

Relatório Técnico Avançado: Implementação Computacional e Análise Normativa de Modelos de Propagação (ITU-R P.1546 e Método Deygout-Assis) no Contexto da Regulação Brasileira

1. Introdução e Contextualização do Cenário de Radiodifusão

O planejamento de redes de radiodifusão e telecomunicações no Brasil atravessa um período de transformação tecnológica e regulatória sem precedentes. A consolidação do Sistema Brasileiro de Televisão Digital Terrestre (SBTVD-T), a migração das emissoras de rádio AM para a faixa de FM e a contínua refarming do espectro para acomodar serviços móveis de banda larga (IMT/5G) exigem um rigor técnico cada vez maior na predição de cobertura e análise de interferência. Neste cenário, a precisão dos algoritmos de propagação de ondas eletromagnéticas deixa de ser apenas uma questão acadêmica para se tornar um requisito legal e operacional crítico.

A Agência Nacional de Telecomunicações (Anatel), através da implementação do sistema **Mosaico**, digitalizou e automatizou os processos de licenciamento e análise técnica. O sucesso na submissão de projetos técnicos — seja para a inclusão de um novo canal no Plano Básico, seja para a alteração de características técnicas de uma estação existente — depende estreitamente da conformidade com modelos de propagação específicos. Entre estes, destacam-se a Recomendação **ITU-R P.1546**, utilizada para cálculos estatísticos de cobertura ponto-área e definição de contornos protegidos, e os métodos determinísticos de difração, especificamente o método de **Deygout**, frequentemente refinado pela correção de curvatura proposta pelo pesquisador brasileiro **Mauro Soares de Assis**.

Este relatório apresenta uma análise exaustiva e detalhada sobre a implementação computacional destes modelos. O objetivo é fornecer a engenheiros, pesquisadores e desenvolvedores de software um guia definitivo que abranja desde os fundamentos físicos e matemáticos até a codificação prática em linguagens de alto nível (Python, MATLAB) e desempenho (C++), tudo sob a ótica das exigências normativas da Anatel (Atos nº 9751, 3114, 3115 e Resolução nº 721).

1.1 A Evolução dos Requisitos Técnicos da Anatel

Historicamente, o planejamento de radiodifusão no Brasil baseava-se em curvas de propagação simplificadas, como as da FCC (Federal Communications Commission) ou a antiga recomendação ITU-R P.370. No entanto, a complexidade do relevo brasileiro — caracterizado por vastos planaltos, serras abruptas (como a Serra do Mar e a Serra da Mantiqueira) e densas áreas urbanas — demonstrou que modelos puramente estatísticos eram insuficientes para

garantir a convivência harmoniosa entre estações, especialmente com a introdução de modulações digitais (OFDM) que possuem requisitos de relação de proteção (D/U) distintos e comportamentos de limiar abruptos ("cliff effect").

A atualização regulatória, consolidada na Resolução nº 721/2020 e seus Atos complementares, institucionalizou o uso da ITU-R P.1546-5 (ou versões superiores) como o padrão ouro para a determinação da intensidade de campo. Paralelamente, para a análise detalhada de enlaces ponto-a-ponto ou verificação de desobstrução em trajetos críticos de interferência, a agência e a engenharia nacional consagraram o uso de métodos que consideram o perfil do terreno. É neste nicho que o método de Deygout, corrigido por Assis para obstáculos arredondados (cilindros), se torna indispensável. Enquanto o método clássico de gume de faca (Knife-Edge) tende a ser otimista demais em terrenos ondulados, subestimando a atenuação e prevendo interferências inexistentes ou coberturas fantasma, a correção de Assis introduz a penalidade física real causada pela difração sobre superfícies convexas, alinhando a predição teórica com as medições de campo.

1.2 Objetivos e Estrutura do Relatório

Este documento foi estruturado para cobrir todas as facetas da solicitação original, expandindo-as para oferecer um manual de referência completo. Abordaremos:

1. **Fundamentação Teórica Profunda:** Uma dissecação das equações de Fresnel, da geometria do elipsoide e das integrais que regem a difração, fornecendo a base para entender por que os modelos funcionam.
2. **A Recomendação ITU-R P.1546:** Análise detalhada dos procedimentos de interpolação, correção de ângulo de clareira do terreno (TCA) e distinção entre trajetos terrestres e marítimos.
3. **O Método Deygout e a Correção de Assis:** Explicação passo a passo do algoritmo recursivo de Deygout e a derivação matemática da correção cilíndrica de Assis.
4. **Implementação Computacional:** Exemplos de código, arquitetura de software e bibliotecas em Python, MATLAB e C++, com foco em desempenho e precisão numérica.
5. **Interface com o Sistema Mosaico:** Detalhamento dos formatos de arquivo, requisitos de upload e parâmetros de engenharia aceitos pela Anatel.
6. **Parâmetros de Proteção e Planejamento:** Tabelas completas e análise das relações de proteção para TV Digital e FM, essenciais para a validação dos cálculos.

