信州大学工学部

学士論文

ポアソン到着に従う出現に対する回収レートの数学的導出

指導教員　西新 幹彦 准教授

学科　電子情報システム工学科
学籍番号　21T2057J
氏名　小池 由樹

2025 年 3 月 27 日

目次

1 はじめに 1

2 問題設定と回収レートの導出 1
2.1 一定の往復時間, バッファ無しの場合 4
2.2 指数分布に従う往復時間, バッファ無しの場合 6
2.3 一定の往復時間, バッファ有りの場合 7
2.4 指数分布に従う往復時間, バッファ有りの場合 9

3 シミュレーションによる検証 11
3.1 一定の往復時間のシミュレーション結果 12
3.2 指数分布に従う往復時間のシミュレーション結果 13

4 収束速度に関する考察 15
4.1 一定の往復時間の場合 15
4.2 指数分布に従う往復時間の場合 15

5 まとめ 16

謝辞 16

参考文献 16

付録 A ソースコード 18
A.1 一定の往復時間, バッファ無しの場合 18
A.2 指数分布に従う往復時間, バッファ無しの場合 18
A.3 一定の往復時間, バッファ有りの場合 19
A.4 指数分布に従う往復時間, バッファ有りの場合 20

1　はじめに

駅のタクシー乗り場で待機しているタクシーは, 顧客を乗せて指定された目的地に移動する. 目的地に到着したら顧客をタクシーから降ろす. そして再び駅のタクシー乗り場に戻り, 新たな顧客を乗せて目的地に移動する. この繰り返しである. このようなタクシーの事例というのは, 顧客, タクシー, タクシー乗り場の 3 つの要素から構成される一つのシステムだと捉えることができる. ここで単位時間あたりに乗車させた顧客の人数のことを回収レートと呼ぶ. 回収レートの値は, そのシステムのパフォーマンスを評価するための一つの指標となる. そのため回収レートの値を導出することは, 今後のシステムを改善するために非常に有用である. さらに市内バスや病院の待合室といった例も同様なシステムだと捉えることができる. このような身近な事例にも適用可能であるため, 回収レートを導出してシステムのパフォーマンスを評価する手法は, 汎用性が高いことが窺える.

本研究では, 基本的な問題設定に対して回収レートの導出を行った. 直線上に異なる 2 点 O, A をとり, 点 O には空き箱を用意する. 点 A ではランダムにボールが出現するものとする. 直線上には点 A で出現したボールを回収する人が存在し, 点 A と点 O の間を往復している. この人物は, 点 A で出現したボールを受け取り, 点 O まで移動して空き箱の中にボールを入れる. そして再び点 A に戻り, 同様な動作を繰り返す. そして, 単位時間あたりに空き箱の中に回収したボールの平均個数を回収レートと呼ぶ. このとき, 出現したボールがすぐに消えるかどうか, 人の往復時間が一定か否かによって 4 パターンの問題を考えることができる. 本論文ではこれら 4 つのパターンの問題について, 回収レートの値を数学的に導出する. さらに, これらのシミュレーションを行い, 導出した数学的な回収レートの値が一致することを確認する.

2　問題設定と回収レートの導出

直線上に異なる 2 点 O, A がある. 点 O には空き箱が用意されており, 点 A ではレート λ[個/s] のポアソン過程に従ってボールが出現する. 直線上には点 O と点 A の間を移動する人がいる (図 1). この人物は最初に点 A で静止している.

点 A で静止している人は, 点 A で出現したボールを受け取る. このときの時刻を $t = 0$ とする. ボールを受け取った人は, そのボールを保持したまま, 点 O に移動を開始する (図 2). 点 O に到着したら空き箱の中にボールを入れ, 点 A に向かって移動を開始する (図 3). ここで点 A から点 O を経て点 A に戻るまでの移動を往復と呼び, その移動時間を往復時間 [s] と呼ぶ.

ここで往復している間に出現するボールの挙動は, 点 A にバッファがあるかどうかで 2 通りのパターンを考える.

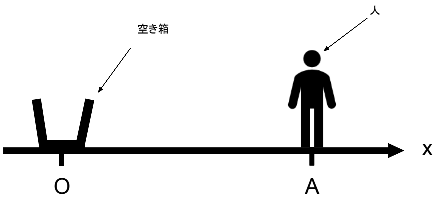

図 1: 空き箱と人

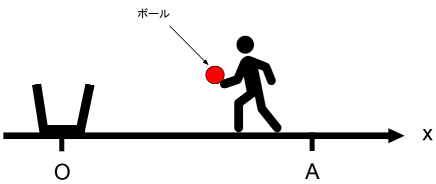

図 2: 点 A→O への人の移動

(i). 点 A にバッファが無い場合：

- 出現したボールはその場で消滅する.

(ii). 点 A にバッファが有る場合：

- 出現したボールは, バッファに 1 個だけ格納できる.
- ただし, バッファがすでにボールを格納している場合, 新たに出現したボールは消滅する.

その後, 往復を終了し点 A に到着したとき, 再びバッファの有無によって次の挙動が決まる.

(i). 点 A にバッファが無い場合：

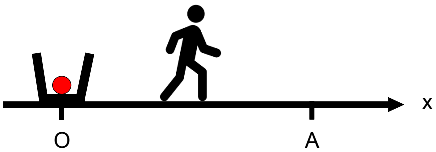

図 3: 点 O→A への人の移動

- 次のボールが出現するまで待機し, そのボールを受け取る.

