica clássica;
- c) elaborar uma forma de contornar o problema da parametrização das curvas geodésicas – lados das poligonais estudadas;
- d) estabelecer um modelo matemático para o cálculo de áreas das poligonais geodésicas;
- e) apresentar uma complementação que indique como adaptar o método de cálculo de áreas para o caso da poligonal ser composta por linhas loxodrômicas;
- f) apresentar um aplicativo que realize todos os cálculos propostos pelos modelos citados a partir de arquivos de coordenadas X,Y,Z tipo ASCII (American Standard Code for Information Interchange);

Área de Estudos

Para avaliar a precisão do método, foram criadas oito poligonais geodésicas. A de maior área entre todas é formada por cinco pontos no Equador espaçados de 72° , de forma que a região calculada possa ser comparada à área de metade do elipsóide, da mesma forma que o ORACLE apresenta a precisão de sua função SDO_AREA. As outras poligonais foram estabelecidas de forma a comparar áreas de grandes e pequenas dimensões à solução comercial CARIS, que apresenta um módulo de cálculo geodésico bastante consistente. Além

disso, os resultados da plataforma CARIS e do programa elaborado na dissertação são confrontados com valores passíveis de cálculos algébricos, quando as poligonais são delimitadas pela linha do Equador e por arcos de meridiano, que são linhas geodésicas de parametrização conhecidas.

Materiais e Métodos

Foram utilizados alguns conceitos matemáticos para desenvolver o algoritmo de cálculo de área das poligonais, a citar: operador gradiente; multiplicadores de Lagrange; método de Newton-Raphson; determinante Jacobiano; integral de superfície; comprimento de arco de curva; integração de linha; soluções direta e inversa de Vincenty das curvas geodésicas sobre o elipsóide; ajustamento paramétrico; adaptação da equação da área dos trapézios de Gauss; e método numérico de Simpson para integração. Para implementação do programa foi utilizado o Microsoft Visual Basic 2008 Express Edition, sobre o Framework .NET 3.5, plataforma gratuita de desenvolvimento da Microsoft. Foi extraída uma imagem do Google Earth, com o Estado de Minas Gerais vetorizado, para ser utilizada como pano de fundo do aplicativo criado no Visual Basic 2008. Como computador, foi utilizado um notebook com processador Intel Core Duo T2350@1.86GHz, com 1GB de memória, disco rígido de 80GB e Sistema Operacional Windows XP Professional SP3. Para confecção das figuras foi utilizado o GIMP 2.4.3 (*GNU Image Manipulation Program*), sistema gratuito de edição de imagens. Alguns módulos de cálculo foram desenvolvidos usando a plataforma Mathcad 14, da empresa Parametric Technology Corporation, para posterior implementação no Visual Basic 2008. Por fim, foi empregada a solução Geocalc 3.09 da empresa Blue Marble Geographics para calcular as coordenadas de alguns pontos em uma projeção equivalente.

Apresentação do Trabalho

Esse trabalho está dividido em 7 capítulos. No primeiro é apresentada uma panorâmica geral da dissertação, os objetivos estabelecidos, uma delineação acerca dos testes realizados, bem como os materiais e métodos empregados.

O Capítulo 2 apresenta uma revisão de conceitos e técnicas que serão necessários ao longo do desenvolvimento do trabalho.

O Capítulo 3 trata da transformação de dados adquiridos por rastreadores GPS (coordenadas nativas em X,Y,Z) em coordenadas geodésicas, curvilíneas, bem como em

geocêntricas. Para tanto, é apresentada a dedução da parametrização elipsoidal, sobre a qual todo o trabalho será edificado.

O Capítulo 4 remete ao desenvolvimento da modelagem matemática para o cálculo das áreas dos triângulos elipsoidais polares, que serão o cerne para o posterior cálculo das áreas das poligonais geodésicas.

O Capítulo 5 apresenta um fluxo de integração dos Capítulos 2 e 3, culminando com a rotina de cálculo das áreas das poligonais geodésicas. Também é apresentada uma rotina adaptada ao caso de poligonais cujos lados sejam linhas loxodrômicas.

O Capítulo 6 apresenta resultados obtidos pelo programa implementado, comparando-os com os valores calculados pela rotina geodésica presente na solução comercial CARIS HPD, além de confrontar ambos os resultados com valores passíveis de cálculos analíticos, como áreas de semi-fusos ou a área de metade do elipsóide de referência do Sistema Geodésico WGS-84.

Finalmente, o Capítulo 7 apresenta as conclusões acerca do trabalho, bem como oportunidades de melhoria e propostas de investigação científica.

1 FUNDAMENTAÇÃO MATEMÁTICA E GEODÉSICA

1.1 Polinômio de Taylor

Admita uma função f que possua derivadas até a ordem n no ponto $x = 0$, sendo $n \geq 1$. Para determinar um polinômio P que seja coincidente com f no ponto $x = 0$, assim como as respectivas derivadas até a ordem n , devem ser verificadas $n + 1$ condições (APOSTOL, 1968):

$$P(0)=f(0), \quad P'(0)=f'(0), \quad \dots, \quad P^{(n)}(0)=f^{(n)}(0) \quad (2.1)$$

Assim, um polinômio $P(x)$ de grau n pode ser representado por:

$$P(x) = c_0 + c_1x + c_2x^2 + \dots + c_nx^n \quad (2.2)$$

Os coeficientes c_n devem ser determinados a partir das condições estabelecidas pelas n derivadas no ponto $x = 0$. Após derivar f k vezes, obtém-se $P^{(k)}(0) = k! c_k$, donde resulta:

$$c_k = \frac{P^{(k)}(0)}{k!} = \frac{f^{(k)}(0)}{k!} \quad (2.3)$$

Admite-se que $k = 0, 1, 2, \dots, n$. Assim, o polinômio é definido por:

$$P(x) = \sum_{k=0}^n \frac{f^{(k)}(0)}{k!} x^k \quad (2.4)$$

A expansão acima é conhecida como polinômio de Maclaurin. De forma análoga ao procedimento encaminhado até aqui, pode-se assumir um polinômio de grau $\leq n$ que coincida com f e suas n primeiras derivadas no ponto $x = a$. Assim, obtém-se o polinômio de Taylor, abaixo representado:

$$P(x) = \sum_{k=0}^n \frac{f^{(k)}(a)}{k!} (x - a)^k \quad (2.5)$$

A utilização desses polinômios em engenharia é de largo espectro, pois muitas vezes é melhor trabalhar com expansões polinomiais do que com funções de modelagem originais. Isso ocorrerá ao longo deste trabalho quando da integração de uma função cuja função resultado não é conhecida. Para contornar esse problema, a função integranda será expandida pelo polinômio de Taylor, uma vez que a integração de um polinômio de grau n terá como resultado outro polinômio de grau $n + 1$ facilmente obtível.

Cabe ressaltar que o erro associado à aproximação é dependente do número de termos que venham a ser empregados. Outro ponto importante é que dependendo da complexidade das derivações sucessivas, em alguns casos será necessário o emprego de métodos numéricos para determinação de seus valores no ponto a .

1.2 Vetor Gradiente

O $\nabla f(a)$ é um vetor cujas componentes são derivadas parciais de f no ponto a (APOSTOL, 1969), conforme abaixo:

$$\nabla f(a) = (D_1 f(a), \dots, D_n f(a)) \quad (2.6)$$

No espaço tridimensional, temos:

$$\nabla f(x, y, z) = \left(\frac{\partial f(x, y, z)}{\partial x}, \frac{\partial f(x, y, z)}{\partial y}, \frac{\partial f(x, y, z)}{\partial z} \right) \quad (2.7)$$

Uma aplicação prática do vetor gradiente é ele ser normal ao plano tangente à superfície de nível $f(x, y, z) = c$, onde c é uma constante, conforme a Figura 1:

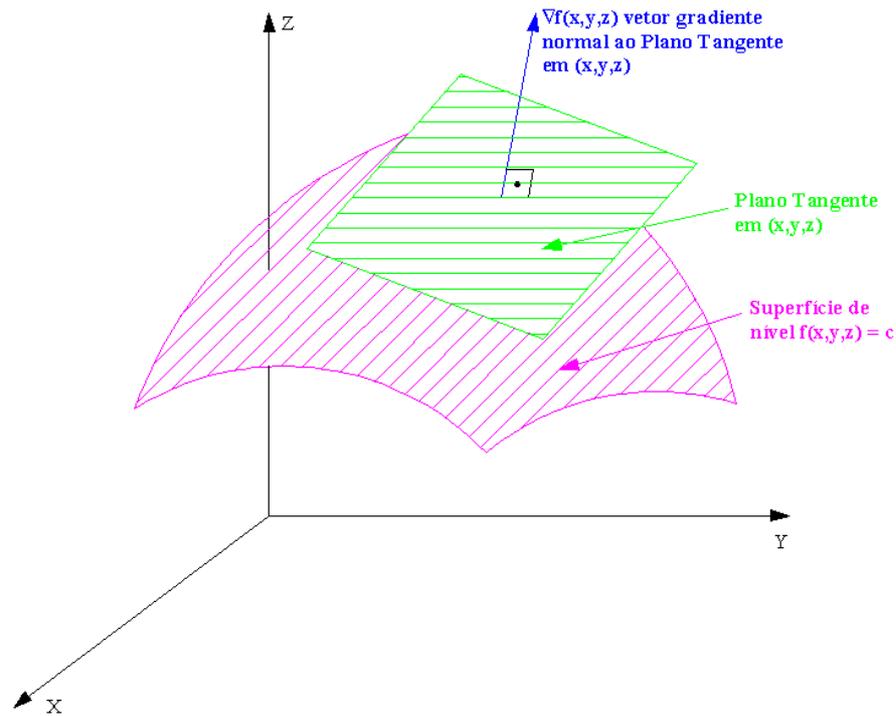


Figura 1 - Vetor Gradiente ∇f em um Ponto Qualquer (x,y,z) é Normal ao Plano Tangente à Superfície de Nível no Mesmo Ponto
Fonte: APOSTOL, 1969

Quando se deseja reduzir um ponto de coordenadas (X,Y,Z) geocêntricas às equivalentes coordenadas curvilíneas geodésicas sobre o elipsóide adotado, isso deve ser feito ao longo do gradiente no ponto homólogo sobre o elipsóide. Essa metodologia será empregada posteriormente nesta dissertação.

1.3 Matriz Jacobiana

A matriz Jacobiana $m \times n$ $Df(a)$ é definida em cada ponto onde as $m \cdot n$ derivadas parciais $D_j f_k(a)$ existem (Apostol, 1969), sendo $j = 1, 2, \dots, n$ e $k = 1, 2, \dots, m$. Assim, tem-se:

$$Df(a) = \begin{bmatrix} D_1 f_1(a) & D_2 f_1(a) & \cdots & D_n f_1(a) \\ D_1 f_2(a) & D_2 f_2(a) & \cdots & D_n f_2(a) \\ \vdots & \ddots & & \vdots \\ D_1 f_m(a) & D_2 f_m(a) & \cdots & D_n f_m(a) \end{bmatrix} \quad (2.8)$$

Quando da mudança de variáveis em integrais duplas, o determinante da matriz Jacobiana é utilizado.

Seja uma integral da forma $\iint_S f(x,y)dx dy$ ao longo de uma região S no plano xy . Para transformá-la em outra integral dupla $\iint_T F(u,v)dudv$ que seja ao longo de uma região T no plano uv , é necessário estabelecer o relacionamento entre S e T e as funções $f(x,y)$ e $F(x,y)$ (Apostol, 1969). Para conectar x,y com u,v é necessário o emprego de duas funções de mapeamento X e Y , conforme abaixo:

$$x = X(u, v) \quad y = Y(u, v) \quad (2.9)$$

Assim, a fórmula para transformar integrais duplas será (APOSTOL, 1969):

$$\iint_S f(x,y)dx dy = \iint_T f[X(u, v), Y(u, v)]|J(u, v)|dudv \quad (2.10)$$

onde $|J(u, v)|$ será o determinante da matriz Jacobiana, que no caso será expressa por:

$$J(u, v) = \begin{bmatrix} \frac{\partial X}{\partial u} & \frac{\partial Y}{\partial u} \\ \frac{\partial X}{\partial v} & \frac{\partial Y}{\partial v} \end{bmatrix} \quad (2.11)$$

O conceito da matriz Jacobiana para funções de mapeamento é importante para o completo entendimento da técnica de cálculo de áreas que será apresentada nesta dissertação. O mapeamento será feito entre coordenadas X,Y,Z (geocêntricas) e u,v (geodésicas), um pouco mais complexo do que o caso ora apresentado.

1.4. Multiplicadores de Lagrange

Problemas de máximo e mínimo com restrições são normalmente bastante complexos (APOSTOL, 1969). Quando o conjunto de restrições tem uma estrutura relativamente simples, é possível empregar o método dos multiplicadores de Lagrange.

Seja um determinado campo escalar $f(x_1, \dots, x_n)$, que possua um extremo relativo quando submetido à m restrições abaixo:

$$g_1(x_1, \dots, x_n) = 0, \dots, g_m(x_1, \dots, x_n) = 0 \quad (2.12)$$

Com $m < n$ existirão m escalares $\lambda_1, \dots, \lambda_m$ tal que, em cada ponto extremo, tem-se:

$$\nabla f = \lambda_1 \nabla g_1 + \dots + \lambda_m \nabla g_m \quad (2.13)$$

Para determinar os pontos extremos (Apostol, 1969), deve-se considerar o sistema de $m + n$ equações obtidas ao tomar as m equações de restrições em (2.12) com as n equações escalares determinadas pela relação vetorial (2.13). Essas equações devem ser resolvidas, quando possível, para $n + m$ variáveis x_1, \dots, x_n e $\lambda_1, \dots, \lambda_m$. Os pontos (x_1, \dots, x_n) nos quais os extremos relativos ocorrem são determinados entre as soluções para as equações apresentadas.

Já os escalares $\lambda_1, \dots, \lambda_m$ que foram introduzidos para auxiliar na resolução deste tipo de problema são chamados multiplicadores de Lagrange. Um multiplicador é introduzido para cada restrição. O campo escalar f e as funções de restrição g_1, \dots, g_n devem ser diferenciáveis. Para que o método seja válido, a condição $m < n$ deve ser observada. Caso contrário, todos os determinantes Jacobianos das funções de restrição serão nulos para os valores extremos em questão (APOSTOL, 1969).

A fim de resolver o problema de redução das coordenadas geocêntricas X, Y, Z ao elipsóide do WGS-84, de forma a transformar o terno citado em coordenadas geodésicas curvilíneas, será empregada a técnica dos multiplicadores de Lagrange.

1.5. Comprimento de arco de curva

Seja $\alpha'(t)$ o vetor velocidade da curva com vetor posicional $\alpha(t)$ e $\|\alpha'(t)\|$ a respectiva grandeza. Se $\alpha'(t)$ é contínuo em um intervalo $[a,b]$, a curva é retificável (possui comprimento) e seu comprimento $s(t)$ é dado por (APOSTOL, 1969):

$$s(u) = \int_a^u \|\alpha'(t)\| dt \quad (2.14)$$

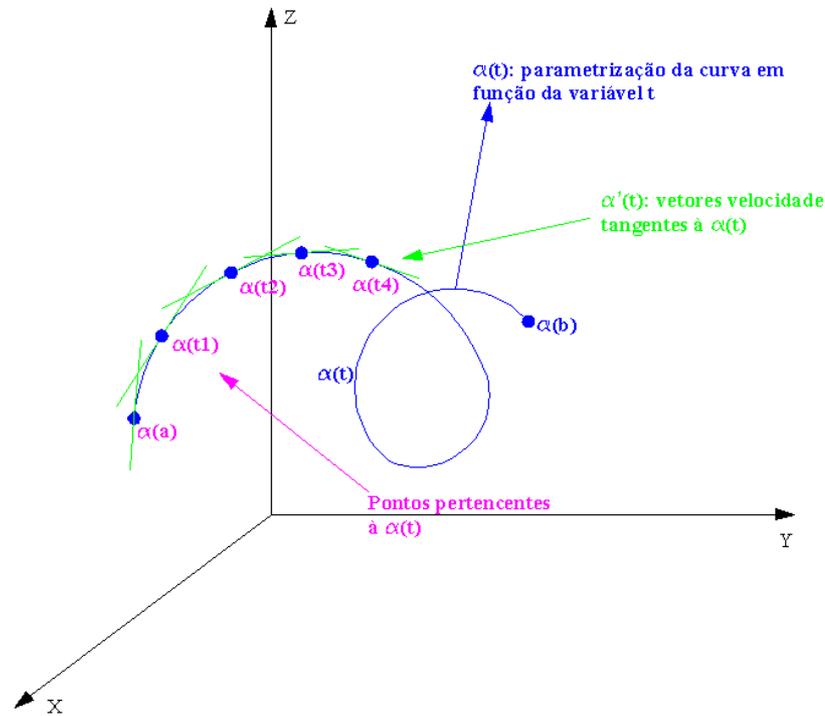


Figura 2 – Comprimento do Arco de Curva

Uma forma de melhor compreender o acima exposto seria a seguinte: como o vetor velocidade por definição é tangente à curva em um ponto qualquer $t \in [a,b]$, conforme a Figura 2, o somatório discreto de todos os módulos desses vetores seria tanto mais próximo do comprimento real da curva quanto maior fosse o número de pontos empregados no intervalo. Se as distâncias entre estes pontos fossem a taxa incremental dt , o somatório seria substituído pela integral descrita em (2.14), cujo resultado seria o exato comprimento da curva. Esse conceito será empregado para calcular a área total do elipsóide de referência.

1.6 Integrais de superfície

Seja $S = r(T)$ uma superfície paramétrica descrita por uma função vetorial r definida em uma região T no plano uv . Um retângulo de área $\Delta u \Delta v$ em T será mapeada por r em um paralelogramo curvilíneo em S com área praticamente igual à equação abaixo (APOSTOL, 1969):

$$\left\| \frac{\partial r}{\partial u} \times \frac{\partial r}{\partial v} \right\| \Delta u \Delta v \quad (2.15)$$

A área de S descrita por $a(S)$ é definida pela integral dupla abaixo (APOSTOL, 1969):

$$a(S) = \iint \left\| \frac{\partial \mathbf{r}}{\partial u} \times \frac{\partial \mathbf{r}}{\partial v} \right\| du dv \quad (2.16)$$

Na Figura 3 encontra-se representado um exemplo do mapeamento acima exposto:

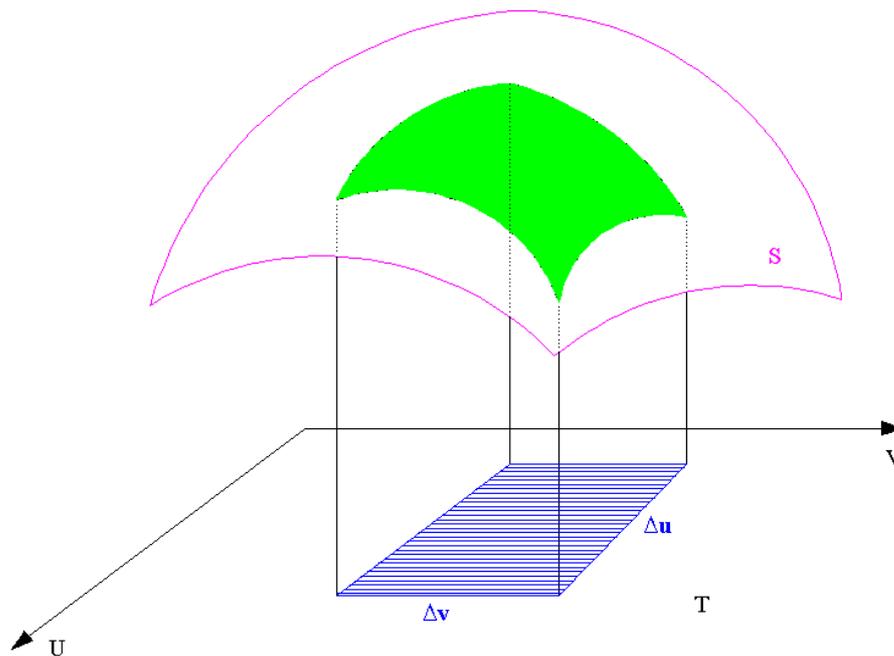


Figura 3 – A Superfície T é a Projeção de S no Plano uv

Fonte: APOSTOL, 1969

A equação (2.16) será a base de implementação para a metodologia do cálculo de áreas de poligonais geodésicas a ser desenvolvida neste trabalho.

1.7 Método de Newton-Raphson

Seja uma determinada função $y = f(x)$, com $x, y \in \mathbb{R}$. Supondo-se que tal função possua pelo menos uma raiz real, deseja-se encontrar um valor de x para o qual $f(x) = 0$. A Figura 4 mostra como seria uma solução para este tipo de problema.

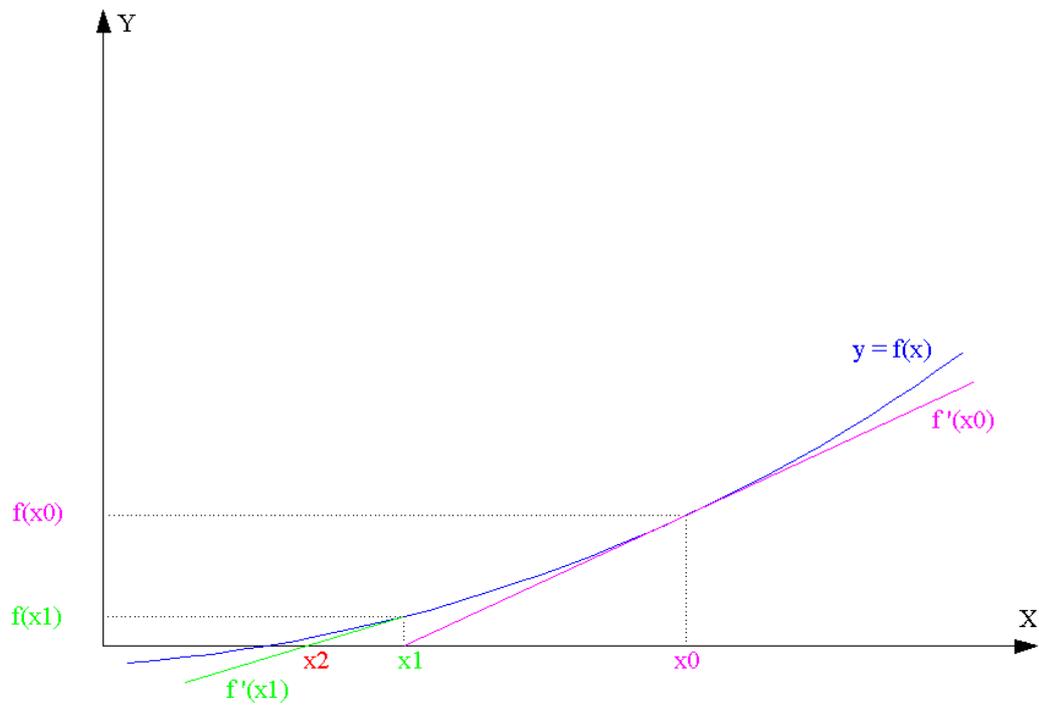


Figura 4 – Interpretação Geométrica do Método de Newton-Raphson

Com base na Figura 4, obtém-se:

$$f'(x_0) = \frac{f(x_0)}{x_0 - x_1} \quad (2.17)$$

$$f'(x_1) = \frac{f(x_1)}{x_1 - x_2} \quad (2.18)$$

⋮

$$f'(x_k) = \frac{f(x_k)}{x_k - x_{k+1}} \quad (2.19)$$

Por fim, tem-se (SANTOS, 1972):

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (2.20)$$

Deve-se estabelecer um critério de parada para o método, pois o mesmo necessita de iterações para que a equação (2.20) forneça a raiz buscada.

Sejam ε e δ números suficientemente pequenos, a parada deve ocorrer quando $|x_{k+1} - x_k| \leq \varepsilon$ ou $|f(x_k)| \leq \delta$.

O método de Newton-Raphson será empregado na redução das coordenadas X,Y,Z geocêntricas ao elipsóide de referência pela direção do vetor gradiente no ponto de redução sobre o elipsóide.

1.8 Ajustamento paramétrico

Sejam n equações não-lineares com u variáveis. Caso as quantidades observadas presentes no vetor L possam ser explicitamente expressadas como funções do vetor de parâmetros X, o modelo matemático poderá ser expresso por (WELLS, KRAKIWSKY, 1997):

$$F(X) = L \quad (2.21)$$

Para linearizar F(X), recorre-se às Séries de Taylor conforme (2.5), obtendo:

$$F(X) = F(X_0) + \left. \frac{dF}{dX} \right|_{X=X_0} (X - X_0) \quad (2.22)$$

Analogamente à (2.21) tem-se:

$$F(X_0) = L_0 \quad (2.23)$$

$$F(X) - F(X_0) = \Delta L \quad (2.24)$$

$$A = \left. \frac{dF}{dX} \right|_{X=X_0} \quad (2.25)$$

$$(X - X_0) = \Delta X \quad (2.26)$$

Por fim, com o modelo já linearizado:

$$A \Delta X = \Delta L \quad (2.27)$$

A solução final será dada por:

$$X = X_0 + \Delta X \quad (2.28)$$

Tal modelagem é conhecida como método paramétrico para modelos não-lineares. O vetor de funções representa n equações (uma por cada observação armazenada no vetor L), com u variáveis e grau de liberdade ($n - u$).

