Zadanie 2

Procesem Wienera (ruchem Browna) nazywamy proces stochastyczny $B = (B_t)_{t\geq 0}$, taki że:

- $B_0 = 0$.
- B ma niezależne przyrosty.
- $B_t - B_s \sim N(0,\, t-s)$, dla $0 \leq s < t$.
- B ma trajektorie ciągłe z prawdopodobieństwem 1.

W zadaniu będziemy posługiwać się jego zmodyfikowaną wersją:

$$B_t^x = B_t^0 + x,$$

Gdzie B_t^0 to klasyczny ruch Browna. Oznacza to tyle, że proces B_t^x rozpoczyna się w punkcie x. Chcąc uzyskać jego numeryczne przybliżenie przy pomocy symulacji komputerowej, wykorzystamy następujący algorytm:

Algorytm(0)

1. $B_0^x = x$
2. $B_{i+1}^x = B_i^x + dt^{\frac{1}{2}}\psi_i$

gdzie $\psi_i \sim N(0,1)$, $\quad i.i.d$, natomiast dt to najmniejszy krok czasowy.

Będziemy badać średni czas wyjścia z przedziału $[a,b]$, dla $x \in [a,b]$. Oznaczmy go jako

$$\mathbb{E}\tau_x, \text{ gdzie } \tau_x := \inf\{t \geq 0 : B_t^x \notin (a,b)\}.$$

Ponieważ na komputerze nie jest możliwe wykonanie symulacji procesu Wienera w czasie ciągłym, ustalamy najmniejszy krok czasowy dt, potrzebny do zastosowania metody numerycznej. Aby móc jak najlepiej oszacować średni czas wyjścia, powtarzamy n razy symulację dla danego x.

Algorytm(1):

1. Ustal punkt startowy x, najmniejszy krok czasowy dt, liczbę powtórzeń symulacji n oraz granice przedziału a i b.
2. Jeżeli $x = a$ lub $x = b$ zwróć 0.
3. Ustal $T = 0$
4. Generuj trajektorię procesu Wienera B_t^x dopóki $B_t^x \leq a$ lub $B_t^x \geq b$.
5. Zwiększ T o ilość kroków czsowych dt, które minęły do tego momentu.
6. Powtórz kroki 4-5 n razy.
7. Zwróć $\frac{T}{n}$.

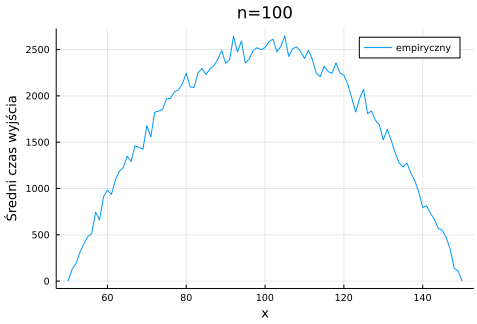
Chcąc zbadać zależność średniego czasu wyjścia od odległości punktu startowego x do granic przedziału $[a,b]$ musimy ustalić dx, czyli różnicę między kolejnymi początkowymi $x \in [a,b]$.

Algorytm(2):

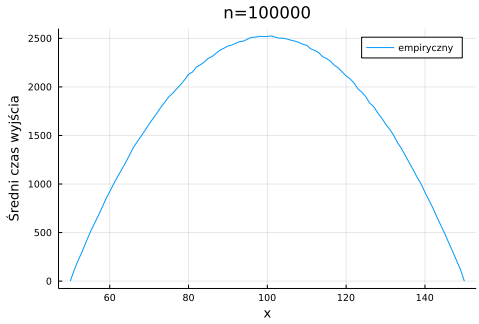
1. Ustal różnicę dx, najmniejszy krok czasowy dt, liczbę powtórzeń symulacji n, oraz granice przedziału a i b.

