



Technische Universiteit Delft  
Faculteit Elektrotechniek, Wiskunde en Informatica  
Delft Institute of Applied Mathematics

**Een maat voor de vorm van een vervuilingspluim in  
de aardbodem**

**(Engelse titel: An indicator for the shape of a  
contamination plume in the earth's subsurface)**

Verslag ten behoeve van het  
Delft Institute for Applied Mathematics  
als onderdeel ter verkrijging

van de graad van

**BACHELOR OF SCIENCE  
in  
TECHNISCHE WISKUNDE**

door

**Annegreet van Opbroek**

**Delft, Nederland  
Mei 2009**



**BSc verslag TECHNISCHE WISKUNDE**

**“Een maat voor de vorm van een vervuilingspluim in  
de aardbodem”**

**(Engelse titel: “An indicator for the shape of a  
contamination plume in the earth’s subsurface”)**

Annegreet van Opbroek

**Technische Universiteit Delft**

**Begeleider**

Prof.dr. F.M. Dekking

**Overige commissieleden**

Prof.dr.ir. A.W. Heemink

Dr. J.G. Spandaw of Drs. A. Verweij

Mei, 2009

Delft



# Inhoudsopgave

|           |                                                                                                                                     |           |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>1</b>  | <b>Voorwoord</b>                                                                                                                    | <b>7</b>  |
| <b>2</b>  | <b>Inleiding</b>                                                                                                                    | <b>9</b>  |
| <b>3</b>  | <b>Advectie, dispersie en adsorptie</b>                                                                                             | <b>11</b> |
| 3.1       | Bewegingen van vervuilingsdeeltjes . . . . .                                                                                        | 11        |
| 3.2       | Deeltjes en hun toestanden . . . . .                                                                                                | 11        |
| 3.3       | Een eerste simulatie . . . . .                                                                                                      | 12        |
| <b>4</b>  | <b>Twee modellen voor de verspreiding</b>                                                                                           | <b>17</b> |
| 4.1       | Model I: Variatie in stapgrootte . . . . .                                                                                          | 17        |
| 4.2       | Model II: Variatie in kansen . . . . .                                                                                              | 18        |
| 4.3       | Simulaties met $\gamma = \sqrt{10}$ en $p_f(0) = \frac{a}{a+b}$ . . . . .                                                           | 19        |
| <b>5</b>  | <b>Een maat voor de verspreiding van vervuilingsdeeltjes</b>                                                                        | <b>23</b> |
| 5.1       | Variantie van de pluim . . . . .                                                                                                    | 23        |
| 5.2       | Bepalen van $\text{Var}(\mathbf{S}_Y(\mathbf{n}) \mathbf{S}_X(\mathbf{n}) = \mathbf{x})$ . . . . .                                  | 24        |
| 5.2.1     | Bepalen van $P(\mathbf{S}_Y(\mathbf{n}) = \mathbf{y}, \mathbf{S}_X(\mathbf{n}) = \mathbf{x}   \mathbf{K}_n = \mathbf{k})$ . . . . . | 25        |
| 5.2.2     | Bepalen van $P(\mathbf{K}_n = \mathbf{k})$ . . . . .                                                                                | 27        |
| 5.2.3     | Bepalen van $P(\mathbf{S}_X(\mathbf{n}) = \mathbf{x}   \mathbf{K}_n = \mathbf{k})$ . . . . .                                        | 27        |
| 5.2.4     | Bepalen van grenzen . . . . .                                                                                                       | 28        |
| 5.3       | Uitkomsten met behulp van Matlab . . . . .                                                                                          | 29        |
| 5.4       | Bepalen van $P(\mathbf{K}_n = \mathbf{k})$ voor $a + b \neq 1$ . . . . .                                                            | 32        |
| <b>6</b>  | <b>De symmetrie in de variantie rondom <math>n</math></b>                                                                           | <b>37</b> |
| 6.1       | Reductie tot symmetrie in de kansen . . . . .                                                                                       | 37        |
| 6.2       | Bewijs van de symmetrie . . . . .                                                                                                   | 38        |
| <b>7</b>  | <b>De empirische variantie</b>                                                                                                      | <b>41</b> |
| 7.1       | Simple sampling . . . . .                                                                                                           | 41        |
| 7.2       | Importance sampling . . . . .                                                                                                       | 43        |
| 7.2.1     | Hoe importance sampling werkt . . . . .                                                                                             | 43        |
| 7.2.2     | Toepassen van importance sampling op de empirische variantie . . . . .                                                              | 43        |
| 7.2.3     | De variantie van importance sampling voor een<br>binomiale verdeling . . . . .                                                      | 46        |
| 7.3       | Verschillende simulaties . . . . .                                                                                                  | 47        |
| <b>8</b>  | <b>Een andere maat voor de verspreiding van deeltjes</b>                                                                            | <b>53</b> |
| 8.1       | Vermenigvuldiging met $P(\mathbf{S}_X(\mathbf{n}) = \mathbf{x})$ . . . . .                                                          | 53        |
| <b>9</b>  | <b>Samenvatting</b>                                                                                                                 | <b>57</b> |
| <b>10</b> | <b>Conclusie en aanbevelingen</b>                                                                                                   | <b>59</b> |