Seja um vetor de resíduos \bar{V} tal que:

$$\bar{V} = A \bar{\Delta X} - \Delta L \quad (2.29)$$

Segundo o critério dos mínimos quadrados, o melhor estimador \bar{X} obtido a partir do cálculo de $\bar{\Delta X}$ é o que irá minimizar a soma dos quadrados dos resíduos ponderados (Wells, Krakiwsky, 1997). A ponderação será dada por meio de uma matriz quadrada de pesos P . Assim:

$$\bar{V}^T P \bar{V} = \text{mínimo} \quad (2.30)$$

A combinação de (2.29) e (2.30) produz:

$$\phi = (A \bar{\Delta X} - \Delta L)^T P (A \bar{\Delta X} - \Delta L) = \text{mínimo} \quad (2.31)$$

Para minimizar a função em (2.31) resulta:

$$\frac{d\phi}{d\bar{\Delta X}} = \frac{d\phi}{d(A \bar{\Delta X} - \Delta L)} \frac{d(A \bar{\Delta X} - \Delta L)}{d\bar{\Delta X}} = 0 \quad (2.32)$$

$$2(A \bar{\Delta X} - \Delta L)^T P A = 0 \quad (2.33)$$

$$(A^T P A) \bar{\Delta X} - A^T P \Delta L = 0 \quad (2.34)$$

As equações em (2.34) são conhecidas como equações normais. O valor de $\overline{\Delta X}$ será dado por (WELLS, KRAKIWSKY, 1997):

$$\overline{\Delta X} = (A^T P A)^{-1} A^T P \Delta L \quad (2.35)$$

A matriz A pode ser encarada como a matriz Jacobiana, com o número de linhas igual ao número de observações e o número de colunas igual ao número de variáveis do modelo a ser ajustado com as observações coletadas. Para implementar a solução não linear apresentada acima, deve-se lançar mão de recursos computacionais, pois serão necessárias algumas ou mesmo muitas iterações, quantidade esta intrinsecamente relacionada ao X_0 (vetor inicial adotado). Algumas vezes pode ser interessante o emprego de um modelo linear simplificado para carregar seu resultado como X_0 no modelo não-linear mais completo, pois isso facilita a convergência.

Para um modelo linear, \overline{X} seria dado por:

$$\overline{X} = (A^T P A)^{-1} A^T P L \quad (2.36)$$

A matriz A continuará sendo encarada como a matriz Jacobiana, observando-se que as derivadas parciais de modelos lineares só podem possuir valores constantes (iguais ou diferentes de zero). Assim, somente uma iteração será necessária para calcular o melhor estimador \overline{X} .

O modelo paramétrico será empregado nesta dissertação quando da simplificação de uma equação de difícil integração por um polinômio, no momento em que o nível de complexidade da expansão em Taylor atingir um limite de cálculo intransponível, em virtude das derivações sucessivas, para os recursos computacionais empregados no desenvolvimento deste trabalho.

1.9 Solução Direta e Inversa de Vincenty

O trabalho desenvolvido por T. Vincenty e publicado na Survey Review número 176 foi empregado nesta dissertação como parte da modelagem utilizada na parametrização das linhas geodésicas. O erro no modelo de Vincenty seria menor que 1mm para medição de distância de linhas de alguns poucos centímetros a 20.000km. Tal método exige a implementação de rotinas iterativas, mas de rápida convergência. Os dados de entrada para a solução inversa encontram-se a seguir:

- a) a: semi-eixo maior do elipsóide;
- b) f: achatamento do elipsóide;
- c) φ_1 : latitude geodésica do ponto de partida, positiva ao norte do Equador;
- d) λ_1 : longitude geodésica/geocêntrica do ponto de partida, positiva a leste do Meridiano de Greenwich;
- e) φ_2 : latitude geodésica do ponto de chegada, positiva ao norte do Equador;
- f) λ_2 : longitude geodésica/geocêntrica do ponto de chegada, , positiva a leste do Meridiano de Greenwich.

O algoritmo implementado para a solução inversa (Vincenty, 1975) encontra-se no Apêndice A.

Esse algoritmo calculará a distância geodésica s e o azimute α_1 entre os pontos de coordenadas geodésicas (φ_1, λ_1) e (φ_2, λ_2) . Para melhor entendimento, a função Saida2 calculará os valores de s e α_1 , fornecendo seus resultados em um vetor. Assim, quando houver referência a Saida2₀ estará sendo calculado o valor de s , bem como Saida2₁ fará referência à α_1 . Na implementação em Visual Basic 2008, foram criadas duas funções, uma que calculava geodesicamente a distância e a outra, o azimute, com base na Função Saida2.

Os dados de entrada para a solução direta encontram-se abaixo:

- g) a: semi-eixo maior do elipsóide;
- h) f: achatamento do elipsóide;
- i) φ_1 : latitude geodésica do ponto de partida, positiva ao norte do Equador;
- j) λ_1 : longitude geodésica/geocêntrica do ponto de partida;
- k) α_1 : azimute geodésico do ponto 1 para o ponto 2;
- l) s : distância geodésica entre os pontos 1 e 2.

O algoritmo implementado para a solução direta (Vincenty, 1975) encontra-se no Apêndice B.

Esse algoritmo calculará as coordenadas de (φ_2, λ_2) a partir do ponto de coordenadas geodésicas (φ_1, λ_1) , distância geodésica s e o azimute geodésico α_1 orientado do primeiro para o segundo ponto. Análogo à função Saida2, armazena-se os valores de φ_2 e λ_2 em um vetor, com Saida1₀ calculando φ_2 e Saida2₁ calculando λ_2 . Na implementação em Visual Basic 2008, foram criadas duas funções, uma que calculava a latitude geodésica e outra, a longitude geodésica.

Para garantir que os valores sejam corretamente computados em ambos os algoritmos, é necessária uma atenção maior ao modo como a linguagem de codificação calcula o arco-tangente. Normalmente os valores calculados estão no intervalo $[-\pi, \pi]$. Para avaliação das latitudes isso não causa problema, mas os azimutes e as longitudes dependem da correta

aferição destes valores. Assim, de forma genérica, caso se deseje calcular o arco-tangente cujo valor seria α , deveriam ser adotados os procedimentos abaixo:

Se $Y > 0$ e $X > 0$ então

$$\alpha = \text{atan}\left(\frac{Y}{X}\right)$$

Se não

$$\text{se } Y < 0 \text{ e } (X > 0 \text{ ou } X < 0) \text{ então } \alpha = \pi + \text{atan}\left(\frac{Y}{X}\right)$$

$$\text{se } Y > 0 \text{ e } X < 0 \text{ então } \alpha = 2\pi + \text{atan}\left(\frac{Y}{X}\right)$$

$$\text{se } Y = 0 \text{ e } X > 0 \text{ então } \alpha = \frac{\pi}{2}$$

$$\text{se } Y = 0 \text{ e } X < 0 \text{ então } \alpha = \frac{3\pi}{2}$$

$$\text{se } Y < 0 \text{ e } X = 0 \text{ então } \alpha = \pi$$

$$\text{se } Y > 0 \text{ e } X = 0 \text{ então } \alpha = 0$$

Fim Se

As soluções direta e inversa de Vincenty serão apresentadas nesta dissertação de forma a ser possível densificar a quantidade de pontos pertencentes a uma linha geodésica que una dois vértices da poligonal.

1.10 Equação das áreas dos trapézios de Gauss

Seja uma poligonal plana de n lados na qual se deseja calcular sua área S , conforme a Figura 5.

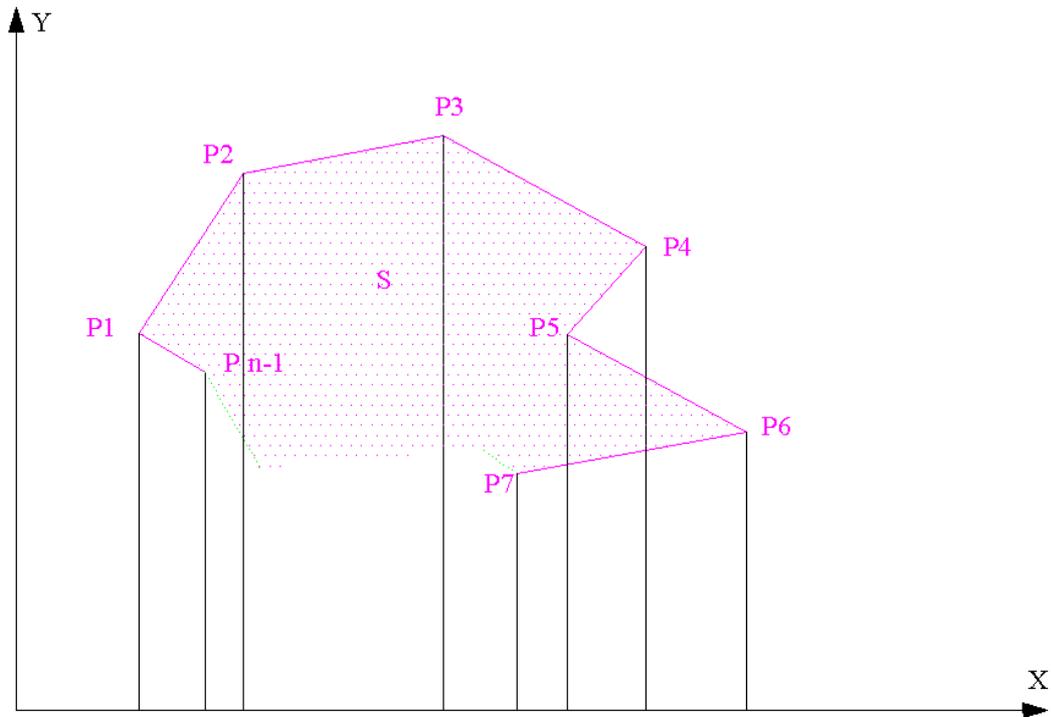


Figura 5 – Cálculo da Área S de uma Poligonal Plana de n Lados

A área total S , conforme o método de Gauss para cálculo de áreas de poligonais, será dada pelo somatório das áreas dos trapézios retangulares formados pelos lados da poligonal, suas respectivas projeções sobre o eixo X , tendo por bases os valores das coordenadas Y dos pontos.

Para o cálculo da área do trapézio retângulo compreendido entre os pontos (X_k, Y_k) e (X_{k+1}, Y_{k+1}) , o primeiro passo é calcular os coeficientes da parametrização do segmento de reta que os une. Assim:

$$y=c_1 x+c_2 \quad (2.37)$$

$$Y_k=c_1 X_k+c_2 \quad (2.38)$$

$$Y_{k+1}=c_1 X_{k+1}+c_2 \quad (2.39)$$

$$c1 = \frac{Y_{k+1} - Y_k}{X_{k+1} - X_k} \quad (2.40)$$

$$c2 = \frac{X_{k+1} Y_k - X_k Y_{k+1}}{X_{k+1} - X_k} \quad (2.41)$$

A partir da parametrização acima, o cálculo de uma área de um trapézio retângulo na superfície plana S, com base em (2.16), será dada por:

$$\text{Area} = \int_{X_k}^{X_{k+1}} \int_0^{c1 x + c2} \left\| \frac{\partial S}{\partial x} \times \frac{\partial S}{\partial y} \right\| dy dx \quad (2.42)$$

Como S é um plano, o primeiro vetor das derivadas parciais será (1, 0, 0) e o segundo (0, 1, 0). Assim o produto vetorial entre eles será (0, 0, 1), cujo módulo será igual a 1. Logo, a área será expressa por:

$$\text{Area} = \int_{X_k}^{X_{k+1}} \int_0^{c1 x + c2} dy dx \quad (2.43)$$

$$\text{Area} = \int_{X_k}^{X_{k+1}} c1 x + c2 dx \quad (2.44)$$

$$\text{Area} = c1 \left. \frac{x^2}{2} \right|_{X_k}^{X_{k+1}} + c2 x \Big|_{X_k}^{X_{k+1}} \quad (2.45)$$

$$\text{Area} = c1 \frac{X_{k+1}^2 - X_k^2}{2} + c2 (X_{k+1} - X_k) \quad (2.46)$$

$$\text{Area} = c1 \frac{(X_{k+1} - X_k)(X_{k+1} + X_k)}{2} + c2 (X_{k+1} - X_k) \quad (2.47)$$

$$\text{Area} = \frac{Y_{k+1} - Y_k}{X_{k+1} - X_k} \frac{(X_{k+1} - X_k)(X_{k+1} + X_k)}{2} + \frac{X_{k+1} Y_k - X_k Y_{k+1}}{X_{k+1} - X_k} (X_{k+1} - X_k) \quad (2.48)$$

$$\text{Area} = \frac{(X_{k+1} - X_k)(Y_{k+1} + Y_k)}{2} \quad (2.49)$$

O cálculo da área total da poligonal plana será dado pelo somatório das áreas trapezoidais em (2.49), o que leva a:

$$\text{Area}_{\text{TOTAL}} = \sum_{k=1}^n \frac{(X_{k+1} - X_k)(Y_{k+1} + Y_k)}{2} = \sum_{k=1}^n \frac{(X_{k+1}Y_{k+1} - X_kY_{k+1} + X_{k+1}Y_k - X_kY_k)}{2} \quad (2.50)$$

Tem-se então:

$$\text{Area}_{\text{TOTAL}} = \sum_{k=1}^n \frac{(X_{k+1}Y_k - X_kY_{k+1})}{2} \quad (2.51)$$

Os termos $X_{k+1}Y_{k+1}$ e X_kY_k do somatório são cancelados ao somar a área de um trapézio com a área dos trapézios adjacentes a ele. Por fim, é necessário ressaltar que o primeiro e o n -ésimo ponto sejam o mesmo, ou seja, ambos possuam as mesmas coordenadas, de forma a garantir um polígono fechado.

Uma adaptação do método acima descrito será utilizada para o cálculo das áreas das poligonais geodésicas.

1.11 Integração Numérica pelo Método de Simpson

No escopo da matemática pura ou aplicada muitos problemas levam à novas funções, cujas propriedades ainda não foram estudadas ou cujos valores ainda não foram tabulados (Apostol, 1969). Tais funções podem ser expressas por integrais, conforme abaixo:

$$F(t) = \int f(t)dt \quad (2.52)$$

Para apresentar uma solução numérica, aproxima-se $f(t)$ por outra função que seja facilmente integrável, garantindo assim um valor aproximado para $F(t)$ no intervalo $[a,b]$. Existem alguns métodos que foram desenvolvidos para essa tarefa, sendo que o escolhido para esta dissertação, por adequação de resultados com a função $f(t)$ empregada, foi o Método de Simpson. Neste método, o intervalo $[a,b]$ é dividido em n partes iguais, sendo que cada subintervalo é interpolado por uma equação parabólica de 2º grau. O valor de n deve ser par. Assim, obtem-se as equações abaixo (SANTOS, 1972):

$$h = \frac{b - a}{n} \quad (2.53)$$

$$F(b) - F(a) = \int_a^b f(t)dt = \sum_{i=1}^{n/2} \frac{h}{3} (f_{2i} + f_{2i-1} + f_{2i-2}) \quad (2.54)$$

Torna-se possível, desta forma, encontrar o valor numérico para integração de $f(t)$ no intervalo $[a,b]$. Tal método será empregado na modelagem apresentada nos Capítulos 3 e 4.

2 MODELAGEM DA GEOMETRIA ELIPSOIDAL PARA O TRATAMENTO MATEMÁTICO DAS COORDENADAS RASTREADAS NO SISTEMA GPS

2.1 Redução das coordenadas XGPS, YGPS e ZGPS ao elipsóide

O cálculo das áreas geodésicas é realizado sobre o elipsóide do sistema geodésico adotado como referência. Assim, como o escopo deste trabalho é focado no sistema WGS-84, a metodologia ora apresentada será pautada em um elipsóide de revolução geocêntrico.

Para projetar as coordenadas geocêntricas XGPS, YGPS e ZGPS obtidas do processamento dos dados coletados por um rastreador GPS, deve-se observar a menor distância entre o ponto rastreado e o elipsóide. Como a menor distância entre um ponto e uma superfície é dada pelo segmento de reta que une o ponto e a perpendicular ao elipsóide, tal reta terá a mesma direção que o vetor gradiente ao elipsóide no ponto projetado. Portanto, seja a equação do elipsóide S, abaixo representada:

$$\left(\frac{X}{a}\right)^2 + \left(\frac{Y}{a}\right)^2 + \left(\frac{Z}{b}\right)^2 = 1 \quad (3.1)$$

Os valores de a e b representam o elipsóide de revolução geocêntrico adotado, bem como X, Y e Z as coordenadas do ponto projetado a partir dos dados processados obtidos pelo rastreador. O vetor gradiente a esta superfície será:

$$\vec{\nabla}S = \left(\frac{2X}{a^2}, \frac{2Y}{a^2}, \frac{2Z}{b^2}\right) \quad (3.2)$$

Como o vetor gradiente representa a direção do segmento de reta que une os pontos (XGPS, YGPS, ZGPS) e (X, Y, Z), tem-se:

$$m \left(\frac{X}{a^2}, \frac{Y}{a^2}, \frac{Z}{b^2}\right) = (X - XGPS, Y - YGPS, Z - ZGPS) \quad (3.3)$$

Admite-se que m seja uma constante diferente de zero. Assim, não há necessidade de representar o vetor gradiente, já multiplicado por m, com suas coordenadas multiplicadas por dois. Dessa forma, chega-se a:

$$X = \frac{XGPS}{1 - \frac{m}{a^2}} \quad (3.4)$$

$$Y = \frac{YGPS}{1 - \frac{m}{a^2}} \quad (3.5)$$

$$Z = \frac{ZGPS}{1 - \frac{m}{b^2}} \quad (3.6)$$

Combinando (3.1) com (3.4), (3.5) e (3.6), chega-se a:

$$\left(\frac{a \text{ XGPS}}{a^2 - m}\right)^2 + \left(\frac{a \text{ YGPS}}{a^2 - m}\right)^2 + \left(\frac{b \text{ ZGPS}}{b^2 - m}\right)^2 = 1 \quad (3.7)$$

Para garantir que a determinação do valor de m na equação (3.7) e das coordenadas em (3.4), (3.5) e (3.6) especifiquem o ponto cuja distância h entre ele e o ponto de coordenadas (XGPS, YGPS, ZGPS) seja mínima, tem-se:

$$h^2 = (X - \text{XGPS})^2 + (Z - \text{ZGPS})^2 + (Z - \text{ZGPS})^2 = \text{mínimo} \quad (3.8)$$

Utilizando o conceito dos Multiplicadores de Lagrange como na equação (2.13), combinando (3.8) e (3.1) por meio da constante ML , diferente de zero, obtém-se:

$$\Omega = (X - \text{XGPS})^2 + (Z - \text{ZGPS})^2 + (Z - \text{ZGPS})^2 + ML \left(1 - \left(\frac{X}{a}\right)^2 - \left(\frac{Y}{a}\right)^2 - \left(\frac{Z}{b}\right)^2 \right) \quad (3.9)$$

$$\frac{\partial \Omega}{\partial X} = 2(X - \text{XGPS}) - 2ML \frac{X}{a^2} = 0 \quad (3.10)$$

$$X = \frac{\text{XGPS}}{1 - \frac{ML}{a^2}} \quad (3.11)$$

$$\frac{\partial \Omega}{\partial Y} = 2(Y - \text{YGPS}) - 2ML \frac{Y}{a^2} = 0 \quad (3.12)$$

$$Y = \frac{\text{YGPS}}{1 - \frac{ML}{a^2}} \quad (3.13)$$

$$\frac{\partial \Omega}{\partial Z} = 2(Z - Z_{GPS}) - 2ML \frac{Z}{b^2} = 0 \quad (3.14)$$

$$Z = \frac{Z_{GPS}}{1 - \frac{ML}{b^2}} \quad (3.15)$$

Combinando (3.11), (3.13) e (3.15) com (3.1), chega-se a:

$$\left(\frac{a X_{GPS}}{a^2 - ML}\right)^2 + \left(\frac{a Y_{GPS}}{a^2 - ML}\right)^2 + \left(\frac{b Z_{GPS}}{b^2 - ML}\right)^2 = 1 \quad (3.16)$$

Como (3.16) é idêntica a (3.7), está provado que o gradiente garante a direção do segmento de reta que representa a menor distância entre o ponto rastreado e o elipsóide geocêntrico adotado. O valor de ML, ou m, permitirá determinar as coordenadas do ponto projetado.

Como a equação (3.16) é uma função não linear e o cálculo de suas raízes algebricamente seria de difícil determinação, recorreu-se ao método numérico de Newton-Raphson para tal. Ao examinar a equação (2.20), verificou-se a necessidade da determinação da função derivada de (3.16), em relação à sua variável ML:

$$f(ML) = \left(\frac{a X_{GPS}}{a^2 - ML}\right)^2 + \left(\frac{a Y_{GPS}}{a^2 - ML}\right)^2 + \left(\frac{b Z_{GPS}}{b^2 - ML}\right)^2 - 1 \quad (3.17)$$

$$D(ML) = \frac{df(ML)}{dML} = \frac{2 a^2 X_{GPS}^2}{(a^2 - ML)^3} + \frac{2 a^2 Y_{GPS}^2}{(a^2 - ML)^3} + \frac{2 b^2 Z_{GPS}^2}{(b^2 - ML)^3} \quad (3.18)$$

Por fim, ao combinar (3.17) e (3.18) com (2.20), obtém-se:

$$ML_{k+1} = ML_k - \frac{f(ML_k)}{D(ML_k)} \quad (3.19)$$

Notou-se que um valor inicial adequado para ML seria igual a b (semi-eixo menor). Tal valor inicial, quando aplicado em (3.19), produz rápida convergência. O critério de parada utilizado foi quando a condição $f(ML_k) \leq f(ML_{k+1})$ fosse atendida. Pela natureza da função $f(ML)$ no campo dos reais, ela será sempre positiva, não havendo a necessidade de utilização

de módulo para comparação dos valores no crivo de parada. Na maioria dos casos testados, com três reiterações já seria possível encontrar numericamente a solução.

Ressalta-se que outros métodos de pesquisa operacional poderiam ser empregados, mas o de Newton-Raphson forneceu uma solução consistente e de rápida convergência. O valor h também é conhecido como altitude geométrica ou elipsoidal.

2.2 Parametrização do elipsóide por coordenadas curvilíneas

No intuito de parametrizar o elipsóide geocêntrico de revolução em um sistema de coordenadas curvilíneo, dois passos consecutivos foram adotados:

- a) determinação de um sistema de coordenadas geodésicas;
- b) determinação de um sistema de coordenadas geocêntricas.

A parametrização geocêntrica será empregada posteriormente nesta dissertação para o cálculo das áreas das poligonais geodésicas.

Sabendo-se que as coordenadas X , Y e Z foram determinadas pelo método numérico anterior, deseja-se agora exprimir cada um dos componentes do terno de coordenadas em função das latitudes e longitudes geodésicas. Para este trabalho, as latitudes geodésicas assumirão valores entre -90° e $+90^\circ$, tendo como 0° a linha do Equador, sendo positiva ao norte desta linha e negativa ao sul dela. Já as longitudes geodésicas estarão entre 0° e 360° , sendo contado a partir do Meridiano de Greenwich, crescendo para leste do mesmo.

