How Much Money Do You Need in a Digital Currency Exchange Pool?
The Jackson Liquidity Framework Explained Simply

Lewis Jackson
Lewis Jackson Ventures (LJV)

December 2025

Abstract

Imagine a digital vending machine that exchanges one currency for another instantly. How much money does it need to hold to work properly? Too little, and it runs out. Too much, and money sits idle. This guide explains a new framework—the Jackson Liquidity Framework—that helps banks and payment systems figure out exactly how much money to put in these digital exchange machines (called AMMs) so they work reliably without wasting resources.

For permission requests, licensing inquiries, or commercial use, please contact:
Lewis Jackson Ventures Ltd. Email: lewis@jackson.ventures

Disclaimer:
This guide is provided for informational purposes only and does not constitute financial, legal, regulatory, or investment advice. Banks and payment systems should consult with appropriate experts before implementing any new technology or framework.

Contents

1 What This Guide Is About

1.1 The Big Picture

Money is going digital, and payments are getting faster. Instead of waiting days for international transfers, new systems promise instant settlement—your money arrives in seconds, not days.

But there's a catch: instant settlement needs instant liquidity. That means having money ready and waiting in the right place at the right time. And figuring out how much money to have ready is surprisingly complicated.

This guide explains a new framework that solves this problem.

1.2 Who This Is For

This guide is written for:

- Anyone curious about how digital currency exchange works
- Business leaders evaluating payment technology
- Journalists covering financial technology
- Students learning about modern finance
- Regulators who need to understand the basics before diving into technical details

You don't need any background in finance or mathematics to understand this guide.

2 The Problem: Running Out of Money in Digital Exchanges

2.1 How Digital Currency Exchanges Work

Think of an Automated Market Maker (AMM) like a robot currency exchange booth. It holds two currencies—let's say US dollars and euros—and swaps them automatically for anyone who asks.

A real example: XRP Ledger (XRPL)

The XRP Ledger has built-in AMM pools (using the XLS-30 standard) that work exactly like this. For instance, a bank might set up an XRP/USD pool:

- The pool holds 1 million XRP and \$2 million USD
- When someone wants to send money from the US to Japan, they might swap USD → XRP → JPY
- The swap happens in 3-5 seconds (XRPL's settlement time)
- The pool automatically adjusts prices based on supply and demand

This is just one example—similar systems exist on Ethereum, Stellar, and other platforms. But XRPL is particularly relevant because it was designed specifically for institutional payments, not just cryptocurrency trading.

The basic math:

The AMM maintains a constant relationship between the two currencies:

$$R_A \cdot R_B = k$$

Where:

- R_A = amount of currency A (e.g., dollars or XRP)
- R_B = amount of currency B (e.g., euros or USD)
- k = a constant number that never changes

For example:

- The booth holds \$1 million and €1 million
- So $k = 1{,}000{,}000 \times 1{,}000{,}000 = 1{,}000{,}000{,}000{,}000$
- You want to swap \$1,000 for euros
- The booth gives you euros and takes your dollars
- Now it has \$1,001,000 and €999,000.999...
- Notice: $1{,}001{,}000 \times 999{,}000.999 \approx 1{,}000{,}000{,}000{,}000$ (still k!)

How trades work:

When you trade amount q of currency A for currency B:

$$R'_A = R_A + q$$

$$R'_B = \frac{k}{R_A + q}$$

The amount of B you receive is:

$$\Delta B = R_B - R'_B = R_B - \frac{k}{R_A + q}$$

What is slippage?

Slippage is how much worse your price is compared to the "fair" price:

$$S(q) = \frac{q}{R_A + q}$$

This happens thousands of times per day.

2.2 Why These Exchanges Can Run Into Trouble

Three main problems can happen:

Problem 1: One-Way Traffic

Imagine more people want to swap dollars for euros than the other way around. The booth keeps losing euros and gaining dollars. Eventually, it might run low on euros and can't make trades anymore—or it has to charge really high prices for the few euros it has left.

Problem 2: Rush Hour

Payments don't arrive evenly throughout the day. Sometimes there are quiet periods, and sometimes there are busy periods (like when banks settle accounts at specific times each day). During rush hour, the booth might run out of money even if things balance out by the end of the day.

