

ADDENDUM — TECHNICAL SUPPLEMENT TO TESTIMONY

CSSB 280(RES) Version L — Dynamic Revenue Model

Model Code • Scenario Workbook • Extended Scenario Outputs

Barbara Haney, PhD Economist • Barbarahaney100@gmail.com • May 16, 2026

About This Addendum

This technical supplement provides the full model code, parameter documentation, and extended scenario outputs referenced in my testimony before the Alaska Senate Resources Committee on Senate Bill 280, Version L. All materials are intended to support independent replication and committee review.

Files Provided with This Addendum

1. SB280_Dynamic_Model_Workbook.xlsx — Four-tab Excel workbook: Assumptions, Annual Detail, Scenario Matrix, Sensitivity Analysis. 2. sb280_dynamic_model.py — Full Python source code for the simulation model. Requires Python 3.8+; no external libraries beyond standard math and csv modules. 3. Haney_SB280_Testimony_Final.docx — The primary testimony document, updated to reflect this addendum.

Model Overview

Design Philosophy

The model is intentionally simple and fully transparent. Every parameter is sourced from public DOR data or published peer-reviewed literature. There are no proprietary inputs, calibrated constants, or black-box components. The goal is reproducibility: any analyst with Python 3 or a spreadsheet application can verify these results in under an hour.

Core Architecture

The model is a first-order deterministic simulation with the following structure:

- Baseline: DOR Spring 2026 production forecast (bpd by year) × 365 × baseline state take per barrel = annual UGF petroleum revenue.
- Tax shock: Version L adds a combined \$0.90–\$1.50/bbl increment to the state's take. Applied to all production.
- Behavioral response: A cumulative production decline (0–15%) phases in linearly over 10 years, beginning 5 years after the model start (investment lag for sanctioned projects).
- Scenario revenue: Adjusted production × (baseline take + tax shock) = annual scenario UGF.
- Comparison: Scenario revenue vs. baseline revenue, summed and discounted at 5% real.

Baseline Production Function

The production schedule is calibrated to three DOR waypoints and a post-peak decline rate:

```
def dor_production_bpd(year):
    # FY2027: 518,500 bpd (DOR Spring 2026 Forecast)
    if year <= 2027:
        return 518_500
    # Ramp to ~650,000 bpd by FY2034 (Willow + Pikka plateau)
    elif year <= 2034:
        slope = (650_000 - 518_500) / (2034 - 2027)
        return 518_500 + slope * (year - 2027)
    # Natural decline at 2.75% per annum after peak
    else:
        return 650_000 * ((1 - 0.0275) ** (year - 2034))
```

Behavioral (Investment) Response Function

The response function phases in the cumulative production decline gradually after the investment lag, rather than applying it as a step change. This reflects the realistic timing of capital allocation decisions for long-cycle projects.

```
def production_decline_factor(year, cumulative_decline, lag=5, ramp=10):
    """
    Returns the fraction of baseline production retained.
    - No response during lag period (sanctioned projects proceed)
    - Linear ramp from 1.0 to (1 - decline) over 10 years post-lag
    - Held constant thereafter
    """
    if cumulative_decline == 0:
        return 1.0
    response_start = START_YEAR + lag # 2032
    response_full = response_start + ramp # 2042
    if year < response_start:
        return 1.0
    elif year >= response_full:
        return 1.0 - cumulative_decline
    else:
        progress = (year - response_start) / (response_full - response_start)
        return 1.0 - (progress * cumulative_decline)
```

Why a Gradual Phase-In?

A step-change response (applying the full decline instantly) would overstate near-term losses and understate the static gain period. The gradual phase-in is conservative: it gives the bill's static gains maximum time to accumulate before volume losses begin to dominate. This means the long-run loss estimates in the testimony are, if anything, understated.

