

DESENVOLVIMENTO DE UM APLICATIVO PARA CÁLCULO DO ESFORÇO DE TRAÇÃO EM FERRAMENTAS ESTREITAS

ROGÉRIO RAMOS WEYMAR¹; MICHEL LOPES PAGNOSSIN²; CARLOS RICARDO TREIN²; ROBERTO LILLES TAVARES MACHADO², FABRÍCIO ARDAIS MEDEIROS²; ANTÔNIO LILLES TAVARES MACHADO³

¹Universidade Federal de Pelotas – rogerioweymar@gmail.com

²Universidade Federal de Pelotas – michelpagnossin@gmail.com

³Universidade Federal de Pelotas – antoniolilles@gmail.com (orientador)

1. INTRODUÇÃO

O preparo do solo é uma etapa essencial na agricultura, especialmente em Planossolos. Esses solos são caracterizados por uma camada superficial arenosa sobre um horizonte subsuperficial denso e de baixa permeabilidade, o que impõe desafios significativos para o manejo agrícola. O cálculo preciso do esforço de tração, necessário para as operações de preparo do solo nessas condições, é fundamental para otimizar o uso de energia, reduzir custos operacionais e minimizar a degradação do solo. A equação universal de mobilização do solo, proposta por Reece (1965), é amplamente utilizada para prever o esforço de tração em diferentes tipos de solo, sendo simples e eficiente. Modelos matemáticos como os de McKyes e Ali (1977), Godwin & Spoor (1977) e Grisso (1980) também permitem prever comportamentos que envolvem fatores complexos de múltiplas variáveis, as quais atuam entre si durante a ruptura do solo e produzem diversos efeitos simultâneos. Morinigo (2023) desenvolveu um aplicativo para telefone celular, baseado nos modelos de Reece (1965) e Godwin & Spoor (1977), obtendo bons resultados em termos de aplicabilidade. No entanto, a aplicação prática desses modelos no campo é frequentemente limitada pela complexidade dos cálculos e pela variabilidade das condições do solo. Com o intuito de superar esses desafios e democratizar o uso desses modelos para o cálculo do esforço de tração de ferramentas estreitas (como hastes de escarificadores e sulcadores de semeadoras), este trabalho apresenta o desenvolvimento de dois aplicativos: um em Python e outro em JavaScript, ambos baseados no modelo de Reece (1965), considerado um dos mais eficientes e simples (MACHADO, 2001). O objetivo é otimizar as operações de preparo do solo, proporcionando o cálculo preciso do esforço de tração, melhorando a eficiência energética e a qualidade do manejo do solo. Esses aplicativos visam preencher uma lacuna importante, oferecendo ferramentas simples e acessíveis tanto para agricultores quanto para engenheiros agrícolas e agrônomos.

2. METODOLOGIA

Esta pesquisa segue uma abordagem aplicada, focada no desenvolvimento de uma ferramenta prática para o cálculo do esforço de tração em ferramentas agrícolas estreitas, utilizando um modelo matemático consagrado. A pesquisa foi quantitativa, baseado no modelo proposto por Reece (1965), para calcular o esforço de tração. O processo metodológico foi estruturado em três etapas principais: implementação do modelo matemático, desenvolvimento do aplicativo em Python e JavaScript, e validação do sistema final.

A equação universal de mobilização do solo de Reece (1965) foi adotada como base para o cálculo do esforço de tração, devido à sua comprovada capacidade de prever o comportamento de ferramentas agrícolas estreitas em diferentes tipos de solo (MACHADO, 2001).

A linguagem Python foi utilizada na fase inicial do desenvolvimento, devido à sua eficiência em cálculos científicos e rápida prototipagem. O aplicativo foi inicialmente desenvolvido e validado no ambiente Google Colab (Fig.1), permitindo a prototipagem do modelo matemático, e a verificação ágil das fórmulas envolvidas no cálculo.