Problem 3: Splitting Money Across Multiple Booths

If you split $1 million across 10 smaller exchange booths instead of one big booth, each booth only has $100,000. Smaller booths run out of money much faster. It turns out that running 10 small booths requires way more total money than running one big booth—sometimes 3 times as much!

2.3 Why This Matters

If these digital exchanges don't have enough money:

- Payments get stuck or rejected
- Exchange rates become unpredictable

- The system becomes unreliable
- People lose trust and stop using it

But if exchanges hold too much money:

- Money sits idle instead of being put to work elsewhere
- The system becomes unnecessarily expensive to run
- Banks and payment companies waste resources

The Jackson Liquidity Framework helps find the "just right" amount.

3 The Solution: The Jackson Liquidity Framework

3.1 What Is the Jackson Liquidity Framework?

The Jackson Liquidity Framework (JLF) is like a sophisticated calculator that tells you exactly how much money your digital exchange booth needs. It considers all the problems we mentioned above and gives you a precise answer.

It has four main tools:

3.2 Tool #1: The Jackson Liquidity Requirement (JLR)

What it does: Tells you the minimum amount of money you need.

Think of it like: A minimum inventory requirement. Just like a grocery store needs to know how much milk to stock, a digital exchange needs to know how much currency to hold.

What it considers:

- How busy is your exchange? (More trades = more money needed)
- How much does traffic go one direction vs. the other?
- What happens during rush hour?
- How much price movement can you tolerate?

The formula:

The JLR combines four components:

$$\text{JLR} = \max\left\{R^{\text{slip}},\, R_A^{\text{VaR}} + R_B^{\text{VaR}},\, R_A^{\text{intraday}} + R_B^{\text{intraday}},\, R^{\text{Basel}}\right\}$$

Where:

- R^{slip} = reserves needed to keep slippage below tolerance ε
- $R_A^{\text{VaR}}, R_B^{\text{VaR}}$ = reserves to cover directional flow risk
- $R_A^{\text{intraday}}, R_B^{\text{intraday}}$ = reserves for rush hour peaks
- R^{Basel} = reserves to comply with banking regulations (LCR, NSFR)

The answer: A specific dollar amount—like "You need $20.4 million to run this exchange safely."

3.3 Tool #2: The Jackson Stability Invariant (JSI)

What it does: Draws a line between "safe" and "unsafe" amounts of money.

Think of it like: A fuel gauge warning light. It tells you, "If you go below this line, you're in the danger zone."

The formula:

The JSI defines a stability boundary:

$$\frac{R_A \cdot R_B}{\sigma_I \cdot \sigma \cdot \sqrt{T}} \geq K(\varepsilon, \rho)$$

Where:

- R_A, R_B = your reserves in currencies A and B
- σ_I = how volatile the imbalance is
- σ = how volatile exchange rates are
- T = time period
- $K(\varepsilon, \rho)$ = a threshold based on your slippage tolerance and correlation

What it shows:

- Above the line: Your exchange can handle normal market conditions
- Below the line: Small problems can cause big disruptions

Why it matters: Even if you think you have enough money, JSI tells you if you're cutting it too close.

3.4 Tool #3: The Jackson Liquidity Surface (JLS)

What it does: Shows how your money needs change under different conditions.

Think of it like: A weather map showing how much heating fuel you need at different temperatures. When it's colder (busier traffic), you need more fuel (money).

The formula:

The JLS is a function mapping corridor conditions to liquidity needs:

$$\text{JLR} = f(\lambda, \sigma_X, p, \varepsilon, \sigma)$$

Where:

- λ = arrival rate (how many payments per day)
- σ_X = payment size volatility (how much sizes vary)
- p = directional bias (how one-sided the traffic is)
- ε = slippage tolerance (price impact you'll accept)
- σ = exchange rate volatility

What it reveals:

- If traffic doubles, money needs more than double
- Small changes in one-way traffic can require big changes in money
- You can see how different scenarios affect your needs

Why it's useful: You can plan ahead. "If we expect traffic to increase by 50%, we'll need this much more money."

3.5 Tool #4: The Jackson Corridor Coefficient (J-score)

What it does: Gives you a real-time stress meter for your exchange.

Think of it like: A speedometer, but for exchange stress instead of speed.