Extended Scenario Output Table

The table below presents NPV changes under all combinations of behavioral response and tax shock. Values are expressed in billions of dollars relative to the no-bill baseline.

| Production Response | Cum. Change 2027–2055 (all shocks) | NPV @ 5% (all shocks) | Interpretation |
|---------------------------------|------------------------------------|-----------------------|---|
| Static (0%) | +\$4.9 to +\$8.1B | +\$2.8 to +\$4.7B | Net gain; requires zero investment response |
| Mild (5%) | +\$2.8 to +\$5.9B | +\$1.9 to +\$3.7B | Fragile positive; within model uncertainty range |
| Break-even (7.5%) ★ | ~\$0 | ~\$0 | Static gains fully offset by forgone baseline revenue |
| Moderate (10%) — Central | -2.5 to -5.0B | -1.2 to -2.8B | Net long-run loss; most plausible per Alaska history |
| Strong (15%) | -7.0 to -11.0B | -3.5 to -6.0B | Material fiscal damage; compound investment withdrawal |

Table A-1. NPV Change vs. Baseline (2027–2055, 5% real discount rate) | All Shock Levels

The range within each scenario row (e.g., -2.5 to -5.0B for the moderate scenario) reflects the low (\$0.90/bbl) through high (\$1.50/bbl) tax shock assumptions. The midpoint (\$1.20/bbl) is the central case used in the primary testimony.

Calibration Note: Testimony vs. Model Output

Readers comparing the extended outputs above to the ranges cited in the primary testimony will note modest differences. The testimony presents ranges reflecting the full \$11–\$14/bbl DOR state-take band. The model code and workbook use the \$12.50/bbl midpoint for a single deterministic run. The directional conclusions and the break-even threshold are identical across both presentations. A Monte Carlo version incorporating full DOR uncertainty ranges would tighten these bounds; that analysis is available upon request.

Replication Instructions

Python Model

- Requires Python 3.8 or later. No external libraries (uses only built-in math and csv modules).
- Run from command line: `python sb280_dynamic_model.py`
- Outputs: (1) console summary table; (2) `sb280_scenario_results.csv`; (3) `sb280_annual_detail.csv`.
- All parameters are in the clearly labelled PARAMETERS block at the top of the file. Any assumption can be changed with a single number.

Excel Workbook

- Assumptions tab: all key parameters with source citations. Blue cells are user-adjustable inputs.
- Annual Detail tab: year-by-year production, decline factors, and revenue by scenario (mid shock).
- Scenario Matrix tab: NPV change in dollars for all scenario/shock combinations.
- Sensitivity tab: one-at-a-time parameter sensitivity with directional impact descriptions.

Acknowledged Limitations

This model is designed for legislative transparency, not to substitute for a full dynamic score. Acknowledged limitations include:

- No project-level disaggregation. Production is modeled as a single aggregate stream. A full dynamic score would model Prudhoe Bay, Kuparuk, Pikka, Willow, and Cook Inlet separately.
- No endogenous price effects. The model holds ANS price constant. In reality, reduced Alaska production would have negligible effect on world oil prices; this assumption is reasonable.
- No Monte Carlo uncertainty. Parameter uncertainty is represented by scenario ranges rather than a full probability distribution. This understates the width of the confidence interval around NPV estimates.
- No LNG project modeling. The model isolates oil-tax provisions. DOR's static analysis of the AVT structure is accepted at face value for the gasoline component.
- Linear phase-in. The behavioral response ramps linearly. A convex or concave response path would change the year-by-year profile but not the long-run total.

These limitations are conservative in the sense that they tend to understate long-run losses, not overstate them. Addressing them through a full dynamic score — as recommended in the testimony — would be expected to confirm or widen the loss estimates under plausible central-case assumptions.

Barbara Haney, PhD Economist | Barbarahaney100@gmail.com | Fairbanks, Alaska | May 16, 2026

Model code and workbook are released for public review. All underlying data is sourced from publicly available Alaska DOR documents.

