Reflections on Statistical Interpretation in CSSM-Based Geodynamic Modeling
Reviewer's Commentary

The revised manuscript makes several technically sound and well-supported observations regarding the influence of material strength and basal friction on the development of back-thrusts. These conclusions are consistent with critical taper theory and the results of prior experimental and numerical studies.
However, I wish to offer a reflection on a deeper methodological concern—one that goes beyond this particular paper and touches on a broader issue in the geodynamic modeling community.

The Nature of the System Under Study

The fold-and-thrust belt systems modelled here are governed by non-linear stress redistribution, critical taper dynamics, and path-dependent failure processes. The introduction of a critical state soil model, combined with large-strain elasto-plastic behavior, suggests the authors are simulating a system with characteristics of a complex system: sensitive to initial and boundary conditions, driven by conservation laws, and capable of emergent behavior such as spontaneous fault reorganization or back-thrust formation.
This implies the system is not purely deterministic, nor entirely stochastic, but something in between—a hallmark of what some would call a self-organized critical system.

The Limits of Simple Statistical Tools

In this context, the authors' use of Excel's CORREL function to evaluate pairwise correlations between ranked output categories (1 = none, 2 = weak, 3 = strong) and input parameters is problematic. While the intention to quantify parameter influence is commendable, this method:

Treats a ranked ordinal variable as though it were interval-scale,
Ignores the covariance and interdependence among inputs,
Implies univariate influence in a system whose behavior is inherently multivariate and emergent.

Such an approach risks drawing overly simplistic conclusions from what is fundamentally a high-dimensional and nonlinear process.

A Better Framework: Multivariate Thinking

Rather than correlating parameters with ranked outputs, I would encourage the use of multivariate regression analysis (linear or nonlinear), augmented with:

Variance Inflation Factor (VIF): to diagnose multicollinearity,
Standardized coefficients: to explore relative influences in standardized space,
SHAP values (Shapley Additive Explanations): to assess feature importance in a model-agnostic and interpretable way.

These tools do not “solve” the problem of emergent behavior—but they can help map the multidimensional parameter space and identify joint patterns of influence without assuming linear causality.

Further Caveat

The issue of scale—specifically the distinction between ordinal and interval-scale variables—is not fully resolved by the statistical methods used here. The outcome variable against which model parameters are analyzed is a dimensionless rank of back-thrust expression. These values are discrete and carry no inherent metric scale. As such, standard correlation or regression methods may not be strictly appropriate.

Proper treatment of ordinal outcomes typically requires alternative methods, such as ordinal logistic regression, which account for the ordering without assuming uniform spacing between categories. Another possibility would be to recast the outcome variable in terms of a continuous quantity—such as the amount of back-thrust displacement—if that data is available.

Nevertheless, the multivariate analysis performed here can still provide qualitative insight into patterns of parameter influence. The results should be interpreted in that light: as exploratory rather than predictive or definitive.

On Fairness and the State of the Field

It would be unfair to single out this paper. On the contrary, I recognize that this manuscript is ahead of many others in attempting quantitative sensitivity analysis at all. However, this step also exposes the statistical shallowness of common practice in geodynamic simulation work. Across the literature, we routinely simulate complex systems and then analyze them with tools that assume simplicity.

My intention is not to criticize this paper alone, but to suggest that we—as a community—need to reconsider how we interpret simulation results when the systems themselves are governed by emergent phenomena.

Conclusion

The conclusions of this study regarding material strength, basal friction, and thrust development are reasonable. But the statistical methods used to rank and relate model outcomes to input parameters do not reflect the complexity of the underlying system. In systems where emergent behavior is expected, we should be cautious in assigning causality and instead adopt tools that respect the interdependent and dynamic nature of such models.

Appendix: Code Reference

I have prepared a Python implementation for multivariate regression analysis, variance inflation assessment, partial correlation calculation, and SHAP-based feature attribution.

```
import numpy as np
import pandas as pd
import statsmodels.api as sm
from sklearn.preprocessing import StandardScaler
from statsmodels.stats.outliers_influence import variance_inflation_factor
from scipy.stats import pearsonr
import shap

# === Load Data from CSV ===
# Replace 'data.csv' with your actual filename
# File should have columns: X1, X2, X3, X4, Y
df = pd.read_csv('data.csv')

# === A) Correlation Matrix ===
correlation_matrix = df.corr()

# === B) Standardized Regression Coefficients ===
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df[['X1', 'X2', 'X3', 'X4']])
Y_scaled = scaler.fit_transform(df[['Y']])

X_scaled_with_const = sm.add_constant(X_scaled)
model = sm.OLS(Y_scaled, X_scaled_with_const).fit()
standardized_coefficients = model.params[1:]

# === C) Variance Inflation Factor (VIF) ===
vif_data = pd.DataFrame()
vif_data["Feature"] = ['X1', 'X2', 'X3', 'X4']
vif_data["VIF"] = [variance_inflation_factor(X_scaled_with_const, i+1) for i in range(4)]

# === D) Partial Correlations ===
partial_corrs = {}
for col in ['X1', 'X2', 'X3', 'X4']:
other_vars = [c for c in ['X1', 'X2', 'X3', 'X4'] if c != col]
model = sm.OLS(df[col], sm.add_constant(df[other_vars])).fit()
residuals_x = model.resid
model_y = sm.OLS(df['Y'], sm.add_constant(df[other_vars])).fit()
residuals_y = model_y.resid
partial_corrs[col] = pearsonr(residuals_x, residuals_y)[0]

# === E) SHAP Values ===
X = sm.add_constant(df[['X1', 'X2', 'X3', 'X4']])
ols_model = sm.OLS(df['Y'], X).fit()

def model_predict(X_input):
```

```
return ols_model.predict(sm.add_constant(X_input))

explainer = shap.Explainer(model_predict, df[['X1', 'X2', 'X3', 'X4']])
shap_values = explainer(df[['X1', 'X2', 'X3', 'X4']])
shap_importance = np.abs(shap_values.values).mean(axis=0)

# === Summary Results ===
print("\n--- Correlation Matrix ---")
print(correlation_matrix)

print("\n--- Standardized Coefficients ---")
print(standardized_coefficients)

print("\n--- Variance Inflation Factor (VIF) ---")
print(vif_data)

print("\n--- Partial Correlations ---")
for key, value in partial_corrs.items():
print(f"{key}: {value:.3f}")

print("\n--- SHAP Feature Importance (Mean Absolute) ---")
for feature, value in zip(['X1', 'X2', 'X3', 'X4'], shap_importance):
print(f"{feature}: {value:.3f}")
```

***to use the code
Prepare a CSV file (e.g. data.csv) with columns labeled exactly as: X1, X2, X3, X4, Y.
Each row is one data point.
Save it in the same folder as the script or adjust the file path accordingly.

Other points about the paper

The original series of suggestions I made have been fairly thoroughly implemented so presentation and ease of reading are now significantly improved.