Seja o vetor gradiente da superfície S – equação (3.2) – no ponto O . O mesmo encontra-se representado na Figura 6. Como o segmento OP , na Figura 6, é definido pela interseção do vetor gradiente com o plano do Equador no ponto P , tem-se:

$$\overrightarrow{OP} = k \overrightarrow{\nabla S} \quad (3.20)$$

$$\overrightarrow{OP} = k \left(\frac{2x_0}{a^2}, \frac{2y_0}{a^2}, \frac{2z_0}{b^2} \right) \quad (3.21)$$

$$(X_1 - x_0, Y_1 - y_0, 0 - z_0) = k \left(\frac{2x_0}{a^2}, \frac{2y_0}{a^2}, \frac{2z_0}{b^2} \right) \quad (3.22)$$

$$-k \frac{2z_0}{b^2} = -z_0 \quad (3.23)$$

$$k = \frac{b^2}{2} \quad (3.24)$$

$$P = \left(x_0 \left(\frac{a^2 - b^2}{a^2} \right), y_0 \left(\frac{a^2 - b^2}{a^2} \right), 0 \right) \quad (3.25)$$

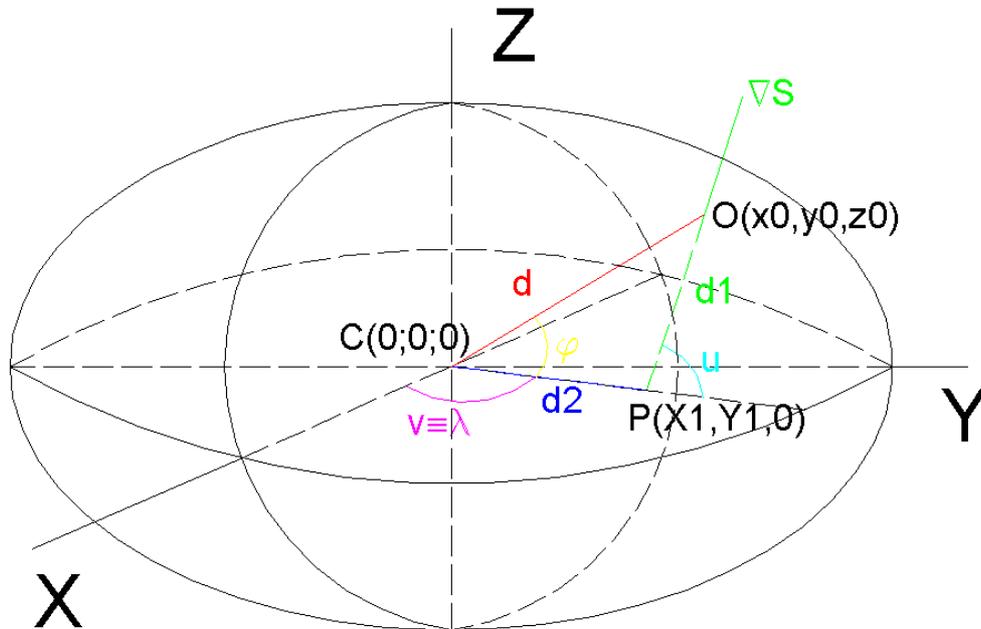


Figura 6 – Parametrização Elipsoidal Geodésica e Geocêntrica

Ao observar a Figura 6, verificam-se diversas relações entre d , $d2$ e as latitudes e longitudes geodésicas e geocêntricas, a seguir:

$$d = \|\vec{CO}\| = \sqrt{(x_0)^2 + (y_0)^2 + (z_0)^2} \quad (3.26)$$

$$x_0^2 + y_0^2 = d^2 \cos^2(\varphi) \quad (3.27)$$

$$d2 = \|\vec{CP}\| = \sqrt{X1^2 + Y1^2} \quad (3.28)$$

A partir de (3.25) e (3.28), tem-se:

$$d2 = \left(\frac{a^2 - b^2}{a^2} \right) \sqrt{x_0^2 + y_0^2} \quad (3.29)$$

Combinando (3.27) e (3.29), chega-se a:

$$d_2 = \left(\frac{a^2 - b^2}{a^2} \right) d \cos(\varphi) \quad (3.30)$$

Novamente na Figura 6, verifica-se agora a relação entre d , d_1 abaixo:

$$d_1 = \|\overrightarrow{PO}\| = \sqrt{(x_0 - X_1)^2 + (y_0 - Y_1)^2 + (z_0 - 0)^2} \quad (3.31)$$

Combinando (3.25) com (3.31):

$$d_1 = \sqrt{\left(x_0 \frac{b^2}{a^2}\right)^2 + \left(y_0 \frac{b^2}{a^2}\right)^2 + (z_0)^2} \quad (3.32)$$

Combinando (3.32) com (3.1) e (3.27):

$$d_1 = \frac{b}{a} \sqrt{\left(\frac{b}{a} d \cos(\varphi)\right)^2 + \left(a^2 - (d \cos(\varphi))^2\right)} \quad (3.33)$$

Por trigonometria elementar chega-se a:

$$\frac{\text{sen}(180^\circ - u)}{d} = \frac{\text{sen}(\varphi)}{d_1} \quad (3.34)$$

$$v = \lambda \quad (3.35)$$

Ao observar a Figura 6, chega-se à seguinte parametrização geodésica inicial para X , Y , Z :

$$X = (d_2 + d_1 \cos(u)) \cos(v) \quad (3.36)$$

$$Y = (d_2 + d_1 \cos(u)) \text{sen}(v) \quad (3.37)$$

$$Z = d_1 \text{sen}(u) \quad (3.38)$$

Combinando (3.26) com (3.1):

$$d = b \sqrt{\left(\frac{x_0}{b}\right)^2 + \left(\frac{y_0}{b}\right)^2 + 1 - \left(\frac{x_0}{a}\right)^2 - \left(\frac{y_0}{a}\right)^2} \quad (3.39)$$

$$d = b \sqrt{\frac{(a^2 - b^2)(x_0^2 + y_0^2)}{a^2 b^2} + 1} \quad (3.40)$$

Combinando (3.40) com (3.27):

$$d^2 = b^2 \left(\frac{(a^2 - b^2)d^2 \cos^2(\varphi)}{a^2 b^2} + 1 \right) \quad (3.41)$$

$$d^2 \left(1 - \frac{(a^2 - b^2) \cos^2(\varphi)}{a^2} \right) = b^2 \quad (3.42)$$

$$d = \frac{a b}{\sqrt{(a \operatorname{sen}(\varphi))^2 + (b \operatorname{cos}(\varphi))^2}} \quad (3.43)$$

Assim, é possível reescrever a parametrização para X, Y e Z em função das coordenadas curvilíneas geocêntricas:

$$X = \left(d_2 + d_1 \sqrt{1 - \left(\frac{d \operatorname{sen}(\varphi)}{d_1} \right)^2} \right) \operatorname{cos}(\lambda) = d \operatorname{cos}(\varphi) \operatorname{cos}(\lambda) \quad (3.44)$$

$$Y = \left(d_2 + d_1 \sqrt{1 - \left(\frac{d \operatorname{sen}(\varphi)}{d_1} \right)^2} \right) \operatorname{sen}(\lambda) = d \operatorname{cos}(\varphi) \operatorname{sen}(\lambda) \quad (3.45)$$

$$Z = d \operatorname{sen}(\varphi) \quad (3.46)$$

3 METODOLOGIA DE APOIO PARA O CÁLCULO DAS ÁREAS DE TRIÂNGULOS ELIPSOIDAIIS POLARES

3.1 Cálculo da área total do elipsóide

De posse da parametrização elipsoidal geodésica desenvolvida no capítulo anterior, cumpre calcular a área total do elipsóide que será posteriormente utilizada para determinar a precisão do método para o cálculo de poligonais geodésicas. Para garantir a consistência do valor calculado, foram utilizados dois métodos, a fim de verificar se os valores das áreas elipsoidais seriam semelhantes entre si.

O primeiro método empregado para calcular a área total do elipsóide foi elaborado partindo da idéia que uma determinada elipse meridiana, com semi-eixo maior a (horizontal) e menor b (vertical), se rotacionada em torno de seu eixo menor geraria a área total do elipsóide de referência. Assim, baseado na Figura 7 abaixo, deve-se, inicialmente, estabelecer uma parametrização para a elipse.

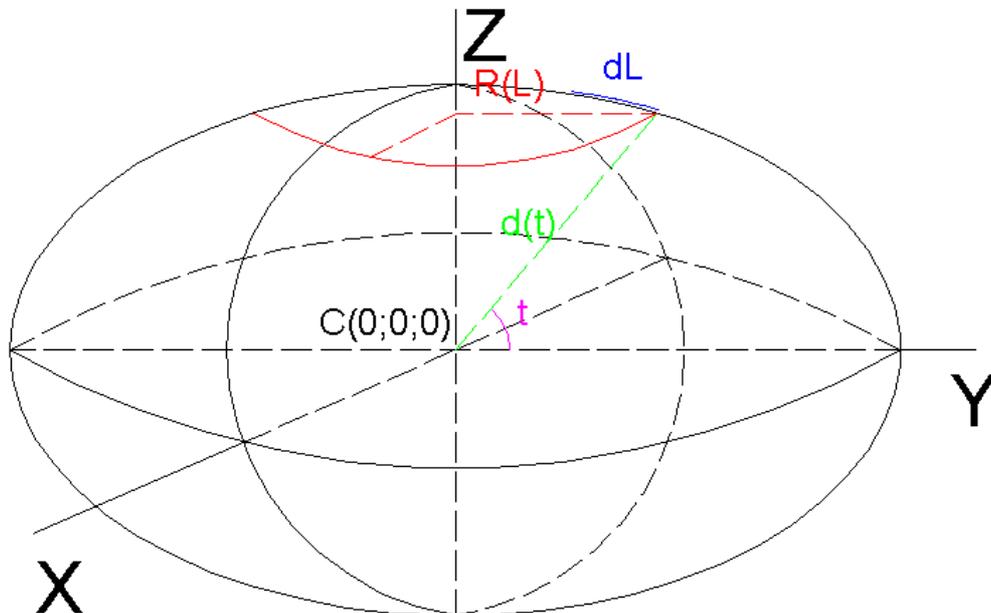


Figura 7 – Elipsóide Gerado a Partir da Rotação da Elipse Meridiana

Abaixo se encontra a metodologia para calcular a área do elipsóide:

$$\text{Area1} = \int 2\pi R(L) dL \quad (4.1)$$

Seja a parametrização para a elipse abaixo, onde $d(t)$, com base em (3.43), pode ser expresso por:

$$(d(t) \cos(t), d(t) \sin(t)) \quad (4.2)$$

$$d(t) = \frac{a b}{\sqrt{(a \sin(t))^2 + (b \cos(t))^2}} \quad (4.3)$$

Com base em (2.14), o comprimento da elipse meridiana L será dado por:

$$L = \int \sqrt{\left(\cos(t) \frac{d}{dt}(d(t)) - d(t) \sin(t)\right)^2 + \left(\sin(t) \frac{d}{dt}(d(t)) + d(t) \cos(t)\right)^2} dt \quad (4.4)$$

$$dL = \sqrt{\left(\cos(t) \frac{d}{dt}(d(t)) - d(t) \sin(t)\right)^2 + \left(\sin(t) \frac{d}{dt}(d(t)) + d(t) \cos(t)\right)^2} dt \quad (4.5)$$

$$R(t) = d(t) \cos(t) \quad (4.6)$$

Assim, a área total será dada por:

$$\text{Area1} = 2 \left[\int_{0^\circ}^{90^\circ} (2\pi(d(t) \cos(t))) \cdot \right.$$

$$\left. \cdot \sqrt{\left(\cos(t) \frac{d}{dt}(d(t)) - d(t) \sin(t)\right)^2 + \left(\sin(t) \frac{d}{dt}(d(t)) + d(t) \cos(t)\right)^2} dt \right] \quad (4.7)$$

A área total calculada por este método foi de 510065621724088,44 m², tendo o resultado desta integração sido obtido numericamente pelo método adaptativo no programa Mathcad versão 14.0.0.163.

Para confirmar o método acima, foi utilizado o conceito das integrais de superfície. Desta forma, para calcular a área total do elipsóide, recorre-se à (2.16) e às parametrizações geocêntricas elipsoidais em (3.44), (3.45) e (3.46), conforme a seguir:

$$\text{Area2} = \iint \left\| \frac{\partial}{\partial \varphi} \mathbf{r}(\varphi, \lambda) \times \frac{\partial}{\partial \lambda} \mathbf{r}(\varphi, \lambda) \right\| d\varphi d\lambda \quad (4.8)$$

$$\mathbf{r}(\varphi, \lambda) = (X(\varphi, \lambda), Y(\varphi, \lambda), Z(\varphi, \lambda)) \quad (4.9)$$

$$X(\varphi, \lambda) = \left(d_2 + d_1 \sqrt{1 - \left(\frac{d \operatorname{sen}(\varphi)}{d_1} \right)^2} \right) \cos(\lambda) \quad (4.10)$$

$$Y(\varphi, \lambda) = \left(d_2 + d_1 \sqrt{1 - \left(\frac{d \operatorname{sen}(\varphi)}{d_1} \right)^2} \right) \operatorname{sen}(\lambda) \quad (4.11)$$

$$Z(\varphi, \lambda) = d \operatorname{sen}(\varphi) \quad (4.12)$$

$$\frac{\partial}{\partial \lambda} X(\varphi, \lambda) = -d \operatorname{sen}(\lambda) \cos(\varphi) \quad (4.13)$$

$$\frac{\partial}{\partial \varphi} X(\varphi, \lambda) = \frac{-d^3 \operatorname{sen}(\varphi) \cos(\lambda)}{b^2} \quad (4.14)$$

$$\frac{\partial}{\partial \lambda} Y(\varphi, \lambda) = d \cos(\lambda) \cos(\varphi) \quad (4.15)$$

$$\frac{\partial}{\partial \varphi} Y(\varphi, \lambda) = \frac{-d^3 \operatorname{sen}(\varphi) \operatorname{sen}(\lambda)}{b^2} \quad (4.16)$$

$$\frac{\partial}{\partial \lambda} Z(\varphi, \lambda) = 0 \quad (4.17)$$

$$\frac{\partial}{\partial \varphi} Z(\varphi, \lambda) = \frac{-d^3 \cos(\varphi)}{a^2} \quad (4.18)$$

$$\left\| \frac{\partial}{\partial \varphi} \mathbf{r}(\varphi, \lambda) \times \frac{\partial}{\partial \lambda} \mathbf{r}(\varphi, \lambda) \right\| = \left\| \begin{array}{ccc} \frac{\partial}{\partial \varphi} X(\varphi, \lambda) & \frac{\partial}{\partial \varphi} Y(\varphi, \lambda) & \frac{\partial}{\partial \varphi} Z(\varphi, \lambda) \\ \frac{\partial}{\partial \lambda} X(\varphi, \lambda) & \frac{\partial}{\partial \lambda} Y(\varphi, \lambda) & \frac{\partial}{\partial \lambda} Z(\varphi, \lambda) \end{array} \right\| \quad (4.19)$$

$$\left\| \frac{\partial}{\partial \varphi} r(\varphi, \lambda) \times \frac{\partial}{\partial \lambda} r(\varphi, \lambda) \right\| = \sqrt{(A(\varphi, \lambda))^2 + (B(\varphi, \lambda))^2 + (C(\varphi, \lambda))^2} \quad (4.20)$$

$$A(\varphi, \lambda) = \frac{\partial}{\partial \lambda} Z(\varphi, \lambda) \frac{\partial}{\partial \varphi} Y(\varphi, \lambda) - \frac{\partial}{\partial \varphi} Z(\varphi, \lambda) \frac{\partial}{\partial \lambda} Y(\varphi, \lambda) \quad (4.21)$$

$$A(\varphi, \lambda) = \frac{-d^4 \cos^2(\varphi) \cos(\lambda)}{a^2} \quad (4.22)$$

$$B(\varphi, \lambda) = \frac{\partial}{\partial \lambda} Z(\varphi, \lambda) \frac{\partial}{\partial \varphi} X(\varphi, \lambda) - \frac{\partial}{\partial \varphi} Z(\varphi, \lambda) \frac{\partial}{\partial \lambda} X(\varphi, \lambda) \quad (4.23)$$

$$B(\varphi, \lambda) = \frac{d^4 \cos^2(\varphi) \sin(\lambda)}{a^2} \quad (4.24)$$

$$C(\varphi, \lambda) = \frac{\partial}{\partial \lambda} X(\varphi, \lambda) \frac{\partial}{\partial \varphi} Y(\varphi, \lambda) - \frac{\partial}{\partial \varphi} X(\varphi, \lambda) \frac{\partial}{\partial \lambda} Y(\varphi, \lambda) \quad (4.25)$$

$$C(\varphi, \lambda) = \frac{d^4 \cos(\varphi) \sin(\varphi)}{b^2} \quad (4.26)$$

$$\left\| \frac{\partial}{\partial \varphi} r(\varphi, \lambda) \times \frac{\partial}{\partial \lambda} r(\varphi, \lambda) \right\| = \frac{d^4 \cos(\varphi) \sqrt{b^4 \cos^2(\varphi) + a^4 \sin^2(\varphi)}}{(a b)^2} \quad (4.27)$$

Assim, o valor para Area2 será dado por:

$$\text{Area2} = 8 \int_{0^\circ}^{90^\circ} \int_{-90^\circ}^{0^\circ} \frac{d^4 \cos(\varphi) \sqrt{b^4 \cos^2(\varphi) + a^4 \sin^2(\varphi)}}{(a b)^2} d\varphi d\lambda \quad (4.28)$$

Esse valor também foi calculado numericamente, chegando ao resultado igual a 510065621724088,56 m², levando à diferença entre os resultados de 0,125 m². Esta diferença entre os resultados pode ser eliminada quando a equação (4.28) é rearranjada abaixo:

$$\text{Area2} = 8 \int_{0^\circ}^{90^\circ} \int_{-90^\circ}^{0^\circ} \frac{d^4 \cos(\varphi) \sqrt{\left(\frac{b}{a}\right)^4 \cos^2(\varphi) + \sin^2(\varphi)}}{b^2} d\varphi d\lambda \quad (4.29)$$

Resultando a área do elipsóide de referência em 510065621724088,44 m². É possível encontrar valor semelhante empregando formulações clássicas de Geodésia, como 510065621,7 km² para o elipsóide GRS80 (RAPP, 1991).

3.2 Área de triângulos elipsoidais polares

Com base no que foi apresentado sobre o método de Gauss para o cálculo das áreas de poligonais planas, surge uma idéia: adaptar o método do plano para o elipsóide. Como o sistema de coordenadas do método de Gauss é plano-retangular, a área entre o lado da poligonal e o eixo horizontal de referência será dada pelo cálculo da área de um trapézio retangular. O somatório das diversas áreas será igual à área da poligonal plana, conforme (2.51). No caso elipsoidal, em que o sistema de coordenadas é curvilíneo, o lado de uma poligonal será balizado por dois meridianos, um em cada extremo. Como os meridianos convergem em direção aos pólos, a área plana trapezoidal entre o lado da poligonal e o eixo horizontal definido terá seu equivalente no elipsóide na figura de um triângulo polar. Para realização deste trabalho foi escolhido o Pólo Sul como referência.

Assim, uma pequena modificação em (4.28) será suficiente para adequar a equação ao triângulo elipsoidal polar, conforme abaixo:

$$\text{Area} = \int_{\lambda_1}^{\lambda_2} \int_{-90^\circ}^{\varphi(\lambda)} \frac{d^4 \cos(\varphi) \sqrt{b^4 \cos^2(\varphi) + a^4 \sin^2(\varphi)}}{(a b)^2} d\varphi d\lambda \quad (4.30)$$

A função $\varphi(\lambda)$ representa a necessidade de expressar o lado da poligonal em função apenas da longitude. Como ainda não há parametrização das linhas geodésicas, especialmente para o sistema de coordenadas geocêntricas deste trabalho, recorre-se ao adensamento da quantidade de pontos que compõem a linha geodésica. Assim, a distância entre os pontos permitirá que seja empregada alguma geometria mais simples, cuja parametrização permita a resolução da integral em (4.30).

Com base nos algoritmos apresentados para o cálculo direto e indireto de Vincenty, foram criadas quatro funções: DistGeod; Azimute; Latitude; e Longitude. DistGeod calcula a distância geodésica entre dois pontos e Azimute calcula o azimute geodésico avante entre dois pontos. Ambas as funções são baseadas no método indireto.

Já pelo método direto, de posse das coordenadas geodésicas de um ponto, da distância entre ele e o próximo ponto e de seu azimute, é possível calcular as coordenadas do segundo

ponto por meio das funções Latitude e Longitude. Todas estas funções estão presentes no Apêndice C do trabalho que contem o código da implementação em Visual Basic 2008. A utilização de tais funções será apresentada posteriormente na metodologia.

3.3 Simplificação do cálculo das áreas dos triângulos elipsoidais polares

A partir da equação (4.30), verifica-se que a função para o cálculo das áreas seria simplificada se a função expressa em (4.27) fosse polinomial. Para tanto, foi utilizado o polinômio de Maclaurin, como em (2.4), gerando a função teste(φ):

$$\text{teste}(\varphi) = \sum_{k=0}^n (a_k \varphi^k) \quad (4.31)$$

A expansão foi realizada sobre a função em (4.27), dividida por a^2 , de forma a simplificar os cálculos. Por meio de operações algébricas, chega-se a:

$$\frac{\sqrt{\left(\frac{a^4 - b^4}{b^4}\right) \frac{\text{sen}^2(2\varphi)}{4} + 1 - \text{sen}^2(\varphi)}}{\left(\left(\frac{a^2 - b^2}{b^2}\right) \text{sen}^2(\varphi) + 1\right)^2} \quad (4.32)$$

Criou-se, então, a função func(φ), com base nos parâmetros do WGS-84:

$$\text{func}(\varphi) = \frac{\sqrt{(0,0033811035752230055068)\text{sen}^2(2\varphi) + 1 - \text{sen}^2(\varphi)}}{\left((0,006739496742276433587)\text{sen}^2(\varphi) + 1\right)^2} \quad (4.33)$$

A seguir, encontram-se os coeficientes a_k pares obtidos até o grau 32 a partir de (4.33):

Tabela 1 – Coeficientes Obtidos pelo Polinômio de Maclaurin para Transformar a (4.27) em um Polinômio

Coeficientes	Valores
a0	1
a2	-0,50671678633410685616
a4	0,047286239591878347278
a6	-0,0031125429187103130978
a8	0,00031055439304701349728
a10	-0,000031104822192129267989
a12	0,0000024964137564367445597
a14	-1,6692337896406958674e-7
a16	8,263766056881320984e-9
a18	-1,2049727257425490291e-10
a20	-1,2362870861489275808e-10
a22	2,5099753377400407442e-11
a24	-3,9319612844930537355e-12
a26	5,5176876672567971383e-13
a28	-7,2890259262632540836e-14
a30	-9,2655755873024710722e-15
a32	1,1469660967064846388e-15

O cálculo dos coeficientes foi limitado ao grau 32 em virtude do Mathcad 14 não conseguir, no computador usado para elaboração da dissertação, realizar a expansão além dele. Além do grau 32 o programa se corrompia e fechava. Os coeficientes ímpares foram todos nulos.

Para avaliar a precisão obtida com a função teste(φ), foram feitas comparações de resultados entre ela e (4.27). Verificou-se que quanto mais afastado do Equador, maiores eram as diferenças, sendo que em -90° a equação (4.27) valia 0 e teste(φ) era igual a $-5,0043 \cdot 10^{-10}$. Tais valores, à primeira vista, seriam muito pequenos para representarem problemas. Porém, ao realizar uma integração da função (4.27) e da função teste(φ) entre os limites de -90° e 90° , foram encontradas diferenças da ordem de $-1850,484375 \text{ m}^2$ a partir da equação abaixo:

$$\left(a^2 \int_{-90^\circ}^{90^\circ} \text{teste}(\varphi) d\varphi \right) - \left(\int_{-90^\circ}^{90^\circ} \frac{d^4 \cos(\varphi) \sqrt{b^4 \cos^2(\varphi) + a^4 \sin^2(\varphi)}}{(a b)^2} d\varphi \right) \quad (4.34)$$

A idéia desta avaliação vem do fato que a solução da integração dupla da equação (4.30) passar primeiro pela integração em relação a $d\varphi$. Além disso, o limite superior de integração $\varphi(\lambda)$ será de 90° , pois as latitudes geocêntricas assumem valores entre -90° e 90° , permitindo que a integração se desenvolva por toda a amplitude das latitudes.

Como teste(φ), com grau 32, ainda não era um modelo satisfatório, conforme relatado acima, para substituir a função em (4.27), foi utilizado o ajustamento paramétrico por mínimos quadrados, de forma a obter coeficientes que permitissem melhorar a modelagem polinomial. Para tanto, foi criado um vetor Lat no qual foram armazenados, por meio de uma

estrutura de repetição com passo de 1 segundo de arco, os valores das latitudes no intervalo entre -90° e 90° , conforme o Apêndice C.

Também foi criado um vetor Função (Apêndice D) no qual foram armazenados os valores da diferença entre as funções teste(φ) e func(φ), ou seja, as observações do ajustamento. Tal vetor teve seus valores alocados por meio de estrutura de repetição com passo de 1 segundo, na qual φ assumiu os valores representados pelo vetor Lat. Dessa forma, foi possível representar a dispersão entre teste(φ) e func(φ) na total amplitude das latitudes geocêntricas.

De posse do vetor das observações Função, foi necessário criar a matriz A que representaria a matriz Jacobiana, presente em (2.36), para que fosse possível resolver a determinação dos coeficientes. Elaborou-se novamente uma estrutura de repetição, sendo que a quantidade de colunas da matriz A foi implementada para ser dependente do grau do polinômio a ser buscado (Apêndice E).

Com a matriz A e o vetor Função, é possível calcular os coeficientes para o polinômio. Abaixo, encontra-se uma tabela com o grau dos novos polinômios e os valores dos erros calculados, conforme (4.34):

Tabela 2 – Erros Calculados pela (4.34), de Acordo com o Grau do Polinômio

Polinômios (grau)	Erros
grau 32	-1850,484375
grau 34	-11,234375
grau 36	-0,125
grau 38	-0,015625
grau 40	0

Assim, verifica-se que o polinômio de grau 40 atende com suficiência a simplificação almejada. Os coeficientes para o grau 40 constam da Tabela 3.

Tabela 3 – Coeficientes Obtidos pelo Ajustamento Paramétrico para Complementar o Polinômio de Maclaurin

Coeficientes	Valores
a34	1,3524360829030762e-16
a36	-1,1013989265482367e-17
a38	-7,366332867999105e-19
a40	2,586160539607493e-19

Por fim, foi calculada a área da superfície total do elipsóide utilizando o polinômio encontrado, obtendo-se o valor igual a 510065621724088,5.

Ao observar (4.30), verifica-se que ainda há necessidade de simplificar o limite superior da primeira integração, a função $\varphi(\lambda)$.