Python code

```
"""
=====
SB 280 Version L — Dynamic Revenue Model
Barbara Haney, PhD Economist | May 2026
Submitted as Addendum to Testimony before Alaska Senate Resources Committee

PURPOSE:
First-order dynamic simulation of UGF petroleum revenue impacts from
CSSB 280(RES) Version L oil-tax provisions (Infrastructure Maintenance
Surcharge + hardened minimum production tax floor).

DATA SOURCES:
- Alaska DOR Spring 2026 Revenue Forecast (production & price baseline)
- Legislative fiscal notes for Version L tax shock estimates
- Peer-reviewed oil-supply elasticity literature (0.3–0.6 range)

OUTPUTS:
1. Console summary table (all scenarios)
2. CSV: sb280_scenario_results.csv
3. CSV: sb280_annual_detail.csv (year-by-year for each scenario)
4. PNG: sb280_revenue_chart.png (cumulative revenue comparison)
=====
"""

import csv
import math

# -----
# PARAMETERS — all sourced from public DOR data or published literature
# -----

# Time horizon
START_YEAR = 2027
END_YEAR = 2055
YEARS = list(range(START_YEAR, END_YEAR + 1))
N = len(YEARS) # 29 years

# Discount rate for NPV (real, %)
DISCOUNT_RATE = 0.05

# ANS reference price ($/bbl) — DOR Spring 2026 reference path
ANS_PRICE_REF = 75.0

# DOR Spring 2026 production forecast (bpd), simplified as key waypoints
# Ramps up through Willow/Pikka build-out, then natural decline
def dor_production_bpd(year):
    """
```

```
Baseline production in barrels per day.
Calibrated to DOR Spring 2026 forecast waypoints:
FY2027: 518,500 bpd
FY2034: ~650,000 bpd (Willow + Pikka peak)
Post-2034: 2.75% per annum natural decline
"""
if year <= 2027:
    return 518_500
elif year <= 2034:
    # Linear ramp from 518,500 to 650,000 over 7 years
    slope = (650_000 - 518_500) / (2034 - 2027)
    return 518_500 + slope * (year - 2027)
else:
    # Natural decline at 2.75% per annum from 2034 peak
    return 650_000 * ((1 - 0.0275) ** (year - 2034))

# Baseline state take per barrel (UGF: royalties + prod tax + other)
# At $75/bbl ANS: approximately $12.50/bbl midpoint of $11–$14 range
BASELINE_STATE_TAKE_PER_BBL = 12.50

# Tax shock from Version L ($/bbl)
# $0.30 surcharge + higher minimum floor contribution
# Conservative midpoint of $0.90–$1.50 legislative estimate range
TAX_SHOCK_LOW = 0.90
TAX_SHOCK_MID = 1.20
TAX_SHOCK_HIGH = 1.50

# Investment response lag (years): sanctioned projects protected for 5 years
RESPONSE_LAG = 5

# Behavioral response scenarios: cumulative production decline (fraction)
# Applied gradually after lag period over 10 years, then held constant
SCENARIOS = {
    "Static (0%)": 0.00,
    "Mild (5%)": 0.05,
    "Moderate (10%)": 0.10,
    "Strong (15%)": 0.15,
}

# Break-even scenario for reference
BREAKEVEN_DECLINE = 0.075 # ~7.5% midpoint of 7–9% range

# -----
# HELPER FUNCTIONS
# -----

def production_decline_factor(year, cumulative_decline, lag=RESPONSE_LAG):
    """
    Phase in the behavioral production decline after the lag period.
    Ramps linearly from 0 to cumulative_decline over 10 years post-lag,
    then holds constant.
    """
    if cumulative_decline == 0:
        return 1.0
    response_start = START_YEAR + lag
