

Symulacja procesów rzadkich przy pomocy równania typu Master, algorytm Gillespie

Metody Monte Carlo w fizyce
Nanoinżynieria materiałów

Łukasz Ruba



Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie

18 kwietnia 2024

Spis treści

| | | |
|---|-------------------|---|
| | Cel ćwiczenia | 3 |
| 1 | Wstęp teoretyczny | 3 |
| 2 | Metodyka | 4 |
| 3 | Wyniki | 5 |
| 4 | Wnioski | 7 |
| | Literatura | 8 |

Cel ćwiczenia

Celem ćwiczenia było obliczenie zawartości składników reakcji chemicznej o dobrze zdefiniowanych parametrach kinetycznych przy użyciu algorytmu Gillespie.

1 Wstęp teoretyczny

W poniższym sprawozdaniu rozpatrywano układ do którego dodawano ze stałą szybkością k_1 substrat x_1 oraz z szybkością k_2 substrat x_2 . Chcąc, aby odwzorowywało to układy chemiczne założono, że cząsteczki x_1 oraz x_2 mogą wejść ze sobą w reakcję tworząc składnik x_3 z szybkością k_3 [1]:



Składnik x_3 usuwany jest z szybkością zależną od ilości x_3 skalowanej pewną stałą k_4 . Przy pomocy układu równań różniczkowych można zapisać dynamikę zachodzących procesów:

$$\begin{cases} x_1 + x_2 \xrightarrow{k_3} x_3 \\ x_3 \xrightarrow{k_4} 0 \end{cases} \Rightarrow \frac{dx_3}{dt} = k_3 x_1 x_2 - k_4 x_3 \Rightarrow \frac{dx_3}{dt} = \Gamma_3(t) - \Gamma_4(t), \quad (2)$$

$$\begin{cases} x_1 \xrightarrow{\Gamma_3} 0 \\ 0 \xrightarrow{k_1} 1 \end{cases} \Rightarrow \frac{dx_1}{dt} = -k_3 x_1 x_2 + k_1 \Rightarrow \frac{dx_1}{dt} = -\Gamma_3(t) + \Gamma_1(t), \quad (3)$$

$$\begin{cases} x_2 \xrightarrow{\Gamma_3} 0 \\ 0 \xrightarrow{k_2} 1 \end{cases} \Rightarrow \frac{dx_2}{dt} = -k_3 x_1 x_2 + k_2 \Rightarrow \frac{dx_2}{dt} = -\Gamma_3(t) + \Gamma_2(t). \quad (4)$$

Można zauważyć, że każde równanie ([2, 3, 4]) ma postać typu Master. Częstość zmian danych substratów Γ_i zależy zatem od ich ilości oraz wartości szybkości k_i . Porównując środkową i prawą część układu równań można wyrazić częstości jako:

$$\Gamma_1(t) = k_1, \quad (5)$$

$$\Gamma_2(t) = k_2, \quad (6)$$

$$\Gamma_3(t) = k_3 x_1 x_2, \quad (7)$$

$$\Gamma_4(t) = k_4 x_3. \quad (8)$$

W algorytmie możemy zmiany te wyrazić w sposób dyskretny i losowy jako:

$$\Gamma_1 : \quad x_1 \rightarrow x_1 + 1, \quad (9)$$

$$\Gamma_2 : \quad x_2 \rightarrow x_2 + 1, \quad (10)$$

$$\Gamma_3 : \quad x_1 \rightarrow x_1 - 1, \quad x_2 \rightarrow x_2 - 1, \quad x_3 \rightarrow x_3 + 1, \quad (11)$$

$$\Gamma_4 : \quad x_3 \rightarrow x_3 - 1, \quad (12)$$

przy założeniu, że poszczególnych składników x_1 , x_2 i x_3 w układzie jest niewiele i opisywane są niewielkimi liczbami naturalnymi.

Zmiana stanu układu może wiązać się ze zmianą ilości pojedynczego składnika jak również kilku składników, w zależności od charakteru zdarzenia. Jest to zatem złożony proces stochastyczny, w którym fluktuacje mogą silnie wpłynąć na jego dynamikę.