Sejam dois pontos 1 e 2 adjacentes, consecutivos e suficientemente próximos, pertencentes ao lado de uma poligonal geodésica. Admitindo-se que $\varphi(\lambda)$ seja a parametrização da pequena geodésica formada pelos dois pontos, deseja-se aproximar esta curva, de parametrização desconhecida, por um arco de elipse. Para tanto, assume-se que tal elipse será definida pela interseção entre um plano que passe pelo geocentro, ou seja, coordenadas (0, 0, 0) e pelos pontos 1 e 2, pertencentes ao elipsóide. Assim, o produto vetorial seguinte determina os coeficientes do plano:

$$\begin{vmatrix} X1 & Y1 & Z1 \\ X2 & Y2 & Z2 \\ i & j & k \end{vmatrix} = (\text{coef1}, \text{coef2}, \text{coef3}) \quad (4.35)$$

$$\text{coef1} = Y1 Z2 - Z1 Y2 \quad (4.36)$$

$$\text{coef2} = Z1 X2 - X1 Z2 \quad (4.37)$$

$$\text{coef3} = X1 Y2 - Y1 X2 \quad (4.38)$$

$$\text{coef1} X + \text{coef2} Y + \text{coef3} Z = 0 \quad (4.39)$$

O termo de translação é igual a zero, pois o plano passa pelo geocentro. Combinando (4.39) com (3.45), (3.46) e (3.47), obtem-se:

$$\varphi(\lambda) = \text{atan} \left(\frac{-\text{coef1} \cos(\lambda) - \text{coef2} \text{sen}(\lambda)}{\text{coef3}} \right) \quad (4.40)$$

Com a equação (4.40) é possível realizar o cálculo das áreas dos triângulos elipsoidais e, portanto, das áreas das poligonais geodésicas. A forma genérica para o cálculo será dada por:

$$\text{Area} = \int_{\lambda_1}^{\lambda_2} \sum_{k=0}^n \left\{ \frac{a_k}{k+1} \left[\text{atan} \left(\frac{-\text{coef1} \cos(\lambda) - \text{coef2} \text{sen}(\lambda)}{\text{coef3}} \right) \right]^{k+1} \right\} d\lambda \quad (4.41)$$

A integral descrita em (4.41) pode ser numericamente resolvida pelo Método de Simpson, empregando a equação (2.54).

$$\int_{\lambda_1}^{\lambda_2} \int_{-90^\circ}^{\text{atan}\left(\frac{-\text{coef1} \cos(\lambda) - \text{coef2} \sin(\lambda)}{\text{coef3}}\right)} \sum_{k=0}^n (a_k \varphi^k) d\varphi d\lambda$$

$$\sum \int_{\lambda_1}^{\lambda_2} \int_{-90^\circ}^{\text{atan}\left(\frac{-\text{coef1} \cos(\lambda) - \text{coef2} \sin(\lambda)}{\text{coef3}}\right)} \sum_{k=0}^n (a_k \varphi^k) d\varphi d\lambda$$

4 METODOLOGIA

4.1 Preparação dos dados

Partindo do exposto até agora, cumpre organizar os modelos apresentados, de forma a criar um fluxo de trabalho. O primeiro passo é em relação aos dados de entrada no sistema. Da forma que foi elaborado, este projeto admite entrada em coordenadas (XGPS, YGPS, ZGPS), nativas do sistema GPS, ou coordenadas geodésicas (u, v). Tais coordenadas deverão ser fornecidas por meio de arquivo ASCII, separadas por espaços simples, com cada ponto em uma linha. No caso das coordenadas (XGPS, YGPS, ZGPS) o formato de cada linha será: XXXXXX.XXX YYYYYY.YYY ZZZZZZ.ZZZ. Já o formato das coordenadas geodésicas será uu.uuuuuu vv.vvvvvv. O programa elaborado prevê entrada de coordenadas apenas no formato do terno cartesiano (XGPS, YGPS e ZGPS), por ser o caso mais completo de estruturação.

4.1.1 Coordenadas de entrada (XGPS, YGPS, ZGPS)

Usou-se uma estrutura de repetição desenvolvida pelo autor, criando a função Resultado, com base no apresentado no Capítulo 3, utilizando as equações (3.17) e (3.18), apresentada no Apêndice F.

Com o valor obtido pela função Resultado, obtem-se o valor ML, necessário para calcular os valores das equações (3.11), (3.13) e (3.15). Assim, determinam-se as coordenadas (X, Y, Z) reduzidas ao elipsóide de referência. Para calcular as coordenadas geodésicas curvilíneas (u,v) homólogas dos pontos sobre o elipsóide, empregam-se as equações (3.32), (3.34), (3.35) e (3.43), obtendo-se:

$$v = \text{atan} \left(\frac{Y}{X} \right) \quad (5.1)$$

$$\phi = \text{asin} \left(\frac{Z}{\sqrt{(X)^2 + (Y)^2 + (Z)^2}} \right) \quad (5.2)$$

$$u = \text{asin} \left(\frac{d}{d_1} \sin(\phi) \right) \quad (5.3)$$

É necessário observar o exposto no algoritmo Saida1, sobre o cálculo do arco-tangente para as coordenadas de longitudes geodésicas na equação (5.1), a seguir:

$$v = \begin{cases} \operatorname{atan}\left(\frac{Y}{X}\right) & \text{se } X > 0 \wedge Y > 0 \\ \text{se não} \\ 180^\circ + \operatorname{atan}\left(\frac{Y}{X}\right) & \text{se } X < 0 \wedge Y > 0 \\ 180^\circ + \operatorname{atan}\left(\frac{Y}{X}\right) & \text{se } X < 0 \wedge Y < 0 \\ 360^\circ + \operatorname{atan}\left(\frac{Y}{X}\right) & \text{se } X > 0 \wedge Y < 0 \\ 90^\circ & \text{se } X = 0 \wedge Y > 0 \\ 270^\circ & \text{se } X = 0 \wedge Y < 0 \\ 180^\circ & \text{se } X < 0 \wedge Y = 0 \\ 0^\circ & \text{se } X > 0 \wedge Y = 0 \\ 2410^\circ & \text{se } X = 0 \wedge Y = 0 \end{cases} \quad (5.4)$$

Relembra-se que o par de coordenadas curvilíneas (ϕ, λ) é geocêntrico. Foi empregado um recurso para identificação de pontos que estivessem localizados nos pólos norte ou sul. Caso isso ocorresse, o valor de v seria 2410° . Esse artifício será entendido mais adiante, mas adianta-se que com ele será possível determinar um único valor de longitude a ser utilizado em um ponto que coincida com um pólo, ao invés de qualquer valor, como poderia se imaginar, devido à convergência meridiana.

Carrega-se a matriz Geodésicos, de número de linhas n igual ao número de pontos da poligonal, e número de colunas igual a dois, conforme abaixo. A primeira coluna da matriz possuirá as latitudes geodésicas, e a segunda coluna, as longitudes, conforme abaixo:

$$\text{Geodésicos}_{n,0} = u_n \quad (5.5)$$

$$\text{Geodésicos}_{n,1} = v_n \quad (5.6)$$

Pode-se observar que as longitudes calculadas acima em (5.4) são necessariamente positivas, variando de 0° a 360° .

4.1.2 Coordenadas de entrada (u,v)

No tópico anterior foi apresentado o encaminhamento para que a massa de dados em (XGPS, YGPS, ZGPS) fosse convertida em (u,v) . Caso o usuário resolvesse entrar com a

massa de dados já em coordenadas curvilíneas geodésicas, haveria apenas a necessidade de preparação das mesmas, de forma que as longitudes variassem no intervalo de 0° a 360° , como no método anterior. Para tanto, carregar-se-ia a matriz Geodésicos diretamente com os dados do arquivo ASCII, conforme formatação já apresentada.

Cria-se um vetor apenas com as longitudes, determina-se o menor valor e diminui-se esse valor de todos os elementos da segunda coluna da matriz Geodésicos, armazenado os novos valores, já positivos, como as longitudes geodésicas a serem empregadas. Esses procedimentos estão abaixo descritos, sendo n o número de pontos da poligonal:

$$\text{Long} = \begin{array}{l} i = 0 \\ \text{para } i = 0 \text{ até } n - 1 \\ A_i = \text{Geodésicos}_{i,1} \\ i = i + 1 \\ \text{fim para} \\ A \end{array} \quad (5.7)$$

$$\text{min} = \begin{array}{l} A = 0 \\ i = 0 \\ \text{para } i = 0 \text{ até } n - 1 \\ A = \text{Long}_n \text{ se } \text{Long}_n < A \\ i = i + 1 \\ \text{fim para} \\ A \end{array} \quad (5.8)$$

$$\text{Geodésicos} = \begin{array}{l} i = 0 \\ \text{para } i = 0 \text{ até } n - 1 \\ A_{i,0} = \text{Geodésicos}_{i,0} \\ A_{i,1} = \text{Geodésicos}_{i,1} - \text{min} \\ i = i + 1 \\ \text{fim para} \\ A \end{array} \quad (5.9)$$

Utilizando o programa Saida2 é possível calcular a distância e o azimute geodésicos entre dois pontos. Assim, são criadas duas funções para discriminação destes cálculos, abaixo descritas:

$$\text{DistGeod}(\phi_1, \lambda_1, \phi_2, \lambda_2) = \text{Saida2}(\phi_1, \lambda_1, \phi_2, \lambda_2)_0 \quad (5.10)$$

$$\text{Azimute}(\phi_1, \lambda_1, \phi_2, \lambda_2) = \text{Saida2}(\phi_1, \lambda_1, \phi_2, \lambda_2)_1 \quad (5.11)$$

Usando a função DistGeod , armazena-se um vetor com as distâncias geodésicas entre os pontos:

$$\text{Distâncias} = \begin{array}{l} i = 0 \\ \text{para } i = 0 \text{ até } n - 1 \\ A_i = \text{DistGeod}(\phi_i, \lambda_i, \phi_{i+1}, \lambda_{i+1}) \\ i = i + 1 \\ \text{fim para} \\ A \end{array} \quad (5.12)$$

O mesmo processo acima é realizado com a função Azimute , armazenando os valores dos azimutes geodésicos em um vetor:

$$\text{Az} = \begin{array}{l} i = 0 \\ \text{para } i = 0 \text{ até } n - 1 \\ A_i = \text{Azimute}(\phi_i, \lambda_i, \phi_{i+1}, \lambda_{i+1}) \\ i = i + 1 \\ \text{fim para} \\ A \end{array} \quad (5.13)$$

4.2 Determinação das coordenadas dos pontos pertencentes a uma linha geodésica

Estando as distâncias e os azimutes geodésicos entre os pontos armazenados nos vetores Distâncias e Az , cumpre apresentar a metodologia para determinação dos pontos que pertencem à linha geodésica que une os vértices consecutivos de uma dada poligonal. A determinação de tais pontos permite criar linhas geodésicas de menor comprimento entre eles. Isso servirá para criar a parametrização elíptica, demonstrada no Capítulo 4, que será empregada para o efetivo cálculo de (4.41). O somatório de todos os triângulos elipsoidais determinará a área geodésica que se deseja conhecer.

Para realizar o adensamento de pontos pertencentes à linha geodésica, empregou-se uma composição dos Métodos Direto e Inverso de Vincenty.

Seja uma linha geodésica partindo do ponto A para o ponto B. Calcula-se a distância geodésica entre A e B, bem como o azimute geodésico.

Para determinar pontos que pertençam à linha geodésica entre A e B, deve-se, a partir das coordenadas geodésicas do ponto A, empregar o Método Direto, admitindo-se que o azimute entre o ponto A e o subsequente a ser determinado será igual ao azimute entre A e B, e a distância deverá ser menor que a calculada entre A e B. Para dividir a linha geodésica em segmentos de praticamente mesmo comprimento, estabeleceu-se um valor de passo, que seria

empregado de acordo com o tamanho da linha geodésica. Tal passo variaria automaticamente entre 20 km até 1 m, atendendo assim poligonais de diferentes comprimentos de lados. Ressalta-se que uma mesma poligonal pode ter passos diferentes, de acordo com o comprimento de seus lados geodésicos.

A metodologia acima descrita gerou uma matriz de coordenadas geodésicas com duas colunas: a primeira para as latitudes; e a segunda para as longitudes. Essa matriz foi chamada de Geodésicos2, presente no Apêndice G. O valor de linPontos é preenchido de acordo com o número de linhas da matriz Geodésicos.

Para verificar o grau de adequação do adensamento de pontos realizado, calcula-se a diferença entre os perímetros das poligonais armazenadas em Geodésicos e Geodésicos2. Para tanto, utiliza-se a equação abaixo, sendo linPontos2 a quantidade de linhas da matriz Geodésicos2:

$$\Delta\text{Perímetro} = \left[\sum_{j=0}^{\text{linPontos}} \text{Distâncias}_i - \sum_{j=0}^{\text{linPontos2}} \text{DistGeod}(\text{Geodésicos2}_{i,0}, \text{Geodésicos2}_{i,1}, \text{Geodésicos2}_{i+1,0}, \text{Geodésicos2}_{i+1,1}) \right] \quad (5.14)$$

Em todos os casos avaliados, o erro cometido é menor que 1 mm, mesmo para poligonais com perímetros superiores a 20.000 km.

A partir das coordenadas geodésicas espaçadas agora de acordo com o passo escolhido para cada lado, devem-se calcular as respectivas coordenadas geocêntricas, sendo estas armazenadas em uma matriz chamada Geocêntricos (Apêndice I).

4.3 Cálculo da área da poligonal geodésica

Com a matriz Geocêntricos preenchida, é possível calcular a área da poligonal geodésica somando-se os valores calculados em (4.41) para todos os pertencentes à referida matriz. Antes, porém, é necessário criar algumas funções. A primeira é chamada de Primitiva40, sendo esta a integração da função descrita em (4.31).

$$\text{Primitiva40}(\varphi) = \sum_{k=0}^{41} \left(\frac{a_k}{k+1} \varphi^{k+1} \right) \quad (5.15)$$

Os termos pares de (5.15) assumem valores nulos. Em seguida, cria-se a função Elipse, baseada em (4.40):

$$\begin{aligned} \text{Elipse}(\phi_1, \lambda_1, \phi_2, \lambda_2, \lambda) &= & (5.16) \\ &= \text{atan}\left(\frac{-\text{coef1}(\phi_1, \lambda_1, \phi_2, \lambda_2) \cos(\lambda) - \text{coef2}(\phi_1, \lambda_1, \phi_2, \lambda_2) \text{sen}(\lambda)}{\text{coef3}(\phi_1, \lambda_1, \phi_2, \lambda_2)}\right) \end{aligned}$$

Relembra-se que os valores dos coeficientes em (5.16) dependem das coordenadas geocêntricas de pares de pontos adjacentes (consecutivos) armazenados na matriz Geocêntricos.

Cria-se então a função s, conforme abaixo:

$$\begin{aligned} s(\phi_1, \lambda_1, \phi_2, \lambda_2, \lambda) &= [\text{Primitiva40}(\text{Elipse}(\phi_1, \lambda_1, \phi_2, \lambda_2, \lambda)) - & (5.17) \\ & - \text{Primitiva40}(-90^\circ)] \end{aligned}$$

Por fim, determina-se a função Simpson, para o cálculo da integral definida em (4.41), conforme abaixo:

$$\text{Simpson}(\phi_1, \lambda_1, \phi_2, \lambda_2) = \begin{cases} n = 10 \\ h = \frac{\lambda_2 - \lambda_1}{n} \\ T = \frac{h}{3} \sum_{i=1}^{n/2} [s(\phi_1, \lambda_1, \phi_2, \lambda_2, \lambda_1 + 2hi) + \\ + 4s[\phi_1, \lambda_1, \phi_2, \lambda_2, \lambda_1 + (2i - 1)h] + \\ + s[\phi_1, \lambda_1, \phi_2, \lambda_2, \lambda_1 + (2i - 2)h]] \\ a^2 T \end{cases} \quad (5.28)$$

Para calcular a área da poligonal geodésica, segue-se o procedimento abaixo discriminado, armazenando seu valor na variável Areas:

$$\begin{array}{l}
\text{Areas} = 0 \\
\text{para } i \text{ de } 0 \text{ até } (\text{linPontos2} - 2) \\
\text{Areas} = \text{Areas} + 0 \text{ se} \\
\text{Azimute}(\text{Geodésicos2}_{i,0}, \text{Geodésicos2}_{i,1}, \\
, \text{Geodésicos2}_{i+1,0}, \text{Geodésicos2}_{i+1,1}) = 0^{\circ} \\
\text{senão } \text{Areas} = \text{Areas} + 0 \text{ se} \\
\text{Azimute}(\text{Geodésicos2}_{i,0}, \text{Geodésicos2}_{i,1}, \\
, \text{Geodésicos2}_{i+1,0}, \text{Geodésicos2}_{i+1,1}) = 180^{\circ} \\
\text{senão } \text{Areas} = \text{Areas} + \\
+\text{Simpson}(\text{Geocêntricos}_{i,0}, \text{Geocêntricos}_{i,1}, \\
, \text{Geocêntricos}_{i+1,0}, \text{Geocêntricos}_{i+1,1}) \text{ se} \\
\text{DistGeod}(\text{Geodésicos2}_{i,0}, \text{Geodésicos2}_{i,1}, \\
, \text{Geodésicos2}_{i+1,0}, \text{Geodésicos2}_{i+1,1}) \neq 0 \\
\text{senão } \text{Areas} = \text{Areas} + 0 \\
\text{Fim para}
\end{array} \tag{5.19}$$

Em (5.19), quando dois pontos adjacentes pertencem a um meridiano, ou seja, seu azimute geodésico é 0° ou 180° , o lado formado por estes pontos terá contribuição nula na área total da poligonal geodésica.

4.4 Adaptação do método para linhas loxodrômicas

O cálculo da área de uma poligonal sobre o elipsóide depende da geometria das linhas que compõem tal poligonal. No item 5.3 foi esclarecido como realizar tal cálculo para uma poligonal composta de linhas geodésicas. Porém, como a cartografia no Brasil está pautada em sistemas projetivos conformes, quer sejam eles a projeção UTM, empregada pelo IBGE e pela DSG, ou a projeção de Mercator, utilizada para construção das cartas náuticas pela DHN e aeronáuticas pelo ICA, existem limites que são traçados como poligonais retas sobre tais sistemas.

Sendo projeções conformes, a manutenção dos ângulos, característica construtiva do modelo matemático empregado, acarreta que tais poligonais retas, delimitadoras de áreas, sejam poligonais loxodrômicas. Portanto, resta apresentar um modelo adaptado para tal condição geométrica.

Para realizar o procedimento de adensamento da linha loxodrômica, seria possível utilizar um princípio semelhante ao empregado na linha geodésica: a combinação dos Métodos Direto e Inverso de Vincenty, porém empregando a manutenção do azimute entre os pontos extremos A e B de forma distinta, sem fixar a origem em A. Começando no primeiro ponto calculado a partir de A, o Método Direto deveria ser empregado com base nesse ponto, usando suas

coordenadas como parâmetros de entrada, bem como o azimute entre A e B e o passo estipulado. Contudo, tal locação partiria de um preceito equivocado, pois estaria pautada na semelhança entre a linha geodésica e a loxodrômica para distâncias curtas, o que pode variar de acordo com a latitude dos pontos. Esta metodologia talvez funcionasse, mas optou-se pela locação dos pontos na projeção de Mercator por ser, aparentemente, mais simples e definitivamente mais confiável. Somente o adensamento dos pontos foi feito sobre a referida projeção, mas o cálculo da área foi realizado sobre o elipsóide.

A projeção de Mercator sobre o elipsóide é de mais fácil modelagem que a UTM, não havendo necessidade de preocupação com fusos, em razão de seu cilindro ser equatorial. O primeiro passo é uma avaliação da amplitude das latitudes dos pontos da poligonal. Ao escolher o valor máximo e o valor mínimo, toma-se a média entre eles e calcula-se a latitude de referência, chamada ϕ_0 . Para determinar as coordenadas planas em Mercator a partir das coordenadas geodésicas, tem-se:

$$X_{\text{merc}}(\lambda) = a\lambda \frac{\cos(\phi_0)}{\sqrt{1 - (e_1 \cdot \sin(\phi_0))^2}} \quad (5.20)$$

$$Y_{\text{merc}}(\phi) = a \cdot \ln \left(\tan \left(\frac{\pi}{4} + \frac{\phi}{2} \right) \left(\frac{1 - e_1 \cdot \sin(\phi)}{1 + e_1 \cdot \sin(\phi)} \right)^{\frac{e_1}{2}} \right) \frac{\cos(\phi_0)}{\sqrt{1 - (e_1 \cdot \sin(\phi_0))^2}} \quad (5.21)$$

O valor de e_1 corresponde à primeira excentricidade. Um artifício empregado para contornar problemas nos pólos foi que se ϕ fosse menor que $-89,999999999999^\circ$, $Y_{\text{merc}}(\phi)$ seria calculado usando este valor para latitude. Analogamente, se ϕ fosse maior que $89,999999999999^\circ$, $Y_{\text{merc}}(\phi)$ seria calculado usando este valor para latitude.

Para determinação das coordenadas geodésicas a partir das coordenadas planas na projeção de Mercator, criaram-se as funções longlox e latlox . A inversão para obter a longitude geodésica do ponto pertencente à linha loxodrômica é bastante simples, como pode se observar em (5.20). Porém, o cálculo da latitude é mais complexo, havendo necessidade de criar uma função Func . Tal função admitiria como raiz o valor a ser encontrado da latitude do ponto. Para escapar da obtenção analítica da raiz de Func , recorre-se ao Método de Newton-Raphson, sendo necessário o cálculo da derivada de Func no ponto de interesse, criando-se a função DerivFunc . Definem-se as funções abaixo:

$$\text{Func}(Y_{\text{merc}}, \phi) = e^{\frac{Y_{\text{merc}} \sqrt{1 - (e_1 \cdot \sin(\phi_0))^2}}{a \cdot \cos(\phi_0)}} - \tan\left(\frac{\pi}{4} + \frac{\phi}{2}\right) \left(\frac{1 - e_1 \cdot \sin(\phi)}{1 + e_1 \cdot \sin(\phi)}\right)^{\frac{e_1}{2}} \quad (5.22)$$

$$\text{DerivFunc}(Y_{\text{merc}}, \phi) = \frac{\text{Func}(Y_{\text{merc}}, \phi + 10^{-5}) - \text{Func}(Y_{\text{merc}}, \phi - 10^{-5})}{2 \cdot 10^{-5}} \quad (5.23)$$

$$\text{longlox}(X_{\text{merc}}) = X_{\text{merc}} \frac{\sqrt{1 - (e_1 \cdot \sin(\phi_0))^2}}{a \cdot \cos(\phi_0)} \quad (5.24)$$

$$\text{latlox}(Y_{\text{merc}1}) = \left\{ \begin{array}{l} \phi_{00} = 0 \\ \text{sinal} = 0 \\ \text{se } Y_{\text{merc}1} > 0 \text{ então} \\ Y_{\text{merc}1} = -Y_{\text{merc}1} \\ \text{sinal} = 1 \\ \text{para contador de 1 até 10} \\ \text{parar se } |\text{Func}(Y_{\text{merc}1}, \phi_{00})| < 10^{-15} \\ \phi_{00} = \phi_{00} - \frac{\text{Func}(Y_{\text{merc}1}, \phi_{00})}{\text{DerivFunc}(Y_{\text{merc}1}, \phi_{00})} \\ \text{contador} = \text{contador} + 1 \\ \text{Fim para} \\ \phi_{00} = -90^{\circ} \text{ se } Y_{\text{merc}1} < Y_{\text{merc}}(-89,9999999999999^{\circ}) \\ \phi_{00} = -\phi_{00} \text{ se } \text{sinal} = 1 \\ \text{se não } \phi_{00} = \phi_{00} \end{array} \right. \quad (5.25)$$

Cumpra agora calcular e armazenar as distâncias e os azimutes, calculados sobre a projeção de Mercator, entre os pontos adjacentes que se encontram na matriz Geodésicos. Para tanto, são criados dois vetores, o Distâncias_loxo e o Az_loxo:

$$\text{Distâncias_loxo} = \left\{ \begin{array}{l} \text{para } i \text{ de } 0 \text{ até } \text{linPontos} - 2 \\ \text{Dist} = \left(X_{\text{merc}}(\text{Geodésicos}_{i+1,1}) - X_{\text{merc}}(\text{Geodésicos}_{i,1}) \right)^2 + \\ + \left(Y_{\text{merc}}(\text{Geodésicos}_{i+1,0}) - Y_{\text{merc}}(\text{Geodésicos}_{i,0}) \right)^2 \\ \text{Distâncias_loxo}_i = \sqrt{\text{Dist}} \\ \text{Fim para} \end{array} \right. \quad (5.26)$$

$$\text{Az_loxo} = \left\{ \begin{array}{l} \text{para } i \text{ de } 0 \text{ até } \text{linPontos} - 2 \\ \text{Az_loxo}_i = \text{atan} \left(\frac{Y_{\text{merc}}(\text{Geodésicos}_{i+1,0}) - Y_{\text{merc}}(\text{Geodésicos}_{i,0})}{X_{\text{merc}}(\text{Geodésicos}_{i+1,1}) - X_{\text{merc}}(\text{Geodésicos}_{i,1})} \right) \\ \text{Fim para} \end{array} \right. \quad (5.27)$$

Face à modelagem em (5.27), é necessário que seja considerado o exposto na equação (5.4), quando do cálculo do arco-tangente. Apenas a última condição, que cria o valor de 2410° deve ser desconsiderada em (5.27).

De forma análoga à que foi empregada para criação da matriz Geodésicos2, determina-se a matriz Loxodrômicos, a qual armazenará as coordenadas dos pontos adensados pertencentes aos lados da poligonal loxodrômica (Apêndice H).

Com a matriz Loxodrômicos preenchida, cumpre calcular as coordenadas geocêntricas homólogas na matriz Geocêntricos_Loxo (Apêndice J), de forma análoga à matriz Geocêntricos. O valor de linPontos3 equivale ao número de linhas da matriz Loxodrômicos.