The formula:

The J-score compresses corridor conditions into a single number:

$$J = \frac{\lambda \cdot \mathbb{E}[X] \cdot \sigma_I}{R_A + R_B}$$

Where:

- λ = arrival rate (payments per day)
- $\mathbb{E}[X]$ = average payment size
- σ_I = imbalance volatility
- $R_A + R_B$ = total reserves available

How to read it:

- **Green zone (J < 1.1M):** Everything is fine
- **Yellow zone (1.1M – 1.3M):** Keep an eye on things
- **Red zone (J > 1.3M):** You might need to add more money soon

Why it matters: You get early warnings before problems happen, not after.

4 Real Examples: What We Learned from Testing

We ran computer simulations to test these ideas. Think of it like a video game where we ran thousands of "what if" scenarios. Here's what we found.

4.1 Test #1: Normal Day

Setup: We simulated a normal business day with typical traffic patterns.

What happened:

- Most trades went smoothly
- Traffic was fairly balanced (55% one direction, 45% the other)
- The exchange needed about $20.4 million to operate safely
- The stress meter (J-score) stayed in the green zone most of the time

Key lesson: Even on a normal day, you need substantial reserves. You can't just "wing it" with whatever cash you have lying around.

4.2 Test #2: Rush Hour Chaos

Setup: We simulated what happens when payments bunch up—like everyone rushing to make payments at the same time.

What happened:

- Short bursts of intense activity
- During busy periods, the exchange needed 20–40% more money
- Traffic going one direction increased to 60% (instead of 55%)
- The stress meter spiked into yellow and red zones during rushes

Key lesson: Looking at averages isn't enough. You need to plan for the busiest moments, not just typical moments.

Real-world parallel: This is like how roads need to be sized for rush hour, not average traffic. A road that works fine most of the day might be a parking lot at 5 PM.

4.3 Test #3: The Problem with Multiple Small Exchanges

Setup: We compared having one big exchange versus splitting money across multiple smaller exchanges.

What happened:

- One big exchange: $20.4 million needed (our baseline)
- Three smaller exchanges: $61.1 million needed (3× as much!)
- Ten tiny exchanges: $203.8 million needed (10× as much!)

Key lesson: Fragmenting your money is incredibly expensive. It's much more efficient to pool money together than to split it up.

Real-world parallel: This is like water reservoirs. One big reservoir is more efficient than ten small ones because each small one needs its own safety buffer.

4.4 Visual Summary

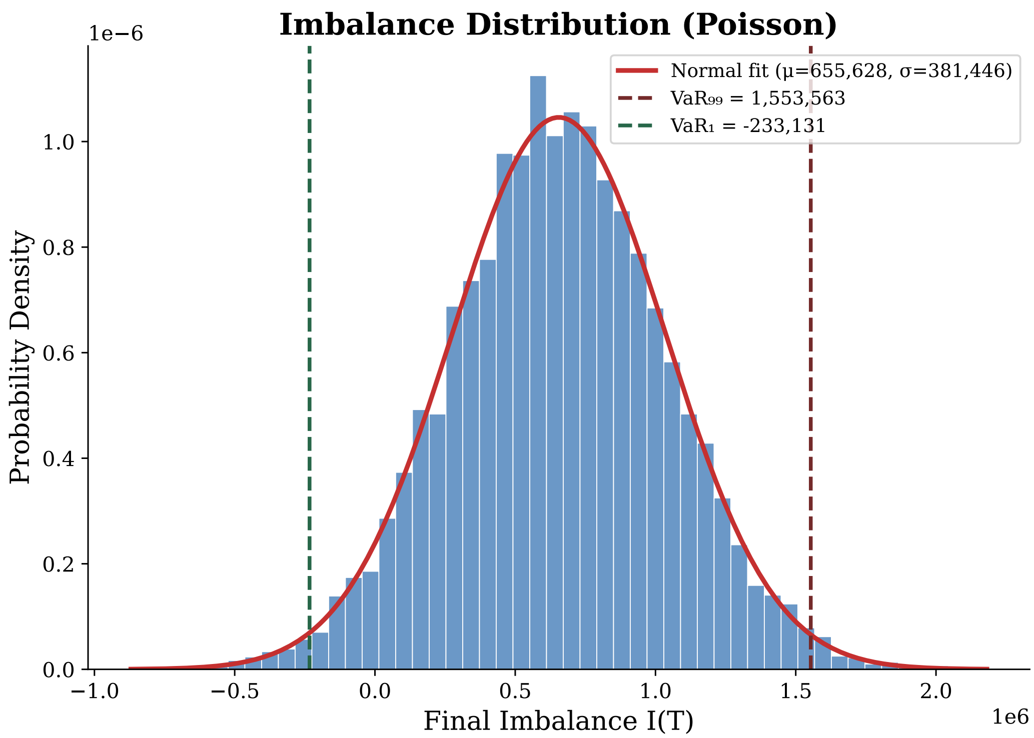

Figure 1: How money imbalance looks over many simulated days. Most days are near the middle, but extreme days happen too—and you need enough money to handle those.