2. Recomendação ITU-R P.1546: O Padrão para Predição Ponto-Área

A Recomendação ITU-R P.1546, intitulada "Método para previsões ponto-área para serviços terrestres na faixa de frequências de 30 MHz a 4.000 MHz", é a espinha dorsal do planejamento de cobertura na Anatel. Diferente de modelos determinísticos que exigem um perfil de terreno detalhado para cada radial, a P.1546 baseia-se em curvas empíricas derivadas de décadas de medições, ajustadas por parâmetros estatísticos e geográficos.

2.1 A Estrutura das Curvas de Propagação

O núcleo da P.1546 não é uma equação única, mas sim uma vasta família de curvas que relacionam a intensidade de campo (E) com a distância (d). A implementação computacional

deste modelo é, essencialmente, um exercício avançado de interpolação multidimensional. As curvas são tabuladas para os seguintes parâmetros discretos:

- **Frequências Nominais (f):** 100, 600 e 2.000 MHz.
- **Alturas Efetivas da Antena Transmissora (h_1):** Valores variando de 10 m a 1.200 m.
- **Porcentagens de Tempo (t):** 50%, 10% e 1%.
- **Tipos de Percurso:** Terra, Mar Frio e Mar Quente.

Para uma dada situação de engenharia — por exemplo, uma estação de FM em 98,5 MHz com antena a 135 metros operando em região mista — o software deve realizar interpolações logarítmicas e lineares para extrair o valor correto.

2.1.1 O Processo de Interpolação

A precisão do modelo depende da ordem correta das interpolações. O algoritmo deve seguir a seguinte hierarquia, conforme definido no Anexo 5 da recomendação :

1. **Interpolação de Altura (h_1):** Para cada frequência nominal e porcentagem de tempo, o campo é interpolado entre as curvas de altura inferior ($h_{\{inf\}}$) e superior ($h_{\{sup\}}$). A altura efetiva h_1 é definida como a altura da antena sobre o nível médio do terreno entre 3 km e 15 km na direção do receptor.
2. **Interpolação de Frequência (f):** Após obter os campos para a altura correta nas frequências nominais (ex: 100 e 600 MHz), realiza-se a interpolação para a frequência de operação.
3. **Interpolação de Distância:** As tabelas fornecem valores para distâncias logarítmicas. A interpolação deve seguir a natureza da propagação (lei do inverso do quadrado ou guias de onda atmosféricos), geralmente linear em $\log(d)$.

2.2 O Ângulo de Clareira do Terreno (TCA)

Uma das inovações mais significativas da P.1546 em relação à P.370 é a introdução da correção pelo Ângulo de Clareira do Terreno (Terrain Clearance Angle - TCA). O TCA, denotado por $\theta_{\{tca\}}$, refina a predição considerando a obstrução local nas imediações do receptor (ou transmissor, dependendo da reciprocidade).

Cálculo do TCA: O TCA não exige o perfil completo do enlace, mas sim uma análise detalhada dos primeiros 16 km a partir da antena em questão. O ângulo é medido em relação à linha horizontal no local da antena:

- $\theta_{\{tca\}} > 0$: Indica que o terreno sobe na direção do trajeto, criando obstrução. A intensidade de campo deve ser atenuada.
- $\theta_{\{tca\}} < 0$: Indica que o terreno desce (ex: transmissor no topo de um morro com vista para um vale). A intensidade de campo pode ser reforçada devido à melhor visibilidade.

A correção aplicada ao campo básico ($E_{\{b\}}$) é dada por uma função não linear de $\theta_{\{tca\}}$, que deve ser implementada com cuidado para evitar descontinuidades. Em implementações de software como a Py1546 e p1546 para MATLAB , a função calc_tca percorre o Modelo Digital de Elevação (MDE) para determinar este ângulo crítico.

2.3 Desafios de Implementação e Bibliotecas Existentes

Para engenheiros que desejam integrar a P.1546 em seus sistemas, a construção das tabelas de dados ("look-up tables") é a etapa mais trabalhosa. Felizmente, existem bibliotecas

open-source e acadêmicas que já digitalizaram essas curvas.

2.3.1 Python: A Biblioteca Py1546 e o Ecossistema itur

A biblioteca Py1546, desenvolvida por eeveetza, é uma implementação de referência da Recomendação ITU-R P.1546-6.