Por fim, para calcular a área da poligonal loxodrômica usando uma metodologia semelhante à empregada no cálculo da variável Areas, cria-se Areas_loxo:

$$\begin{array}{l}
 \text{Areas_loxo} = 0 \\
 \text{para } i \text{ de } 0 \text{ até } (\text{linPontos3} - 2) \\
 \text{Areas_loxo} = \text{Areas_loxo} + \\
 \text{+Simpson}(\text{Geocêntricos_loxo}_{i,0}, \text{Geocêntricos_loxo}_{i,1}, \\
 , \text{Geocêntricos_loxo}_{i+1,0}, \text{Geocêntricos_loxo}_{i+1,1}) \\
 \text{fim para}
 \end{array} \tag{5.28}$$

5 RESULTADOS OBTIDOS

O primeiro teste realizado foi para cinco pontos sobre a linha do Equador, igualmente espaçados. As coordenadas conhecidas destes pontos estão representadas na tabela abaixo:

Tabela 4 – Coordenadas dos Pontos que Compõem a Poligonal 1

Pontos	X	Y	Z
P1	6378137	0	0
P2	1970952,725	6065968,756	0
P3	-5160021,225	3748974,866	0
P4	-5160021,225	-3748974,866	0
P5	1970952,725	-6065968,756	0

As coordenadas na Tabela 4 expressam pontos separados de 72° . Como a linha do Equador é uma linha geodésica, é possível calcular a área de um hemisfério do elipsóide de referência ao considerar a poligonal composta pelos pontos acima como fechada. Para tanto, um último ponto, P6, terá coordenadas iguais ao ponto P1.

Com base no resultado obtido pelas equações (4.7) e (4.29), a área de metade do elipsóide utilizado é igual a $255032810862044,22 \text{ m}^2$. Calculando a mesma área com a metodologia da dissertação, obtém-se $255032810862034,44 \text{ m}^2$.

Para uma área equivalente à metade do planeta, obtém-se precisão de 38 ppq (partes por quatrilhão), sendo calculada em menos de dois segundos na estação de trabalho empregada para a modelagem. Lembra-se que o SGBD ORACLE SPATIAL 11g apresenta um erro de 0,1% ao calcular a mesma área por meio de seu SDO_AREA.

Como a linha do Equador também é uma linha loxodrômica, a mesma metodologia foi aplicada para uma poligonal loxodrômica, obtendo-se valor igual a $255032810862034,22 \text{ m}^2$. Tal resultado confere precisão de 39 ppq.

Para melhor amparar a avaliação dos resultados, compararam-se as áreas das poligonais geodésicas calculadas com os resultados oferecidos pela solução *Hydrographic Production Database* (HPD) da empresa CARIS. O módulo geodésico fornecido pela empresa é de grande precisão, e foi muito utilizado quando do projeto Levantamento da Plataforma Continental (LEPLAC). Foram criadas algumas outras poligonais geodésicas, discriminadas a seguir, tendo suas áreas comparadas com as calculadas pela solução HPD.

Tabela 5 – Coordenadas dos Pontos que Compõem a Poligonal 2

Pontos	Latitude	Longitude
P1	0°	-66°
P2	0°	-65°
P3	-1°	-65°
P4	-1°	-66°
P5	0°	-66°

Tabela 6 – Coordenadas dos Pontos que Compõem a Poligonal 3

Pontos	Latitude	Longitude
P1	-68°	-66°
P2	-68°	-65°
P3	-67°	-65°
P4	-67°	-66°
P5	-68°	-66°

Tabela 7 – Coordenadas dos Pontos que Compõem a Poligonal 4

Pontos	Latitude	Longitude
P1	9°	-53,68°
P2	9°	-52°
P3	-38°	-52°
P4	-38°	-53,68°
P5	9°	-53,68°

Tabela 8 – Coordenadas dos Pontos que Compõem a Poligonal 5

Pontos	Latitude	Longitude
P1	28,6362365°	0°
P2	-21,4366965°	64,2086651°
P3	4,9898034°	119,491462°
P4	33,8288302°	100,7759318°
P5	19,6663563°	77,7414331°
P6	37,5849405°	65,0724588°
P7	40,7225762°	33,4000231°
P8	28,6362365°	0°

Tabela 9 – Coordenadas dos Pontos que Compõem a Poligonal 6

Pontos	Latitude	Longitude
P1	3,5524312°	1,9731329°
P2	0,7075042°	3,6174104°
P3	-1,6097656°	6,1824832°
P4	-2,5361417°	9,0764115°
P5	-3,6601003°	12,2991953°
P6	-4,518658°	15,0615814°
P7	-6,298307°	16,6400878°
P8	-8,4648974°	17,6266542°
P9	-10,4239475°	16,6400878°
P10	-12,8873274°	14,4696415°
P11	-14,6231097°	14,4696415°
P12	-17,2320944°	13,7461595°
P13	-20,6114609°	11,7730265°
P14	-23,1890778°	9,0764115°
P15	-24,5198541°	6,5113387°
P16	-26,2524215°	3,5516393°
P17	-28,5416559°	3,2227838°
P18	-30,6680735°	1,5127352°
P19	-32,9703044°	0,3288555°
P20	-35,1595275°	0°
P21	-36,3922205°	4,2751213°
P22	-35,9121597°	6,5113387°
P23	-33,6880531°	8,0240739°
P24	-31,4607051°	9,7341225°
P25	-29,8114799°	11,7072554°
P26	-27,6668636°	12,3649664°
P27	-26,311713°	15,2588947°
P28	-25,5385499°	15,9823768°
P29	-23,9771183°	18,2843652°
P30	-23,3106254°	20,6521247°
P31	-25,2398229°	23,8091374°
P32	-25,0003052°	27,1634634°
P33	-22,579672°	30,0573918°
P34	-19,9287644°	30,188934°
P35	-17,6109279°	30,1231629°
P36	-15,9000714°	28,4131143°
P37	-15,0709373°	25,9138126°
P38	-14,4949876°	22,5594866°
P39	-12,4998773°	21,9017756°
P40	-10,228564°	23,4145109°
P41	-7,612689°	24,4668484°
P42	-6,2324934°	25,5849571°
P43	-4,1225167°	26,3084392°
P44	-1,4773856°	28,2815721°
P45	1,8328497°	28,9392831°
P46	4,3450857°	27,8869455°
P47	5,3346998°	25,2561016°
P48	4,6751182°	22,16486°
P49	2,0313782°	20,6521247°
P50	0,2440211°	19,5340161°
P51	0,7737129°	17,2977987°
P52	1,7666684°	15,0615814°
P53	2,362203°	12,6938219°
P54	3,6845924°	11,7730265°
P55	5,3346998°	9,7341225°
P56	6,7173811°	8,4844716°
P57	7,4401089°	6,4455676°
P58	7,8338238°	4,9986034°
P59	7,0460357°	3,6174104°
P60	3,5524312°	1,9731329°

Tabela 10 – Coordenadas dos Pontos que Compõem a Poligonal 7

Pontos	Latitude	Longitude
P1	0°	0°
P2	0°	11°
P3	-90°	11°
P4	-90°	0°
P5	0°	0°

Todas as poligonais acima foram criadas para verificar o grau de aderência entre o método geodésico de cálculo de áreas utilizado na plataforma CARIS e o ora apresentado nesta dissertação. A Poligonal 2 foi criada para verificar as discrepâncias em uma poligonal geodésica fechada, de quatro lados iguais a 1° , próxima ao Equador.

A Poligonal 3, baseada na 2, foi elaborada para verificar o comportamento do modelo em latitude mais próxima do Pólo Sul. A Poligonal 4 foi montada com base em um dos valores de área calculados no LEPLAC.

Ainda com base no LEPLAC, foi criada a Poligonal 6, para estimar o grau de aderência entre os valores calculados no CARIS e a metodologia da dissertação. Já a Poligonal 5 foi estruturada para avaliar o comportamento do cálculo de uma área de dimensões continentais.

Por fim, a Poligonal 7 objetivou o avaliar o comportamento do CARIS em relação à área de metade de um fuso de 11° , comparando a precisão da solução comercial e a do modelo da dissertação a um aferidor externo, por meio de adaptação das equações (4.7) e (4.29), cujo valor calculado por elas foi de $7792669220784,69 \text{ m}^2$. Abaixo encontra-se uma tabela que compara os valores de áreas calculados das poligonais 2 a 6 pela metodologia da dissertação, com passo ajustável até 20 quilômetros, os valores calculados pelo CARIS, as precisões internas entre os passos de 200 metros e 20 quilômetros, e as precisões entre o CARIS e a metodologia da dissertação.

Tabela 11 – Cálculo das Áreas das Poligonais e Respectivas Precisões

Poligonal	Área da dissertação	Área pelo CARIS	Precisão entre os passos	CARIS versus dissertação
2	12308778361,46	12308778798,98	0,43ppt	35,55ppb
3	4764521310,60	4764521767,77	22,62ppb	95,95ppb
4	916107764479,79	916107767925,56	4,36ppb	3,76ppb
5	4718727237071,78	47186908662362,30	3,93ppb	7,70ppm
6	6392929465950,95	6392929359041,24	4,65ppb	16,72ppb

Os valores de áreas expressos na Tabela 11 estão em m^2 , ppt significa partes por trilhão, ppb partes por bilhão e ppm partes por milhão. Ao observar os valores das áreas na Tabela 11, bem como as precisões entre o método da solução CARIS e da metodologia da dissertação, fica bastante claro que ambos convergem para soluções praticamente idênticas.

Em relação à simplificação do passos utilizados na dissertação, a penúltima coluna mostra que o adensamento de coordenadas com passo de 200 metros meramente iria aumentar o custo de processamento, não gerando ganho significativo na precisão final do produto. Apesar da Tabela 11 indicar o grande grau de aderência entre a plataforma CARIS e a metodologia da dissertação, cumpre aferir a precisão relativa de ambos contra uma área de valor conhecido, como é o caso da Poligonal 7.

Tabela 12 – Resultados Obtidos para Poligonal 7

Métodos	Area	Precisão em relação à área de meio fuso de 11°	Precisão entre os passos
Dissertação	7792669220784,71	3,38ppq	43ppq
CARIS	7792666987951,76	0,29ppm	-
Integração	7792669220784,69	0	-

Ao avaliar a Tabela 12, é possível verificar que a área calculada pela metodologia da dissertação é mais precisa que a fornecida pelo CARIS, além de destacar, mais uma vez, a precisão interna de resultados, a despeito do emprego de um passo de 200 metros ou de 20 quilômetros, o que garante diminuição sobremaneira do tempo de processamento, sem perda da precisão final.

Por fim, a título de analisar o comportamento da metodologia para uma área muito pequena, criou-se a Poligonal 8.

Tabela 13 – Coordenadas dos Pontos que Compõem a Poligonal 8

Pontos	Latitude	Longitude
P1	0°	0°
P2	0°	0,0001°
P3	0,0001°	0,0001°
P4	0,0001°	0°
P5	0°	0°

A fim de calcular essa mesma área sobre uma projeção equivalente, utilizou-se o recurso do Geocalc 3.09, para calcular os valores de E e N na projeção cilíndrica equivalente, com paralelo padrão e meridiano central iguais a 0°. A utilização direta da projeção é devido ao fato das linhas geodésicas da Poligonal 8 serem tão pequenas, que sua locação como linha reta na projeção não acarreta em erros de avaliação. Assim, a tabela abaixo armazena os valores dos pontos da Poligonal 8 na referida projeção:

Tabela 14 – Coordenadas dos Pontos da Poligonal 8 Projetados

Pontos	N(m)	E(m)
P1	0	0
P2	0	11,13194907932736
P3	11,05742758196920	11,13194907932736
P4	11,05742758196920	0
P5	0	0

Empregando a equação (2.51), obteve-se 123,09072079083099 m². Empregando-se a metodologia, obteve-se 123,09072080254555m². Assim, a precisão interna encontrada foi de

95,17ppt. Dessa forma, fica comprovada a eficácia do método para poligonais com áreas de quase quaisquer dimensões.

6 CONCLUSÕES E RECOMENDAÇÕES

Por meio dos procedimentos apresentados ao longo desta dissertação, forneceu-se um método para calcular as áreas geodésicas diretamente sobre o elipsóide de referência. Assim, a determinação das áreas de imóveis, conforme rege a lei, se torna inequívoca. A implementação prática apresentada no Apêndice C, desenvolvida em Visual Basic 2008, permite que o código seja mais facilmente adaptado a qualquer Sistema de Informações Geográficas. Abaixo, encontra-se uma imagem da interface criada para o cálculo das áreas.

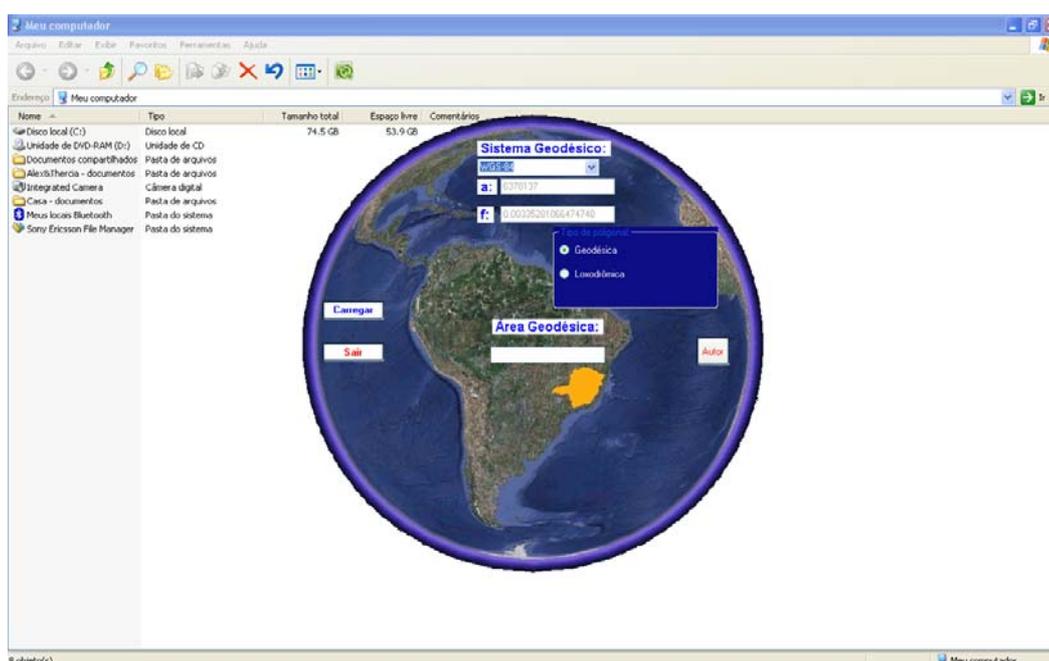


Figura 8 – Aplicativo AreaCalc Desenvolvido na Dissertação

Os resultados alcançados nesta dissertação foram bastante consistentes, porém algumas recomendações para trabalhos futuros podem ser evidenciadas:

- a) como o trabalho foi desenvolvido baseado no Sistema Geodésico WGS-84, seria interessante que o fosse adaptada a metodologia apresentada para o SIRGAS2000, possibilitando a criação de regulamentação para o cálculo de áreas com base no SGB. Cabe ressaltar que a referida adaptação seria apenas para consolidar uma regulamentação legislativa, uma vez que a diferença entre os Sistemas Geodésicos não implicaria em variação significativa das áreas calculadas. Isso facilitaria na tarifação decorrente do cadastro de imóveis rurais e urbanos em bases de dados, uma vez que um único método seria suficiente para definir desde pequenas propriedades até limites de países ou continentes;

- b) outro ponto interessante seria a investigação do problema direto da Geodésia, usando a formulação modificada de Kivioja (Featherstone, Thomas, 2005). Apesar deste autor não acreditar em variações do resultado final, seria um exercício interessante validar o método de adensamento de pontos das linhas geodésicas e verificar o resultado final no cálculo da área da poligonal;
- c) foi verificado que o executável AreaCalc aloca grande quantidade de memória. Provavelmente, isso é devido à declaração dos vetores e matrizes, cujo dimensionamento foi feito para inserção de poligonais de até dez mil pontos. Esse dimensionamento poderia ser reestruturado dinamicamente, de forma a otimizar o gerenciamento de memória pelo aplicativo;
- d) por razões construtivas, o executável AreaCalc somente funciona caso as Opções Regionais estejam configuradas para Inglês e só funciona no Sistema Operacional Windows XP. Uma reestruturação do aplicativo poderia compatibilizá-lo com o Windows Vista, bem como executar os cálculos com as Opções Regionais em Português;
- e) as estimativas de erros foram baseadas em avaliações internas do modelo matemático, por meio da variação do valor de seu passo, comparação com a solução comercial da empresa CARIS, e validação de algumas áreas cujos valores eram passíveis de cálculos algébricos. Em prol de melhorar a estimativa do erro propagado, seria muito interessante o desenvolvimento de uma metodologia que relacionasse o modelo matemático apresentado e a precisão fornecida na determinação de um ponto por um rastreador GPS. Seria possível apresentar um produto que além de calcular a área de uma dada região, poderia estimar o erro total propagado em função da qualidade da aquisição dos dados levantados. Mesmo que este indicador de qualidade fosse armazenado como metadado, ele seria de crucial importância para decisões judiciais.

REFERÊNCIAS

- APOSTOL, Tom M., Cálculo com Funções de uma Variável, com uma Introdução à Álgebra Linear. vol. 1, Ed. Reverté Ltda, 1968.
- APOSTOL, Tom M., Calculus Multi-Variable Calculus and Linear Algebra, with Applications to Differential Equations and Probability. vol. 2, Ed. Wiley International, 1969.
- Blue Mable Geographics, PolygonArea Method, Disponível em: <http://www.bluemarblegeo.com/knowledgebase/geocalc/classdef/ellipsoid/methods/polygonarea_method.htm>. Acesso em: 7 dez 2008.
- GALO, Maurício, MONICO, João F. G., OLIVEIRA, Leonardo C., Cálculo de Áreas de Polígonos sobre o Elipsóide usando Projeções Equivalentes. Curitiba, PR, III Colóquio Brasileiro de Ciências Geodésicas, 2003.
- HALVORSON, Michael, Microsoft Visual Basic 2008, Step by Step, Microsoft Press, 2008.
- IBGE, Frequently Asked Questions – Perguntas mais Frequentes, Disponível em: <<http://www.ibge.gov.br/home/geociências/geodesia/pmrg/faq.shtm>>. Acesso em: 5 jul 2009.
- LÉVESQUE, Serge, COCKBURN, Sara, MCLEAY, Cameron, Modern Development in Geospatial Management in the Field of Marine Cadastre. ABLOS 2008. Disponível em: <http://www.ihp-ohi.net/mtg_docs/com_wg/ABLOS/ABLOS_Conf5/Abstracts/Session6-Paper1-Levesque.pdf>. Acesso em: 10 ago. 2009.
- MANSFIELD, Richard, Visual Basic 2005 Express Edition for Dummies, Ed Wiley Publishing, Inc, 2006.
- MATLAB, AREAIN, Disponível em: <<http://www.mathworks.com/access/helpdesk/help/toolbox/map/index.html?/access/helpdesk/help/toolbox/map/areaint.html>>. Acesso em: 25 maio 2009.
- MELLO, Mauro Pereira de. Teoria das Deformações – Elementos de Geometria Diferencial. Rio de Janeiro: Faculdade de Engenharia da UERJ. 2001. il. Notas de Aula.
- PINTO, Diomara, MORGADO, Maria Cândida Ferreira, Cálculo Diferencial e Integral de Funções de Várias Variáveis, Editora UFRJ/SR-1, 2000.
- RAPP, Richard H., Geometric Geodesy Part 1 - The Ohio State University, Columbus, Ohio, 1991.
- SANTOS, Vitoriano Ruas de Barros, Curso de Cálculo Numérico, Ed Ao Livro Técnico S.A., 1972
- THOMAS, C. M., FEATHERSTONE, W. E., Validation of Vincenty's Formulas for the Geodesic Using a New Fourth-Order Extension of Kivioja's Formula, vol. 131, No. 1, pp. 20-26, 2005. Disponível em : <<http://www.cage.curtin.edu.au/~will/thomas-featherstone.pdf>>. Acesso em: 15 jul 2009.

VINCENY, T., Direct and Inverse Solutions of Geodesics on the Ellipsoid with application of nested equations, Survey Review, XXII, 176, p.88-93, 1975.

WELLS, D. E., KRAKIWSKY, E. J., The Method of Least Squares, Department of Geodesy and Geomatics Engineering, University of New Brunswick, 1997. Lectures Notes 18.

APÊNDICE A – Função Saída2

Função Saída2

variável a como inteiro

variável f como dupla precisão

variável b como dupla precisão

variável φ_1 como dupla precisão

variável φ_2 como dupla precisão

variável λ_1 como dupla precisão

variável λ_2 como dupla precisão

variável L como dupla precisão

variável tanU1 como dupla precisão

variável tanU2 como dupla precisão

variável cosU1 como dupla precisão

variável cosU2 como dupla precisão

variável senU1 como dupla precisão

variável senU2 como dupla precisão

variável λ como dupla precisão

variável λ' como dupla precisão

variável $\text{sen}\sigma$ como dupla precisão

variável $\text{cos}\sigma$ como dupla precisão

variável σ como dupla precisão

variável $\text{sen}\alpha$ como dupla precisão

variável $\text{cos}2\alpha$ como dupla precisão

variável $\text{cos}2\sigma_m$ como dupla precisão

variável C como dupla precisão

variável u2 como dupla precisão

variável A como dupla precisão

variável B como dupla precisão

variável $\Delta\sigma$ como dupla precisão

variável s como dupla precisão

variável α_1 como dupla precisão

Se $[(\varphi_1 \neq \varphi_2) \text{ e } (\lambda_1 \neq \lambda_2)]$ ou $[(\lambda_1 \neq \lambda_2) \text{ e } (\varphi_1 = \varphi_2)]$ ou $[(\varphi_1 \neq \varphi_2) \text{ e } (\lambda_1 = \lambda_2)]$ então

Se $\lambda_1 = 2410$ então $\lambda_1 = \lambda_2$

Se $\lambda_2 = 2410$ então $\lambda_2 = \lambda_1$

$$b = a(1 - f)$$

$$L = \lambda_2 - \lambda_1$$

$$\tan U_1 = (1 - f)\tan(\varphi_1)$$

$$\tan U_2 = (1 - f)\tan(\varphi_2)$$

$$\cos U_1 = \frac{1}{\sqrt{1 + \tan^2 U_1}}$$

$$\sin U_1 = \tan U_1 \cos U_1$$

$$\cos U_2 = \frac{1}{\sqrt{1 + \tan^2 U_2}}$$

$$\sin U_2 = \tan U_2 \cos U_2$$

$$\lambda = L$$

$$\lambda' = 2\pi$$

$$\text{enquanto } |\lambda - \lambda'| > 10^{-20}$$

$$\sin \sigma = \sqrt{(\cos U_2 \sin(\lambda))^2 + (\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos(\lambda))^2}$$

$$\cos \sigma = \sin U_1 \sin U_2 + \cos U_1 \cos U_2 \cos(\lambda)$$

$$\sigma = \text{atan}\left(\frac{\sin \sigma}{\cos \sigma}\right)$$

$$\sin \alpha = \frac{\cos U_1 \cos U_2 \sin(\lambda)}{\sin(\sigma)}$$

$$\cos 2\alpha = 1 - \sin^2 \alpha$$

$$\cos 2\sigma m = \cos(\sigma) - \frac{2 \sin U_1 \sin U_2}{\cos 2\alpha}$$

$$C = \frac{f}{16} \cos 2\alpha (4 + f(4 - 3(\cos 2\alpha)))$$

$$\lambda' = \lambda$$

$$\lambda = L + (1 - C) f \sin \alpha \left(\sigma + C \sin(\sigma) \left(\cos 2\sigma m + C \cos(\sigma) (-1 + 2(\cos 2\sigma m^2)) \right) \right)$$

fim enquanto

$$u_2 = \left(1 - \left(\frac{\cos U_1 \cos U_2 \sin(\lambda)}{\sin(\sigma)} \right)^2 \right) \left(\frac{a^2 - b^2}{b^2} \right)$$

$$A = 1 + \frac{u_2}{16384} (4096 + u_2(-768 + u_2(320 - 175u_2)))$$

$$B = \frac{u_2}{1024} (256 + u_2(-128 + u_2(74 - 47u_2)))$$

$$\Delta \sigma = B \sin(\sigma) \left(\cos 2\sigma m + \frac{B}{4} (\cos(\sigma) (-1 + 2(\cos 2\sigma m^2))) - \right)$$

$$-\frac{B}{6} \cos 2\sigma m \left((-3 + 4\sin(\sigma)^2) (-3 + 4\cos 2\sigma m^2) \right) \right)$$

$$s = b A (\sigma - \Delta\sigma)$$

$$\alpha_1 = \operatorname{atan} \left(\frac{\cos U_2 \sin(\lambda)}{\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos(\lambda)} \right)$$

Senão

s=0

Fim se

Fim Saida2.