# Hoofdstuk 1

## Voorwoord

Voor u ligt het verslag van mijn Bachelorproject over de tweedimensionale verspreiding van een vervuilingspluim in de bodem. Mijn verslag is voortgekomen uit [3], het artikel van onder andere mijn begeleider. In dit artikel wordt een vervuilingspluim in de bodem beschreven door een groot aantal vervuilingsdeeltjes die een stochastische wandeling uitvoeren. Mijn verslag gaat verder in op de precieze vorm van de uiteindelijke vervuilingspluim.

Bij deze wil ik mijn begeleider, Prof.dr.F.M. Dekking bedanken voor zijn hulp, inzichten en kritische woorden.



## Hoofdstuk 2

# Inleiding

Lang werd gedacht dat vervuilingsdeeltjes die in de bodem terecht komen, na verloop van tijd een Gaussisch patroon gaan vertonen. Dit als gevolg van de grondwaterstroom en dispersie. Het proces dat een vervuilingswolk zou ondergaan, werd gezien als het uit elkaar gaan van de deeltjes, met een translatie veroorzaakt door de grondwaterstroom. Uit onder andere het model in [3] blijkt echter dat dit in het algemeen niet geldt. Een vervuilingswolk blijkt eerder de vorm aan te nemen van een komeet, met een platte kop en een lange staart, dan van een Gaussische kromme. In dit verslag gaan we verder in op de oorsprong van dit fenomeen, dat we zullen onderbouwen met Matlab simulaties. Hierbij beschouwen we de vervuiling als een groot aantal losse vervuilingsdeeltjes die onafhankelijk van elkaar met een bepaalde kans een beweging uitvoeren.

Allereerst zullen we kijken naar wat er allemaal met een vervuilingsdeeltje kan gebeuren en welke situaties we kunnen onderscheiden. Gedurende het verslag zal opvallen dat een model vaak op verschillende manieren bekeken kan worden en dat sommige manieren voor bepaalde analyses handiger zijn dan andere.

Vervolgens zullen we naar de kern gaan van dit verslag, waarin we een maat proberen te vinden die onze waarnemingen kwalitatief ondersteunt, dat de verdeling geen Gaussisch patroon zou betreffen. Ook hierbij is gebruikt gemaakt van Matlab om grote berekeningen uit te voeren. Hierbij zullen we onder andere moeten kijken naar Markov binomiale verdelingen, die duidelijk beschreven worden in [2] en [4].

