2024 年度版

システムプログラミング実験

確率プログラミング　第3回

担当：中嶋 一貴

第 3 回：待ち行列シミュレーション

1　本日の目標

待ち行列は，人などが処理やサービスなどの資源を受けるための利用要求の数理モデルである．例えば，銀行の ATM に並ぶ顧客の列やスーパーのレジに並ぶ客の列が挙げられる．このようなシステムを設計する際には，待ち行列の振る舞いを予測して，処理するもの（例：銀行の場合は ATM，スーパーの場合はレジ）の性能や配置を決めることが重要である．本実験では，ランダムなイベント到着を表す基本的な確率過程の 1 つであるポアソン過程のシミュレータを作成し，待ち行列の特性を解析する．

2　実験の理論概要

本実験では，あるイベント（例：ATM の顧客の到着）を発生する時刻を考え，そのイベントがランダムに発生する事象（図 1）を記述する基本的な確率過程のポアソン分布について学ぶ．

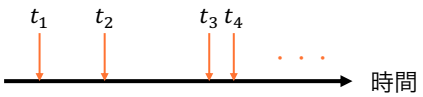

図 1: ランダムなイベント発生の例．

ポアソン過程では，イベント発生に関する以下の仮定を満たす．

1. 異なるイベントは互いに独立に発生する．
2. 微小区間 Δt 内に異なるイベントは発生しない．
3. 時間単位あたりに平均して λ 回のイベントが発生する．つまり，時間間隔 T での発生回数は λT 回である．

このとき，ポアソン過程におけるランダムなイベント発生の規則は，

(a) 時間間隔 T で発生するイベント回数は，平均 λ のポアソン分布に従う（2.2 節参照）．

(b) 発生時刻の間隔は，平均 $\frac{1}{\lambda}$ の指数分布に従う（2.3 節参照）．

なお，上記の 2 つの規則は同値である．すなわち (a) を満たせば (b) が成り立つ（逆も同様）．

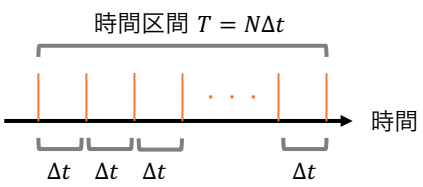

図 2: 時間区間 T の N 分割.

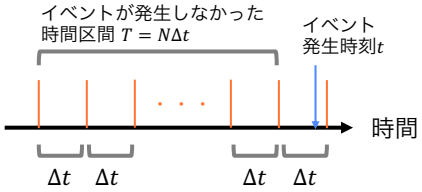

図 3: イベントが発生しない $T = N\Delta t$ 区間と発生区間 $(T, T + \Delta t)$.

2.1 イベント回数の頻度分布

時間区間 T 内のランダムなイベントの発生回数を考える．まず，時間区間 T を微小区間 Δt に N 分割する（図 1）．つまり，$T = N\Delta t$ である．このとき，1 つの区間 Δt でイベントが発生する確率は

$$p = \frac{\lambda T}{N}$$

である．N 個の ΔT 区間でイベントが k 個発生する確率 $P_N(k)$ は次の二項分布に従う．

$$P_N(k) = {}_N C_k p^k (1-p)^{N-k}.$$

$p = \frac{\lambda T}{N}$ を満たしながら $N \to \infty$ とすると，$P_N(k)$ はポアソン分布に収束する．

$$\begin{aligned}
\lim_{N\to\infty} P_N(k) &= \lim_{N\to\infty} {}_N C_k p^k (1-p)^{N-k} \\
&= \lim_{N\to\infty} \frac{N!}{(N-k)!k!} \left(\frac{\lambda T}{N}\right) \left(1 - \frac{\lambda T}{N}\right)^{N-k} \\
&= \frac{(\lambda T)^k}{k!} e^{-\lambda T}. \qquad (1)
\end{aligned}$$

ここで，以下を使った．

$$\lim_{N\to\infty} \left(1 - \frac{a}{N}\right)^N = e^{-a}.$$

なお，ある定数 λ に対して，0 以上の整数を値に取る確率変数 X が

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

を満たすとき，確率変数 X はパラメータ λ のポアソン分布に従うという．

2.2 発生時刻の間隔の分布

あるイベントが発生してからその次のイベントが発生するまでの時間間隔の分布を考える．あるイベントが発生した時刻を 0 とし，次のイベントが発生する時刻を t とする．t' はいずれかの微小区間 Δt 内に含まれる．そこで，イベントが発生しなかった区間とイベントが発生した区間に分割する（図 3）．すなわち，N 個の Δt の区間でイベントが発生せず，その次の区間 Δt でイベントが発生したとする．この確率は

$$(1 - \lambda\Delta t)^N \lambda\Delta t = \left(1 - \frac{\lambda t}{N}\right) \lambda\Delta t$$