APÊNDICE B – Função Saída1

Função Saída1

variável a como inteiro

variável f como dupla precisão

variável b como dupla precisão

variável φ_1 como dupla precisão

variável φ_2 como dupla precisão

variável λ_1 como dupla precisão

variável λ_2 como dupla precisão

variável L como dupla precisão

variável tanU1 como dupla precisão

variável cosU1 como dupla precisão

variável senU1 como dupla precisão

variável σ_1 como dupla precisão

variável λ como dupla precisão

variável λ' como dupla precisão

variável $\text{sen}\sigma$ como dupla precisão

variável $\text{cos}\sigma$ como dupla precisão

variável σ como dupla precisão

variável σ' como dupla precisão

variável $\text{sen}\alpha$ como dupla precisão

variável $\text{cos}\alpha$ como dupla precisão

variável $\text{cos}2\alpha$ como dupla precisão

variável $\text{cos}2\sigma_m$ como dupla precisão

variável C como dupla precisão

variável u2 como dupla precisão

variável A como dupla precisão

variável B como dupla precisão

variável $\Delta\sigma$ como dupla precisão

variável s como dupla precisão

variável α_1 como dupla precisão

$$b = a(1 - f)$$

$$\tan U_1 = (1 - f)\tan(\varphi_1)$$

$$\cos U1 = \frac{1}{\sqrt{1+\tan U1^2}}$$

$$\text{sen}U1 = \tan U1 \cos U1$$

$$\sigma1 = \text{atan}\left(\frac{\tan U1}{\cos(\alpha1)}\right)$$

$$\text{sen}\alpha = \cos U1 \text{sen}(\alpha1)$$

$$\cos 2\alpha = 1 - \text{sen}\alpha^2$$

$$\cos\alpha = \sqrt{1 - \text{sen}\alpha^2}$$

$$u2 = \left(1 - (\cos U1 \text{sen}(\alpha1))^2\right) \left(\frac{a^2 - b^2}{b^2}\right)$$

$$A = 1 + \frac{u2}{16384} \left(4096 + u2(-768 + u2(320 - 175u2))\right)$$

$$B = \frac{u2}{1024} \left(256 + u2(-128 + u2(74 - 47u2))\right)$$

$$\sigma = \frac{s}{bA}$$

$$\sigma' = 2\pi$$

$$\text{enquanto } |\sigma - \sigma'| > 10^{-12}$$

$$\cos 2\sigma m = \cos(2\sigma1 + \sigma)$$

$$\Delta\sigma = B \text{sen}(\sigma) \left(\cos 2\sigma m + \frac{B}{4} (\cos(\sigma) (-1 + 2(\cos 2\sigma m^2)) - \frac{B}{6} \cos 2\sigma m ((-3 + 4\text{sen}(\sigma)^2) (-3 + 4\cos 2\sigma m^2))) \right)$$

$$\sigma' = \sigma$$

$$\sigma = \frac{s}{bA} + \Delta\sigma$$

fim enquanto

$$\phi2 = \text{atan}\left(\frac{\text{sen}U1 \cos(\sigma) + \cos U1 \text{sen}(\sigma) \cos(\alpha1)}{(1-f)\sqrt{(\text{sen}\alpha)^2 + (\text{sen}U1 \text{sen}(\sigma) - \cos U1 \cos(\sigma) \cos(\alpha1))^2}}\right)$$

$$\lambda = \text{atan}\left(\frac{\text{sen}(\sigma) \text{sen}(\alpha1)}{\cos U1 \cos(\sigma) - \text{sen}U1 \text{sen}(\sigma) \cos(\alpha1)}\right)$$

$$C = \frac{f}{16} \cos 2\alpha (4 + f(4 - 3(\cos 2\alpha)))$$

$$L = \lambda - (1 - C) f \text{sen}\alpha \left(\sigma + C \text{sen}(\sigma) (\cos 2\sigma m + C \cos(\sigma) (-1 + 2(\cos 2\sigma m^2))) \right)$$

$$\lambda2 = \lambda1 + L$$

Se $\lambda_2 > 2\pi$ então $\lambda_2 = \left(\frac{\lambda_2}{2\pi} - \text{truncar} \left(\frac{\lambda_2}{2\pi} \right) \right) 2\pi$

Fim Saida1.

APÊNDICE C – Vetor Lat

Vetor Lat

Para contador = 0 até $2*90*3600$

$\text{Lat}[\text{contador}] = \text{contador} * \text{Pi}/(180*3600) - \text{Pi}/2$

contador = contador + 1

Fim Para

APÊNDICE D – Vetor Função

Vetor Função

Para contador = 0 até 2*90*3600

Função[contador] = func(Lat[contador]) - teste(Lat[contador])

contador = contador + 1

Fim Para

APÊNDICE E – Matriz A

Matriz A(grau)

Para contador = 0 até 2*90*3600

contador2 = 34

Para contador2 = 34 até grau

Se (contador2/2) – truncar(contador2/2) <> 0 então

contador2 = contador2 + 1

Fim se

$A[\text{contador}, ((\text{contador2}-34)/2)] = (\text{Lat}[\text{contador}])^{\text{contador2}}$

contador = contador + 1

Fim Para

APÊNDICE F – Função Resultado

Função Resultado

$X = b$

contadorbásico = 0

enquanto contadorbásico < 100

$X_{ini} = X$

$$X = X_{ini} + \frac{-f(X_{ini})}{D(X_{ini})}$$

parar se $|f(X_{ini})| \leq |f(X)|$

contadorbásico = contadorbásico + 1

fim enquanto

$X_2 = X$

reiteraçoofina = 0

enquanto reiteraçoofina \leq 100

contador_{reiteraçoofina} = 0

enquanto contador_{reiteraçoofina} < 100

$X_{ini2} = X_2$

$$X_2 = X_{ini2} + \frac{-f(X_{ini2})}{D(X_{ini2})} 10^{-reiteraçoofina}$$

parar se $|f(X_{ini2})| \leq |f(X_2)|$

contador_{reiteraçoofina} = contador_{reiteraçoofina} + 1

fim enquanto

reiteraçoofina = reiteraçoofina + 1

fim enquanto

fim Função Resultado

APÊNDICE G – Matriz Geodésicos2

Matriz Geodésicos2

$i = 0$

contador = 0

enquanto $i \leq \text{linPontos} - 2$

passo = 20000 se $|\text{Distâncias}_i| > 20000$

se não passo = 10000 se $|\text{Distâncias}_i| > 10000$

se não passo = 5000 se $|\text{Distâncias}_i| > 5000$

se não passo = 1000 se $|\text{Distâncias}_i| > 1000$

se não passo = 500 se $|\text{Distâncias}_i| > 500$

se não passo = 200 se $|\text{Distâncias}_i| > 200$

se não passo = 100 se $|\text{Distâncias}_i| > 100$

se não passo = 50 se $|\text{Distâncias}_i| > 50$

se não passo = 10 se $|\text{Distâncias}_i| > 10$

se não passo = 1

dist = -passo

enquanto $\text{Distâncias}_i - \text{dist} > \text{passo}$

dist = dist + passo se $\text{Distâncias}_i - \text{dist} > 2 \cdot \text{passo}$

se não dist = Distâncias_i

$\alpha = \text{Az}_i$

$\text{Geodésicos2}_{\text{contador},0} = \text{Saida1}(\text{Geodésicos}_{i,0}, \text{Geodésicos}_{i,1}, \alpha, \text{dist})_0$ se

$|\text{Saida1}(\text{Geodésicos}_{i,0}, \text{Geodésicos}_{i,1}, \alpha, \text{dist})_0| > 10^{-15}$

$\text{Geodésicos2}_{\text{contador},0} = \text{Geodésicos}_{i+1,0}$ se dist = Distâncias_i

se não $\text{Geodésicos2}_{\text{contador},0} = 0$

$\text{Geodésicos2}_{\text{contador},1} = \text{Geodésicos}_{i+1,1}$ se $\text{Geodésicos}_{i,1} = 2410^\circ$

Fim se

$\text{Geodésicos2}_{\text{contador},1} = \text{Geodésicos}_{i,1}$ se $\text{Geodésicos}_{i+1,1} = 2410^\circ$

se não

$\text{Geodésicos2}_{\text{contador},1} = \text{Saida1}(\text{Geodésicos}_{i,0}, \text{Geodésicos}_{i,1}, \alpha, \text{dist})_1$ se

$|\text{Saida1}(\text{Geodésicos}_{i,0}, \text{Geodésicos}_{i,1}, \alpha, \text{dist})_1| > 10^{-15}$

$\text{Geodésicos2}_{\text{contador},1} = \text{Geodésicos}_{i+1,1}$ se dist = Distâncias_i

se não $\text{Geodésicos2}_{\text{contador},1} = 0$

```
Geodésicos2contador,1 = 3600 se Geodésicos2contador,1 = 00 ∧ contador > 0 ∧  
∧ Geodésicos2contador-1,1 > 3590  
contador = contador + 1  
fim enquanto  
i = i + 1  
fim enquanto
```

APÊNDICE H – Matriz Loxodrômicos

Matriz Loxodrômicos

$i = 0$

contador = 0

enquanto $i \leq \text{linPontos} - 2$

passo = 20000 se $|\text{Distâncias_loxo}_i| > 20000$

se não passo = 10000 se $|\text{Distâncias_loxo}_i| > 10000$

se não passo = 5000 se $|\text{Distâncias_loxo}_i| > 5000$

se não passo = 1000 se $|\text{Distâncias_loxo}_i| > 1000$

se não passo = 500 se $|\text{Distâncias_loxo}_i| > 500$

se não passo = 200 se $|\text{Distâncias_loxo}_i| > 200$

se não passo = 100 se $|\text{Distâncias_loxo}_i| > 100$

se não passo = 50 se $|\text{Distâncias_loxo}_i| > 50$

se não passo = 10 se $|\text{Distâncias_loxo}_i| > 10$

se não passo = 1

dist = $-\text{passo}$

enquanto $\text{Distâncias_loxo}_i - \text{dist} > \text{passo}$

dist = dist + passo se $\text{Distâncias_loxo}_i - \text{dist} > 2 \cdot \text{passo}$

se não dist = Distâncias_loxo_i

$\alpha = \text{Az_loxo}_i$

$\text{Loxodrômicos}_{\text{contador},0} = \text{latlox}(\text{Ymerc}(\text{Geodésicos}_{i,0} + \text{dist} \cdot \cos(\alpha)))$ se

$|\text{latlox}(\text{Ymerc}(\text{Geodésicos}_{i,0} + \cos(\alpha)))| > 10^{-15}$

$\text{Loxodrômicos}_{\text{contador},0} = \text{Geodésicos}_{i+1,0}$ se dist = Distâncias_loxo_i

se não $\text{Loxodrômicos}_{\text{contador},0} = 0$

$\text{Loxodrômicos}_{\text{contador},1} = \text{Geodésicos}_{i+1,1}$ se $\text{Geodésicos}_{i,1} = 2410^\circ$

Fim se

$\text{Loxodrômicos}_{\text{contador},1} = \text{Geodésicos}_{i,1}$ se $\text{Geodésicos}_{i+1,1} = 2410^\circ$

se não

$\text{Loxodrômicos}_{\text{contador},1} = \text{longlox}(\text{Xmerc}(\text{Geodésicos}_{i,1} + \text{dist} \cdot \sin(\alpha)))$ se

$|\text{longlox}(\text{Xmerc}(\text{Geodésicos}_{i,1} + \text{dist} \cdot \sin(\alpha)))| > 10^{-15}$

$\text{Loxodrômicos}_{\text{contador},1} = \text{Geodésicos}_{i+1,1}$ se dist = Distâncias_loxo_i

se não $\text{Loxodrômicos}_{\text{contador},1} = 0$

$\text{Loxodrômicos}_{\text{contador},1} = 360^0$ se $\text{Loxodrômicos}_{\text{contador},1} = 0^0 \wedge \text{contador} > 0 \wedge$

$\wedge \text{Loxodrômicos}_{\text{contador}-1,1} > 359^0$

$\text{contador} = \text{contador} + 1$

fim enquanto

$i = i + 1$

fim enquanto

APÊNDICE I – Matriz Geocêntricos

Matriz Geocêntricos

$i = 0$

enquanto $i \leq \text{linPontos2} - 1$

$$\text{Geocêntricos}_{\text{contador},0} = \text{atan} \left[\left(\frac{b}{a} \right)^2 \tan(\text{Geodésicos2}_{\text{contador},0}) \right]$$

$$\text{Geocêntricos}_{\text{contador},1} = \text{Geodésicos2}_{\text{contador},1}$$

$i = i + 1$

fim enquanto

APÊNDICE J – Matriz Geocêntricos_loxo

Matriz Geocêntricos_loxo

i = 0

enquanto i ≤ linPontos3 – 1

$$\text{Geocêntricos_lox0}_{\text{contador},0} = \text{atan} \left[\left(\frac{b}{a} \right)^2 \tan(\text{Loxodrômicos}_{\text{contador},0}) \right]$$

$$\text{Geocêntricos_lox0}_{\text{contador},1} = \text{Geodésicos2}_{\text{contador},1}$$

i = i + 1

fim enquanto

APÊNDICE K – Código Fonte do Aplicativo AreaCalc

Public Class Form1

Function Deriv(ByVal k As Double, ByVal XGPS As Double, ByVal YGPS As Double, ByVal ZGPS As Double, ByVal a As Double, ByVal b As Double) As Double

$$\text{Deriv} = ((2 * a ^ 2) * ((XGPS ^ 2) / ((a ^ 2) - k) ^ 3)) + ((2 * a ^ 2) * ((YGPS ^ 2) / ((a ^ 2) - k) ^ 3)) + ((2 * b ^ 2) * ((ZGPS ^ 2) / ((b ^ 2) - k) ^ 3))$$

End Function

Function g(ByVal k As Double, ByVal XGPS As Double, ByVal YGPS As Double, ByVal ZGPS As Double, ByVal a As Double, ByVal b As Double) As Double

$$g = (((a * XGPS) / ((a ^ 2) - k)) ^ 2 + ((a * YGPS) / ((a ^ 2) - k)) ^ 2 + ((b * ZGPS) / ((b ^ 2) - k)) ^ 2) - 1$$

End Function

Function Resultado(ByVal XGPS As Double, ByVal YGPS As Double, ByVal ZGPS As Double, ByVal a As Double, ByVal b As Double) As Double

Dim X0 As Double

Dim contadorbásico As Integer

Dim X As Double

Dim Xini As Double

Dim X2 As Double

Dim reiteraçãofina As Integer

Dim contador(100) As Integer

Dim Xini2 As Double

X0 = b

contadorbásico = 0

X = X0

Do While contadorbásico < 100

 Xini = X

 X = Xini + (-g(Xini, XGPS, YGPS, ZGPS, a, b)) / (Deriv(Xini, XGPS, YGPS, ZGPS, a, b))

 If g(Xini, XGPS, YGPS, ZGPS, a, b) <= g(X, XGPS, YGPS, ZGPS, a, b) Then

 contadorbásico = 99

 End If

 contadorbásico = contadorbásico + 1

Loop

```

X2 = X
reiteraçãoфина = 0
Do While reiteraçãoфина < 100
    contador(reiteraçãoфина) = 0
    Do While contador(reiteraçãoфина) < 100
        Xini2 = X2
        X2 = Xini2 + ((-g(Xini, XGPS, YGPS, ZGPS, a, b)) / (Deriv(Xini, XGPS,
YGPS, ZGPS, a, b))) * 10 ^ (-reiteraçãoфина)
        If g(Xini2, XGPS, YGPS, ZGPS, a, b) <= g(X2, XGPS, YGPS, ZGPS, a, b)
Then
            contador(reiteraçãoфина) = 99
        End If
        contador(reiteraçãoфина) = contador(reiteraçãoфина) + 1
    Loop
    reiteraçãoфина = reiteraçãoфина + 1
Loop
Resultado = X2
End Function
Function X(ByVal XGPS As Double, ByVal YGPS As Double, ByVal ZGPS As
Double, ByVal a As Double, ByVal b As Double) As Double
    X = XGPS / (1 - (Resultado(XGPS, YGPS, ZGPS, a, b) / (a ^ 2)))
End Function
Function Y(ByVal XGPS As Double, ByVal YGPS As Double, ByVal ZGPS As
Double, ByVal a As Double, ByVal b As Double) As Double
    Y = YGPS / (1 - (Resultado(XGPS, YGPS, ZGPS, a, b) / (a ^ 2)))
End Function
Function Z(ByVal XGPS As Double, ByVal YGPS As Double, ByVal ZGPS As
Double, ByVal a As Double, ByVal b As Double) As Double
    Z = ZGPS / (1 - (Resultado(XGPS, YGPS, ZGPS, a, b) / (b ^ 2)))
End Function
Function v(ByVal XGPS As Double, ByVal YGPS As Double, ByVal ZGPS As
Double, ByVal a As Double, ByVal b As Double) As Double
    If X(XGPS, YGPS, ZGPS, a, b) > 0 And Y(XGPS, YGPS, ZGPS, a, b) > 0 Then
        v = Math.Atan(Y(XGPS, YGPS, ZGPS, a, b) / X(XGPS, YGPS, ZGPS, a, b))
    Else

```

```

If X(XGPS, YGPS, ZGPS, a, b) < 0 And Y(XGPS, YGPS, ZGPS, a, b) > 0 Then
    v = Math.PI + Math.Atan(Y(XGPS, YGPS, ZGPS, a, b) / X(XGPS, YGPS,
ZGPS, a, b))
End If
If X(XGPS, YGPS, ZGPS, a, b) < 0 And Y(XGPS, YGPS, ZGPS, a, b) < 0 Then
    v = Math.PI + Math.Atan(Y(XGPS, YGPS, ZGPS, a, b) / X(XGPS, YGPS,
ZGPS, a, b))
End If
If X(XGPS, YGPS, ZGPS, a, b) > 0 And Y(XGPS, YGPS, ZGPS, a, b) < 0 Then
    v = 2 * Math.PI + Math.Atan(Y(XGPS, YGPS, ZGPS, a, b) / X(XGPS, YGPS,
ZGPS, a, b))
End If
If X(XGPS, YGPS, ZGPS, a, b) = 0 And Y(XGPS, YGPS, ZGPS, a, b) > 0 Then
    v = Math.PI / 2
End If
If X(XGPS, YGPS, ZGPS, a, b) = 0 And Y(XGPS, YGPS, ZGPS, a, b) < 0 Then
    v = 3 * Math.PI / 2
End If
If X(XGPS, YGPS, ZGPS, a, b) < 0 And Y(XGPS, YGPS, ZGPS, a, b) = 0 Then
    v = Math.PI
End If
If X(XGPS, YGPS, ZGPS, a, b) > 0 And Y(XGPS, YGPS, ZGPS, a, b) = 0 Then
    v = 0
End If
If X(XGPS, YGPS, ZGPS, a, b) = 0 And Y(XGPS, YGPS, ZGPS, a, b) = 0 Then
    v = 2410
End If
End If
End Function
Function phi(ByVal XGPS As Double, ByVal YGPS As Double, ByVal ZGPS As
Double, ByVal a As Double, ByVal b As Double) As Double
    phi = Math.Asin((Z(XGPS, YGPS, ZGPS, a, b)) / (((X(XGPS, YGPS, ZGPS, a, b))
^ 2) + ((Y(XGPS, YGPS, ZGPS, a, b)) ^ 2) + ((Z(XGPS, YGPS, ZGPS, a, b)) ^ 2)) ^ 0.5))
End Function

```

Function d1(ByVal XGPS As Double, ByVal YGPS As Double, ByVal ZGPS As Double, ByVal a As Double, ByVal b As Double) As Double

$$d1 = (((b \wedge 2) * X(XGPS, YGPS, ZGPS, a, b) / (a \wedge 2)) \wedge 2 + ((b \wedge 2) * Y(XGPS, YGPS, ZGPS, a, b) / (a \wedge 2)) \wedge 2 + (Z(XGPS, YGPS, ZGPS, a, b) \wedge 2) \wedge 0.5$$

End Function

Function d(ByVal XGPS As Double, ByVal YGPS As Double, ByVal ZGPS As Double, ByVal a As Double, ByVal b As Double) As Double

$$d = (a * b) / (((a * \text{Math.Sin}(\text{phi}(XGPS, YGPS, ZGPS, a, b))) \wedge 2) + ((b * \text{Math.Cos}(\text{phi}(XGPS, YGPS, ZGPS, a, b))) \wedge 2)) \wedge 0.5$$

End Function

Function u(ByVal XGPS As Double, ByVal YGPS As Double, ByVal ZGPS As Double, ByVal a As Double, ByVal b As Double) As Double

If Z(XGPS, YGPS, ZGPS, a, b) = 0 Then

u = 0

Else

If X(XGPS, YGPS, ZGPS, a, b) = 0 And Y(XGPS, YGPS, ZGPS, a, b) = 0 And Z(XGPS, YGPS, ZGPS, a, b) < 0 Then

u = -Math.PI / 2

Else

If X(XGPS, YGPS, ZGPS, a, b) = 0 And Y(XGPS, YGPS, ZGPS, a, b) = 0 And Z(XGPS, YGPS, ZGPS, a, b) > 0 Then

u = Math.PI / 2

Else

u = Math.Asin((((d(XGPS, YGPS, ZGPS, a, b)) / (d1(XGPS, YGPS, ZGPS, a, b))) * (Math.Sin(phi(XGPS, YGPS, ZGPS, a, b))))

End If

End If

End If

End Function

Function Distgeod(ByVal phi1 As Double, ByVal lambda1 As Double, ByVal phi2 As Double, ByVal lambda2 As Double, ByVal a As Double, ByVal f As Double) As Double

Dim b As Double

Dim L As Double

Dim tanU1 As Double

Dim tanU2 As Double

Dim cosU1 As Double
 Dim sinU1 As Double
 Dim cosU2 As Double
 Dim sinU2 As Double
 Dim lambda As Double
 Dim lambdalinha As Double
 Dim sinsigma As Double
 Dim cossigma As Double
 Dim sigma As Double
 Dim sinalfa As Double
 Dim cos2alfa As Double
 Dim cos2sigmam As Double
 Dim C As Double
 Dim u2 As Double
 Dim azão As Double
 Dim bezão As Double
 Dim deltasigma As Double
 Dim s As Double
 Dim Az1 As Double

If ((phi1 <> phi2) And (lambda1 <> lambda2)) Or ((phi1 <> phi2) And (lambda1 = lambda2)) Or ((phi1 = phi2) And (lambda1 <> lambda2)) Then

 If lambda1 = 2410 Then

 lambda1 = lambda2

 End If

 If lambda2 = 2410 Then

 lambda2 = lambda1

 End If

 b = a * (1 - (f))

 L = lambda2 - lambda1

 tanU1 = (1 - f) * Math.Tan(phi1)

 tanU2 = (1 - f) * Math.Tan(phi2)

 cosU1 = 1 / ((1 + ((tanU1) ^ 2)) ^ 0.5)

 sinU1 = tanU1 * cosU1

 cosU2 = 1 / ((1 + ((tanU2) ^ 2)) ^ 0.5)

 sinU2 = tanU2 * cosU2

```

lambda = L
lambdalinha = 2 * Math.PI
While Math.Abs(lambda - lambdalinha) > 10 ^ -12
    sinsigma = (((cosU2 * Math.Sin(lambda)) ^ 2) + ((cosU1 * sinU2 - sinU1 *
cosU2 * Math.Cos(lambda)) ^ 2)) ^ 0.5
    cossigma = sinU1 * sinU2 + cosU1 * cosU2 * Math.Cos(lambda)
    If (cossigma > 0) And (sinsigma > 0) Then
        sigma = Math.Atan(sinsigma / cossigma)
    Else
        If (cossigma < 0) And (sinsigma <> 0) Then
            sigma = Math.PI + Math.Atan(sinsigma / cossigma)
        End If
        If (cossigma > 0) And (sinsigma < 0) Then
            sigma = 2 * Math.PI + Math.Atan(sinsigma / cossigma)
        End If
        If (cossigma = 0) And (sinsigma > 0) Then
            sigma = Math.PI / 2
        End If
        If (cossigma = 0) And (sinsigma < 0) Then
            sigma = 3 * Math.PI / 2
        End If
        If (cossigma < 0) And (sinsigma = 0) Then
            sigma = Math.PI
        End If
        If (cossigma > 0) And (sinsigma = 0) Then
            sigma = 0
        End If
    End If
End If
sinalfa = (cosU1 * cosU2 * Math.Sin(lambda)) / (Math.Sin(sigma))
cos2alfa = 1 - (sinalfa ^ 2)
cos2sigmam = Math.Cos(sigma) - ((2 * sinU1 * sinU2) / (cos2alfa))
C = ((f) / (16)) * (cos2alfa) * ((4) + ((f) * ((4) - ((3) * (cos2alfa))))))
lambdalinha = lambda
lambda = L + (1 - C) * (f) * (sinalfa) * ((sigma) + ((C) * (Math.Sin(sigma)) *
((cos2sigmam) + ((C) * (Math.Cos(sigma)) * ((-1) + ((2) * ((cos2sigmam) ^ 2)))))))

```

End While

$u2 = ((1) - (((\cos U1) * (\cos U2) * (\text{Math.Sin}(\lambda))) / (\text{Math.Sin}(\sigma))) ^ 2))$
 $* (((a) ^ 2) - ((b) ^ 2)) / ((b) ^ 2)$

$az\tilde{a}o = 1 + ((u2) / (16384)) * ((4096) + ((u2) * (-768) + ((u2) * ((320) - (175 * u2))))))$

$bez\tilde{a}o = ((u2) / (1024)) * ((256) + ((u2) * (-128) + ((u2) * ((74) - (47 * u2))))))$

$delt\sigma = (bez\tilde{a}o) * (\text{Math.Sin}(\sigma)) * ((\cos 2\sigma) + ((bez\tilde{a}o) / 4) * ((\text{Math.Cos}(\sigma)) * (-1) + ((2) * (\cos 2\sigma) ^ 2))) - (((bez\tilde{a}o) / (6)) * (\cos 2\sigma) * (((-3) + (4) * (\text{Math.Sin}(\sigma)) ^ 2)) * ((-3) + ((4) * (\cos 2\sigma) ^ 2))))))$