W celu symulowania takich procesów metodą Monte Carlo można posłużyć się algorytmem Gillespie. Na wstępie należy założyć, że dynamika rozważanego procesu ma charakter losowy, a szybkości zachodzących zmian opisywane są za pomocą odpowiadających im częstości: $\{\Gamma_1, \Gamma_2, \dots, \Gamma_n\}$ - co de facto oznacza liczbę realizacji danego stanu na jednostkę czasu. Algorytm Gillespie jest iteracyjny i w każdej iteracji oblicza się:

1. sumę częstości wszystkich procesów:

$$\Gamma_{max} = \sum_{i=1}^n \Gamma_i, \quad (13)$$

2. losuje się przedział czasu Δt , w którym nie zachodzą zmiany w układzie:

$$U_1 \sim U(0, 1) \quad \rightarrow \quad \Delta t = -\frac{1}{\Gamma_{max}} \ln(U_1), \quad (14)$$

3. po czasie Δt następuje zmiana stanu układu, w sposób losowy określany jest numer zdarzenia m :

$$U_2 \sim U(0, 1) \quad \rightarrow \quad m = \min \left\{ s; \sum_{i=1}^s \frac{\Gamma_i}{\Gamma_{max}} > U_2, \quad s = 1, 2, 3, \dots, n \right\}, \quad (15)$$

4. na podstawie informacji o numerze zdarzenia, określa się jego rodzaj i dokonuje się zmiany stanu układu na podstawie równań [9, 10, 11, 12],

5. daną iterację kończy się zmieniając aktualny czas symulacji:

$$t + \Delta t \rightarrow t \quad (16)$$

6. symulację kończy się, gdy zachodzi warunek: $t > t_{max}$.

2 Metodyka

Opisany algorytm we wstępie teoretycznym (1) zaimplementowano w kodzie napisanym w języku *C++*.

Wykonano symulację dla parametrów: $k_1 = 1$, $k_2 = 1$, $k_3 = 0.001$, $k_4 = 0.01$ oraz wartości początkowych składników $x_1(t=0) = 120$, $x_2(t=0) = 80$, $x_3(t=0) = 1$. Czas t_{max} wynosił 200 sekund.

W trakcie trwania algorytmu zapisywano wartości każdego substratu w każdej chwili czasowej. Dodatkowo, chcąc jak najlepiej estymować wartość x_3 wykonywano kilka powtórzeń algorytmu, a następnie uśredniano jego wartości dla $N = 50$ binów czasowych, które obliczano jako:

$$\delta t = \frac{t_{max}}{N}. \quad (17)$$

Wartość średnią x_3 obliczono jako:

$$\langle x_3 \rangle = \frac{1}{P} \sum_i^P h_1(i), \quad (18)$$

gdzie P to liczba powtórzeń algorytmu, a ciąg h_1 to suma wartości x_3 w danym binie czasowym podzielona przez ilość zliczeń w tym binie.