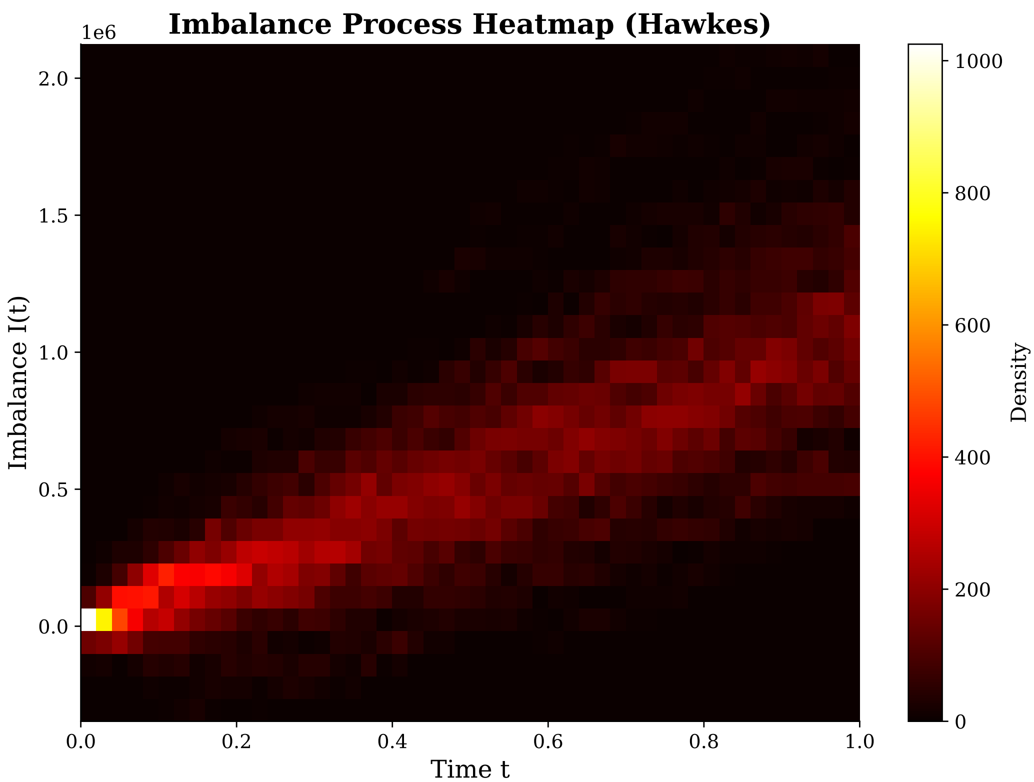

Figure 2: What rush hour looks like. The bright yellow spots show moments when the exchange is under heavy stress. This is when you most need adequate reserves.