(ii). 点 A にバッファが有る場合：

- バッファの中身を確認する.
  - ボールが格納されていれば, それを受け取る.
  - バッファが空の場合, 次のボールが出現するまで待機し, そのボールを受け取る.

点 A でボールを受け取ることができたら, 点 O に移動を開始する. そして, 同様の動作を繰り返す. このとき十分な時刻 t が経過したときの単位時間あたりに回収したボールの平均個数を回収レート h[個/s] とする. 本研究では, 以下の条件に基づいて, 異なる 4 つのパターンの問題設定を行い, 各問題における回収レートを導出する.

- 往復時間が一定の場合と指数分布に従う場合.
- 点 A にバッファが有る場合と無い場合.

ここで上記の問題設定というのは, 以下のように対応させることによって待ち行列システムとして表すことができる (図 4, 図 5).

1. ボールの出現 $\longrightarrow$ ボールの到着
2. 人の往復時間 $\longrightarrow$ サーバーのサービス時間
3. 点 A における人の待機時間 $\longrightarrow$ サーバーのアイドル時間
4. ボールの消失 $\longrightarrow$ ボールの棄却

このとき点 A で出現したボールを人が往復して回収するプロセスは, システムに到着したボールをサーバーで処理するプロセスと同等だと捉えることができる. したがって, ボールを

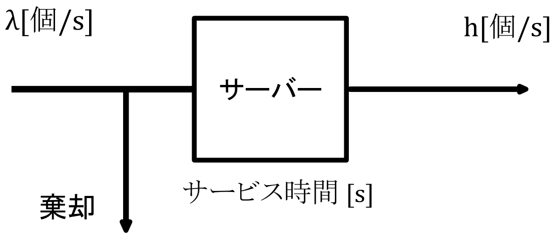

図 4: バッファ無しの待ち行列システム

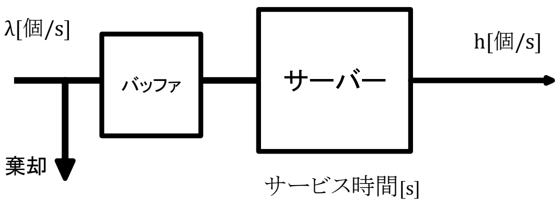

図 5: バッファ有りの待ち行列システム

回収する問題における回収レートを導出することは, 待ち行列システムにおける単位時間あたりにサーバーで処理したボールの平均個数を導出することと同義である. 以降では, 回収レートを導出するにあたって待ち行列システムに変換して考えていく.

2.1 一定の往復時間, バッファ無しの場合

往復時間が一定の t_0 で, バッファが無い場合を考える. これを問題 1 とする.

この問題は図 6 のような待ち行列システムで表され, 時間の経過に伴ってサービス時間とアイドル時間 (サーバーが稼働していない時間) が交互に繰り返される. このサイクルの内, サー

ビス時間とアイドル時間の 1 サイクルが 1 個のボールを回収する時間を表している (図 7). ここでサービス時間は一定値 t_0 となる.

一方アイドル時間は, サービス時間が終了してから, 次のボールが到着するまでの待機時間を表している. ここでボールの到着はレート λ のポアソン過程に従うため, 到着間隔はパラメータ λ の指数分布に従う. 指数分布の無記憶性より, 前のボールがいつ出現したのかに関わらず, サービス時間が終了してから次のボールが到着するまでの待機時間は指数分布に従う. すなわちアイドル時間はパラメータ λ の指数分布に従う. つまりアイドル時間の期待値は

$$\begin{aligned}\int_{-\infty}^{\infty} x\lambda e^{-\lambda x}dx &= \lambda\int_{0}^{\infty} xe^{-\lambda x}dx \\ &= \lambda\left[-\frac{1}{\lambda}xe^{-\lambda x}\right]_0^{\infty} - \lambda\int_0^{\infty}\left(-\frac{1}{\lambda}\right)e^{-\lambda x}dx \\ &= 0 + \int_0^{\infty} e^{-\lambda x}dx \\ &= \left[-\frac{1}{\lambda}e^{-\lambda x}\right]_0^{\infty} \\ &= \frac{1}{\lambda} \end{aligned} \tag{1}$$

となる. 以上より 1 個のボールを回収するのにかかる時間の期待値は $t_0 + \frac{1}{\lambda}$ なので, 回収レート h_1 は

$$h_1 = \frac{1}{t_0 + \frac{1}{\lambda}} \tag{2}$$

で表される. これが問題 1 における回収レートである.