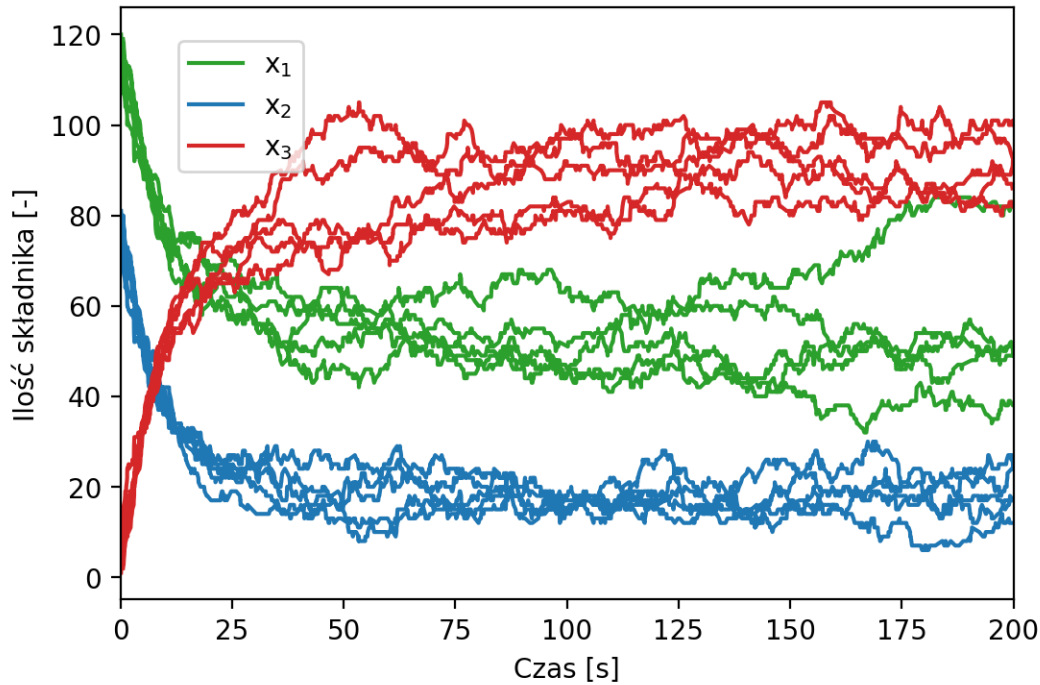
Ponadto, obliczono wartości odchylenia standardowego dla każdego punktu:

$$\sigma = \sqrt{\frac{\langle x_3^{(2)} \rangle - \langle x_3^{(1)} \rangle^2}{P}}. \quad (19)$$

Wykonano dwie symulacje, dla $P = 5$ oraz $P = 100$. Wszystkie grafiki wykonany przy pomocy biblioteki *matplotlib* w *Python*.

3 Wyniki

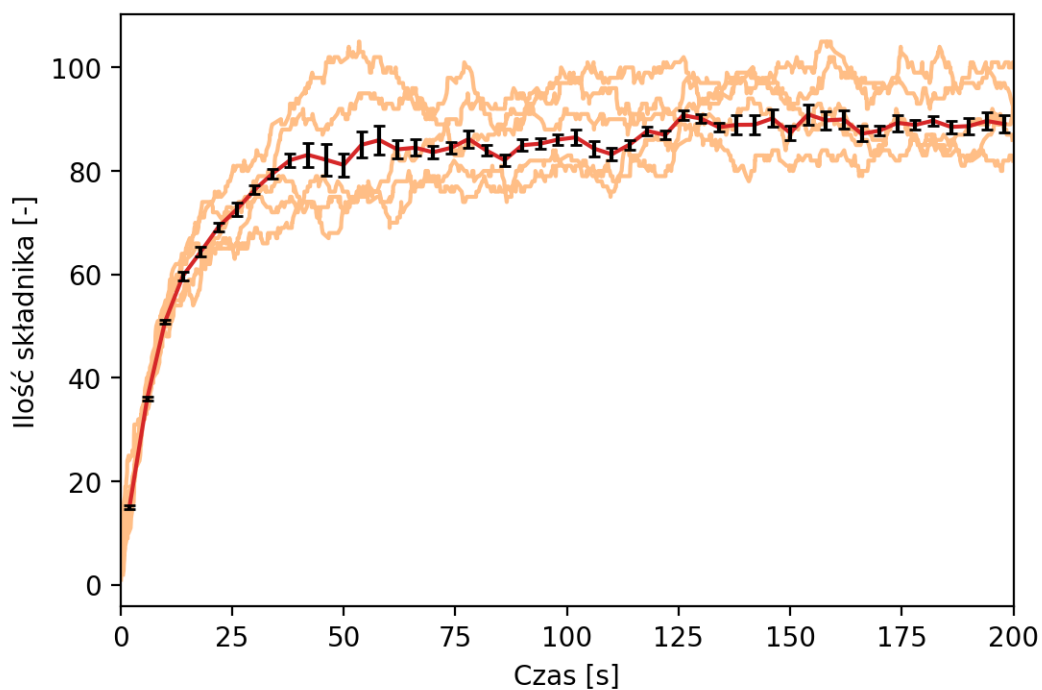
Na Rysunku 1 przedstawiono wynik symulacji z pięciu przebiegów algorytmu Gillespie. Każda symulacja startowała z tym samym zestawem parametrów wymienionych w sekcji 2. Ilości każdego składnika w danej chwili czasowej różniły się od siebie przy każdym podejściu, ale można zauważyć że trendy krzywych dla x_1 , x_2 i x_3 były takie same: zawartość x_1 oraz x_2 malała, zaś x_3 rosło przez około 50 sekund. Następnie wartości substratów fluktuowały wokół pewnej stałej wartości.