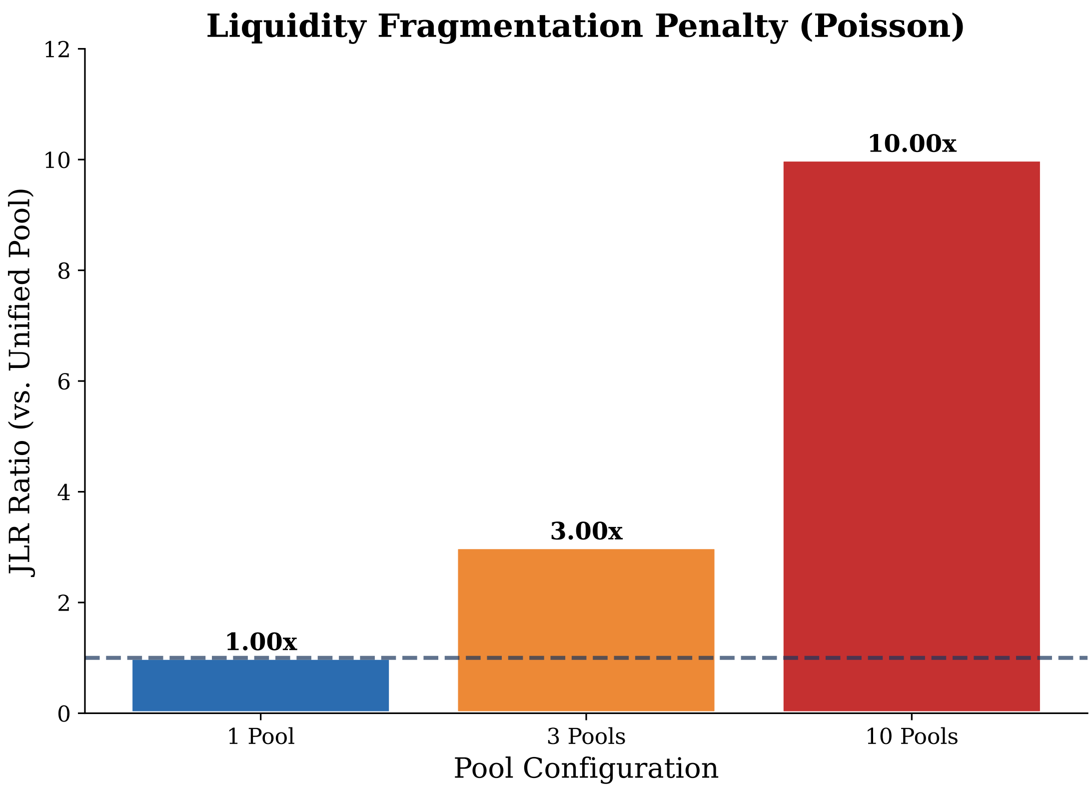

Figure 3: The cost of splitting up your money. Notice how the bars get dramatically higher as you split into more exchanges—this is why consolidation is better.

5 Practical Questions Answered

5.1 Who Would Use This Framework?

Central banks: When setting up digital currency systems that exchange between different countries' digital currencies.

Large banks: When operating payment systems that move money between accounts instantly.

Payment companies: Like PayPal or Stripe, when offering instant currency conversion.

Blockchain payment networks: Platforms using built-in AMM protocols for settlement:

- **XRP Ledger (XRPL):** Financial institutions using XRPL's XLS-30 AMM standard for cross-border payments can apply this framework to figure out exactly how much XRP, USD, EUR, or other currencies to hold in their liquidity pools. Since XRPL settles transactions in 3-5 seconds, it's the perfect example of instant settlement where this framework really matters.

- **Ethereum-based platforms:** Companies offering stablecoin swaps or tokenized asset exchanges can use this framework to manage liquidity professionally.

- **Stellar payment corridors:** Organizations using Stellar for remittances can apply these methods to size their liquidity pools correctly.

Cryptocurrency platforms: Exchanges running AMMs that swap between different cryptocurrencies or between crypto and regular money.

The framework works especially well for XRPL-based corridors because XRPL was designed specifically for payments (not just trading), and its 3-5 second finality creates exactly the instant-settlement environment where getting liquidity right is critical.

5.2 Does This Replace Existing Rules?

No, it complements them.

Think of it this way: Banks already have rules about keeping enough cash on hand. Those rules were written for old-fashioned banking where you might wait days for transactions to complete.

The Jackson Liquidity Framework adds new rules specifically for instant, digital exchanges. It's an addition, not a replacement.

5.3 How Much Does It Cost to Implement?

Implementing the framework itself is mostly about calculation and monitoring—it's a method, not a physical thing you buy.

The cost is really about having enough money in your exchange:

- If you're already holding enough money, there's no additional cost

- If you're not holding enough, you'll need to add more (but you should have been doing that anyway for safety)

- The monitoring dashboard is software that tracks your numbers in real-time

Think of it like installing speedometers in cars. The speedometer itself isn't expensive—the "cost" is that you might discover you were going too fast and need to slow down (or in this case, add more money to be safe).

5.4 What If I Ignore This and Just Guess?

Bad things can happen:

If you hold too little money:

- Your exchange might fail during busy periods
- Customers get frustrated and leave
- Regulators might shut you down for being unsafe
- You might have to close temporarily to add more funds

If you hold too much money:

- You're wasting resources that could be earning money elsewhere
- Your system costs more to operate than necessary
- You're less competitive than competitors who sized things correctly

The framework helps you hit the sweet spot: enough to be safe, but not so much that you're wasteful.