* **Instalação e Estrutura:** A biblioteca pode ser instalada via pip diretamente do repositório git. A estrutura do código é modular, separando as tabelas de dados da lógica de interpolação.
bash python -m pip install "git+https://github.com/eeveetza/Py1546/#egg=Py1546"

- **Exemplo de Uso Prático (Python):** O código abaixo ilustra como calcular a perda básica de transmissão (L_b) para um cenário típico de radiodifusão FM no Brasil.

```
from Py1546 import P1546
import numpy as np

# Parâmetros do Cenário
freq_mhz = 98.5          # Frequência da Rádio FM
time_pct = 50.0           # 50% do tempo (Serviço)
h_eff_tx = 150.0          # Altura efetiva do transmissor (m)
h_rx = 10.0               # Altura do receptor (m) - Típico outdoor
clutter_rx = 10.0          # Altura representativa do clutter
                           # (Rural/Suburbano)
path_type = 'Land'        # Trajeto terrestre

# Vetor de distâncias (km) - Ex: Radial de 1km a 100km
distances = np.arange(1, 101, 1.0)

# Definição do vetor de zonas (todas 'Land' neste exemplo)
path_zones = [path_type for _ in distances]

# Flag para informação de terreno (0 = sem perfil detalhado)
terrain_info = 0

# Cálculo
# A função retorna Campo (E) e Perda Básica (Lb)
E_field, L_basic = P1546.bt_loss(
    f=freq_mhz,
    t=time_pct,
    heff=h_eff_tx,
    h2=h_rx,
    R2=clutter_rx,
    area=path_type,
    d_v=distances,
    path_c=path_zones,
    pathinfo=terrain_info
)

print(f"Campo a 50km: {E_field:.2f} dBuV/m")
```

Além da Py1546, a biblioteca itur fornece implementações de recomendações auxiliares, como a P.453 (Índice de Refração) e P.837 (Precipitação), permitindo criar um ecossistema completo de predição climática e de propagação.

2.3.2 MATLAB: Ferramentas da Comunidade

No ambiente MATLAB, amplamente utilizado em universidades brasileiras (como Inatel, UnB, PUC-Rio), a implementação P1546FieldStrMixed.m é a referência. Ela se destaca pela capacidade de lidar com **percursos mistos** (terra-mar), aplicando o método de Millington implicitamente, o que é crucial para estações costeiras em cidades como Rio de Janeiro, Salvador ou Florianópolis.

O código MATLAB permite a passagem de argumentos opcionais como pares Nome-Valor, facilitando a configuração de parâmetros como TCA, Clutter Height e Location Percentage ($q\%$, que difere da porcentagem de tempo). A função validateP1546.m incluída no repositório é essencial para garantir que a implementação esteja em conformidade com os valores de teste oficiais da UIT.

3. Difração Determinística: O Método Deygout e a Correção de Assis

Enquanto a P.1546 oferece uma visão macroscópica e estatística, a engenharia de enlaces críticos e a análise de "zonas de sombra" exigem uma abordagem determinística. Aqui, o perfil exato do terreno entre transmissor e receptor é analisado para calcular a perda por difração.

3.1 Fundamentos da Difração por Gume de Faca (Knife-Edge)

A base de todos os modelos de difração é o princípio de Huygens-Fresnel. Quando uma onda eletromagnética encontra um obstáculo, cada ponto da frente de onda age como uma fonte secundária. A interferência dessas ondas secundárias no receptor determina a atenuação. O parâmetro fundamental é o parâmetro de difração de Fresnel-Kirchhoff, \lnu :

Onde:

- h é a altura do obstáculo acima da linha de visada (LOS). Se $h < 0$, há desobstrução (mas ainda pode haver perda se estiver dentro da 1ª zona de Fresnel).
- d_1 e d_2 são as distâncias do obstáculo aos terminais.
- λ é o comprimento de onda.

A perda por um único gume de faca, $J(\lnu)$, é dada pela aproximação de Lee (amplamente usada em código por evitar integrais complexas) :

- Se $\lnu \leq -0.7$: $J(\lnu) = 0$ dB.
- Se $-0.7 < \lnu \leq 0$: $J(\lnu) = 20 \log(0.5 - 0.62\lnu)$ dB.
- Se $0 < \lnu \leq 1$: $J(\lnu) = 20 \log(0.5 \cdot e^{-0.95\lnu})$ dB.
- Se $\lnu > 1$: $J(\lnu) = 20 \log(0.225 / \lnu)$ dB.