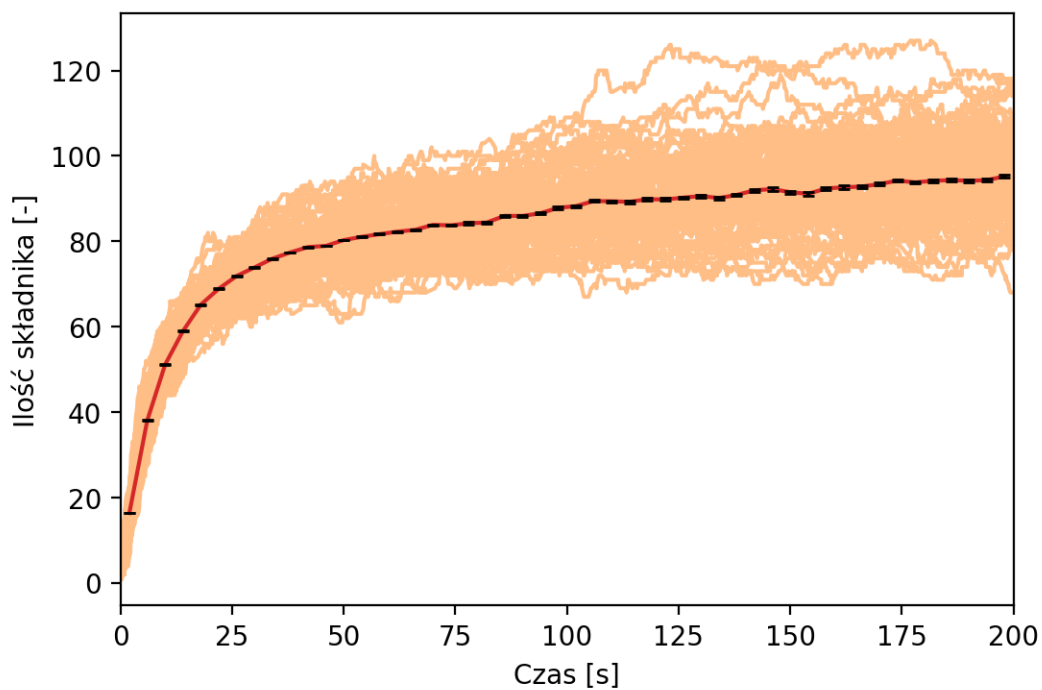
Rysunek 1: Ilość składników x_1 , x_2 i x_3 dla pięciu symulacji algorytmem Gillespie.

Na Rysunku 2 przedstawiono krzywą z uśredniania pięciu przebiegów zawartości składnika x_3 (wzór [18]) w układzie. Zgodnie ze wzorem [19] obliczono w każdym punkcie wartość odchylenia standardowego, które reprezentowane jest przez czarne słupki. Już pięć symulacji pozwoliło znacznie wygładzić linię x_3 , która nie fluktuuje tak mocno jak pojedyncze przebiegi.

W kolejnym kroku dokonano uśrednienia zawartości składnika x_3 dla 100 symulacji. Wyniki zaprezentowano na Rysunku 3. Krzywa jest bardzo gładka, a odchylenia standardowe bardzo małe. W celu sprawdzenia poprawności obliczania odchylenia standardowego całą operację przeprowadzono kilkakrotnie dla różnych ziaren generatora liczb pseudolosowych podstawowego C++ oraz generatora Mersenne Twister i w każdym przypadku otrzymywano małe wartości σ_{x_3} . Wyniki z których utworzono wykres z Rys. 3 przedstawiono w Tabeli 1.