6 The Future: What Comes Next

6.1 Where This Is Heading

Digital currencies and instant payments are the future. This isn't some distant possibility—it's happening now:

- Many countries are testing or launching digital versions of their currencies (called CBDCs—Central Bank Digital Currencies)
- Banks are upgrading to instant payment systems
- International payments that used to take days are becoming instant

As this transition happens, figuring out liquidity becomes critical. The Jackson Liquidity Framework provides the tools to do it right.

6.2 Tools Being Built

Lewis Jackson Ventures is building a monitoring dashboard that puts these ideas into practice:

- Real-time displays showing your stress meter (J-score)
- Automatic alerts when you're approaching danger zones
- Scenario planning: "What if traffic increases by 50%?"
- Reports you can show to regulators or management

Think of it like a fitness tracker, but for financial exchanges instead of your body.

6.3 Want to Learn More?

For technical readers: There's a detailed technical whitepaper with all the mathematics and formulas.

For institutions: Contact Lewis Jackson Ventures to discuss implementing this framework for your payment system.

For researchers: Collaboration opportunities are available for academics studying liquidity in digital payment systems.

Contact: Email: lewis@jackson.ventures

Follow on social media:

- YouTube – `https://www.youtube.com/@LewisWJackson`
- LinkedIn – `https://www.linkedin.com/in/lewisjacksonli/`
- X (Twitter) – `https://x.com/WhatSayLew`
- Instagram – `https://www.instagram.com/lewis.w.jackson`

7 Summary: The Key Takeaways

If you remember nothing else from this guide, remember these five points:

1. **Instant payments need instant liquidity.** You can't fake it—you need real money available right now.

2. **Guessing is dangerous and expensive.** Too little money causes failures. Too much wastes resources. You need a proper method to calculate the right amount.
3. **The Jackson Liquidity Framework provides that method.** It's like a sophisticated calculator specifically designed for digital currency exchanges.
4. **Rush hour matters more than averages.** You need to plan for the busiest moments, not typical moments.
5. **Bigger is better than fragmented.** One large exchange is much more efficient than many small ones—sometimes 3–10× more efficient.

As the world moves toward instant digital payments, frameworks like this one will become essential infrastructure—as fundamental as accounting rules or safety standards.

8 Glossary: Terms You Might Encounter

AMM (Automated Market Maker): A robot currency exchange booth. It automatically swaps one currency for another based on fixed rules.

Liquidity: Money that's available to use right now. "High liquidity" means lots of available money. "Low liquidity" means money is tight.

Fragmentation: Splitting money across multiple exchanges instead of pooling it in one place.

Imbalance: When more money flows one direction than the other. Like a seesaw that's heavier on one side.

VaR (Value at Risk): A way of measuring "how bad could things get?" For example, "99% of the time, we won't lose more than $1 million" means your VaR is $1 million.

Slippage: The difference between the price you expect and the price you actually get. Like buying concert tickets that say $50 but end up costing $60 after fees.

J-score: The stress meter number. Higher means more stressed, lower means more comfortable.

Clustering: When events bunch up together instead of being evenly spread out. Like everyone trying to leave a stadium at the same time instead of trickling out gradually.

Corridor: A pathway for payments between two currencies. For example, the "USD-EUR corridor" is the system that handles dollar-to-euro exchanges.

9 Mathematical Appendix: The Complete Formulas

This section provides all the mathematical formulas for those who want to see the complete technical details. Don't worry if the math looks intimidating—the main guide already explained everything you need to know!

9.1 Payment Flow Model

Payments arrive following either a Poisson process or a Hawkes process.

Poisson arrivals:

$$N(t) \sim \text{Poisson}(\lambda t)$$

Hawkes arrivals (with clustering):

$$\lambda(t) = \mu + \alpha \sum_{t_i < t} \beta e^{-\beta(t - t_i)}$$

Where μ is the baseline rate, α is the self-excitation, and β is the decay rate.