$s = b * az\tilde{a}o * (\sigma - \text{delt}\sigma)$

If $\phi_2 = 0$ And $\phi_1 = 0$ Then

$s = a * \text{Math.Abs}(\lambda_2 - \lambda_1)$

End If

If $(\cos U1 * \sin U2 - \sin U1 * \cos U2 * \text{Math.Cos}(\lambda) > 0)$ And $(\cos U2 * \text{Math.Sin}(\lambda) > 0)$ Then

$Az_1 = \text{Math.Atan}((\cos U2 * \text{Math.Sin}(\lambda)) / (\cos U1 * \sin U2 - \sin U1 * \cos U2 * \text{Math.Cos}(\lambda)))$

Else

If $(\cos U1 * \sin U2 - \sin U1 * \cos U2 * \text{Math.Cos}(\lambda) < 0)$ And $(\cos U2 * \text{Math.Sin}(\lambda) < 0)$ Then

$Az_1 = \text{Math.PI} + \text{Math.Atan}((\cos U2 * \text{Math.Sin}(\lambda)) / (\cos U1 * \sin U2 - \sin U1 * \cos U2 * \text{Math.Cos}(\lambda)))$

End If

If $(\cos U1 * \sin U2 - \sin U1 * \cos U2 * \text{Math.Cos}(\lambda) > 0)$ And $(\cos U2 * \text{Math.Sin}(\lambda) < 0)$ Then

$Az_1 = 2 * \text{Math.PI} + \text{Math.Atan}((\cos U2 * \text{Math.Sin}(\lambda)) / (\cos U1 * \sin U2 - \sin U1 * \cos U2 * \text{Math.Cos}(\lambda)))$

End If

If $(\cos U1 * \sin U2 - \sin U1 * \cos U2 * \text{Math.Cos}(\lambda) = 0)$ And $(\cos U2 * \text{Math.Sin}(\lambda) > 0)$ Then

$Az_1 = \text{Math.PI} / 2$

End If

If $(\cos U1 * \sin U2 - \sin U1 * \cos U2 * \text{Math.Cos}(\lambda) = 0)$ And $(\cos U2 * \text{Math.Sin}(\lambda) < 0)$ Then

$Az_1 = 3 * \text{Math.PI} / 2$

```

End If
If (cosU1 * sinU2 - sinU1 * cosU2 * Math.Cos(lambda) < 0) And (cosU2 *
Math.Sin(lambda) = 0) Then
    Az1 = Math.PI
End If
If (cosU1 * sinU2 - sinU1 * cosU2 * Math.Cos(lambda) > 0) And (cosU2 *
Math.Sin(lambda) = 0) Then
    Az1 = 0
End If
End If
Else
    s = 0
End If
Distgeod = s
End Function

Function Azimute(ByVal phi1 As Double, ByVal lambda1 As Double, ByVal phi2 As
Double, ByVal lambda2 As Double, ByVal a As Double, ByVal f As Double) As Double
    Dim b As Double
    Dim L As Double
    Dim tanU1 As Double
    Dim tanU2 As Double
    Dim cosU1 As Double
    Dim sinU1 As Double
    Dim cosU2 As Double
    Dim sinU2 As Double
    Dim lambda As Double
    Dim lambdalinha As Double
    Dim sinsigma As Double
    Dim cossigma As Double
    Dim sigma As Double
    Dim sinalfa As Double
    Dim cos2alfa As Double
    Dim cos2sigmam As Double
    Dim C As Double
    Dim u2 As Double

```

Dim azão As Double

Dim bezão As Double

Dim deltasigma As Double

Dim s As Double

Dim Az1 As Double

If ((phi1 <> phi2) And (lambda1 <> lambda2)) Or ((phi1 <> phi2) And (lambda1 = lambda2)) Or ((phi1 = phi2) And (lambda1 <> lambda2)) Then

 If lambda1 = 2410 Then

 lambda1 = lambda2

 End If

 If lambda2 = 2410 Then

 lambda2 = lambda1

 End If

 b = a * (1 - (f))

 L = lambda2 - lambda1

 tanU1 = (1 - f) * Math.Tan(phi1)

 tanU2 = (1 - f) * Math.Tan(phi2)

 cosU1 = 1 / ((1 + ((tanU1) ^ 2)) ^ 0.5)

 sinU1 = tanU1 * cosU1

 cosU2 = 1 / ((1 + ((tanU2) ^ 2)) ^ 0.5)

 sinU2 = tanU2 * cosU2

 lambda = L

 lambdalinha = 2 * Math.PI

 While Math.Abs(lambda - lambdalinha) > 10 ^ -12

 sinsigma = (((cosU2 * Math.Sin(lambda)) ^ 2) + ((cosU1 * sinU2 - sinU1 * cosU2 * Math.Cos(lambda)) ^ 2)) ^ 0.5

 cossigma = sinU1 * sinU2 + cosU1 * cosU2 * Math.Cos(lambda)

 If (cossigma > 0) And (sinsigma > 0) Then

 sigma = Math.Atan(sinsigma / cossigma)

 Else

 If (cossigma < 0) And (sinsigma <> 0) Then

 sigma = Math.PI + Math.Atan(sinsigma / cossigma)

 End If

 If (cossigma > 0) And (sinsigma < 0) Then

 sigma = 2 * Math.PI + Math.Atan(sinsigma / cossigma)

```

End If
If (cossigma = 0) And (sinsigma > 0) Then
    sigma = Math.PI / 2
End If
If (cossigma = 0) And (sinsigma < 0) Then
    sigma = 3 * Math.PI / 2
End If
If (cossigma < 0) And (sinsigma = 0) Then
    sigma = Math.PI
End If
If (cossigma > 0) And (sinsigma = 0) Then
    sigma = 0
End If
End If
End If
sinalfa = (cosU1 * cosU2 * Math.Sin(lambda)) / (Math.Sin(sigma))
cos2alfa = 1 - (sinalfa ^ 2)
cos2sigmam = Math.Cos(sigma) - ((2 * sinU1 * sinU2) / (cos2alfa))
C = ((f) / (16)) * (cos2alfa) * ((4) + ((f) * ((4) - ((3) * (cos2alfa))))))
lambdalinha = lambda
lambda = L + (1 - C) * (f) * (sinalfa) * ((sigma) + ((C) * (Math.Sin(sigma)) *
((cos2sigmam) + ((C) * (Math.Cos(sigma)) * ((-1) + ((2) * ((cos2sigmam) ^ 2)))))))
End While
u2 = ((1) - (((cosU1) * (cosU2) * (Math.Sin(lambda))) / (Math.Sin(sigma))) ^ 2))
* (((a) ^ 2) - ((b) ^ 2)) / ((b) ^ 2)
azão = 1 + ((u2) / (16384)) * ((4096) + ((u2) * ((-768) + ((u2) * ((320) - (175 *
u2))))))
bezão = ((u2) / (1024)) * ((256) + ((u2) * ((-128) + ((u2) * ((74) - (47 * u2))))))
deltasigma = (bezão) * (Math.Sin(sigma)) * ((cos2sigmam) + ((bezão / 4) *
((Math.Cos(sigma)) * ((-1) + ((2) * (cos2sigmam) ^ 2))) - (((bezão) / (6)) * (cos2sigmam) *
(((3) + (4) * ((Math.Sin(sigma)) ^ 2)) * ((-3) + ((4) * ((cos2sigmam) ^ 2)))))))
s = b * azão * (sigma - deltasigma)
If phi2 = 0 And phi1 = 0 Then
    s = a * Math.Abs(lambda2 - lambda1)
End If

```

If $(\cos U_1 * \sin U_2 - \sin U_1 * \cos U_2 * \text{Math.Cos}(\lambda) > 0)$ And $(\cos U_2 * \text{Math.Sin}(\lambda) > 0)$ Then

$Az_1 = \text{Math.Atan}((\cos U_2 * \text{Math.Sin}(\lambda)) / (\cos U_1 * \sin U_2 - \sin U_1 * \cos U_2 * \text{Math.Cos}(\lambda)))$

Else

If $(\cos U_1 * \sin U_2 - \sin U_1 * \cos U_2 * \text{Math.Cos}(\lambda) < 0)$ And $(\cos U_2 * \text{Math.Sin}(\lambda) < 0)$ Then

$Az_1 = \text{Math.PI} + \text{Math.Atan}((\cos U_2 * \text{Math.Sin}(\lambda)) / (\cos U_1 * \sin U_2 - \sin U_1 * \cos U_2 * \text{Math.Cos}(\lambda)))$

End If

If $(\cos U_1 * \sin U_2 - \sin U_1 * \cos U_2 * \text{Math.Cos}(\lambda) > 0)$ And $(\cos U_2 * \text{Math.Sin}(\lambda) < 0)$ Then

$Az_1 = 2 * \text{Math.PI} + \text{Math.Atan}((\cos U_2 * \text{Math.Sin}(\lambda)) / (\cos U_1 * \sin U_2 - \sin U_1 * \cos U_2 * \text{Math.Cos}(\lambda)))$

End If

If $(\cos U_1 * \sin U_2 - \sin U_1 * \cos U_2 * \text{Math.Cos}(\lambda) = 0)$ And $(\cos U_2 * \text{Math.Sin}(\lambda) > 0)$ Then

$Az_1 = \text{Math.PI} / 2$

End If

If $\phi_1 = 0$ And $\phi_2 = 0$ And $\lambda_2 - \lambda_1 > 0$ Then

$Az_1 = \text{Math.PI} / 2$

End If

If $(\cos U_1 * \sin U_2 - \sin U_1 * \cos U_2 * \text{Math.Cos}(\lambda) = 0)$ And $(\cos U_2 * \text{Math.Sin}(\lambda) < 0)$ Then

$Az_1 = 3 * \text{Math.PI} / 2$

End If

If $\phi_1 = 0$ And $\phi_2 = 0$ And $\lambda_2 - \lambda_1 < 0$ Then

$Az_1 = 3 * \text{Math.PI} / 2$

End If

If $(\cos U_1 * \sin U_2 - \sin U_1 * \cos U_2 * \text{Math.Cos}(\lambda) < 0)$ And $(\cos U_2 * \text{Math.Sin}(\lambda) = 0)$ Then

$Az_1 = \text{Math.PI}$

End If

If $\lambda_1 = \lambda_2$ And $\phi_2 - \phi_1 < 0$ Then

$Az_1 = \text{Math.PI}$

```

End If
If (cosU1 * sinU2 - sinU1 * cosU2 * Math.Cos(lambda) > 0) And (cosU2 *
Math.Sin(lambda) = 0) Then
    Az1 = 0
End If
If lambda1 = lambda2 And phi2 - phi1 > 0 Then
    Az1 = 0
End If
End If
Else
    s = 0
End If
Azimute = Az1
End Function
Function Lat2(ByVal phi1 As Double, ByVal lambda1 As Double, ByVal alfa1 As
Double, ByVal dist As Double, ByVal a As Double, ByVal f As Double) As Double
    Dim b As Double
    Dim tanU1 As Double
    Dim cosU1 As Double
    Dim sinU1 As Double
    Dim sigma1 As Double
    Dim sigma As Double
    Dim sigmalinha As Double
    Dim sinalfa As Double
    Dim cosalfa As Double
    Dim cos2alfa As Double
    Dim cos2sigmam As Double
    Dim u2 As Double
    Dim azão As Double
    Dim bezão As Double
    Dim deltasigma As Double
    b = a * (1 - f)
    tanU1 = (1 - f) * Math.Tan(phi1)
    cosU1 = 1 / ((1 + ((tanU1) ^ 2)) ^ 0.5)
    sinU1 = tanU1 * cosU1

```

```

If Math.Cos(alfa1) > 0 And tanU1 > 0 Then
    sigma1 = Math.Atan(tanU1 / Math.Cos(alfa1))
Else
    If Math.Cos(alfa1) < 0 And tanU1 <> 0 Then
        sigma1 = Math.PI + Math.Atan(tanU1 / Math.Cos(alfa1))
    End If
    If Math.Cos(alfa1) > 0 And tanU1 < 0 Then
        sigma1 = 2 * Math.PI + Math.Atan(tanU1 / Math.Cos(alfa1))
    End If
    If Math.Cos(alfa1) = 0 And tanU1 > 0 Then
        sigma1 = Math.PI / 2
    End If
    If Math.Cos(alfa1) = 0 And tanU1 < 0 Then
        sigma1 = 3 * Math.PI / 2
    End If
    If Math.Cos(alfa1) < 0 And tanU1 = 0 Then
        sigma1 = Math.PI
    End If
    If Math.Cos(alfa1) > 0 And tanU1 = 0 Then
        sigma1 = 0
    End If
End If
End If
sinalfa = cosU1 * Math.Sin(alfa1)
cos2alfa = 1 - (sinalfa ^ 2)
cosalfa = (1 - (sinalfa ^ 2)) ^ 0.5
u2 = ((1) - (((cosU1) * (Math.Sin(alfa1))) ^ 2)) * (((a) ^ 2) - ((b) ^ 2)) / ((b) ^ 2)
azão = 1 + ((u2) / (16384)) * ((4096) + ((u2) * ((-768) + ((u2) * ((320) - (175 *
u2))))))
bezão = ((u2) / (1024)) * ((256) + ((u2) * ((-128) + ((u2) * ((74) - (47 * u2))))))
sigma = dist / (b * azão)
sigmalinha = 2 * Math.PI
While Math.Abs(sigma - sigmalinha) > 10 ^ -12
    cos2sigmam = Math.Cos(2 * sigma1 + sigma)

```

```

deltasigma = (bezão) * (Math.Sin(sigma)) * ((cos2sigmam) + ((bezão / 4) *
(((Math.Cos(sigma)) * (-1) + ((2) * (cos2sigmam) ^ 2))) - (((bezão) / (6)) * (cos2sigmam) *
(((-3) + (4) * ((Math.Sin(sigma)) ^ 2)) * ((-3) + ((4) * ((cos2sigmam) ^ 2))))))

```

```

sigmalinha = sigma

```

```

sigma = (dist / (b * azão)) + deltasigma

```

```

End While

```

```

Lat2 = Math.Atan((((sinU1) * (Math.Cos(sigma))) + ((cosU1) * (Math.Sin(sigma)) *
(Math.Cos(alfa1)))) / (((1) - (f)) * (((sinalfa) ^ 2) + (((sinU1) * (Math.Sin(sigma))) -
((cosU1) * (Math.Cos(sigma)) * (Math.Cos(alfa1)))) ^ 2)) ^ 0.5))

```

```

End Function

```

```

Function Long2(ByVal phi1 As Double, ByVal lambda1 As Double, ByVal alfa1 As
Double, ByVal dist As Double, ByVal a As Double, ByVal f As Double) As Double

```

```

Dim b As Double

```

```

Dim L As Double

```

```

Dim tanU1 As Double

```

```

Dim cosU1 As Double

```

```

Dim sinU1 As Double

```

```

Dim sigma1 As Double

```

```

Dim sigma As Double

```

```

Dim sigmalinha As Double

```

```

Dim lambda As Double

```

```

Dim sinalfa As Double

```

```

Dim cosalfa As Double

```

```

Dim cos2alfa As Double

```

```

Dim cos2sigmam As Double

```

```

Dim C As Double

```

```

Dim u2 As Double

```

```

Dim azão As Double

```

```

Dim bezão As Double

```

```

Dim deltasigma As Double

```

```

b = a * (1 - f)

```

```

tanU1 = (1 - f) * Math.Tan(phi1)

```

```

cosU1 = 1 / ((1 + ((tanU1) ^ 2)) ^ 0.5)

```

```

sinU1 = tanU1 * cosU1

```

```

If Math.Cos(alfa1) > 0 And tanU1 > 0 Then

```

```

sigma1 = Math.Atan(tanU1 / Math.Cos(alfa1))
Else
  If Math.Cos(alfa1) < 0 And tanU1 <> 0 Then
    sigma1 = Math.PI + Math.Atan(tanU1 / Math.Cos(alfa1))
  End If
  If Math.Cos(alfa1) > 0 And tanU1 < 0 Then
    sigma1 = 2 * Math.PI + Math.Atan(tanU1 / Math.Cos(alfa1))
  End If
  If Math.Cos(alfa1) = 0 And tanU1 > 0 Then
    sigma1 = Math.PI / 2
  End If
  If Math.Cos(alfa1) = 0 And tanU1 < 0 Then
    sigma1 = 3 * Math.PI / 2
  End If
  If Math.Cos(alfa1) < 0 And tanU1 = 0 Then
    sigma1 = Math.PI
  End If
  If Math.Cos(alfa1) > 0 And tanU1 = 0 Then
    sigma1 = 0
  End If
End If
sinalfa = cosU1 * Math.Sin(alfa1)
cos2alfa = 1 - (sinalfa ^ 2)
cosalfa = (1 - (sinalfa ^ 2)) ^ 0.5
u2 = ((1) - (((cosU1) * (Math.Sin(alfa1))) ^ 2)) * (((a) ^ 2) - ((b) ^ 2)) / ((b) ^ 2)
azão = 1 + ((u2) / (16384)) * ((4096) + ((u2) * ((-768) + ((u2) * ((320) - (175 *
u2))))))
bezão = ((u2) / (1024)) * ((256) + ((u2) * ((-128) + ((u2) * ((74) - (47 * u2))))))
sigma = dist / (b * azão)
sigmalinha = 2 * Math.PI
While Math.Abs(sigma - sigmalinha) > 10 ^ -12
  cos2sigmam = Math.Cos(2 * sigma1 + sigma)
  deltasigma = (bezão) * (Math.Sin(sigma)) * ((cos2sigmam) + ((bezão / 4) *
(((Math.Cos(sigma)) * (-1) + ((2) * (cos2sigmam) ^ 2))) - (((bezão) / (6)) * (cos2sigmam) *
(((-3) + (4) * ((Math.Sin(sigma)) ^ 2)) * ((-3) + ((4) * ((cos2sigmam) ^ 2))))))

```

```

    sigmalinha = sigma
    sigma = (dist / (b * azão)) + deltasigma
End While
If ((cosU1 * Math.Cos(sigma)) - (sinU1 * Math.Sin(sigma) * Math.Cos(alfa1)) > 0)
And ((Math.Sin(sigma) * Math.Sin(alfa1) > 0)) Then
    lambda = Math.Atan((Math.Sin(sigma) * Math.Sin(alfa1)) / ((cosU1 *
Math.Cos(sigma)) - (sinU1 * Math.Sin(sigma) * Math.Cos(alfa1))))
Else
    If ((cosU1 * Math.Cos(sigma)) - (sinU1 * Math.Sin(sigma) * Math.Cos(alfa1)) <
0) And ((Math.Sin(sigma) * Math.Sin(alfa1) <> 0)) Then
        lambda = Math.PI + Math.Atan((Math.Sin(sigma) * Math.Sin(alfa1)) / ((cosU1
* Math.Cos(sigma)) - (sinU1 * Math.Sin(sigma) * Math.Cos(alfa1))))
    End If
    If ((cosU1 * Math.Cos(sigma)) - (sinU1 * Math.Sin(sigma) * Math.Cos(alfa1)) >
0) And ((Math.Sin(sigma) * Math.Sin(alfa1) < 0)) Then
        lambda = 2 * Math.PI + Math.Atan((Math.Sin(sigma) * Math.Sin(alfa1)) /
((cosU1 * Math.Cos(sigma)) - (sinU1 * Math.Sin(sigma) * Math.Cos(alfa1))))
    End If
    If ((cosU1 * Math.Cos(sigma)) - (sinU1 * Math.Sin(sigma) * Math.Cos(alfa1)) =
0) And ((Math.Sin(sigma) * Math.Sin(alfa1) > 0)) Then
        lambda = Math.PI / 2
    End If
    If ((cosU1 * Math.Cos(sigma)) - (sinU1 * Math.Sin(sigma) * Math.Cos(alfa1)) =
0) And ((Math.Sin(sigma) * Math.Sin(alfa1) < 0)) Then
        lambda = 3 * Math.PI / 2
    End If
    If ((cosU1 * Math.Cos(sigma)) - (sinU1 * Math.Sin(sigma) * Math.Cos(alfa1)) <
0) And ((Math.Sin(sigma) * Math.Sin(alfa1) = 0)) Then
        lambda = Math.PI
    End If
    If ((cosU1 * Math.Cos(sigma)) - (sinU1 * Math.Sin(sigma) * Math.Cos(alfa1)) >
0) And ((Math.Sin(sigma) * Math.Sin(alfa1) = 0)) Then
        lambda = 0
    End If
End If

```

```

C = ((f) / (16)) * (cos2alfa) * ((4) + ((f) * ((4) - ((3) * (cos2alfa))))))
L = lambda - (1 - C) * (f) * (sinalfa) * ((sigma) + ((C) * (Math.Sin(sigma)) *
((cos2sigmam) + ((C) * (Math.Cos(sigma)) * ((-1) + ((2) * ((cos2sigmam) ^ 2)))))))
long2 = lambda1 + L
If long2 > 2 * Math.PI Then
    long2 = (((long2 / (2 * Math.PI))) - (Math.Truncate(long2 / (2 * Math.PI)))) * 2 *
Math.PI
End If
End Function
Function d2(ByVal phi As Double, ByVal a As Double, ByVal b As Double) As
Double
    d2 = (a * b) / (((a * Math.Sin(phi)) ^ 2) + ((b * Math.Cos(phi)) ^ 2)) ^ 0.5
End Function
Function X1(ByVal phi1 As Double, ByVal lambda1 As Double, ByVal a As Double,
ByVal b As Double) As Double
    X1 = (d2(phi1, a, b)) * (Math.Cos(phi1)) * (Math.Cos(lambda1))
End Function
Function Y1(ByVal phi1 As Double, ByVal lambda1 As Double, ByVal a As Double,
ByVal b As Double) As Double
    Y1 = (d2(phi1, a, b)) * (Math.Cos(phi1)) * (Math.Sin(lambda1))
End Function
Function Z1(ByVal phi1 As Double, ByVal lambda1 As Double, ByVal a As Double,
ByVal b As Double) As Double
    Z1 = (d2(phi1, a, b)) * (Math.Sin(phi1))
End Function
Function X2(ByVal phi2 As Double, ByVal lambda2 As Double, ByVal a As Double,
ByVal b As Double) As Double
    X2 = (d2(phi2, a, b)) * (Math.Cos(phi2)) * (Math.Cos(lambda2))
End Function
Function Y2(ByVal phi2 As Double, ByVal lambda2 As Double, ByVal a As Double,
ByVal b As Double) As Double
    Y2 = (d2(phi2, a, b)) * (Math.Cos(phi2)) * (Math.Sin(lambda2))
End Function
Function Z2(ByVal phi2 As Double, ByVal lambda2 As Double, ByVal a As Double,
ByVal b As Double) As Double

```

$$Z2 = (d2(\text{phi}2, a, b)) * (\text{Math.Sin}(\text{phi}2))$$

End Function

Function coef1(ByVal phi1 As Double, ByVal lambda1 As Double, ByVal phi2 As Double, ByVal lambda2 As Double, ByVal a As Double, ByVal b As Double) As Double

$$\text{coef1} = (\text{Y1}(\text{phi}1, \text{lambda}1, a, b)) * (\text{Z2}(\text{phi}2, \text{lambda}2, a, b)) - (\text{Z1}(\text{phi}1, \text{lambda}1, a, b)) * (\text{Y2}(\text{phi}2, \text{lambda}2, a, b))$$

End Function

Function coef2(ByVal phi1 As Double, ByVal lambda1 As Double, ByVal phi2 As Double, ByVal lambda2 As Double, ByVal a As Double, ByVal b As Double) As Double

$$\text{coef2} = (\text{Z1}(\text{phi}1, \text{lambda}1, a, b)) * (\text{X2}(\text{phi}2, \text{lambda}2, a, b)) - (\text{X1}(\text{phi}1, \text{lambda}1, a, b)) * (\text{Z2}(\text{phi}2, \text{lambda}2, a, b))$$

End Function

Function coef3(ByVal phi1 As Double, ByVal lambda1 As Double, ByVal phi2 As Double, ByVal lambda2 As Double, ByVal a As Double, ByVal b As Double) As Double

$$\text{coef3} = (\text{X1}(\text{phi}1, \text{lambda}1, a, b)) * (\text{Y2}(\text{phi}2, \text{lambda}2, a, b)) - (\text{Y1}(\text{phi}1, \text{lambda}1, a, b)) * (\text{X2}(\text{phi}2, \text{lambda}2, a, b))$$

End Function

Function Elipse(ByVal phi1 As Double, ByVal lambda1 As Double, ByVal phi2 As Double, ByVal lambda2 As Double, ByVal lambda As Double, ByVal a As Double, ByVal b As Double) As Double

If coef3(phi1, lambda1, phi2, lambda2, a, b) <> 0 Then

$$\text{Elipse} = \text{Math.Atan}(\frac{((- \text{coef1}(\text{phi}1, \text{lambda}1, \text{phi}2, \text{lambda}2, a, b)) * (\text{Math.Cos}(\text{lambda}))) - ((\text{coef2}(\text{phi}1, \text{lambda}1, \text{phi}2, \text{lambda}2, a, b)) * (\text{Math.Sin}(\text{lambda})))}{(\text{coef3}(\text{phi}1, \text{lambda}1, \text{phi}2, \text{lambda}2, a, b))})$$

