

Sample Workflow Calculation Review

Payment Settlement Reconciliation

Synthetic example — no client or production data

One financially important rule. Explicit assumptions. A defined failure boundary. Expected outputs your workflow owner can check before trusting the rule in production.

This representative sample shows the structure of SumWise's **one-time \$99 Workflow Calculation Review** for one supported calculation rule. It is separate from paid Compute API access unless access is explicitly arranged.

01 The business rule

A payment processor deposits a net settlement after refunds, processing fees and chargebacks. Flag the settlement for review when the actual bank deposit differs from the expected net by **more than \$0.05**.

1. Calculate expected net settlement.
2. Subtract expected net from the actual bank deposit to get signed variance.
3. Take the absolute value.
4. **MATCH** when absolute variance is at most 5 cents.
5. **NEEDS REVIEW** when absolute variance is more than 5 cents.

The inclusive boundary matters.

5 cents: MATCH · 6 cents: NEEDS REVIEW

```
absolute_variance_cents <= 5
```

02 Integer cents, one currency

Convert all money to **integer cents before calculation**. This avoids unnecessary decimal/binary floating-point behavior in the reconciliation rule. All values in one calculation must represent the same currency.

Input	Meaning	Required contract
gross_sales_cents	Gross captured sales	Integer; 0 to 100,000,000,000
refunds_cents	Refunded amount	Integer; 0 to gross sales
processing_fees_cents	Processor fees	Integer >= 0
chargebacks_cents	Disputes / chargebacks	Integer >= 0
actual_deposit_cents	Bank deposit received	Integer >= 0
tolerance_cents	Allowed absolute variance	Integer >= 0; sample value 5

Calculation specification

```

expected_net_cents = gross_sales_cents - refunds_cents
                    - processing_fees_cents - chargebacks_cents
variance_cents = actual_deposit_cents - expected_net_cents
absolute_variance_cents = abs(variance_cents)

```

```

MATCH          when absolute_variance_cents <= tolerance_cents
NEEDS REVIEW  when absolute_variance_cents > tolerance_cents

```

SumWise Compute integration pattern

The current public API is Evaluate-only: POST /v1/evaluate. Build a fixed expression from validated integers. Braces below are **documentation placeholders, not SumWise expression syntax**.

```

abs({actual_deposit_cents} - ({gross_sales_cents}
- {refunds_cents} - {processing_fees_cents} - {chargebacks_cents}))

```

Concrete first synthetic request:

```

{"expression":"abs(1034250 - (1075000 - 25000 - 12750 - 3000))"}

```

Expected numerical result: **0**. With validated integer inputs:

```

result.type      = integer
result.exactness = exact
result.value     = "0"

```

Compute returns the exact absolute numerical variance. The customer's workflow IF/branch step decides `absolute_variance_cents <= tolerance_cents`; the API does not perform that business branch. Validate and interpret the returned string as an integer before comparison; do not compare numeric strings lexicographically or lose integer precision.

See the [current API documentation](#) for authentication, response shape and limits.

03 Check both sides of the boundary

All ten cases use **tolerance_cents = 5**. Amounts below are dollar-formatted for reading; calculations use integer cents. Signed variance is actual minus expected. The decision uses **absolute variance**.

01 · Exact match		MATCH
Gross sales	\$10,750.00	
Refunds	\$250.00	
Processing fees	\$127.50	
Chargebacks	\$30.00	
Actual deposit	\$10,342.50	
Expected net	\$10,342.50	
Signed variance	\$0.00	
Absolute variance	\$0.00	

02 · +1 cent		MATCH
Gross sales	\$10,750.00	
Refunds	\$250.00	
Processing fees	\$127.50	
Chargebacks	\$30.00	
Actual deposit	\$10,342.51	
Expected net	\$10,342.50	
Signed variance	\$0.01	
Absolute variance	\$0.01	

03 · -1 cent		MATCH
Gross sales	\$10,750.00	
Refunds	\$250.00	
Processing fees	\$127.50	
Chargebacks	\$30.00	
Actual deposit	\$10,342.49	
Expected net	\$10,342.50	
Signed variance	-\$0.01	
Absolute variance	\$0.01	

04 · +5-cent boundary		MATCH
Gross sales	\$10,750.00	
Refunds	\$250.00	
Processing fees	\$127.50	
Chargebacks	\$30.00	
Actual deposit	\$10,342.55	
Expected net	\$10,342.50	
Signed variance	\$0.05	
Absolute variance	\$0.05	

05 · -5-cent boundary		MATCH
Gross sales	\$10,750.00	
Refunds	\$250.00	
Processing fees	\$127.50	
Chargebacks	\$30.00	
Actual deposit	\$10,342.45	
Expected net	\$10,342.50	
Signed variance	-\$0.05	
Absolute variance	\$0.05	

06 · +6 cents		NEEDS REVIEW
Gross sales	\$10,750.00	
Refunds	\$250.00	
Processing fees	\$127.50	
Chargebacks	\$30.00	
Actual deposit	\$10,342.56	
Expected net	\$10,342.50	
Signed variance	\$0.06	
Absolute variance	\$0.06	

Zero activity, mixed deductions & larger amounts

The tolerance remains 5 cents, including the zero-activity case. All money inputs in that case are zero. Full inputs are shown for every test.