Tenslotte zullen we de resultaten toetsen met behulp van uitkomsten van Matlab simulaties van de vervuilingspluimen. Hierbij zullen twee statistische methoden de revue passeren; simple sampling en importance sampling. We sluiten af met de conclusies en aanbevelingen.



## Hoofdstuk 3

# Advectie, dispersie en adsorptie

We zullen gaan kijken naar een stochastisch model voor de tweedimensionale verspreiding van vervuiling in de bodem, zoals besproken in [3]. We beginnen met een overzicht van dit stochastische model en de verschillende fenomenen waarmee vervuilingsdeeltjes in onze aardbodem te maken krijgen. Ook zullen we alvast een simulatie maken met behulp van Matlab, om wat inzicht te krijgen in de situatie.

### 3.1 Bewegingen van vervuilingsdeeltjes

Om het verspreiden van vervuiling in de bodem met een stochastisch model te kunnen beschrijven, zullen we kijken naar een groot aantal vervuilingsdeeltjes. Deze vervuilingsdeeltjes kunnen in verschillende toestanden verkeren, die we met behulp van stochastiek kunnen beschrijven. Deze verschillende toestanden zijn onder te brengen in “vrij” en “geadsorbeerd”. De deeltjes kunnen namelijk door de bodem gedurende een tijdje geadsorbeerd worden en later weer vrijgelaten worden. Daarbij hebben vrije deeltjes te maken met twee fenomenen. Ten eerste advectie, waarbij de deeltjes meestromen met de grondwaterstroom en ten tweede dispersie, waarbij de deeltjes zich willekeurig verspreiden in alle richtingen. We hebben dus drie verschillende bewegingen, maar hoe gaan we dit modelleren?

### 3.2 Deeltjes en hun toestanden

We zullen een vervuilingspluim modelleren met een groot aantal losse deeltjes,  $N$  die onafhankelijk van elkaar bewegen. Wat we willen is de vervuilingspluim volgen gedurende een tijdsinterval  $[0, t]$ , waarin  $t$  groot genoeg is om in een evenwichtstoestand terecht te komen. Daartoe discretiseren we de tijd met  $n$  tijdsintervallen met lengte  $\Delta t = \frac{t}{n}$ . Waarbij ieder deeltje aan het begin van een tijdstap met een bepaalde kans wisselt van de ene toestand naar de andere, maar tijdens een tijdstap niet wisselt.

Waar we uiteindelijk naar willen kijken zijn de x- en y-posities van de deeltjes na  $n$  tijdstappen,  $S_X(n)$  en  $S_Y(n)$ , hiertoe zullen we moeten kijken naar de verschillende toestanden waarin de deeltjes kunnen verkeren. Deeltjes kunnen gedurende een tijdstap vrij of geadsorbeerd zijn, we zullen dit noteren als  $f$  (‘free’) en  $a$  (‘adsorbed’). We gaan er vanuit dat een geadsorbeerd deeltje niet beweegt en dat vrije deeltjes te maken hebben met zowel advectie als dispersie. Om meer te weten te komen over het gedrag van een deeltje, is het nodig te kijken naar het aantal keren dat een deeltje vrij is geweest in  $n$  tijdstappen, dit is  $K_n$ . We kiezen ervoor om de verdeling van  $K_n$  te beschrijven met een Markov keten met twee toestanden,  $f$  en  $a$ . De overgangsmatrix ziet er als volgt uit:

$$\begin{bmatrix} p_{ff} & p_{fa} \\ p_{af} & p_{aa} \end{bmatrix} = \begin{bmatrix} 1-b & b \\ a & 1-a \end{bmatrix}. \quad (3.1)$$

De kans dat een geadsorbeerd deeltje op tijdstip  $i\Delta t$  ( $i \in \mathbb{N}$ ) wisselt naar vrij is dus  $a$ , en andersom is  $b$  de kans dat een vrij deeltje wisselt naar geadsorbeerd.