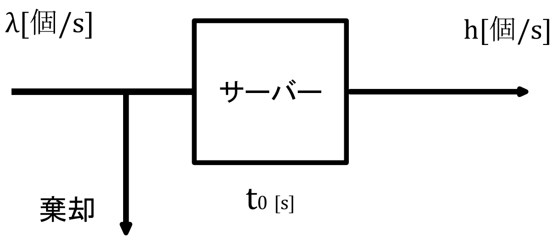

図 6: 待ち行列システム (1)

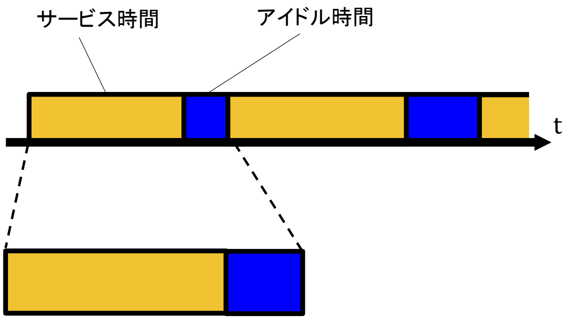

図 7: サービス時間とアイドル時間 (1)

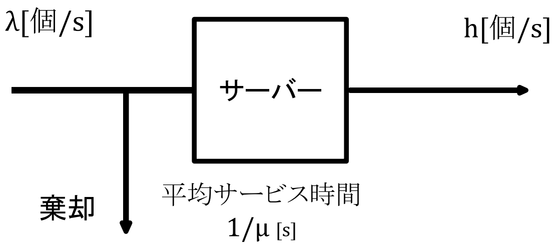

図 8: 待ち行列システム (2)

2.2 指数分布に従う往復時間, バッファ無しの場合

往復時間がパラメータ μ の指数分布に従い, バッファが無い場合を考える. これを問題 2 とする.

この問題は図 8 のような待ち行列システムとして表され, 問題 1 と同様に時間の経過に伴ってサーバーのサービス時間とアイドル時間が交互に繰り返される. そしてサービス時間とアイドル時間の 1 サイクルが 1 個のボールを回収する時間を表している. サービス時間は, 問題文

の条件よりパラメータ μ の指数分布に従う．つまりサービス時間の期待値は

$$\begin{aligned}\int_{-\infty}^{\infty} x\mu e^{-\mu x}dx &= \mu\int_{0}^{\infty} xe^{-\mu x}dx \\ &= \frac{1}{\mu}\end{aligned} \tag{3}$$

となる．一方アイドル時間は問題 1 の場合と同様にパラメータ λ の指数分布に従う．つまりアイドル時間の期待値は

$$\int_{-\infty}^{\infty} x\lambda e^{-\lambda x}dx = \frac{1}{\lambda} \tag{4}$$

で表される．以上より, 1 個のボールを回収するのにかかる時間の期待値は $\frac{1}{\mu}+\frac{1}{\lambda}$ なので, 回収レート h_2 は

$$h_2 = \frac{1}{\frac{1}{\mu}+\frac{1}{\lambda}} \tag{5}$$

で表される．これが問題 2 の回収レートである．

2.3 一定の往復時間, バッファ有りの場合

往復時間が一定の t_0 で, 点 A にバッファが有る場合を考える．これを問題 3 とする．

この問題は図 9 のような待ち行列システムとして表され, 時間の経過に伴ってサービス時間とアイドル時間が交互に繰り返されるとは限らない．サービス時間が連続して続く場合がある．つまり, ボール 1 個を回収するのにかかる時間には, アイドル時間がゼロのときとゼロでないときの 2 パターンが生じる (図 10).

アイドル時間がゼロになる場合, 1 個のボールを回収するのにかかる時間の期待値は, サービス時間と等しい．すなわち, 一定時間 t_0 である．アイドル時間がゼロにならない場合, 1 個のボールを回収するのにかかる時間の期待値は, サービス時間とアイドル時間の期待値で表され

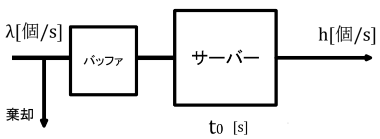

図 9: 待ち行列システム (3)

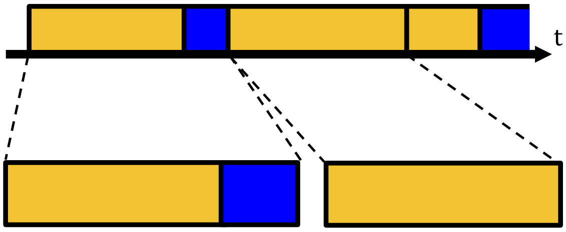

図 10: サービス時間とアイドル時間 (2)

る．ここでサービス時間は先ほどと同様に一定値 t_0 となる．一方, アイドル時間は, 問題 1, 問題 2 と同様にパラメータ λ の指数分布に従うので, 期待値は $\frac{1}{\lambda}$ となる．以上より 1 個のボールを回収するのにかかる時間の期待値は $t_0 + \frac{1}{\lambda}$ と表される．

次に, アイドル時間がゼロになる場合とゼロにならない場合のそれぞれの確率を求める．アイドル時間がゼロになる場合というのは, サービス時間 t_0 の間にバッファの中にボールが格納されている場合, すなわちサービス時間 t_0 の間にボールが 1 個以上到着する場合であると考えることができる．また, アイドル時間がゼロにならない場合というのは, サービス時間 t_0 の間バッファが空の場合, すなわち, サービス時間 t_0 の間にボールが 0 個到着する場合であると考えることができる．

ここでボールの到着は到着レート λ のポアソン過程に従っており, ポアソン過程の到着回数の確率分布はポアソン分布に従っている．ポアソン分布の定義より, 到着レート λ のポアソン過程に従って到着するボールが時間 t_0 の間に k 個到着する確率は

$$P(X = k) = \frac{(\lambda t_0)^k e^{-\lambda t_0}}{k!}, \quad k = 0, 1, 2, \ldots \tag{6}$$