07 · Zero activity	MATCH	08 · Mixed deductions	MATCH
Gross sales	\$0.00	Gross sales	\$2,500.00
Refunds	\$0.00	Refunds	\$100.01
Processing fees	\$0.00	Processing fees	\$74.99
Chargebacks	\$0.00	Chargebacks	\$150.00
Actual deposit	\$0.00	Actual deposit	\$2,175.03
Expected net	\$0.00	Expected net	\$2,175.00
Signed variance	\$0.00	Signed variance	\$0.03
Absolute variance	\$0.00	Absolute variance	\$0.03

09 · Large settlement	MATCH	10 · Material mismatch	NEEDS REVIEW
Gross sales	\$123,456.78	Gross sales	\$5,000.00
Refunds	\$1,234.56	Refunds	\$0.00
Processing fees	\$2,345.67	Processing fees	\$150.00
Chargebacks	\$456.78	Chargebacks	\$0.00
Actual deposit	\$119,419.72	Actual deposit	\$4,849.00
Expected net	\$119,419.77	Expected net	\$4,850.00
Signed variance	-\$0.05	Signed variance	-\$1.00
Absolute variance	\$0.05	Absolute variance	\$1.00

8 MATCH · 2 NEEDS REVIEW

These are synthetic expected outcomes, not customer results or a claim of live API execution.

04 Invalid is never a match

Missing required amount

Do not evaluate. Route to input error / review; missing values must never silently become zero.

Negative refund

Reject under this business contract.

Non-integer cents

Reject or normalize explicitly upstream. Never silently round fractional cents.

Deductions exceed gross

Reject under this sample unless the business explicitly allows negative settlements.

Mixed currencies

Reject. Currency conversion is outside this rule.

Edge conditions to preserve

- Preserve signed variance until the final `abs()` operation.
- `<= 5` is intentional. Replacing it with `< 5` changes the rule.
- API or transport failure must never become a MATCH.
- With validated integer-cent inputs, an unexpected approximate result is an integration/contract problem, not a successful reconciliation.

n8n implementation note

1. Input / source system
2. Validate required fields and convert money to integer cents
3. Build the fixed SumWise expression from validated integers
4. SumWise Compute / POST /v1/evaluate
5. Validate HTTP/result shape and exact integer result
6. IF `absolute_variance_cents <= tolerance_cents`

True → **MATCH**

False → **NEEDS REVIEW**

A failure at validation or the service step takes the error/review path, never the MATCH branch. Store the issued API key in your integration's credential facility. **Never put a key in expression text, workflow logs, screenshots or browser/client-side code.**

05 What the review is designed to catch

- Reversed variance sign
- Forgotten deduction
- < versus <= boundary mistakes
- Dollars/cents mixups
- Unnecessary decimal floating-point behavior
- Missing values becoming zero
- Malformed inputs
- Calculation-service failure treated as successful reconciliation
- The same rule implemented differently across workflows

For this absolute-variance decision, reversing the sign alone does not change MATCH versus NEEDS REVIEW. Recording signed variance still exposes the reversal and keeps reporting and later rules consistent.

For one supported rule, the \$99 review can provide

- Written calculation specification, assumptions and input contract
- Edge and failure conditions
- 8–12+ synthetic test cases with expected outputs
- A short n8n / Make / HTTP integration note where relevant
- An explicit in-scope / out-of-scope statement
- A fit check before purchase

Useful when

A numerical rule matters enough that its behavior should be explicit, tested and reusable across workflows or systems.

Possibly not worthwhile when

The rule is trivial, exists in one low-risk location, and your current workflow already handles it clearly and reliably.

Scope of this sample

This is a **synthetic sample deliverable, not a paid-client case study**, testimonial or evidence of an engagement. A real review is scoped to your supported rule after a no-data fit check. It is not a security/compliance certification or a guarantee of ROI or production suitability.

The **\$99 Workflow Calculation Review is a one-time service**. Hosted Compute private-alpha/API access is separate unless included under separate agreed terms. Currency conversion, production system access and broader custom engineering are outside this sample's scope.

Have one calculation you want to harden?

Send the rule or formula first. Avery will confirm whether it fits SumWise before you purchase the \$99 review. **The fit decision happens before payment.**

Use synthetic inputs. Do not send passwords, API keys, bank credentials or private production records for the initial fit check.

avery@sumwisecalc.com · sumwisecalc.com/compute/