3.2 O Algoritmo de Deygout (1966)

Em terrenos reais, raramente existe apenas um obstáculo. O método de Deygout propõe uma solução hierárquica para múltiplos obstáculos:

- Busca do Obstáculo Principal:** Calcula-se ν para todos os pontos do perfil. O ponto com o maior ν é o obstáculo principal (M).
- Cálculo da Perda Principal:** Calcula-se a perda L_M considerando M como um gume de faca isolado.
- Subdivisão:** O perfil é dividido em dois segmentos: Transmissor $\rightarrow M$ e $M \rightarrow$ Receptor.
- Recursividade:** O processo é repetido para encontrar o obstáculo secundário em cada segmento. A perda desses obstáculos secundários é adicionada à perda principal (com correções empíricas em algumas variantes).

O método de Deygout é considerado mais preciso que o método de Bullington (que simplifica o terreno para um único obstáculo equivalente) e mais robusto que o Epstein-Peterson para obstáculos dominantes.

3.3 A Correção de Assis (1971): O Problema da Convexidade

A grande limitação do modelo de gume de faca é assumir que o topo da montanha é infinitamente fino. No Brasil, o relevo é frequentemente composto por morros arredondados ("meias-laranjas") e chapadões. Nesses casos, a onda não apenas difrata na aresta, mas desliza sobre a superfície curva, sofrendo uma atenuação adicional significativa.

Mauro Soares de Assis, em seu artigo seminal "A simplified solution to the problem of multiple diffraction over rounded obstacles" (IEEE Transactions on Antennas and Propagation, 1971), propôs integrar a teoria da difração por cilindros ao método de Deygout.

3.3.1 Formulação Matemática da Correção

A correção de Assis substitui a perda do gume de faca (L_{ke}) por uma perda total (L_{total}) que inclui um termo de curvatura (L_{cil}):

O cálculo do termo de curvatura baseia-se na Recomendação ITU-R P.526 para cilindros. A aproximação utilizada por Assis e consolidada na engenharia brasileira é dada por:

$$\$ L_{cil} \approx 11.7 \cdot \theta \cdot \sqrt{\frac{\pi R}{\lambda}} \quad (\text{para } \theta \cdot \sqrt{\frac{2\pi R}{\lambda}} < 1) \$$$

Ou, numa formulação mais robusta derivada das curvas da ITU-R P.526, a perda adicional (excesso sobre o gume de faca) depende de dois parâmetros adimensionais:

- Parâmetro de Curvatura (m):** Nota: Esta é uma simplificação. A forma rigorosa usa o raio da Terra equivalente. A formulação mais direta usada em implementações computacionais para o termo adicional de curvatura (em dB) é: Onde ρ é um índice de curvatura normalizado.

No entanto, a implementação "clássica" da correção de Assis em software de engenharia segue o seguinte algoritmo híbrido:

- Identificar o obstáculo dominante via **Deygout**.
- Extrair os pontos do terreno no entorno do pico dominante (janela de análise).
- Ajustar uma parábola ($y = ax^2 + bx + c$) a esses pontos para estimar o **Raio de Curvatura (R)** no vértice.
- Calcular o ângulo de difração total θ (soma dos ângulos de depressão de ambos os lados).
- Calcular a perda adicional de curvatura L_{add} usando a fórmula simplificada: (Atenção: Os coeficientes exatos variam levemente entre a publicação original de 1971 e adaptações modernas, mas o termo $(R/\lambda)^{1/3}$ é a assinatura física da

correção).

4. Estratégias de Implementação Computacional (Código)

Nesta seção, apresentamos como traduzir a teoria acima em código funcional. Focaremos na estrutura dos dados e nos algoritmos críticos.

