Shaikh & Tonak (1994) Code Explainer

Methodology, Formula, and Code Side-by-Side

September 28, 2025

Abstract

This document provides a detailed breakdown of the key formulas and methodologies in Shaikh & Tonak (1994), showing the original book descriptions, the mathematical formulation in LaTeX, and the corresponding Python implementation.

1 Unified Capital Stock (K)

Book Description The book uses specific capital stock measures (KK for 1958–1973, K for 1974–1989) without interpolation between periods.

$$K_t = \begin{cases} KK_t & \text{if } t \leq 1973 \\ K_t & \text{if } t \geq 1974 \end{cases}$$

Python Implementation

```python
def create_unified_capital_series(self, df):
    K_unified = pd.Series(index=df.index, dtype=float, name='
    ↪ K_unified')
    # Use KK for 1958-1973
    for year in df.index:
        if year <= 1973:
            K_unified.loc[year] = df.loc[year, 'KK']
    # Use K for 1974-1989
    for year in df.index:
        if year >= 1974:
            K_unified.loc[year] = df.loc[year, 'K']
    return K_unified
```

2 Profit Rate (r)

Book Description The profit rate r' is reproduced with very small error using $r_t = SP_t/(K_t\, u_t)$. This suggests the operational definition in Table 5.4 matches an SP-based construction, not the textbook $s'/(1 + c')$ identity.

$$r_t = \frac{SP_t}{K_t \times u_t}$$

Python Implementation

```python
def calculate_marxian_profit_rate(self, df):
    SP = df.get('SP')
    K_unified = self.create_unified_capital_series(df)
    u = df.get('u')

    denominator = K_unified * u
    mask = (SP.notna()) & (denominator != 0)
    r_exact = pd.Series(index=df.index, dtype=float)
    r_exact.loc[mask] = SP.loc[mask] / denominator.loc[mask]

    return r_exact
```

3 Organic Composition of Capital (c')

Book Description The book's tables provide the organic composition of capital directly as c'. The replication uses these values without recalculation.

$$q_t = c'_t$$

Python Implementation

```python
def calculate_organic_composition(self, df):
    c_exact = df.get("c'")
    return c_exact
```

4 Rate of Surplus Value (s')

Book Description Similar to the organic composition, the rate of surplus value is provided directly in the book's tables as s'.

$$s'_t = \frac{S_t}{V_t} \quad \text{(from book data)}$$

Python Implementation

```python
def calculate_surplus_value_rate(self, df):
    svv_exact = df.get("s'")
    return svv_exact
```