と表すことができる．すなわち, それぞれの確率は

$$P(X > 0) = 1 - e^{-\lambda t_0} \qquad \text{(アイドル時間がゼロになる場合)} \tag{7}$$

$$P(X = 0) = e^{-\lambda t_0} \qquad \text{(アイドル時間がゼロにならない場合)} \tag{8}$$

と表される．以上より 1 個のボールを処理するのにかかる時間の期待値は

$$e^{-\lambda t_0} \times \left(t_0 + \frac{1}{\lambda} \right) + (1 - e^{-\lambda t_0}) \times t_0$$

となるから, 回収レート h_3 は

$$
\begin{aligned}
h_3 &= \frac{1}{e^{-\lambda t_0} \times \left(t_0 + \frac{1}{\lambda}\right) + (1 - e^{-\lambda t_0}) \times t_0} \\
&= \frac{1}{t_0 + e^{-\lambda t_0} \times \frac{1}{\lambda}}
\end{aligned} \tag{9}
$$

と表せる．これが問題 3 の回収レートである．

2.4　指数分布に従う往復時間, バッファ有りの場合

往復時間がパラメータ μ の指数分布に従い, バッファが有る場合を問題 4 とする．この問題は図 11 のような待ち行列システムとして表すことができる．本問題では, 回収レートを導出するにあたって, ベルヌーイ過程に基づいた手法を用いる．

まず時間軸を非常に短い間隔 Δ で区切る．一つの間隔をタイムスロットと呼ぶ (図 12)．各タイムスロットでは独立的確率試行が行われる．システムに到着がある確率を $p = \lambda\Delta$, サーバーにサービスを終了する確率を $q = \mu\Delta$ と定義する．

次にシステムの状態を定義する. システムの状態は次の 3 つに分けることができる.

- 状態 0: サーバーとバッファにボールが両方とも空である
- 状態 1: サーバーが 1 個のボールを処理中であり, バッファが空である
- 状態 2: サーバーが 1 個のボールを処理中, バッファにボールが 1 個格納されている

このとき状態遷移図は図 13 と表される．次に状態 1, 状態 2 の状態確率を求める．各状態の確

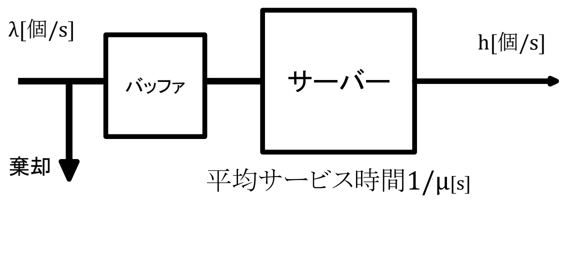

図 11: 待ち行列システム (4)

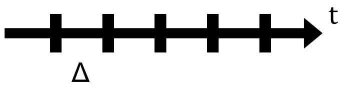

図 12: タイムスロット

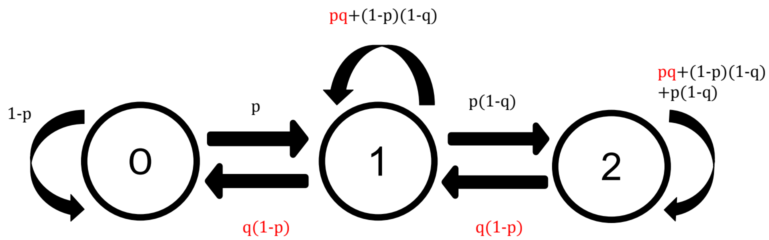

図 13: 状態遷移図

率を P_0, P_1, P_2 とおくと状態遷移方程式は

$$\begin{cases} pP_0 = q(1-p)P_1 \\ p(1-q)P_1 = q(1-p)P_2 \\ P_0 + P_1 + P_2 = 1 \end{cases} \tag{10}$$

となる．この方程式を解くと

$$P_1 = \frac{\frac{p}{q(1-p)}}{1+\frac{p}{q(1-p)}+\frac{p^2(1-q)}{q^2(1-p)^2}} \tag{11}$$

$$P_2 = \frac{\frac{p^2(1-q)}{q^2(1-p)^2}}{1+\frac{p}{q(1-p)}+\frac{p^2(1-q)}{q^2(1-p)^2}} \tag{12}$$

となる．

ここで図 13 の状態遷移図では, 時間 Δ ごとに状態を遷移する．さらに, ボールを回収する遷移は赤色の箇所に相当する．以上より, 時間 Δ あたりに回収されるモノの個数の期待値 $\bar{N}$ は

$$\begin{aligned}\bar{N} &= pqP_1 + q(1-p)P_1 + pqP_2 + q(1-p)P_2 = (P_1+P_2)q \\ &= \frac{(\frac{p}{q(1-p)}+\frac{p^2(1-q)}{q^2(1-p)^2})q}{1+\frac{p}{q(1-p)}+\frac{p^2(1-q)}{q^2(1-p)^2}} \end{aligned} \tag{13}$$