4.1 Python: Implementação Deygout com Assis

Python é ideal para prototipagem rápida e análise de dados. Usaremos numpy para vetorização e scipy.optimize para o ajuste do raio de curvatura.

```
import numpy as np
from scipy.optimize import curve_fit

# Constantes Físicas
VELOCIDADE_LUZ = 3e8

def calcular_lambda(frequencia_mhz):
    return VELOCIDADE_LUZ / (frequencia_mhz * 1e6)

def perda_gume_faca_lee(nu):
    """
    Aproximação de Lee para a integral de Fresnel (Loss em dB).
    """
    if nu <= -0.7:
        return 0.0
    elif nu <= 0.0:
        return 6.0 + 9.0 * nu + 1.27 * (nu**2)
    elif nu <= 1.0:
        return 6.0 + 9.0 * nu - 1.27 * (nu**2)
    else:
        return 13.0 + 20.0 * np.log10(nu)

def estimar_raio_curvatura(distancias, alturas, indice_pico,
janela=5):
    """
    Estima o raio R ajustando uma parábola aos pontos locais do
    terreno.
    """
    y = ax^2 + bx + c -> R = 1 / |2a|
    """
    inicio = max(0, indice_pico - janela)
    fim = min(len(distancias), indice_pico + janela + 1)

    x_local = distancias[inicio:fim]
    y_local = alturas[inicio:fim]
```

```

if len(x_local) < 3:
    return 0.0 # Pontos insuficientes, trata como gume de faca

# Função quadrática para ajuste
def parabola(x, a, b, c):
    return a * x**2 + b * x + c

try:
    # Normaliza X para evitar instabilidade numérica
    x_norm = x_local - x_local[len(x_local)//2]
    popt, _ = curve_fit(parabola, x_norm, y_local)
    a = popt

    if abs(a) < 1e-10: # Terreno plano
        return 0.0

    R = 1.0 / abs(2 * a)
    return R
except:
    return 0.0

def correcao_assis(R, lambda_val, theta_rad):
    """
    Calcula a perda adicional por curvatura (Baseado em Assis/ITU-R P.526).
    """
    if R <= 1.0: # Raio desprezível, trata como gume
        return 0.0

    # Termo Chi da ITU-R P.526 (simplificado para Assis)
    # A fórmula exata pode envolver funções de Bessel/Hankel para precisão total,
    # mas a aproximação logarítmica é padrão em engenharia de radiodifusão.

    termo_geometrico = np.power( (np.pi * R) / lambda_val, 1/3 )
    chi = 11.7 * termo_geometrico * theta_rad

    # Perda adicional em dB
    # Se Chi for muito pequeno, a perda tende a zero.
    loss_curvatura = 6.0 * np.log10(1 + chi)

    return max(0.0, loss_curvatura)

def deygout_recursivo(dist, alt, tx_h, rx_h, lambda_val,
perfil_completo_d, perfil_completo_h):
    """
    Algoritmo de Deygout com injeção da Correção de Assis.

```

```

"""
nPontos = len(dist)
if nPontos <= 2:
    return 0.0

# 1. Definir Linha de Visada (LOS)
dTotal = dist[-1] - dist
inclina = (rx_h - tx_h) / dTotal
los = tx_h + inclina * (dist - dist)

# 2. Calcular Nu de Fresnel para todos os pontos
hObs = alt - los # Altura do obstáculo acima da LOS
d1 = dist - dist
d2 = dist[-1] - dist

# Evita divisão por zero nas pontas
with np.errstate(divide='ignore', invalid='ignore'):
    nuVec = hObs * np.sqrt(2 * (d1 + d2) / (lambda_val * d1 * d2))
nuVec = -99; nuVec[-1] = -99 # Ignora terminais

# 3. Encontrar Obstáculo Dominante
idxMax = np.argmax(nuVec)
nuMax = nuVec[idxMax]

if nuMax < -0.7: # Critério de desobstrução (0.6 da 1ª Zona de Fresnel)
    return 0.0

# 4. Calcular Perda Base (Knife Edge)
perdaPrincipal = perda_gume_faca_lee(nuMax)

# 5. --- APLICAÇÃO DA CORREÇÃO DE ASSIS ---
# Recupera índice global para análise de terreno
idxGlobal = np.where(perfilCompleto_d == dist[idxMax])

# Estima Raio (R)
R = estimar_raio_curvatura(perfilCompleto_d, perfilCompleto_h,
idxGlobal)

# Calcula ângulo de difração (theta)
# Theta ~ (hObs / d1) + (hObs / d2) para ângulos pequenos
theta = np.arctan(hObs[idxMax] / d1[idxMax]) +
np.arctan(hObs[idxMax] / d2[idxMax])

perdaCurvatura = correcao_assis(R, lambda_val, theta)
perdaNoNo = perdaPrincipal + perdaCurvatura
# -----

```

```

# 6. Recursão (Esquerda e Direita)
# Sub-trajeto TX -> Obstáculo
perda_esq = deygout_recursivo(dist[:idx_max+1], alt[:idx_max+1],
                                tx_h, alt[idx_max], lambda_val,
perfil_completo_d, perfil_completo_h)

# Sub-trajeto Obstáculo -> RX
perda_dir = deygout_recursivo(dist[idx_max:], alt[idx_max:],
                                alt[idx_max], rx_h, lambda_val,
perfil_completo_d, perfil_completo_h)

return perda_no_no + perda_esq + perda_dir

# Wrapper de Exemplo
def calcular_enlace(perfil_d, perfil_h, freq_mhz):
    # Alturas de antena (ex: Torre de 50m e Rx de 10m)
    tx_abs = perfil_h + 50.0
    rx_abs = perfil_h[-1] + 10.0
    lamb = calcular_lambda(freq_mhz)

    perda_difracao = deygout_recursivo(perfil_d, perfil_h, tx_abs,
rx_abs, lamb, perfil_d, perfil_h)

    # Perda Espaço Livre (FSPL)
    d_km = (perfil_d[-1] - perfil_d) / 1000.0
    fspl = 32.4 + 20*np.log10(d_km) + 20*np.log10(freq_mhz)

    return fspl + perda_difracao