で与えられる．なお，1 つの区間 Δt でイベントが発生する確率は $p = \frac{\lambda T}{N} = \lambda \Delta t$ である．$N \to \infty$ とすれば，微小区間 $(T, T + \Delta t]$ でイベントが発生する確率 $p(t)dt$ は

$$\begin{aligned} p(t)dt &= \lim_{N \to \infty} \left(1 - \frac{\lambda t}{N}\right)^N \lambda \Delta t \\ &= \lambda e^{-\lambda t} dt \end{aligned}$$

で与えられる．つまり，ランダムなイベントの発生間隔の確率密度関数 $p(t)$ はパラメータ λ の指数分布に従う．

課題 3-1: ポアソン過程のイベントクラスの作成

ポアソン過程のイベントクラス ‘Event’ を作成せよ．なお，作成するクラスは以下を満たすこと．

1. ‘Event’ クラスに 2 つのメンバ変数を追加する：（1）イベントの ID を指す ‘id’ （2）イベントの発生時刻を指す ‘time’．‘id’ は int 型，‘time’ は double 型とする．なお，両方とも公開メンバ変数でよい．

2. コンストラクタ ‘Event(const int i, const double t)’ を定義する．コンストラクタとは，クラスオブジェクトを初期化するために，オブジェクト生成時に呼び出される関数である．ここでのコンストラクタは，‘i’ と ‘t’ を引数にとり，イベント ID が ‘i’, 発生時刻が ‘t’ の ‘Event’ オブジェクトを作成するように定義せよ．

3. デストラクタ ‘∼Event()’ を定義する．デストラクタは，オブジェクトが破棄される時に呼び出される関数である．なお，今回は ‘Event’ クラスのメンバ変数が int や double の通常型のため，関数内の処理には特に何も書く必要はない．つまり，‘∼Event() = {}’ とすればよい．

レポートには，作成したクラスのソースファイル名を言及すること．例えばファイル名が ‘xyz.cpp’ の場合，‘xyz.cpp を参照のこと．’ と書けばよい．ファイルが複数ある場合，全て言及すること．なお，考察を書く必要はない．

課題 3-2: ポアソン過程に従うイベント列を返す関数の作成

引数に実数 $\lambda > 0$ と 整数 n を受け取り，ポアソン過程に従う n 個の ‘Event’ オブジェクトから成るリストを返す関数 ‘poisson_exp’ を作成せよ．

なお，関数 ‘poisson_exp’ 内の処理が以下の手順に従うように設計するとよい．

1. 時刻を表す double 型の変数 ‘t’ を 0 に初期化する．
2. int 型の変数 ‘i’ とし，$\mathrm{i} = 0, \ldots, n-1$ に対して以下の処理を順番に行う．
   (a) パラメータ λ の指数分布に従う乱数 ‘rnd’ を 1 個生成する．生成方法は第 1 回の課題 1-1 で学んだ通りである．なお，‘rnd’ はポアソン過程の連続するイベント間の時間間隔に対応する（2.2 章を参照）．
   (b) ‘Event(const int i, const double t)’ 関数を用いて，イベント ID が ‘i’, 発生時刻が ‘t’ の Event オブジェクトを生成する．適切な箇所で ‘t’ に ‘rnd’ を足す必要があることに注意せよ．
   (c) ‘Event’ オブジェクトのリストに生成した Event オブジェクトを追加する．‘Event’ オブジェクトのリストは ‘std::vector<Event>’ で表現するとよい．

レポートには，作成した関数のソースファイル名を言及すること．例えばファイル名が ‘xyz.cpp’ の場合，‘xyz.cpp を参照のこと．’ と書けばよい．ファイルが複数ある場合，全て言及すること．なお，考察を書く必要はない．

課題 3-3: ポアソン過程のシミュレーション

ポアソン過程に従うイベントを $n = 100,000$ 個発生させ，時間区間 $T = 1$ のイベント発生回数 k の確率分布を考察せよ．レポートには，$\lambda = 1.0, 1.5, 2.0$ のそれぞれについて以下の 2 つを報告すること．

- 時間区間 $T = 1$ のイベント発生回数 k の確率分布のプロット図．$k = 0, 1, 2, 3, 4, 5, 6$ でよい．図 4 にプロット例を示す．
- シミュレーションにより得られた確率分布と，理論値の確率分布との比較の考察．なお，理論値の確率分布については式 (1) を参照せよ．

なお，確率分布を求めるには，以下の手順に従って取り組むとよい．

1. 関数 'poisson_exp' を用いて，パラメータ λ のポアソン過程に従う $n = 100,000$ 個のイベントから成るリストを作成する．
2. 'i' を 0 以上の int 型の整数として，時間区間 [i, i+1) 内に発生したイベントの個数を記録する配列 'num_event_lst' を用意する．
3. 'num_event_lst' をもとにイベント発生回数 k の確率分布を計算する．