となる．つまり単位時間あたりに回収されるボールの個数の期待値は $\frac{\bar{N}}{\Delta}$ と表される．Δ は時間軸を離散的に区切ったものなので, $\Delta \to 0$ とすると

$$\begin{aligned}\lim_{\Delta\to 0}\frac{\bar{N}}{\Delta} &= \lim_{\Delta\to 0}\frac{1}{\Delta}\times\frac{(\frac{p}{q(1-p)}+\frac{p^2(1-q)}{q^2(1-p)^2})q}{1+\frac{p}{q(1-p)}+\frac{p^2(1-q)}{q^2(1-p)^2}} \\ &= \lim_{\Delta\to 0}\frac{1}{\Delta}\times\frac{pq^2(1-p)+p^2q(1-q)}{q^2(1-p)^2+pq(1-p)+p^2(1-q)} \\ &= \lim_{\Delta\to 0}\frac{\lambda\mu^2\Delta^2(1-\lambda\Delta)+\lambda^2\mu\Delta^2(1-\mu\Delta)}{\mu^2\Delta^2(1-\lambda\Delta)^2+\lambda\mu\Delta^2(1-\lambda\Delta)+\lambda^2\Delta^2(1-\mu\Delta)} \\ &= \frac{\lambda\mu(\lambda+\mu)}{\lambda^2+\lambda\mu+\mu^2} \\ &= \frac{1}{\frac{1}{\mu}+\frac{1}{\lambda}-\frac{1}{\lambda+\mu}} \end{aligned} \tag{14}$$

となる．以上より回収レート h_4 は

$$h_4 = \frac{1}{\frac{1}{\mu}+\frac{1}{\lambda}-\frac{1}{\lambda+\mu}} \tag{15}$$

となる．これが問題 4 で求める回収レートである．

3　シミュレーションによる検証

第 2 章で求めた回収レートの理論値が正しいかどうかをシミュレーションによって検証する．シミュレーション時間 n[s] を設定する．このとき $\sum_{k=1}^{m} X_k \leqq n$ が成り立つ範囲で, パラ

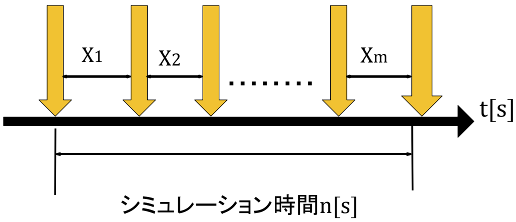

図 14: シミュレーション時間

メータ λ に従う指数乱数 $X_1, X_2, X_3, \cdots, X_m$[s] を生成する (図 14). ここで指数乱数 $X_1, X_2, X_3, \cdots, X_m$ がボールの出現間隔を表している. ボールの出現間隔から, 回収できたボールの個数を計測する. 回収したボールの個数を r 個とすると, 回収レートは $h = \frac{r}{n}$[個/s] と計算できる. パラメータ t_0 と μ の値を固定し, パラメータ λ だけを変化させたときの回収レート h の変化をグラフに表す.

3.1 一定の往復時間のシミュレーション結果

パラメータ値 $n = 10000$, $t_0 = 0.1$ でシミュレーションを行った. h_1 と h_3 のシミュレーション結果はそれぞれ図 15, 図 16 のようになった. 青色の実線が理論値, 赤色の点線がシミュレーション値を表しており, 両者は一致していると言える. ここで本シミュレーションではパ

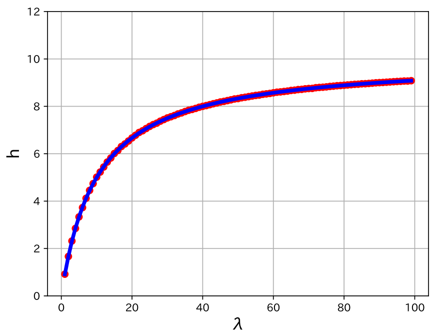

図 15: $n = 10000$, $t_0 = 0.1$ のときの h_1

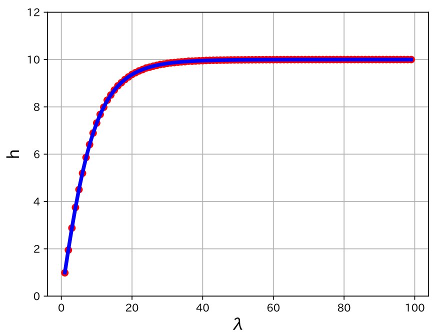

図 16: $n = 10000$, $t_0 = 0.1$ のときの h_3

ラメータ t_0 の値を固定して行った．任意のパラメータ値 t'_0 に対しても, 理論値とシミュレーション値が一致することを示す.

パラメータ $t'_0 = at_0$(a は正の実数) とおく．回収レート h_1, h_3 に t'_0 を代入したものをそれぞれ $\overline{h}_1$, $\overline{h}_3$ とする．このとき $\overline{h}_1$ と $\overline{h}_3$ は

$$\overline{h}_1 = \frac{1}{t' + \frac{1}{\lambda}} = \frac{1}{at_0 + \frac{1}{\lambda}} = \frac{1}{a} \times \frac{1}{t_0 + \frac{1}{a\lambda}} \tag{16}$$

$$\overline{h}_3 = \frac{1}{t' + e^{-\lambda t'} \times \frac{1}{\lambda}} = \frac{1}{at_0 + e^{-\lambda(at_0)} \times \frac{1}{\lambda}} = \frac{1}{a} \times \frac{1}{t_0 + e^{-(a\lambda)t_0} \times \frac{1}{a\lambda}} \tag{17}$$

と表すことができる．ここから $\overline{h}_1$ と $\overline{h}_3$ は, h_1 と h_3 のグラフをそれぞれ x 軸方向に $\frac{1}{a}$ 倍, y 軸方向に $\frac{1}{a}$ 倍したものである．つまり h_1 と h_3 が一致することを検証すれば, $\overline{h}_1$ と $\overline{h}_3$ も一致することが示せる．以上より任意のパラメータ値 t_0 に対してシミュレーション値と理論値が一致することが示せた.