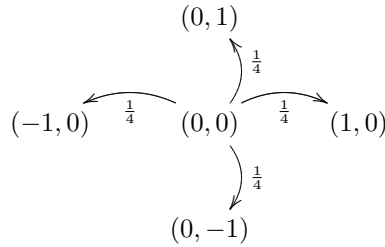
Een geadsorbeerd deeltje beweegt niet, maar wat doen vrije deeltjes? Deeltjes die vrij zijn stromen gedurende de tijdstap mee met het grondwater en daar bovenop voeren ze dispersie uit. Dit laatste betekent dat het deeltje een (kleine) random beweging maakt in het vlak, met gemiddelde 0 in zowel de x- als de y-richting. Een bekende beweging die hier aan voldoet is de stochastische wandeling ofwel random walk. We zullen de dispersie daarom hiermee beschrijven. Voor de x- en y-positie van een deeltje  $i$  dat  $k$  keer vrij is geweest,  $(X_i, Y_i)$  geldt dus dat  $E[X_i|K_n = k] = k$  en  $E[Y_i|K_n = k] = 0$ .

Voor het aantal keren dat een deeltje na  $n$  tijdstappen vrij is geweest, geldt dan  $E[K_n] \approx n \cdot \frac{a}{a+b}$ , immers per tijdstap is een gedeelte van  $\frac{a}{a+b}$  van de deeltjes vrij. Waarom is het dan ongeveer? Simpelweg omdat we rekening moeten houden met de beginverdeling van de deeltjes. Als aan het begin een deel  $\frac{a}{a+b}$  van de deeltjes vrij zouden zijn, zou  $E[K_n]$  precies gelijk zijn aan  $n \frac{a}{a+b}$ .

### 3.3 Een eerste simulatie

We weten nu genoeg van de vervuilingsdeeltjes om alvast een simulatie te doen in Matlab. We beginnen met  $N = 10\,000$  deeltjes en  $n = 100$  tijdstappen. Een vrij deeltje zal in één tijdstap een afstand van 1 meestromen met het grondwater, in de x-richting. Ons assenstelsel is dus zodanig dat de grondwaterstroom precies in de positieve x-richting loopt.

De random beweging die een vrij deeltje maakt door dispersie, modelleren we met behulp van de random walk. We gooien als het ware een muntje op, en laten het deeltje met kans  $p = \frac{1}{4}$  een stapje in de positieve danwel negatieve x-richting of y-richting maken.



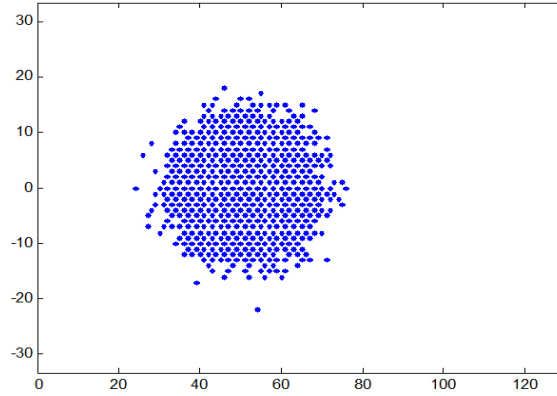
Figuur 3.1: De mogelijke verplaatsingen door dispersie.

Ook moeten we nog vaststellen in welke toestand de deeltjes starten, allemaal vrij, allemaal geadsorbeerd, of een mengsel. We kunnen de verschillende initiële condities gemakkelijk met elkaar vergelijken.

Om een vervuilingspluim vast te kunnen stellen, kunnen we nu het bovenstaande model in Matlab implementeren. Het handigste is hierbij om de posities van alle  $N$  deeltjes bij te houden in een vector van 2 bij  $N$  en per tijdstapje vast te stellen of het deeltje geadsorbeerd of vrij is en welke bewegingen het uitvoert. Hieronder staat een plaatje van het resultaat voor het geval dat  $a = b = 0.5$  en dat de kans dat een deeltje aan het begin vrij is gelijk aan 1,  $p_f(0) = 1$ .