```

4.2 C++: Desempenho para o Sistema Mosaico

Para processar milhares de radiais e gerar os contornos protegidos exigidos pelo Mosaico, a linguagem C++ é superior. A gestão de memória para carregar dados SRTM/ALOS é crítica. Abaixo, um esboço de uma estrutura de classes (PropEngine) otimizada. Recomenda-se o uso de bibliotecas como **GDAL** para leitura dos rasters (GeoTIFF) e **Eigen** para os cálculos matriciais de ajuste de curva (Assis).

```

#include <vector>
#include <cmath>
#include <algorithm>
#include <iostream>

// Constantes
const double SPEED_OF_LIGHT = 299792458.0;

struct TerrainPoint {
    double distance; // metros

```

```

        double elevation; // metros
    };

class DeygoutAssisCalculator {
private:
    double frequency_hz;
    double wavelength;

    double KnifeEdgeLoss(double nu) {
        if (nu <= -0.7) return 0.0;
        if (nu <= 0.0) return 6.0 + 9.0 * nu + 1.27 * nu * nu;
        if (nu <= 1.0) return 6.0 + 9.0 * nu - 1.27 * nu * nu;
        return 13.0 + 20.0 * log10(nu);
    }

    double EstimateRadius(const std::vector<TerrainPoint>& profile,
int peak_idx) {
        // Implementação simplificada de mínimos quadrados para y =
ax^2 + bx + c
        // Em produção, usar Eigen::HouseholderQR
        // Retorna R = 1 / |2a|
        //... (código de álgebra linear omitido por brevidade)
        return 5000.0; // Valor dummy para exemplo (R = 5km)
    }

    double CylinderCorrection(double R, double theta) {
        if (R < 10.0) return 0.0; // Gume de faca
        double term = (M_PI * R) / wavelength;
        double chi = 11.7 * pow(term, 1.0/3.0) * theta;
        return 6.0 * log10(1.0 + chi);
    }
}

public:
    DeygoutAssisCalculator(double freq_mhz) {
        frequency_hz = freq_mhz * 1e6;
        wavelength = SPEED_OF_LIGHT / frequency_hz;
    }

    double Calculate(const std::vector<TerrainPoint>& profile, double
tx_h, double rx_h) {
        return RecursiveDeygout(profile, 0, profile.size() - 1, tx_h,
rx_h);
    }

    double RecursiveDeygout(const std::vector<TerrainPoint>& profile,
int start_idx, int end_idx, double h_start, double h_end) {
        if (end_idx - start_idx < 2) return 0.0;

```

```

        double d_total = profile[end_idx].distance -
profile[start_idx].distance;
        double slope = (h_end - h_start) / d_total;

        double max_nu = -10.0;
        int max_idx = -1;

        // Loop principal - Pode ser paralelizado com OpenMP para
perfis muito longos
        // #pragma omp parallel for reduction(max: max_nu)
        for (int i = start_idx + 1; i < end_idx; ++i) {
            double d1 = profile[i].distance -
profile[start_idx].distance;
            double d2 = profile[end_idx].distance -
profile[i].distance;

            double los_h = h_start + slope * d1;
            double h_obs = profile[i].elevation - los_h;

            double nu = h_obs * sqrt(2.0 * (d1 + d2) / (wavelength *
d1 * d2));

            if (nu > max_nu) {
                max_nu = nu;
                max_idx = i;
            }
        }

        if (max_nu <= -0.7) return 0.0;

        double loss_main = KnifeEdgeLoss(max_nu);

        // Assis Correction
        double R = EstimateRadius(profile, max_idx);
        double d1_main = profile[max_idx].distance -
profile[start_idx].distance;
        double d2_main = profile[end_idx].distance -
profile[max_idx].distance;
        double h_obs_main = profile[max_idx].elevation - (h_start +
slope * d1_main);
        double theta = atan(h_obs_main / d1_main) + atan(h_obs_main / d2_main);

        double loss_curv = CylinderCorrection(R, theta);

        // Recursão
        double loss_left = RecursiveDeygout(profile, start_idx,
max_idx, h_start, profile[max_idx].elevation);

```

```

        double loss_right = RecursiveDeygout(profile, max_idx,
end_idx, profile[max_idx].elevation, h_end);

        return loss_main + loss_curv + loss_left + loss_right;
    }
};

```

4.3 MATLAB: Integração com Toolboxes

Para usuários de MATLAB, a abordagem recomendada é utilizar o Antenna Toolbox e o Mapping Toolbox para a gestão de dados geográficos, integrando scripts personalizados para a lógica de Assis.