3.2 指数分布に従う往復時間のシミュレーション結果

パラメータ値 $n = 10000$, $\mu = 10$ でシミュレーションを行った．h_2 と h_4 のシミュレーション結果はそれぞれ図 17, 図 18 のようになった．青色の実線が理論値, 赤色の点線がシミュレーション値を表しており, 両者は一致していると言える．ここで本シミュレーションではパラメータ μ の値を固定して行った．任意のパラメータ値 μ' に対しても, 理論値とシミュレーション値が一致することを示す.

パラメータ $\mu' = b\mu$(b は正の実数) をおく．回収レート h_2, h_4 に μ' を代入したものをそれ

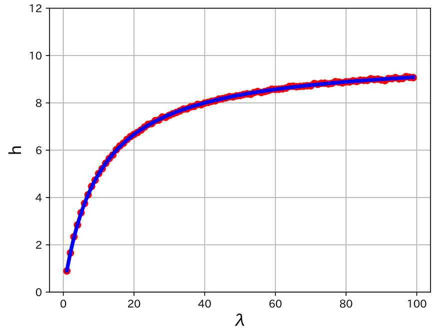

図 17: $n = 10000$, $\mu = 10$ のときの h_2

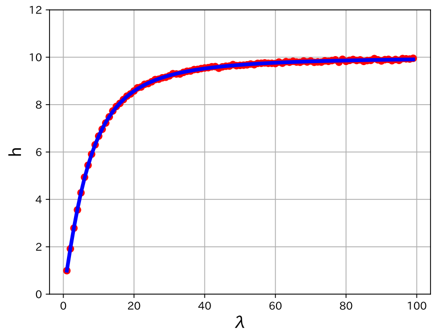

図 18: $n = 10000$, $\mu = 10$ のときの h_4

ぞれ $\overline{h}_2, \overline{h}_4$ とする．このとき $\overline{h}_2$ と $\overline{h}_4$ は

$$\bar{h_2} = \frac{1}{\frac{1}{\mu'} + \frac{1}{\lambda}} = \frac{1}{\frac{1}{b\mu} + \frac{1}{\lambda}} = b \times \frac{1}{\frac{1}{\mu} + \frac{b}{\lambda}} \tag{18}$$

$$\begin{aligned} \bar{h_4} &= \frac{1}{\frac{1}{\mu'} + \frac{1}{\lambda} - \frac{1}{\lambda+\mu'}} = \frac{1}{\frac{1}{b\mu} + \frac{1}{\lambda} - \frac{1}{\lambda+b\mu}} \\ &= b \times \frac{1}{\frac{1}{\mu} + \frac{\lambda}{b} - \frac{1}{\frac{\lambda}{b}+\mu}} \end{aligned} \tag{19}$$

と表すことができる．ここから $\overline{h}_2$ と $\overline{h}_4$ は, h_2 と h_4 のグラフをそれぞれ x 軸方向に $\frac{1}{b}$ 倍, y 軸方向に b 倍したものである．つまり h_2 と h_4 がシミュレーション値と一致することを検証すれば, $\overline{h}_2$ と $\overline{h}_4$ も一致することが示せる．すなわち任意のパラメータ値 μ に対してシミュレーション値と理論値が一致することが示せた．

4　収束速度に関する考察

4.1　一定の往復時間の場合

回収レート h_1 と h_3 について $\lambda \to \infty$ のとき

$$h_1 = \frac{1}{t_0 + \frac{1}{\lambda}} \to \frac{1}{t_0} \tag{20}$$

$$h_3 = \frac{1}{t_0 + e^{-\lambda} \times \frac{1}{\lambda}} \to \frac{1}{t_0} \tag{21}$$

となる．どちらも $\frac{1}{t_0}$ に収束する．次に h_1 と h_3 がどれぐらい早く $\frac{1}{t_0}$ に収束するか求める．h_1, h_3 の極限値 $\frac{1}{t_0}$ との差をそれぞれ $\overline{h}_1$, $\overline{h}_3$ とすると

$$\overline{h}_1 = \frac{1}{t_0} - h_1 \tag{22}$$

$$\overline{h}_3 = \frac{1}{t_0} - h_3 \tag{23}$$

となる．さらに式を変形すると

$$\overline{h}_1 = \frac{1}{t_0} - \frac{1}{t_0 + \frac{1}{\lambda}} = \frac{1}{t_0(\lambda t_0 + 1)} \tag{24}$$

$$\overline{h}_3 = \frac{1}{t_0} - \frac{1}{t_0 + e^{-\lambda t_0} \times \frac{1}{\lambda}} = \frac{1}{t_0(\lambda e^{\lambda t_0} + 1)} \tag{25}$$