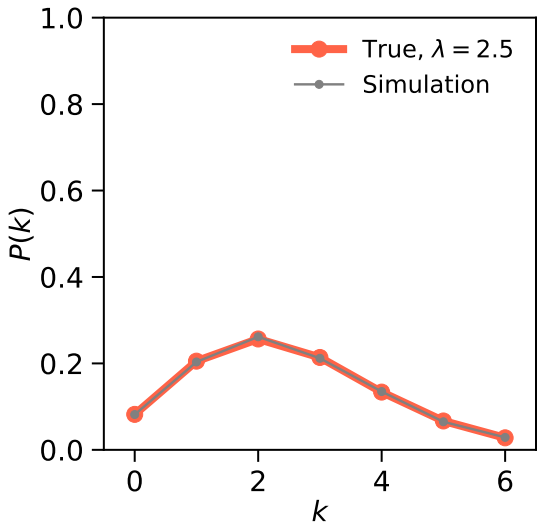

図 4: $\lambda = 2.5$ のときのポアソン分布（'True, $\lambda = 2.5$' のラベル）と，シミュレーションにより得られた時間区間 $T = 1$ 内のイベント発生回数 k の確率分布（'Simulation' のラベル）．2 つの分布はほとんど重なっているため，理論値の分布の線の太さと点の大きさを大きくした．

課題 3-4: バス停における待ち時間

あるバス停では，平均 $1/\lambda$ の指数分布に従う時間間隔でバスが到着する．このバス停に毎日ランダムなタイミングで到着した場合の平均待ち時間がどうなるか，検証せよ．
レポートには以下の 2 つを書くこと．

- $\lambda = 1, 1.5, 2.0$ のそれぞれについて，バス停にランダムなタイミングで到着した回数 m の関数としてバスの平均待ち時間をプロットした図．$1 \leq m \leq 10,000$ とせよ．図 5 にプロット例を示す．
- 平均待ち時間の理論値とシミュレーションで得られた数値結果の比較と考察．理論値は λ の関数として表される（本やウェブで調べること）．

なお，次の手順に従って取り組むとよい．

1. 課題 2-2 で作成した関数 'poisson_exp' を用いて，時間間隔が平均 $1/\lambda$ の指数分布に従う $n = 1,000$ 個のバス到着のイベント列を作成する．
2. バス停にランダムに到着する試行を m 回行うとする．i 回目（$i = 1, \ldots, m$）にバス停にランダムに到着したときのバスの待ち時間 w_i を計算する．$n = 1,000$ 個目のバスの到着時刻を t_l として，区間 $[0, t_l)$ の一様乱数 t_r を生成し，時刻 t_r にバス停に到着したときの待ち時間が w_i である．
3. 各 $m = 1, \ldots, 10,000$ に対して，$w_1, \ldots, w_m$ から $\bar{w}_m = (w_1 + \cdots + w_m)/m$ を計算する．

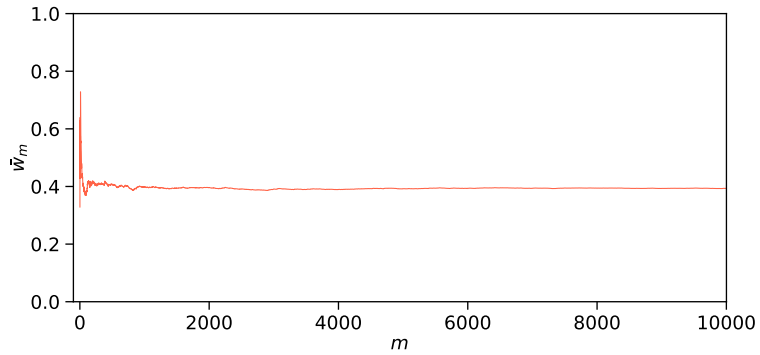

図 5: $\lambda = 2.5$ のときの平均待ち時間 $\bar{w}_m$ のプロット図．

レポート提出方法

以下を kibaco の課題ページから提出すること．

- 課題の内容を記載したレポート．
  - 課題 3-1, 3-2, 3-3, 3-4 の内容は必須である．各課題について，回答が正しくない場合でも，自力でコードの実装・数値結果の出力・結果の検証と考察がされていれば，部分点が付与される．なお，白紙や無回答の場合，その課題の得点は 0 点となる．
  - ファイル名は “apl_pp_xxx_yyy.pdf” とする．xxx には学修番号，yyy には氏名を入力する．例えば学修番号が 21012345 で氏名が Kazuki Nakajima の場合，“apl_pp_21012345_kazuki_nakajima.pdf” とする．漢字やひらがなをファイル名に含めないこと．
- 各課題を解くために使用したソースコード．
  - ソースコードが複数に分かれている場合は，zip ファイルにまとめて提出すること．
  - 各ソースコードが，誰のもので，どの課題に対応しているかわかるようにファイル名を設定すること．例えば，氏名が Kazuki Nakajima で課題 2-1 に対する C++ コードのファイル名は “kadai_2_1_kazuki_nakajima.cpp” とする．

注意事項

- **本課題の提出締め切りは 1 週間後（2024/11/13）の 12:00 である．**
- レポート作成時には，テクニカルライティングの資料を確認し，全体をよく検証してから提出すること．