- **P.1546:** Utilizar a função propagationModel introduzida nas versões recentes, que já suporta modelos ITU. Caso não disponível, o script P1546FieldStrMixed.m do repositório [GitHub - eeveetza/p1546](#) é a alternativa robusta.
- **Deygout:** O MATLAB não possui um "built-in" para Deygout com Assis. O engenheiro deve escrever a função recursiva (similar à lógica Python acima), mas tirando proveito das operações matriciais do MATLAB:

```
% Exemplo de vetorização em MATLAB para cálculo de Nu
D1 = distances - distances(1);
D2 = distances(end) - distances;
LOS = tx_h + (rx_h - tx_h).* (D1./ total_dist);
H_obs = terrain_profile - LOS;
Nu = H_obs.* sqrt(2 * (D1 + D2)./(lambda * D1.* D2));
[max_nu, idx] = max(Nu);
```

5. Regulação da ANATEL e o Sistema Mosaico

A implementação correta desses algoritmos visa, em última instância, a aprovação regulatória. O sistema **Mosaico** da Anatel impõe formatos rígidos e critérios de proteção que o software deve respeitar.

5.1 O Ecossistema Normativo

A **Resolução nº 721/2020** revogou regulamentos antigos e consolidou a estrutura técnica de radiodifusão. Ela é complementada por Atos de Requisitos Técnicos específicos:

- **Ato nº 9751 (TV Digital - SBTVD-T):** Define os parâmetros para TV Digital (ISDB-Tb). Revogou o Ato 3114.
- **Ato nº 3115/8104 (FM):** Define os requisitos para Radiodifusão Sonora em Frequência Modulada.

5.2 Relações de Proteção (D/U)

Um software de propagação deve verificar automaticamente se o sinal desejado (D) supera o sinal indesejado (U) pelas margens (Relações de Proteção) especificadas nas tabelas da

Anatel.

5.2.1 TV Digital (Ato nº 9751)

As relações de proteção variam drasticamente dependendo da natureza do interferente (Analógico ou Digital). Para o planejamento da TV Digital no Brasil, as tabelas críticas são :

Tabela 1: Relações de Proteção D/U para TV Digital (Co-canal e Adjacente)

Cenário de Interferência (Interferente → Vítima)	Co-canal (N)	Adjacente Inferior (N-1)	Adjacente Superior (N+1)
Digital → Digital	+19 dB	-36 dB	-36 dB
Analógico → Digital	+7 dB	-31 dB	-31 dB
Digital → Analógico	+34 dB	-11 dB	-11 dB

Insight Técnico: Note a robustez do sinal digital contra interferência analógica (exige apenas 7 dB de proteção co-canal), contrastando com a fragilidade do sinal analógico, que exige 34 dB de proteção contra um sinal digital. Isso explica a agressividade com que a TV Digital pode ser implantada em canais adjacentes, mas o cuidado extremo necessário ao planejar canais digitais próximos a legados analógicos.

5.2.2 FM (Ato nº 3115)

Para FM, a proteção considera até o segundo canal adjacente (devido à largura de banda de 200 kHz e o desvio de frequência).

Tabela 2: Relações de Proteção D/U para FM

Separação de Frequência	Relação de Proteção (D/U)
Co-canal (0 kHz)	28 a 30 dB
1º Adjacente (lpm 200 kHz)	6 dB
2º Adjacente (lpm 400 kHz)	-40 dB

5.3 Contornos Protegidos e Arquivos de Upload

Para submeter um projeto ao Mosaico, o engenheiro deve calcular o **Contorno Protegido**.

- **Definição:** É o lugar geométrico onde a intensidade de campo atinge o valor mínimo exigido (ex: 66 dBµV/m para FM Classe A, 51 dBµV/m para TV Digital UHF).
- **Curvas:**
 - **Analógico:** Usa-se E(50,50) (50% tempo, 50% locais).
 - **Digital:** Usa-se E(50,90) (90% tempo, 50% locais). A fórmula de conversão usual é $E(50,90) = E(50,50) - (E(50,10) - E(50,50))$ ou interpolações diretas da P.1546.
- **Formato de Arquivo (.txt):** O Mosaico exige o upload dos pontos do contorno. O formato típico é um arquivo de texto simples contendo azimute e distância, ou coordenadas geográficas.
 - **Padrão:** Azimutes de 0 a 355 graus (passo de 5 graus).
 - **Datum: SIRGAS 2000** (Obrigatório). O uso de WGS-84 ou SAD-69 gerará erros de validação posicional.