2. Stwórz pusty wektor y.
3. Stwórz wektor v, początkowych x-ów należących do przedziału $[a,b]$, tak aby $v[i+1] - v[i] = dx$.
4. Dla każdego elementu z wektora v oblicz średni czas wyjścia (Algorytm(1)) i wstaw otrzymany wynik do wektora y.
5. Zwróć y

Mając powyższe informacje, możemy przeprowadzić symulację. Rozważmy przypadki dla $n = 10^3$ i $n = 10^5$, $dt = 0.1$, $dx = 1$, $a = 50$, $b = 150$. Otrzymujemy wyniki widoczne na rys. 1 i rys. 2



Rysunek 1



Rysunek 2

Spróbujemy aproksymować funkcję, która opisuje otrzymane wyniki. Po kształcie wykresu przypuszczamy, że może to być wielomian co najmniej stopnia II na przedziale $[a,b]$. Wiemy również, że jego zera znajdują się w punktach a oraz b, ponieważ czas wyjścia w przypadku gdy $x = a$ lub $x = b$, będzie wynosił 0. Załóżmy więc, że jest to funkcja kwadratowa, po zapisaniu w postaci kwadratowej wygląda następująco:

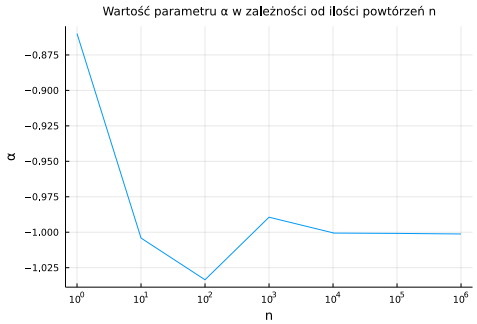
$$f(x) = \alpha(x - a)(x - b),$$

gdzie α jest szukanym współczynnikiem. W rozważanym przez nas przypadku:

$$f(x) = \alpha(x - 50)(x - 150).$$

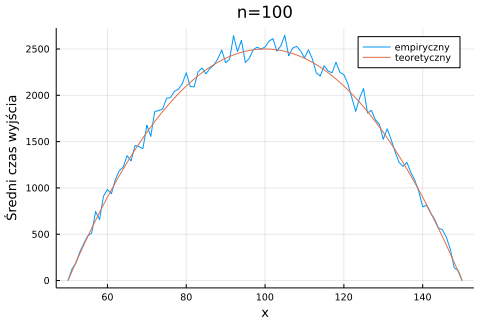
Aby znaleźć współczynnik α użyjemy modułu *numpy.polyfit*. Dla przypadku z $n = 10^3$ otrzymujemy $\alpha = 1.012$, natomiast dla $n = 10^5$, $\alpha = -1.001$. Na wykresie 3 widzimy jak współczynnik α zmienia się wraz ze wzrostem n. Możemy przypuścić, że $\lim_{n\to\infty} c = -1$. Podobnie średnia ze stu powtórzeń algorytmu(2) dla $n = 1000$ wraz z aproksymacją *numpy.polyfit* wskazała, że $\alpha = 1$. Ostatecznie otrzymujemy:

$$f(x) = -(x - 50)(x - 150).$$

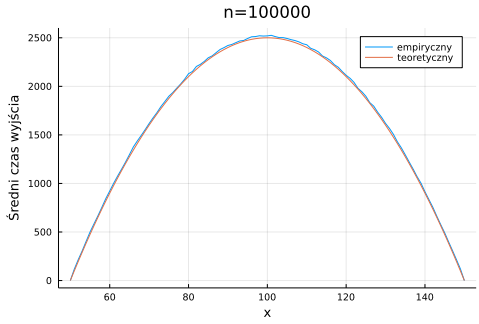


Rysunek 3

Zestawienie funkcji empirycznej z teoretyczną znajduje się na rys. 4 oraz rys. 5.