となる．$\overline{h}_1$ の収束の速さは $O(\frac{1}{\lambda t_0 + 1})$ に比例し, $\lambda \to \infty$ のときに $O(\frac{1}{\lambda})$ の速さで収束する．また $\overline{h}_3$ の収束の速さは $O(\frac{1}{\lambda e^{\lambda t_0} + 1})$ に比例し, $\lambda \to \infty$ のときに $O(\frac{e^{-\lambda t_0}}{\lambda})$ の速さで収束する．すなわち h_1 は $O(\frac{1}{\lambda})$ の速さで収束し, h_3 は $O(\frac{e^{-\lambda t_0}}{\lambda})$ の速さで収束する．h_3 は指数関数的に収束するので, h_1 よりも遥かに早く収束する．

以上より, $\lambda \to \infty$ のとき, バッファ有りの場合の回収レートの方がバッファ無しの場合の回収レートよりも速く収束することがいえる．

4.2　指数分布に従う往復時間の場合

回収レート h_2 と h_4 について $\lambda \to \infty$ のとき

$$h_2 = \frac{1}{\frac{1}{\mu} + \frac{1}{\lambda}} \to \mu \tag{26}$$

$$h_4 = \frac{1}{\frac{1}{\mu} + \frac{1}{\lambda} - \frac{1}{\mu + \lambda}} \to \mu \tag{27}$$

となる．どちらも μ に収束する．

次に h_2 と h_4 の両関数がどれぐらい早く μ に収束するか求める．h_2, h_4 の収束値 μ との差をそれぞれ $\overline{h}_2, \overline{h}_4$ とすると

$$\overline{h}_2 = \mu - h_2 \tag{28}$$

$$\overline{h}_4 = \mu - h_4 \tag{29}$$

となる．さらに式を変形すると

$$\overline{h}_2 = \mu - \frac{1}{\frac{1}{\mu} + \frac{1}{\lambda}} = \frac{\mu}{\frac{1}{\mu}\lambda + 1} \tag{30}$$

$$\overline{h}_4 = \mu - \frac{1}{\frac{1}{\mu} + \frac{1}{\lambda} - \frac{1}{\mu+\lambda}} = \frac{\mu^3}{\lambda^2 + \lambda\mu + \mu^2} \tag{31}$$

となる．$\overline{h}_2$ の収束の速さは $O(\frac{1}{\frac{1}{\mu}\lambda+1})$ に比例し, $\lambda \to \infty$ のときに $O(\frac{1}{\lambda})$ の速さで収束する．また $\overline{h}_4$ の収束の速さは $O(\frac{1}{\lambda^2+\mu\lambda+\mu^2})$ に比例し, $\lambda \to \infty$ のときに $O(\frac{1}{\lambda^2})$ の速さで収束する．すなわち h_2 は $O(\frac{1}{\lambda})$ の速さで収束し, h_4 は $O(\frac{1}{\lambda^2})$ の速さで収束する．すなわち h_4 の方が h_2 よりも早く収束する．

以上より, $\lambda \to \infty$ のとき, バッファ有りの回収レートの方がバッファ無しの回収レートの場合よりも速く収束することが言える．

5 まとめ

本研究では, 往復時間が一定の場合と指数分布に従う場合, また点 A にバッファがある場合とない場合の 4 つのパターン問題設定を行い, その問題を待ち行列システムに変換することによって回収レートを導出した．さらに, シミュレーションを行い, 理論的に導出した回収レートとシミュレーション結果が一致することを確認した．$\lambda \to \infty$ の場合における回収レートの収束速度を比較した結果, バッファ有りの場合の方がバッファ無しの場合に比べて収束が速いことが明らかとなった．今後の展望としては, ボールの出現箇所を増やすなど, 問題設定を拡張し, より複雑なシステムにおける回収レートを導出していきたいと考える．

謝辞

本研究の作成にあたり, 多大なるご指導を賜りました西新幹彦教授に深く感謝申し上げる．

参考文献

[1] Randolph Nelson, Probability, stochastic processes, and queueing theory: the mathematics of computer performance modeling, Springer-Verlag, New York, 1995

付録 A　ソースコード

到着レート λ を変化させたときの回収レート h の変化をグラフにプロットするプログラムを以下に掲載する.

A.1　一定の往復時間, バッファ無しの場合

```
import math
import numpy as np
import matplotlib.pyplot as plt
import japanize_matplotlib
rng = np.random.default_rng()
def func_server_1(t_0,lamda,n):
    s = -(math.log(1 - rng.random())/lamda)
    sum_a_1 = s
    sum_a_2 = s
    sum_return = 1
    while (sum_a_1 <= n):
        if sum_a_2 >= t_0:
            sum_a_2 = 0.0
            sum_return += 1
        s = -(math.log(1 - rng.random())/lamda)
        sum_a_1 += s
        sum_a_2 += s
    h = (sum_return)/n
    return h
x = list(range(1,100,1))
y1 = [func_server_1(0.1,j,10000) for j in x]
y2 = [j/(0.1*j + 1) for j in x]

fig, ax = plt.subplots()
ax.grid()
ax.set_title("",fontsize=20)
ax.set_xlabel("λ",fontsize=16)
ax.set_ylabel("h",fontsize=16)
ax.set_ylim(0,12)

plt.plot(x, y1, marker="o", color = "red", linestyle = "--",label='実測値')
plt.plot(x, y2, color = "blue", linestyle = "solid",label='理論値',linewidth=3.5);
plt.savefig('s_1_ittei_graph.pdf')
plt.legend()
plt.show()
```