ElseIf coef3(phi1, lambda1, phi2, lambda2, a, b) = 0 Then

$$\text{Elipse} = -\text{Math.PI} / 2$$

End If

End Function

Function Primitiva40(ByVal p As Double) As Double

$$\text{Primitiva40} = (-0.50671678633410688 / 3) * (\text{p} \wedge 3)$$

$$\text{Primitiva40} = \text{Primitiva40} + (-0.000031104822192129266 / 11) * (\text{p} \wedge 11)$$

$$\text{Primitiva40} = \text{Primitiva40} + (0.00000000012049727257425491 / 19) * (\text{p} \wedge 19)$$

$$\text{Primitiva40} = \text{Primitiva40} + (-0.00000000012362870861489277 / 21) * (\text{p} \wedge 21)$$

$$\text{Primitiva40} = \text{Primitiva40} + (-0.000000000039319612844930536 / 25) * (\text{p} \wedge 25)$$

$$\text{Primitiva40} = \text{Primitiva40} + (0.0000000082637660568813216 / 17) * (\text{p} \wedge 17)$$

$$\text{Primitiva40} = \text{Primitiva40} + (2.5861605396074929\text{E-}19 / 41) * (p \wedge 41)$$

$$\text{Primitiva40} = \text{Primitiva40} + (0.00000000000005517687667256797 / 27) * (p \wedge 27)$$

$$\text{Primitiva40} = \text{Primitiva40} + (0.00000000000000013524360829030762 / 35) * (p \wedge 35)$$

$$\text{Primitiva40} = \text{Primitiva40} + (0.0003105543930470135 / 9) * (p \wedge 9)$$

$$\text{Primitiva40} = \text{Primitiva40} + (-0.0000000000000072890259262632536 / 29) * (p \wedge 29)$$

$$\text{Primitiva40} = \text{Primitiva40} + (-0.00000000000000011469660967064847 / 33) * (p \wedge 33)$$

$$\text{Primitiva40} = \text{Primitiva40} + (0.000000000000000926557558730247 / 31) * (p \wedge 31)$$

$$\text{Primitiva40} = \text{Primitiva40} + (0.0000024964137564367444 / 13) * (p \wedge 13)$$

$$\text{Primitiva40} = \text{Primitiva40} + (-0.00000016692337896406959 / 15) * (p \wedge 15)$$

$$\text{Primitiva40} = \text{Primitiva40} + (0.047286239591878346 / 5) * (p \wedge 5)$$

$$\text{Primitiva40} = \text{Primitiva40} + (-0.0000000000000000011013989265482367 / 37) * (p \wedge 37)$$

$$\text{Primitiva40} = \text{Primitiva40} + (-0.0031125429187103129 / 7) * (p \wedge 7)$$

$$\text{Primitiva40} = \text{Primitiva40} + (-7.366332867999105\text{E-}19 / 39) * (p \wedge 39)$$

$$\text{Primitiva40} = \text{Primitiva40} + (0.000000000025099753377400407 / 23) * (p \wedge 23)$$

$$\text{Primitiva40} = \text{Primitiva40} + p$$

End Function

Function Ss(ByVal phi1 As Double, ByVal lambda1 As Double, ByVal phi2 As Double, ByVal lambda2 As Double, ByVal lambda As Double, ByVal a As Double, ByVal b As Double) As Double

Ss = Primitiva40(Elipse(phi1, lambda1, phi2, lambda2, lambda, a, b)) - Primitiva40(-Math.PI / 2)

End Function

Function Simpson(ByVal phi1 As Double, ByVal lambda1 As Double, ByVal phi2 As Double, ByVal lambda2 As Double, ByVal a As Double, ByVal b As Double) As Double

Dim n As Integer

Dim h As Double

Dim i As Integer

n = 10

h = (lambda2 - lambda1) / n

Simpson = 0

For i = 1 To n / 2

```
Simpson = Simpson + Ss(phi1, lambda1, phi2, lambda2, lambda1 + 2 * i * h, a, b)
+ 4 * Ss(phi1, lambda1, phi2, lambda2, lambda1 + (2 * i - 1) * h, a, b) + Ss(phi1, lambda1,
phi2, lambda2, lambda1 + (2 * i - 2) * h, a, b)
```

```
Next
```

```
Simpson = Simpson * ((a ^ 2) * (h / 3))
```

```
End Function
```

```
Function Xmerc(ByVal lambda As Double, ByVal phi0 As Double, ByVal a As
Double, ByVal e1 As Double) As Double
```

```
Xmerc = a * lambda * Math.Cos(phi0) / ((1 - ((e1 * Math.Sin(phi0)) ^ 2)) ^ 0.5)
```

```
End Function
```

```
Function Ymerc(ByVal phi As Double, ByVal phi0 As Double, ByVal a As Double,
ByVal e1 As Double) As Double
```

```
If phi <= -89.999999999999 * Math.PI / 180 Then
```

```
Ymerc = (Math.Log((Math.Tan((Math.PI / 4) + (-89.999999999999 * Math.PI /
360))) * (((1 - e1 * Math.Sin(-89.999999999999 * Math.PI / 180)) / (1 + e1 * Math.Sin(-
89.999999999999 * Math.PI / 180))) ^ (e1 / 2)))) * (a * Math.Cos(phi0) / ((1 - ((e1 *
Math.Sin(phi0)) ^ 2)) ^ 0.5))
```

```
ElseIf phi >= 89.999999999999 * Math.PI / 180 Then
```

```
Ymerc = (Math.Log((Math.Tan((Math.PI / 4) + (89.999999999999 * Math.PI /
360))) * (((1 - e1 * Math.Sin(89.999999999999 * Math.PI / 180)) / (1 + e1 *
Math.Sin(89.999999999999 * Math.PI / 180))) ^ (e1 / 2)))) * (a * Math.Cos(phi0) / ((1 - ((e1
* Math.Sin(phi0)) ^ 2)) ^ 0.5))
```

```
Else
```

```
Ymerc = (Math.Log((Math.Tan((Math.PI / 4) + (phi / 2))) * (((1 - e1 *
Math.Sin(phi)) / (1 + e1 * Math.Sin(phi))) ^ (e1 / 2)))) * (a * Math.Cos(phi0) / ((1 - ((e1
* Math.Sin(phi0)) ^ 2)) ^ 0.5))
```

```
End If
```

```
End Function
```

```
Function longlox(ByVal xmerc1 As Double, ByVal phi0 As Double, ByVal a As
Double, ByVal e1 As Double) As Double
```

```
longlox = xmerc1 * ((1 - ((e1 * Math.Sin(phi0)) ^ 2)) ^ 0.5) / (a * Math.Cos(phi0))
```

```
End Function
```

```
Function Func(ByVal ymerc1 As Double, ByVal phi As Double, ByVal phi0 As
Double, ByVal a As Double, ByVal e1 As Double) As Double
```

```

Func = Math.Exp(((1 - ((e1 * Math.Sin(phi0)) ^ 2)) ^ 0.5) * (ymerc1) / (a *
Math.Cos(phi0))) - (((Math.Tan((Math.PI / 4) + (phi / 2))) * (((1 - e1 * Math.Sin(phi)) / (1 +
e1 * Math.Sin(phi))) ^ (e1 / 2))))

```

```

End Function

```

```

Function DerivFunc(ByVal ymerc1 As Double, ByVal phi As Double, ByVal phi0 As
Double, ByVal a As Double, ByVal e1 As Double) As Double

```

```

DerivFunc = (Func(ymerc1, phi + 10 ^ -5, phi0, a, e1) - Func(ymerc1, phi - 10 ^ -5,
phi0, a, e1)) / (2 * (10 ^ -5))

```

```

End Function

```

```

Function latlox(ByVal ymerc1 As Double, ByVal phi0 As Double, ByVal a As Double,
ByVal e1 As Double) As Double

```

```

Dim phi00 As Double

```

```

Dim sinal As Integer

```

```

Dim contador As Integer

```

```

Dim i As Integer

```

```

phi00 = 0

```

```

sinal = 0

```

```

If ymerc1 > 0 Then

```

```

    ymerc1 = -ymerc1

```

```

    sinal = 1

```

```

End If

```

```

For i = 1 To 10

```

```

    phi00 = phi00 - ((Func(ymerc1, phi00, phi0, a, e1)) / (DerivFunc(ymerc1, phi00,
phi0, a, e1)))

```

```

    If Math.Abs(Func(ymerc1, phi00, phi0, a, e1)) < 10 ^ -15 Then

```

```

        contador = 11

```

```

    End If

```

```

    contador = contador + 1

```

```

Next

```

```

If ymerc1 < Ymerc(-89.999999999999 * Math.PI / 180, phi0, a, e1) Then

```

```

    phi00 = -Math.PI / 2

```

```

ElseIf ymerc1 > Ymerc(89.999999999999 * Math.PI / 180, phi0, a, e1) Then

```

```

    phi00 = Math.PI / 2

```

```

End If

```

```

If sinal = 1 Then

```

```

        phi00 = -phi00
    End If
    latlox = phi00
End Function

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    BackColor = Color.Black
    TransparencyKey = BackColor
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    TextBox4.Enabled = False
    TextBox4.Visible = True
End Sub

Private Sub RadioButton1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RadioButton1.Click
    Button2.Visible = True
    Me.Refresh()
End Sub

Private Sub RadioButton2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles RadioButton2.Click
    Button2.Visible = False
    Me.Refresh()
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Dim fs As System.IO.StreamReader
    Dim dataLine As String
    Dim tempArray() As String
    Dim Geodésicos(10000, 2) As Double
    Dim Pontos(10000, 3) As Double
    Dim Distâncias(10000)
    Dim Az(10000)

```

```
Dim Geodésicos2(10000000, 2) As Double
Dim Geocêntricos(10000000, 2) As Double
Dim contador As Integer
Dim i As Int32
Dim j As Int32 = 0
Dim a As Double
Dim f As Double
Dim b As Double
Dim e1 As Double
Dim linPontos As Integer
Dim X0 As Double
Dim dist As Integer
Dim passo As Integer
Dim alfa As Double
Dim coef As Double
Dim Area As Double
Dim Areas As Double
OpenFileDialog1.Filter = "Text files (*.TXT)|*.TXT"
OpenFileDialog1.ShowDialog()
fs = New System.IO.StreamReader(OpenFileDialog1.FileName)
While Not fs.EndOfStream
    dataLine = fs.ReadLine()
    tempArray = dataLine.Split()
    For i = 0 To tempArray.Length - 1
        Try
            Pontos(j, i) = Double.Parse(tempArray(i))
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    Next
    j = j + 1
    linPontos = linPontos + 1
End While
fs.Close()
a = TextBox1.Text
```

```

f = TextBox2.Text
b = a * (1 - f)
e1 = (((a ^ 2) - (b ^ 2)) / (a ^ 2)) ^ 0.5
X0 = b
contador = 0
For contador = 0 To linPontos - 1
    Geodésicos(contador, 0) = u(Pontos(contador, 0), Pontos(contador, 1),
Pontos(contador, 2), a, b)
    Geodésicos(contador, 1) = v(Pontos(contador, 0), Pontos(contador, 1),
Pontos(contador, 2), a, b)
Next
    If (Geodésicos(linPontos - 1, 1) = 0) And (Geodésicos(linPontos - 2, 1) -
Geodésicos(linPontos - 3, 1) > 0) Then
        Geodésicos(linPontos - 1, 1) = 2 * Math.PI
    End If
    contador = 0
    For contador = 0 To linPontos - 2
        Distâncias(contador) = Distgeod(Geodésicos(contador, 0), Geodésicos(contador,
1), Geodésicos(contador + 1, 0), Geodésicos(contador + 1, 1), a, f)
    Next
    contador = 0
    For contador = 0 To linPontos - 2
        Az(contador) = Azimute(Geodésicos(contador, 0), Geodésicos(contador, 1),
Geodésicos(contador + 1, 0), Geodésicos(contador + 1, 1), a, f)
    Next
    i = 0
    contador = 0
    coef = 0
    While i <= (linPontos - 2)
        If Math.Abs(Distâncias(i)) > 20000 Then
            passo = 20000
        Else
            If Math.Abs(Distâncias(i)) > 1000 Then
                passo = 1000
            Else

```



```

    Geodésicos2(contador, 0) = Geodésicos(i + 1, 0)
ElseIf dist = Distâncias(i) Then
    Geodésicos2(contador, 0) = Geodésicos(i + 1, 0)
ElseIf Math.Abs(Lat2(Geodésicos(i, 0), Geodésicos(i, 1), alfa, dist, a, f)) > 10 ^
-15 Then
    Geodésicos2(contador, 0) = Lat2(Geodésicos(i, 0), Geodésicos(i, 1), alfa, dist,
a, f)
    ElseIf (Math.Abs(Lat2(Geodésicos(i, 0), Geodésicos(i, 1), alfa, dist, a, f)) < 10 ^
-15) And (dist <> Distâncias(i)) Then
        Geodésicos2(contador, 0) = 0
    End If
    If Math.Abs(dist - Distâncias(i)) < passo Then
        Geodésicos2(contador, 1) = Geodésicos(i + 1, 1)
    ElseIf Geodésicos(i, 1) = 2410 Then
        Geodésicos2(contador, 1) = Geodésicos(i + 1, 1)
    ElseIf Geodésicos(i + 1, 1) = 2410 Then
        Geodésicos2(contador, 1) = Geodésicos(i, 1)
    ElseIf dist = Distâncias(i) Then
        Geodésicos2(contador, 1) = Geodésicos(i + 1, 1)
    ElseIf (Math.Abs(Long2(Geodésicos(i, 0), Geodésicos(i, 1), alfa, dist, a, f)) > 10
^ -15) And (Long2(Geodésicos(i, 0), Geodésicos(i, 1), alfa, dist, a, f) <> 2410) Then
        Geodésicos2(contador, 1) = Long2(Geodésicos(i, 0), Geodésicos(i, 1), alfa,
dist, a, f)
    End If
    If (Math.Abs(Long2(Geodésicos(i, 0), Geodésicos(i, 1), alfa, dist, a, f)) < 10 ^ -
15) And (Long2(Geodésicos(i, 0), Geodésicos(i, 1), alfa, dist, a, f) <> 2410) And (dist <>
Distâncias(i)) Then
        Geodésicos2(contador, 1) = 0
    End If
    If contador > 0 Then
        If (Geodésicos2(contador, 1) = 0) And (Geodésicos2(contador - 1, 1) > (359 *
Math.PI / 360)) Then
            Geodésicos2(contador, 1) = 2 * Math.PI
        End If
    End If
End If

```

```

        contador = contador + 1
        coef = coef + 1
    End While
    i = i + 1
End While
For contador = 0 To coef - 1
    Geocêntricos(contador, 0) = Math.Atan((1 - e1 ^ 2) *
Math.Tan(Geodésicos2(contador, 0)))
    Geocêntricos(contador, 1) = Geodésicos2(contador, 1)
Next
Area = 0
For contador = 0 To coef - 2
    If Distgeod(Geodésicos2(contador, 0), Geodésicos2(contador, 1),
Geodésicos2(contador + 1, 0), Geodésicos2(contador + 1, 1), a, f) = 0 Then
        Area = Area + 0
    ElseIf Azimute(Geodésicos2(contador, 0), Geodésicos2(contador, 1),
Geodésicos2(contador + 1, 0), Geodésicos2(contador + 1, 1), a, f) = 0 Then
        Area = Area + 0
    ElseIf Azimute(Geodésicos2(contador, 0), Geodésicos2(contador, 1),
Geodésicos2(contador + 1, 0), Geodésicos2(contador + 1, 1), a, f) = Math.PI Then
        Area = Area + 0
    ElseIf Distgeod(Geodésicos2(contador, 0), Geodésicos2(contador, 1),
Geodésicos2(contador + 1, 0), Geodésicos2(contador + 1, 1), a, f) <> 0 Then
        Area = Area + Simpson(Geocêntricos(contador, 0), Geocêntricos(contador, 1),
Geocêntricos(contador + 1, 0), Geocêntricos(contador + 1, 1), a, b)
    End If
Next
Areas = Math.Abs(Area)
TextBox3.Text = Areas
End Sub
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
    Me.Close()
End Sub

```

```
Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
    If ComboBox1.SelectedItem = "WGS-84" Then
        TextBox1.Enabled = False
        TextBox2.Enabled = False
        TextBox1.Text = "6378137"
        TextBox2.Text = "0.00335281066474748"
    End If
    If ComboBox1.SelectedItem = "Outros" Then
        TextBox1.Enabled = True
        TextBox2.Enabled = True
    End If
End Sub
```

```
Private Sub Button2_MouseEnter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.MouseEnter
    Button2.Width = 120
    Me.Refresh()
End Sub
```

```
Private Sub Button2_MouseLeave(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.MouseLeave
    Button2.Width = 75
    Me.Refresh()
End Sub
```

```
Private Sub Button4_MouseEnter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.MouseEnter
    Button4.Width = 120
    Me.Refresh()
End Sub
```

```
Private Sub Button4_MouseLeave(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.MouseLeave
    Button4.Width = 75
    Me.Refresh()
```

End Sub

```
Private Sub Button3_MouseEnter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.MouseEnter
```

```
    Button3.Width = 120
```

```
    Me.Refresh()
```

End Sub

```
Private Sub Button3_MouseLeave(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.MouseLeave
```

```
    Button3.Width = 75
```

```
    Me.Refresh()
```

End Sub

```
Private Sub Button3_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
```

```
    Dim fs As System.IO.StreamReader
```

```
    Dim dataLine As String
```

```
    Dim tempArray() As String
```

```
    Dim Geodésicos(10000000, 2) As Double
```

```
    Dim Pontos(10000000, 3) As Double
```

```
    Dim Distâncias_loxo(10000000) As Double
```

```
    Dim Az_loxo(10000000) As Double
```

```
    Dim Loxodrômicos(100000, 2) As Double
```

```
    Dim Geocêntricos_loxo(1000000, 2) As Double
```

```
    Dim contador As Integer
```

```
    Dim i As Int32
```

```
    Dim j As Int32 = 0
```

```
    Dim a As Double
```

```
    Dim f As Double
```

```
    Dim b As Double
```

```
    Dim e1 As Double
```

```
    Dim linPontos As Integer
```

```
    Dim X0 As Double
```

```
    Dim dist As Integer
```

```
    Dim passo As Integer
```

```

Dim alfa As Double
Dim coef As Double
Dim Area As Double
Dim Areas As Double
Dim latmax As Double
Dim latmin As Double
Dim phi0 As Double
Dim cossigma As Double
Dim sinsigma As Double
OpenFileDialog1.Filter = "Text files (*.TXT)|*.TXT"
OpenFileDialog1.ShowDialog()
fs = New System.IO.StreamReader(OpenFileDialog1.FileName)
While Not fs.EndOfStream
    dataLine = fs.ReadLine()
    tempArray = dataLine.Split()
    For i = 0 To tempArray.Length - 1
        Try
            Pontos(j, i) = Double.Parse(tempArray(i))
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try
    Next
    j = j + 1
    linPontos = linPontos + 1
End While
fs.Close()
a = TextBox1.Text
f = TextBox2.Text
b = a * (1 - f)
e1 = (((a ^ 2) - (b ^ 2)) / (a ^ 2)) ^ 0.5
X0 = b
contador = 0
For contador = 0 To linPontos - 1
    Geodésicos(contador, 0) = u(Pontos(contador, 0), Pontos(contador, 1),
Pontos(contador, 2), a, b)

```

```

    Geodésicos(contador, 1) = v(Pontos(contador, 0), Pontos(contador, 1),
Pontos(contador, 2), a, b)
Next
If (Geodésicos(linPontos - 1, 1) = 0) And (Geodésicos(linPontos - 2, 1) -
Geodésicos(linPontos - 3, 1) > 0) Then
    Geodésicos(linPontos - 1, 1) = 2 * Math.PI
End If
latmax = Geodésicos(0, 0)
latmin = Geodésicos(0, 0)
i = 0
For i = 0 To linPontos - 1
    latmax = latmax
    latmin = latmin
    If (Geodésicos(i, 0) - latmax) >= 0 Then
        latmax = Geodésicos(i, 0)
    End If
    If (Geodésicos(i, 0) - latmin) <= 0 Then
        latmin = Geodésicos(i, 0)
    End If
Next
phi0 = (latmin + latmax) / 2
i = 0
For i = 0 To linPontos - 2
    Distâncias_loxo(i) = (((Xmerc(Geodésicos(i + 1, 1), phi0, a, e1) -
Xmerc(Geodésicos(i, 1), phi0, a, e1)) ^ 2) + ((Ymerc(Geodésicos(i + 1, 0), phi0, a, e1) -
Ymerc(Geodésicos(i, 0), phi0, a, e1)) ^ 2)) ^ 0.5
Next
i = 0
For i = 0 To linPontos - 2
    sinsigma = Xmerc(Geodésicos(i + 1, 1), phi0, a, e1) - Xmerc(Geodésicos(i, 1),
phi0, a, e1)
    cossigma = Ymerc(Geodésicos(i + 1, 0), phi0, a, e1) - Ymerc(Geodésicos(i, 0),
phi0, a, e1)
    If (cossigma > 0) And (sinsigma > 0) Then
        Az_loxo(i) = Math.Atan(sinsigma / cossigma)

```

```

Else
  If (cossigma < 0) And (sinsigma <> 0) Then
    Az_loxo(i) = Math.PI + Math.Atan(sinsigma / cossigma)
  End If
  If (cossigma > 0) And (sinsigma < 0) Then
    Az_loxo(i) = 2 * Math.PI + Math.Atan(sinsigma / cossigma)
  End If
  If (cossigma = 0) And (sinsigma > 0) Then
    Az_loxo(i) = Math.PI / 2
  End If
  If (cossigma = 0) And (sinsigma < 0) Then
    Az_loxo(i) = 3 * Math.PI / 2
  End If
  If (cossigma < 0) And (sinsigma = 0) Then
    Az_loxo(i) = Math.PI
  End If
  If (cossigma > 0) And (sinsigma = 0) Then
    Az_loxo(i) = 0
  End If
End If
Next
i = 0
contador = 0
coef = 0
While i <= (linPontos - 2)
  If Math.Abs(Distâncias_loxo(i)) > 20000 Then
    passo = 20000
  Else
    If Math.Abs(Distâncias_loxo(i)) > 1000 Then
      passo = 1000
    Else
      If Math.Abs(Distâncias_loxo(i)) > 500 Then
        passo = 500
      Else
        If Math.Abs(Distâncias_loxo(i)) > 200 Then

```



```

ElseIf Math.Abs(latlox(Ymerc(Geodésicos(i, 0), phi0, a, e1) + dist *
Math.Cos(alfa), phi0, a, e1)) > 10 ^ -15 Then
    Loxodrômicos(contador, 0) = latlox(Ymerc(Geodésicos(i, 0), phi0, a, e1) +
dist * Math.Cos(alfa), phi0, a, e1)
    ElseIf (Math.Abs(latlox(Ymerc(Geodésicos(i, 0), phi0, a, e1) + dist *
Math.Cos(alfa), phi0, a, e1)) < 10 ^ -15) And (dist <> Distâncias_loxo(i)) Then
        Loxodrômicos(contador, 0) = 0
    End If
    If Math.Abs(dist - Distâncias_loxo(i)) < passo Then
        Loxodrômicos(contador, 1) = Geodésicos(i + 1, 1)
    ElseIf Geodésicos(i, 1) = 2410 Then
        Loxodrômicos(contador, 1) = Geodésicos(i + 1, 1)
    ElseIf Geodésicos(i + 1, 1) = 2410 Then
        Loxodrômicos(contador, 1) = Geodésicos(i, 1)
    ElseIf dist = Distâncias_loxo(i) Then
        Loxodrômicos(contador, 1) = Geodésicos(i + 1, 1)
    ElseIf (Math.Abs(longlox(Xmerc(Geodésicos(i, 1), phi0, a, e1) + dist *
Math.Sin(alfa), phi0, a, e1)) > 10 ^ -15) And (longlox(Xmerc(Geodésicos(i, 1), phi0, a, e1) +
dist * Math.Sin(alfa), phi0, a, e1) <> 2410) Then
        Loxodrômicos(contador, 1) = longlox(Xmerc(Geodésicos(i, 1), phi0, a, e1) +
dist * Math.Sin(alfa), phi0, a, e1)
    End If
    If (longlox(Xmerc(Geodésicos(i, 1), phi0, a, e1) + dist * Math.Sin(alfa), phi0, a,
e1) < 10 ^ -15) And (longlox(Xmerc(Geodésicos(i, 1), phi0, a, e1) + dist * Math.Sin(alfa),
phi0, a, e1) <> 2410) And (dist <> Distâncias_loxo(i)) Then
        Loxodrômicos(contador, 1) = 0
    End If
    If contador > 0 Then
        If (Loxodrômicos(contador, 1) = 0) And (Loxodrômicos(contador - 1, 1) >
(359 * Math.PI / 360)) Then
            Loxodrômicos(contador, 1) = 2 * Math.PI
        End If
    End If
    contador = contador + 1
    coef = coef + 1

```

```

    End While
    i = i + 1
End While
For contador = 0 To coef - 1
    Geocêntricos_loxo(contador, 0) = Math.Atan((1 - e1 ^ 2) *
Math.Tan(Loxodrômicos(contador, 0)))
    Geocêntricos_loxo(contador, 1) = Loxodrômicos(contador, 1)
Next
Area = 0
For r = 0 To coef - 2
    Area = Area + Simpson(Geocêntricos_loxo(r, 0), Geocêntricos_loxo(r, 1),
Geocêntricos_loxo(r + 1, 0), Geocêntricos_loxo(r + 1, 1), a, b)
Next
Areas = Math.Abs(Area)
TextBox3.Text = Areas
End Sub
End Class

```