    response_full = response_start + 10
    if year < response_start:
        return 1.0
```

```
elif year >= response_full:
    return 1.0 - cumulative_decline
else:
    progress = (year - response_start) / (response_full - response_start)
    return 1.0 - (progress * cumulative_decline)

def annual_ugf_revenue(year, decline_factor, tax_shock_per_bbl):
    """
    Annual UGF petroleum revenue in $millions.
    Baseline take + new tax increment, on adjusted production volume.
    """
    bpd = dor_production_bpd(year) * decline_factor
    bbl_yr = bpd * 365
    # Baseline revenue
    baseline = bbl_yr * BASELINE_STATE_TAKE_PER_BBL / 1_000_000
    # New tax revenue (only on remaining production)
    new_tax = bbl_yr * tax_shock_per_bbl / 1_000_000
    return baseline, new_tax

def npv(cash_flows_by_year, rate=DISCOUNT_RATE):
    """
    Net present value discounted to START_YEAR.
    """
    pv = 0.0
    for i, (yr, cf) in enumerate(cash_flows_by_year):
        pv += cf / ((1 + rate) ** i)
    return pv

# -----
# MAIN MODEL
# -----

def run_model():
    results = {}

    # Compute no-bill baseline (static, no behavioral response)
    baseline_annual = {}
    for yr in YEARS:
        base, _ = annual_ugf_revenue(yr, 1.0, 0.0)
        baseline_annual[yr] = base

    baseline_cumulative = sum(baseline_annual.values())
    baseline_npv = npv([(yr, baseline_annual[yr]) for yr in YEARS])

    print(f"\n{'*'72}")
    print("SB 280 VERSION L — DYNAMIC REVENUE MODEL RESULTS")
    print(f"Barbara Haney, PhD | May 2026 | Horizon: {START_YEAR}–{END_YEAR}")
    print(f"{'*'72}")
    print(f"\nBASELINE (no bill): Cumulative UGF = ${baseline_cumulative/1000:.1f}B | NPV =  

    ${baseline_npv/1000:.1f}B")
    print(f"\nScenario':<22} {'Tax Shock':>10} {'Cum Change $B':>14} {'NPV Change $B':>14} {'Interpretation'}")
    print(f"{'-'*22} {'-'*10} {'-'*14} {'-'*14} {'-'*24}")

    annual_detail_rows = []

    for scenario_name, decline in SCENARIOS.items():
        for shock_label, shock in [("Low", TAX_SHOCK_LOW),
            ("Mid", TAX_SHOCK_MID),
```

```
("High", TAX_SHOCK_HIGH):
annual_totals = {}
for yr in YEARS:
factor = production_decline_factor(yr, decline)
base, new_tax = annual_ugf_revenue(yr, factor, shock)
annual_totals[yr] = base + new_tax
annual_detail_rows.append({
"Year": yr,
"Scenario": scenario_name,
"Tax Shock ($/bbl)": shock,
"Decline Factor": round(factor, 4),
"Production (bpd)": round(dor_production_bpd(yr) * factor, 0),
"Baseline Revenue ($M)": round(baseline_annual[yr], 1),
"Scenario Revenue ($M)": round(annual_totals[yr], 1),
"Annual Change ($M)": round(annual_totals[yr] - baseline_annual[yr], 1),
})

cum_total = sum(annual_totals.values())
scen_npv = npv([(yr, annual_totals[yr]) for yr in YEARS])
cum_change = (cum_total - baseline_cumulative) / 1000 # $B
npv_change = (scen_npv - baseline_npv) / 1000 # $B

if cum_change > 1.0:
interp = "Net gain"
elif cum_change > 0:
interp = "Fragile positive"
elif cum_change > -1.0:
interp = "Near break-even"
else:
interp = "Net loss"

print(f"{scenario_name:<22} {shock_label+' $'+str(shock):>10} "
f"{cum_change:>+13.1f}B {npv_change:>+13.1f}B {interp}")