Rysunek 4



Rysunek 5

Przejdziemy teraz do szacowania, że wyjście nastąpiło przez granicę b, czyli:

$$P(B^x_{\tau^x} = b).$$

Jako, że symulacja działa w czasie dyskretnym, do określenia prawdopodobieństwa użyjemy wzoru:

$$P(B^x_{\tau^x} \geq b).$$

Postępujemy podobnie jak w przypadku szukania średniego czasu wyjścia.

Algorytm(3):

1. Ustal punkt startowy x, najmniejszy krok czasowy dt, liczbę powtórzeń symulacji n oraz granice przedziału a i b.
2. Jeżeli $x = a$ zwróć 0.
3. Jeżeli $x = b$ zwróć 1.
4. Ustal $k = 0$
5. Generuj trajektorię procesu Wienera B^x_t dopóki $B^x_t \leq a$ lub $B^x_t \geq b$.
6. Jeśli $w \geq b$ zwiększ k o jeden.
7. Powtórz kroki 5-6 n razy.
8. Zwróć $\frac{k}{n}$.

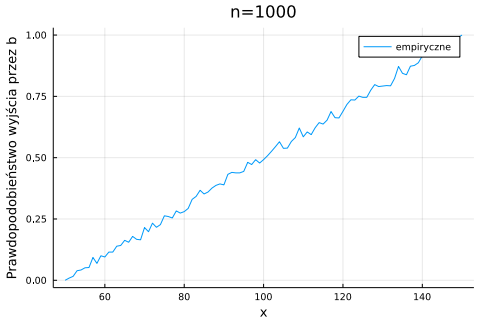
Analogicznie tworzymy algorytm, mający na celu zbadanie zależności między prawdopodobieństwem wyjścia przez b a początkowym $x \in [a,b]$.

Algorytm(4):

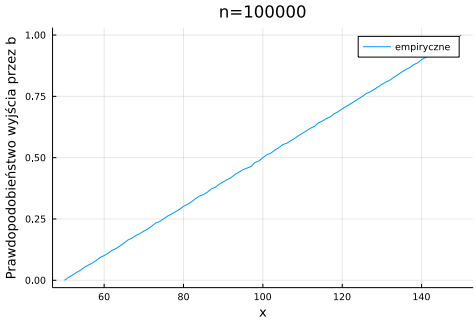
1. Ustal różnicę dx, najmniejszy krok czasowy dt, liczbę powtórzeń symulacji n, oraz granice przedziału a i b.
2. Stwórz pusty wektor y.
3. Stwórz wektor v, początkowych x-ów należących do przedziału $[a,b]$, tak aby $v[i+1] - v[i] = dx$.
4. Dla każdego elementu z wektora v oblicz prawdopodobieństwo wyjścia przez b (Algorytm(3)) i wstaw otrzymany wynik do wektora y.

5. Zwróć y

Korzystając z algorytmu 3 i 4, przeprowadzimy symulację dla $n = 10^3$ i $n = 10^5$, $dt = 0.1$, $dx = 1$, $a = 50$, $b = 150$. Otrzymujemy wyniki widoczne na rys. 6 i rys. 7



Rysunek 6



Rysunek 7

Po kształcie wykresu spodziewamy się, że jest to funkcja liniowa. Kiedy początkowe $x = a$, prawdopodobieństwo wyjścia przez b wynosi 0, natomiast gdy początkowe $x = b$, prawdopodobieństwo to będzie wynosić 1. Znając dwa punkty przez które przechodzi funkcja liniowa $f(x) = \beta x + \gamma$, gdzie β i γ są szukanymi parametrami, możemy wyprowadzić ogólny wzór rozwiązując następujący układ równań:

$$\begin{cases} \beta a + \gamma = 0 \\ \beta b + \gamma = 1 \end{cases} \Leftrightarrow \begin{cases} \beta = \frac{-1}{a-b} \\ \gamma = \frac{a}{a-b} \end{cases}$$

Zatem:

$$f(x) = -\frac{1}{a-b}x + \frac{a}{a-b}$$

Podstawiając nasze dane ($a = 50$, $b = 150$), otrzymujemy:

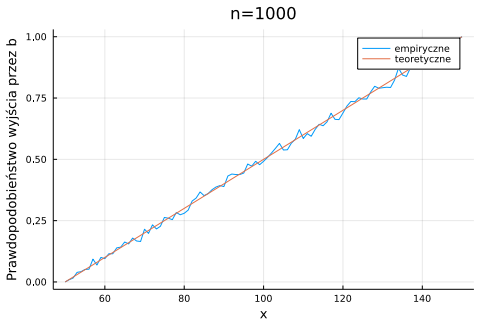
$$f(x) = \frac{1}{100}x - \frac{1}{2}$$

Sprawdzamy, czy moduł *numpy.polyfit* da nam pdobny rezultat, przy jego pomocy otrzymujemy:

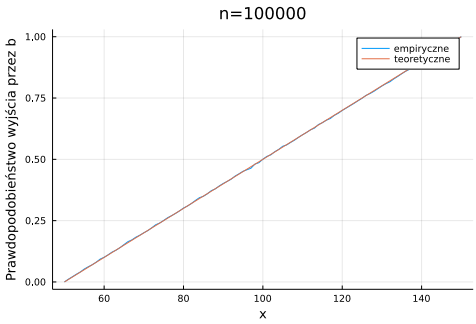
$$f(x) = 0.00997x - 0.497 \quad \text{dla} \quad n = 10^3$$

$$f(x) = 0.01x - 0.5 \quad \text{dla} \quad n = 10^5$$

Co zgadza się z naszymi teoretycznymi obliczeniami (im większe n tym większa dokładność). Na wykresach zestawienie empirycznej i teoretycznej funkcji prezentuje się następująco (rys. 8 i rys. 9):



Rysunek 8



Rysunek 9

Podsumowanie

W zadaniu szukaliśmy zależności, między średnim czasem wyjścia z przedziału [a,b] oraz prawdopodobieństwem, że wyjście to nastąpi przez b, a wartością początkową x. Musimy jednak pamiętać, że przeprowadzając symulację metodą numeryczną, dostaniemy jedynie przybliżone zachowanie procesu Wienera. Aby uzyskać rzeczywisty efekt, nasz minimalny krok czasowy dt musiałby dążyć do zera. Żeby wyniki były jak najbardziej rzetelne, musi zostać spełniony następujący warunek:

$$dt << |a - b|.$$

Należy jednak pamiętać, że dla ustalonego dt, zwiększając różnicę $|a - b|$, zwiększa się również czas wykonywania symulacji, dlatego mając ograniczone warunki, nie powinna być ona zbyt duża. Wykonując więcej powtórzeń (n), również poprawiamy dokładność otrzymanych wyników, jednak tutaj podobnie dobrze jest zachować umiar.

Obserwując średni czas wyjścia, w zależności od początkowego x, widzimy że rośnie on wraz z odległością do bliższej granicy a lub b. Będzie on najdłuższy, kiedy $|x - a| = |x - b|$, czyli dla $x = \frac{a+b}{2}$. Wtedy też prawdopodobieństwo wyjścia przez granicę b:

$$\beta = P(B^x_{\tau^x} = b),$$

jest równe prawdopodobieństwu wyjścia przez granicę a:

$$\alpha = 1 - \beta = P(B^x_{\tau^x} = a),$$

$$\alpha = \beta = \frac{1}{2}.$$

Funkcja autokowariancji

$$\begin{aligned} Cov(B_t,B_s) &= \mathbb{E}(B_s B_t) - \mathbb{E}B_s \mathbb{E}B_t = \\ &= \mathbb{E}(B_s(B_s B_t)) + \mathbb{E}(B_s^2) \\ &= \mathbb{E}B_s \mathbb{E}(B_s B_t) + \mathbb{E}B_s^2 = s \end{aligned}$$

Ogólnie

$$Cov(B_t,B_s) = \min\{s,t\}$$