9.2 Imbalance Process

Net imbalance over time:

$$I(t) = \sum_{i=1}^{N(t)} s_i X_i$$

Where:

- $s_i \in \{+1, -1\}$ indicates direction (buy or sell)
- X_i is the payment size
- $N(t)$ is the number of arrivals by time t

Statistical properties:

Mean imbalance:

$$\mathbb{E}[I(T)] = \lambda T(2p - 1)\mu_X$$

Variance:

$$\text{Var}[I(T)] = \lambda T \left[\sigma_X^2 + 4p(1 - p)\mu_X^2 + (2p - 1)^2 \mu_X^2\right]$$

Where p is the directional probability (e.g., $p = 0.55$ means 55% go one direction).

9.3 Value at Risk (VaR)

The 99% VaR for imbalance:

$$\text{VaR}_{0.99}(I(T)) = \mathbb{E}[I(T)] + 2.326 \cdot \sqrt{\text{Var}[I(T)]}$$

This tells you: ”99% of the time, imbalance won’t exceed this amount.”

9.4 Slippage Constraint

To keep slippage below tolerance ε:

$$F_X\left(\frac{\varepsilon}{1-\varepsilon} R_A\right) \geq 1 - \alpha$$

Where F_X is the cumulative distribution of payment sizes and α is your risk tolerance (e.g., $\alpha = 0.01$ for 99% confidence).

Solving for required reserves:

$$R^{\text{slip}} = \frac{1-\varepsilon}{\varepsilon} \cdot F_X^{-1}(1-\alpha)$$

9.5 Reserve Evolution

Reserves change as trades execute:

$$R_A(t) \approx R_A(0) - \max_{0 \leq s \leq t} (-I(s))^+$$

$$R_B(t) \approx R_B(0) - \max_{0 \leq s \leq t} I(s)^+$$

Where $(x)^+ = \max(0, x)$ means ”positive part only.”

9.6 Jackson Stability Invariant (Complete Form)

The full stability condition:

$$\frac{R_A \cdot R_B}{\sigma_I \cdot \sigma \cdot \sqrt{T}} \geq K(\varepsilon, \rho)$$

Where the threshold K is determined by:

$$K(\varepsilon, \rho) = \Phi^{-1}(1-\varepsilon) \cdot \sqrt{1+\rho^2}$$

And Φ^{-1} is the inverse normal distribution.

9.7 Fragmentation Penalty

For n equal-sized pools vs. 1 unified pool:

$$\text{JLR}_{\text{fragmented}} = n \cdot \text{JLR}_{\text{unified}}$$

This linear scaling explains why:

- 3 pools require 3× the liquidity
- 10 pools require 10× the liquidity

9.8 Complete JLR Formula

The full Jackson Liquidity Requirement:

$$\text{JLR} = \max \left\{ \begin{array}{l} R^{\text{slip}} = \dfrac{1-\varepsilon}{\varepsilon} \cdot F_X^{-1}(1-\alpha) \\ R_A^{\text{VaR}} = -\text{VaR}_{0.01}(-I(T)) \\ R_B^{\text{VaR}} = \text{VaR}_{0.99}(I(T)) \\ R_A^{\text{intraday}} = \max_{0\le t\le T}(-I(t))^+ \\ R_B^{\text{intraday}} = \max_{0\le t\le T} I(t)^+ \\ R^{\text{Basel}} = \max(\text{LCR requirements}, \text{NSFR requirements}) \end{array} \right\}$$

9.9 Parameter Values from Case Studies

Case Study 1 (Poisson):

- $\lambda = 800$ payments/day
- $\mu_X = \ln(5000)$, $\sigma_X = 1.0$
- $p = 0.55$ (directional bias)
- $\sigma = 0.07$ (FX volatility)
- $\varepsilon = 0.005$ (0.5% slippage tolerance)

Case Study 2 (Hawkes):

- $\mu = 600$, $\alpha = 0.30$, $\beta = 2.5$
- $p = 0.60$ (higher bias)
- $\sigma = 0.10$ (higher volatility)

Resulting JLR: \$20,378,543

About the Author

Lewis Jackson is a financial researcher and entrepreneur specializing in liquidity management for next-generation payment systems. Through Lewis Jackson Ventures Ltd., he develops frameworks and tools for central banks, financial institutions, and payment companies navigating the transition to instant settlement and digital currencies.

The Jackson Liquidity Framework represents years of research into how traditional liquidity management principles apply—or don't apply—to automated, algorithmic exchange mechanisms. This work bridges academic research, regulatory requirements, and practical implementation.

Lewis Jackson Ventures Ltd. Developing tools for the future of instant settlement.