Para tornar a ferramenta acessível, a versão final do aplicativo foi implementada em HTML, CSS e JavaScript, (Fig. 2), visando uma solução diretamente no navegador, sem a necessidade de instalação de software adicional. O HTML foi utilizado para estruturar a interface do usuário, permitindo a inserção dos parâmetros de entrada e a exibição dos resultados de forma clara e organizada. O CSS foi empregado para estilizar a interface, utilizando design responsivo para garantir que o aplicativo se adapte a diferentes dispositivos, como desktops, tablets e smartphones, otimizando a experiência do usuário. E o JavaScript foi responsável por implementar a lógica de cálculo do esforço de tração, realizando os cálculos diretamente no navegador e exibindo os resultados de forma instantânea. A escolha de JavaScript se deu pela sua compatibilidade com navegadores modernos e pela capacidade de gerar resultados em tempo real, sem a necessidade de um servidor. Além disso, a biblioteca chart.js foi integrada para gerar gráficos comparativos das variações no esforço de tração em diferentes cenários operacionais, facilitando a visualização e interpretação dos resultados. A validação do aplicativo foi realizada testando diferentes cenários de entrada para verificar a consistência e precisão dos cálculos. A ferramenta foi testada em situações simuladas, variando parâmetros como profundidade, ângulo de ataque e características do solo. Os resultados obtidos foram comparados com valores teóricos e esperados encontrados na literatura.

```

trabalho_antonio_final.ipynb ☆
Arquivo Editar Ver Inserir Ambiente de execução Ferramentas Ajuda Última edição em 8 de setembro

+ Código + Texto

!pip install numpy matplotlib

import numpy as np
import matplotlib.pyplot as plt

# Função para calcular N_gamma
def calculate_N_gamma(alpha, beta, delta, phi):
    alpha_rad = np.radians(alpha)
    beta_rad = np.radians(beta)
    delta_rad = np.radians(delta)
    phi_rad = np.radians(phi)
    N_gamma = (1/2) * (1/np.tan(alpha_rad) + 1/np.tan(beta_rad)) / (np.cos(alpha_rad + delta_rad) + np.sin(alpha_rad + delta_rad) * 1/np.tan(beta_rad + phi_rad))
    return N_gamma

# Função para calcular N_c
def calculate_N_c(alpha, beta, delta, phi):
    alpha_rad = np.radians(alpha)
    beta_rad = np.radians(beta)
    delta_rad = np.radians(delta)
    phi_rad = np.radians(phi)
    N_c = (1 + 1/np.tan(beta_rad) * 1/np.tan(beta_rad + phi_rad)) / (np.cos(alpha_rad + delta_rad) + np.sin(alpha_rad + delta_rad) * 1/np.tan(beta_rad + phi_rad))
    return N_c

# Função para calcular N_q
def calculate_N_q(N_gamma):
    N_q = 2 * N_gamma
    return N_q

# Função para calcular P
def calculate_P(gamma, d, W, N_gamma, C, N_c, q, N_q):
    P = (gamma * 9.81 * d**2 * N_gamma + C * d * N_c + q * d * N_q) * W
    return P

```

Figura 1: printscreen da tela do Colab

Cálculo do Esforço de Tração

Ângulo de atrito interno do solo (°):	Ângulo de ataque da ferramenta (°):
<input type="text"/>	<input type="text"/>
Largura da ferramenta (m):	Profundidade de trabalho (m):
<input type="text"/>	<input type="text"/>
Ângulo de atrito solo-material (°):	Índice de coesão do solo (Pa):
<input type="text"/>	<input type="text"/>
Índice de adesão solo-material (Pa):	Densidade do solo (T/m ³):
<input type="text"/>	<input type="text"/>

Resultado:

Comparação de Medições e Gráficos

Figura 2: printscreen da tela do aplicativo web

3. RESULTADOS E DISCUSSÃO

O desenvolvimento e a validação do aplicativo trouxeram avanços tanto em termos de precisão quanto de praticidade. O aplicativo ofereceu uma alternativa ágil e econômica, permitindo simular diferentes cenários operacionais e ajustar variáveis como o ângulo de ataque e a profundidade de trabalho. Uma das maiores contribuições do aplicativo é sua facilidade de uso. Com uma interface intuitiva e acessível via navegador, o tempo médio para realizar os cálculos foi inferior a dois minutos, representando uma significativa redução em comparação com os métodos tradicionais. Além disso, a implementação de gráficos dinâmicos facilitou a interpretação dos resultados, proporcionando uma visão clara das variações no esforço de tração entre diferentes condições de operação. Essa agilidade, aliada à precisão dos cálculos, torna o aplicativo uma ferramenta acessível tanto para agricultores quanto para engenheiros agrícolas e agrônomos, facilitando a tomada de decisões rápidas e precisas no campo. A adaptabilidade do aplicativo a dispositivos móveis amplia ainda mais sua aplicabilidade, permitindo o uso em diversas plataformas, desde computadores até smartphones.

Apesar dos resultados positivos, a aplicação do modelo matemático precisa ser ampliada para incluir outros tipos de solo e condições operacionais. Embora o desempenho em Planossolos tenha sido satisfatório, ainda é necessária uma validação empírica com dados de campo em diferentes solos, teores de água e profundidades, o que permitirá aprimorar a confiabilidade do aplicativo em uma variedade maior de cenários. Adicionalmente, a inclusão de mais variáveis relacionadas ao solo e às características das ferramentas agrícolas pode aumentar ainda mais a precisão dos cálculos. O aplicativo oferece um ponto de partida promissor para o uso de tecnologia no cálculo do esforço de tração. Futuras expansões podem incluir a integração com sensores de solo, coletando dados em tempo real, ou o uso de inteligência artificial para prever cenários de forma mais dinâmica e precisa. Tais aprimoramentos fariam da ferramenta um recurso ainda mais poderoso no contexto da agricultura de precisão, promovendo maior eficiência e sustentabilidade nas operações agrícolas.

4. CONCLUSÕES

A inovação central deste trabalho reside na simplificação de um processo que envolve cálculos complexos e demorados, oferecendo uma solução prática e acessível tanto para agricultores quanto para engenheiros agrícolas e agrônomos. Ao utilizar tecnologias web, como JavaScript, HTML e CSS, foi possível integrar esses cálculos em uma plataforma intuitiva, eliminando a necessidade de softwares adicionais. Essa abordagem facilita a implementação de soluções diretamente no campo. A principal contribuição deste trabalho está na democratização do cálculo do esforço de tração, tornando-o não apenas mais rápido e acessível, mas também flexível às necessidades dos usuários, sem comprometer a precisão dos resultados.

5. REFERÊNCIAS BIBLIOGRÁFICAS

GODWIN, R. J.; SPOOR. G. **Soil failure with narrow tines**. Journal of Agricultural Engineering Research, 1977. 22(4):213-228.

GRISSE, R. D., PERUMPRAL, J. V., & DESAI, C. S. **A soil e tool interaction model for narrow tillage tools**. St. Joseph, MI 49085: ASAE.p.80-1518. 1980.

MACHADO, A.L.T. **Esforço de tração para ferramentas de hastes com ponteiros estreitas em dois solos do Rio Grande do Sul**. 2001. 176 f. Tese (Doutorado em Ciência do Solo) - Faculdade de Agronomia, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2001.

MCKYES, E.; ALI, O. S. **The cutting of soil by narrow blades**. Journal of Terramechanics, Oxford, v. 14, n. 2, p. 43-58, 1977.

MORINIGO, E.E.V. **Modelo matemático simplificado para previsão do esforço de tração em ferramentas estreitas de mobilização do solo**. 2023. 137p. Tese (Doutorado em Agronomia) – Programa de Pós-Graduação em Sistemas de Produção Agrícola Familiar, Faculdade de Agronomia Eliseu Maciel, Universidade Federal de Pelotas, Pelotas, 2023.

REECE, A. R. **The fundamental equation of earth-moving mechanics**. In Symposium on Earth-Moving Machinery, Proceedings of the Institution of Mechanical Engineers, 179(3F): p.16-22. 1965.