

Optimierung eines Berechnungsalgorithmus eines Modells verpinnter DCB unter Nutzung von ChatGPT-4

Optimization of a computational algorithm of a model of pinned DCB with use of ChatGPT-4

(Bachelor-/ Studien-/ Masterarbeit)

Die Rechendauer der Implementierung analytischer Modelle in einer Programmiersprache mag bei einigen wenigen Ausführungen nicht maßgeblich sein. Sollen aber beispielsweise umfangreiche Parameterstudien oder Optimierungen durchgeführt werden, ergeben sich bei ineffizienter Programmierung teils große Rechenzeiten. Mithilfe von Profiling lässt sich identifizieren, welche Teile des Codes welche Anteile an der Gesamtrechenzeit haben und somit mögliche Ansatzpunkte für Codeoptimierungen bieten. ChatGPT-4 der Firma OpenAI kann bei richtigem Einsatz ein nützliches Werkzeug für die Wissenschaft sein. Gerade beim Programmieren kann ChatGPT hilfreiche Beiträge leisten und Vorgehensweisen vereinfachen.

```
def calculate_constants_eqs(lambda_tilde):
    lambda_1_tilde, lambda_2_tilde = lambda_tilde
    eq1 = C1_sym * C2_sym + 0.5 * W_tilde0_sym
    eq2 = -C1_sym * lambda_1_tilde**2 + C3_sym * lambda_1_tilde**2
    eq3 = (
        -C1_sym * lambda_1_tilde * c_tilde_sym
        + C2_sym * lambda_2_tilde * c_tilde_sym
        + C3_sym * cosh(lambda_1_tilde * c_tilde_sym)
        + C4_sym * sinh(lambda_1_tilde * c_tilde_sym)
        + 0.5
    )
    eq4 = (
        -lambda_2_tilde * c_tilde_sym
        + lambda_2_tilde * lambda_1_tilde * c_tilde_sym
        + lambda_2_tilde * c_tilde_sym
    )
    Constant = solve([eq1, eq2, eq3])
    return Constant

def calculate_cn(
    lambda_tilde,
    Cn_list,
    eqs,
    eq_subs,
    c_tilde_sym,
    c,
    ap_tilde_sym,
    ap,
    subf,
    cn_expr,
    evalf,
):
    Cn_vals = []
    for n in range(1, len(Cn_list)):
        Cn = Cn_list[n]
        Cn_eq = eqs[n][Cn]
        Cn_val = eq_subs(Cn_tilde_sym, c).subs(ap_tilde_sym, ap).subs(
            subf, evalf(Cn_expr)
        )
        Cn_vals.append(Cn_val)
    return Cn_vals[0], Cn_vals[1], Cn_vals[2], Cn_vals[3]
```

In dieser Arbeit soll ein vorliegendes und in Python implementiertes analytisches Modell zur Beschreibung des Modell Risswachstums eines Double-Cantilever-Beams (DCB) mit einer z-Pin Rissüberbrückung optimiert werden. Dazu muss das zugrundeliegende Modell und dessen mathematischer Lösungsansatz verstanden sein und dessen Umsetzung im Code mithilfe von Profilingtools untersucht werden. Der algorithmische Lösungsansatz soll entsprechend optimiert werden. Dazu soll eine objektorientierte Programmierung verwendet werden. Dies erlaubt dann auch die gezielte Nutzung des Modells in parametrischen Studien. Neben der Verbesserung des Algorithmus ist es ein Ziel dieser Arbeit, die Potentiale des Einsatzes von ChatGPT-4 für das Profiling und die Optimierung von Codes zu erschließen und kritisch zu diskutieren.

Es werden gute Mechanikkenntnisse sowie grundlegende Programmierkenntnisse vorausgesetzt