Rysunek 2: Uśredniona wartość składnika x_3 (linia czerwona) z pięciu symulacji wraz z odchyleniami standardowymi (czarne słupki) pokazana na tle przebiegów wszystkich x_3 wziętych do uśrednienia.



Rysunek 3: Uśredniona wartość składnika x_3 (linia czerwona) ze 100 symulacji wraz z odchyleniami standardowymi (czarne słupki) pokazana na tle przebiegów wszystkich x_3 wziętych do uśrednienia.

Tabela 1: Tabela wartości x_3 uśrednionego dla 100 losowań oraz odchyłeń standardowych w każdej chwili czasowej.

| Czas [s] | Zawartość x_3 [-] | Std. [-] | Czas [s] | Zawartość x_3 [-] | Std. [-] |
|----------|---------------------|-----------|----------|---------------------|-----------|
| 2 | 16.7158 | 0.031174 | 102 | 88.148 | 0.209444 |
| 6 | 39.019 | 0.0727149 | 106 | 89.4146 | 0.224197 |
| 10 | 51.4379 | 0.0773506 | 110 | 89.9679 | 0.222085 |
| 14 | 58.7597 | 0.0615479 | 114 | 90.2112 | 0.191932 |
| 18 | 63.5217 | 0.0946704 | 118 | 90.9379 | 0.224614 |
| 22 | 67.6065 | 0.117347 | 122 | 90.916 | 0.166966 |
| 26 | 71.2707 | 0.0958277 | 126 | 91.4566 | 0.159104 |
| 30 | 73.5246 | 0.0868577 | 130 | 91.3202 | 0.207936 |
| 34 | 75.693 | 0.0745055 | 134 | 92.0921 | 0.201651 |
| 38 | 77.0413 | 0.058354 | 138 | 92.3858 | 0.204541 |
| 42 | 78.5679 | 0.0746632 | 142 | 92.694 | 0.16447 |
| 46 | 79.3556 | 0.0933797 | 146 | 92.1401 | 0.152932 |
| 50 | 80.1372 | 0.135418 | 150 | 92.9247 | 0.181775 |
| 54 | 81.313 | 0.172333 | 154 | 92.5397 | 0.236619 |
| 58 | 81.7229 | 0.175939 | 158 | 92.8585 | 0.162129 |
| 62 | 82.2051 | 0.193689 | 162 | 92.4932 | 0.243421 |
| 66 | 83.3433 | 0.160374 | 166 | 93.321 | 0.268438 |
| 70 | 84.2558 | 0.147286 | 170 | 93.2377 | 0.25827 |
| 74 | 84.5209 | 0.173035 | 174 | 94.0531 | 0.177583 |
| 78 | 84.8301 | 0.151128 | 178 | 94.165 | 0.171634 |
| 82 | 85.2806 | 0.144138 | 182 | 95.5473 | 0.168392 |
| 86 | 85.2306 | 0.179809 | 186 | 95.3483 | 0.139026 |
| 90 | 86.0776 | 0.184416 | 190 | 94.7424 | 0.110361 |
| 94 | 86.8297 | 0.170851 | 194 | 94.8728 | 0.0862639 |
| 98 | 87.2731 | 0.191569 | 198 | 95.8079 | 0.129306 |

4 Wnioski

Procesy stochastyczne mogą być przydatne nawet w takich dziedzinach nauki jak badanie kinetyki reakcji chemicznych, czego przykładem jest zaprezentowany algorytm Gillespie. Na jego podstawie udało się wysymulować zawartość substratów x_1 , x_2 i x_3 .

Pokazano, że wielokrotne powtarzanie symulacji pozwala na odsumowanie wyników krzywych reprezentujących zawartości składników w układzie poprzez pozbycie się fluktuacji, będących efektem stochastycznej natury algorytmu. Ponadto, wartości odchylenia standardowego znacząco spadają wraz ze wzrostem uśrednień, co jest zgodne z przypuszczeniami teoretycznymi.

Literatura

- [1] Tomasz Chwiej, dr hab. inż., skrypt do zadania “*Monte Carlo: symulacja procesów rzadkich przy pomocy równania typu Master, algorytm Gillespie*”, AGH, Kraków, 12 kwietnia 2024.