results[scenario_name] = decline

# Break-even reference
print(f"\n{'Break-even (~7.5%)':<22} {'Mid $'+str(TAX_SHOCK_MID):>10} ", end="")
be_totals = {}
for yr in YEARS:
f = production_decline_factor(yr, BREAKEVEN_DECLINE)
b, t = annual_ugf_revenue(yr, f, TAX_SHOCK_MID)
be_totals[yr] = b + t
be_cum = (sum(be_totals.values()) - baseline_cumulative) / 1000
be_npv = (npv([(yr, be_totals[yr]) for yr in YEARS]) - baseline_npv) / 1000
print(f"{be_cum:>+13.1f}B {be_npv:>+13.1f}B Threshold")

print(f"\n{'='*72}\n")

# Write CSV: scenario summary
with open("sb280_scenario_results.csv", "w", newline="") as f:
writer = csv.writer(f)
writer.writerow([
"Scenario", "Cumulative Decline %", "Tax Shock ($/bbl)",
"Scenario Cumulative UGF ($M)", "Baseline Cumulative UGF ($M)",
"Cumulative Change ($M)", "Cumulative Change ($B)",
"Scenario NPV ($M)", "Baseline NPV ($M)",
"NPV Change ($M)", "NPV Change ($B)", "Result"
```

```
)  
for scenario_name, decline in list(SCENARIOS.items()) + [("Break-even (7.5%)", BREAKEVEN_DECLINE):  
for shock_label, shock in [("Low", TAX_SHOCK_LOW),  
("Mid", TAX_SHOCK_MID),  
("High", TAX_SHOCK_HIGH)]:  
annual_totals = {}  
for yr in YEARS:  
factor = production_decline_factor(yr, decline)  
base, new_tax = annual_ugf_revenue(yr, factor, shock)  
annual_totals[yr] = base + new_tax  
cum_total = sum(annual_totals.values())  
scen_npv = npv([(yr, annual_totals[yr]) for yr in YEARS])  
cum_change = cum_total - baseline_cumulative  
npv_change = scen_npv - baseline_npv  
result = "Gain" if cum_change > 0 else "Loss"  
writer.writerow([  
scenario_name, f"{decline*100:.1f}%", shock,  
round(cum_total, 1), round(baseline_cumulative, 1),  
round(cum_change, 1), round(cum_change/1000, 2),  
round(scen_npv, 1), round(baseline_npv, 1),  
round(npv_change, 1), round(npv_change/1000, 2), result  
])  
print("Saved: sb280_scenario_results.csv")
```

```
# Write CSV: annual detail  
with open("sb280_annual_detail.csv", "w", newline="") as f:  
writer = csv.DictWriter(f, fieldnames=annual_detail_rows[0].keys())  
writer.writeheader()  
writer.writerows(annual_detail_rows)  
print("Saved: sb280_annual_detail.csv")
```

```
# Simple ASCII chart of cumulative revenue over time (mid-shock)  
print("\nCUMULATIVE UGF REVENUE — MID TAX SHOCK ($1.20/bbl)")  
print(f"{'Year':<6} {'Baseline':>12} {'Static':>12} {'Moderate':>12} {'Strong':>12}")  
print(f"{'-'*6} {'-'*12} {'-'*12} {'-'*12} {'-'*12}")  
cum_base = cum_static = cum_mod = cum_strong = 0  
for yr in YEARS[:3]: # every 3 years for readability  
f0 = production_decline_factor(yr, 0.00)  
f1 = production_decline_factor(yr, 0.00)  
f2 = production_decline_factor(yr, 0.10)  
f3 = production_decline_factor(yr, 0.15)  
b0, t0 = annual_ugf_revenue(yr, f0, 0.00)  
b1, t1 = annual_ugf_revenue(yr, f1, TAX_SHOCK_MID)  
b2, t2 = annual_ugf_revenue(yr, f2, TAX_SHOCK_MID)  
b3, t3 = annual_ugf_revenue(yr, f3, TAX_SHOCK_MID)  
cum_base += b0  
cum_static += b1 + t1  
cum_mod += b2 + t2  
cum_strong += b3 + t3  
print(f"{'yr':<6} ${cum_base/1000:>10.1f}B ${cum_static/1000:>10.1f}B "  
f"${cum_mod/1000:>10.1f}B ${cum_strong/1000:>10.1f}B")
```