- **Formato TXT:** Geralmente colunas delimitadas: AZIMUTE; DISTANCIA_KM ou LAT; LON.

6. Conclusão

A implementação de ferramentas para cálculos de propagação no Brasil exige uma abordagem híbrida. O uso da **ITU-R P.1546** é mandatório para a definição legal da cobertura (contornos protegidos) e obediência aos ritos da Anatel. Entretanto, a topografia brasileira impõe o uso de métodos determinísticos como **Deygout com correção de Assis** para garantir que a cobertura teórica se traduza em recepção real e para resolver disputas de interferência em áreas de sombra.

Para o desenvolvedor, a escolha da linguagem (Python para agilidade, C++ para volume de dados) deve ser guiada pela escala do projeto. Contudo, a integridade dos dados de entrada (MDE confiável, Datum SIRGAS 2000) e a rigorosa adesão às tabelas de proteção dos Atos nº 9751 e 3115 são os fatores determinantes para o sucesso de qualquer projeto de engenharia de radiodifusão submetido ao sistema Mosaico.

Referências citadas

1. Ato nº 915, de 01 de fevereiro de 2024 - Anatel, <https://informacoes.anatel.gov.br/legislacao/atos-de-requisitos-tecnicos-de-gestao-do-espectro/2024/1920-ato-915>
2. P.1546 : Method for point-to-area predictions for terrestrial services in the frequency range 30 MHz to 4 000 MHz - ITU, <https://www.itu.int/rec/r-rec-p.1546>
3. eeveetza/Py1546: Python Implementation of Recommendation ITU-R P.1546 - GitHub, <https://github.com/eeveetza/Py1546>
4. eeveetza/p1546: MATLAB/Octave Implementation of Recommendation ITU-R P.1546, <https://github.com/eeveetza/p1546>
5. itur · PyPI, <https://pypi.org/project/itur/>
6. Comportamento de modelos de difração sobre múltiplos gumes de faca em VHF e UHF - Biblioteca da SBrT, <https://biblioteca.sbrt.org.br/articlefile/1197.pdf>
7. UNIVERSIDADE FEDERAL FLUMINENSE ESCOLA DE ..., https://app.uff.br/riuff/bitstream/handle/1/10575/dissertacao_andre_vinicius.pdf?sequence=1&isAllowed=y
8. Propagation over irregular terrain with and without vegetation - ITU, https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-P.1145-1990-PDF-E.pdf
9. Diffraction loss - Wireless Communication, <http://www.wirelesscommunication.nl/reference/chaptr03/diffrac.htm>
10. Field testing diffraction - CloudRF, <https://cloudrf.com/field-testing-diffraction/>
11. MANUAL – Propagación de las ondas radioeléctricas en sistemas terrenales móviles terrestres en las bandas de ondas métricas/decimétricas - ITU, https://www.itu.int/dms_pub/itu-r/opb/hdb/R-HDB-44-2002-OAS-MSW-S.doc
12. P.526-5 - Propagation by diffraction - ITU, https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.526-5-199708-S!!PDF-E.pdf
13. Recommendation ITU-R P.526-11, https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.526-11-200910-S%21%21PDF-E.pdf
14. Resolução nº 721, de 11 de fevereiro de 2020 - Anatel, <https://informacoes.anatel.gov.br/legislacao/resolucoes/2020/1383-resolucao-721>
15. Ato nº 9751, de 06 de julho de 2022 - Anatel, <https://informacoes.anatel.gov.br/legislacao/atos-de-requisitos-tecnicos-de-gestao-do-espectro/2022/1688-ato-9751>
16. Ato nº 3115, de 10 de junho de 2020 (REVOGADO) - Anatel, <https://informacoes.anatel.gov.br/legislacao/atos-de-requisitos-tecnicos-de-gestao-do-espectro/>

2020/1491-ato-3115 17. 2022 - Anatel,
<https://informacoes.anatel.gov.br/legislacao/atos-de-requisitos-tecnicos-de-gestao-do-espectro/>
2022 18. Manual de Mapeamento Simplificado para Identificação Preliminar e Priorização de
Talvezes Suscetíveis à Ocorrência,
https://www.gov.br/mdr/pt-br/centrais-de-conteudo/publicacoes/protecao-e-defesa-civil-sedec/2025_12_08_Manual_2ed_v_preliminar.pdf