A.2　指数分布に従う往復時間, バッファ無しの場合

```
import math
import numpy as np
import matplotlib.pyplot as plt
import japanize_matplotlib
rng = np.random.default_rng()

def func_server_1(lamda,service,n):
    s_a = -(math.log(1-rng.random())/lamda)
    sum_a_1 = s_a
    sum_a_2 = s_a
    sum_return = 1
    s_s = -(math.log(1 - rng.random())/service)
    while (sum_a_1 <= n):
        if sum_a_2 >= s_s:
```

```
            sum_return += 1
            sum_a_2 = 0.0
            s_s = -(math.log(1 - rng.random())/service)
        s_a = -(math.log(1 - rng.random())/lamda)
        sum_a_1 += s_a
        sum_a_2 += s_a
    h = (sum_return)/n
    return h

x = list(range(1,100,1))
y1 = [func_server_1(j,10,10000) for j in x]
y2 = [10*j/(10 + j) for j in x]

fig, ax = plt.subplots()
ax.grid()
ax.set_title("",fontsize=20)
ax.set_xlabel("λ",fontsize=16)
ax.set_ylabel("h",fontsize=16)
ax.set_ylim(0,12)
plt.plot(x, y1, marker="o", color = "red", linestyle = "--",label='実測値')
plt.plot(x, y2, color = "blue", linestyle = "solid",label='理論値',linewidth=3.5);
plt.savefig('s_1_sisu_graph.pdf')
plt.legend()
plt.show()
```

A.3 一定の往復時間, バッファ有りの場合

```
import math
import numpy as np
import matplotlib.pyplot as plt
import japanize_matplotlib
rng = np.random.default_rng()
def func_server_1_queue_1(t_0,lamda,n):
    s_a = -(math.log(1 - rng.random())/lamda)
    sum_a_1 = s_a
    sum_a_2 = s_a
    sum_return = 1 #回収された数の合計
    queue = 0 #queue の状態。0のとき空，1のとき満杯
    while (sum_a_1 <= n):
        if sum_a_2 < t_0:
            if queue == 0:
                sum_return += 1
                queue = 1
        else:
            if queue == 0:
                sum_a_2 = 0
                sum_return += 1
            if queue == 1:
                sum_a_2 -= t_0
                sum_return += 1
                if (sum_a_2 - t_0) >= 0:
                    queue = 0
                    sum_a_2 = 0
        s_a = -(math.log(1 - rng.random())/lamda)
        sum_a_1 += s_a
        sum_a_2 += s_a
    h = (sum_return)/n
    return h

x = list(range(1,100,1))
y1 = [func_server_1_queue_1(0.1,j,10000) for j in x]
y2 = [j/(0.1*j + math.exp(-0.1*j)) for j in x]
```

```
fig, ax = plt.subplots()
ax.plot(x, y1, marker="o",color = "red",linestyle = "--",label='実測値')
ax.plot(x, y2, color = "blue", linestyle = "solid",label='理論値',linewidth=3.5)
ax.grid()
ax.set_xlabel("λ",fontsize=16)
ax.set_ylabel("h",fontsize=16)
ax.set_ylim(0,12)
plt.savefig('s_2_ittei_graph.pdf')
plt.legend()
plt.show()
```

A.4 指数分布に従う往復時間, バッファ有りの場合

```
import math
import numpy as np
import matplotlib.pyplot as plt
import japanize_matplotlib
rng = np.random.default_rng()

def func_server_1_queue_1(lamda,service,n):
    s_a = -(math.log(1 - rng.random())/lamda)
    s_s = -(math.log(1 - rng.random())/service)
    sum_a_1 = s_a
    sum_a_2 = s_a
    sum_return = 1 #回収された数の合計
    queue = 0 #queue の状態。0のとき空, 1のとき満杯
    while (sum_a_1 <= n):
        if sum_a_2 < s_s:
            if queue == 0:
                sum_return += 1
                queue = 1
        else:
            if queue == 0:
                sum_a_2 = 0.0
                s_s = -(math.log(1 - rng.random())/service)
                sum_return += 1
            if queue == 1:
                sum_return += 1
                sum_a_2 -= s_s
                s_s = -(math.log(1 - rng.random())/service)
                if sum_a_2 >= s_s:
                    sum_a_2 = 0.0
                    queue = 0
                    s_s = -(math.log(1 - rng.random())/service)
        s_a = -(math.log(1 - rng.random())/lamda)
        sum_a_1 += s_a
        sum_a_2 += s_a
    h = (sum_return)/n
    return h
x = list(range(1,100,1))
y1 = [func_server_1_queue_1(j,10,10000) for j in x]
y2 = [ 10*j*(10 + j)/(math.pow(10,2) + 10*j + math.pow(j,2)) for j in x]


fig, ax = plt.subplots()
plt.plot(x, y1, marker="o", color = "red", linestyle = "--",label='実測値')
plt.plot(x, y2, color = "blue", linestyle = "solid",label='理論値',linewidth=3.5)
ax.set_ylim(0,12)
ax.set_xlabel("λ",fontsize=16)
ax.set_ylabel("h",fontsize=16)
ax.grid()
plt.savefig('s_2_sisu_graph.pdf')
plt.legend()
plt.show()
```