De vervuilingspluim blijkt erg te lijken op een 2 dimensionale Gaussische verdeling, met gemiddelde

$$(E[S_X(n)], E[S_Y(n)]) \approx (\overline{S_X}(100), \overline{S_Y}(100)) \approx (50, 0).$$



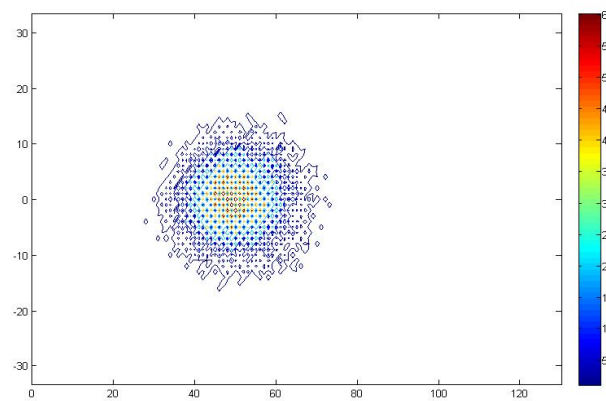
Figuur 3.2: Een vervuilingspluim met  $n = 100$ ,  $a = b = 0.5$  en  $p_f(0) = 1$ .

Hierbij gebruiken we dat

$$\mathbb{E} \left[ \sum_{k=1}^{K_n} X_k \right] = \mathbb{E}[K_n] \cdot \mathbb{E}[X_1] \approx n \frac{a}{a+b} \cdot 1 = 50$$

$$\mathbb{E} \left[ \sum_{k=1}^{K_n} Y_k \right] = \mathbb{E}[K_n] \cdot \mathbb{E}[Y_1] \approx n \cdot 0 = 0.$$

Figuur 3.2 geeft informatie over waar de pluim wel of niet zit, maar we willen ook iets weten over de concentratie van deeltjes in de bodem. Hiertoe kunnen we van alle  $N$  deeltjes hun positie nagaan en deze opslaan in een matrix waarin elk element een positie voorstelt. Matlab kan dan gemakkelijk zelf een contourplot maken. Zo krijgen we figuur 3.3.

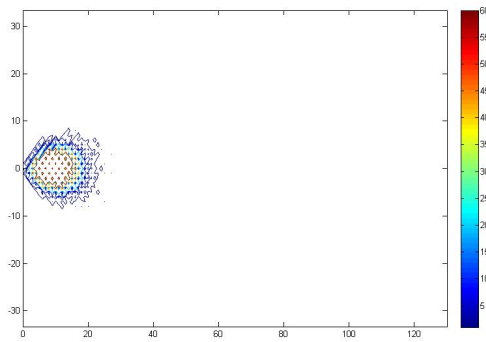


Figuur 3.3: Contourplot van de vervuilingspluim met  $n = 100$ ,  $a = b = 0.5$  en  $p_f(0) = 1$ .

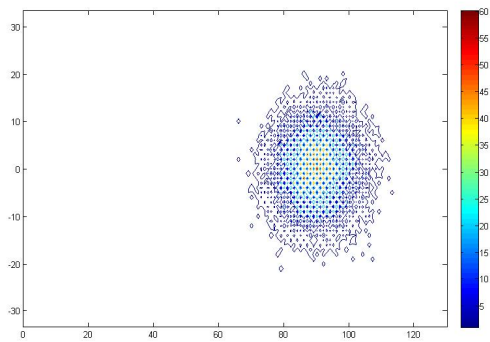
We zien nu dat de meeste deeltjes dicht rond het punt  $(50,0)$  zitten, zoals we ook zouden verwachten. Wat verder opvalt is dat de deeltjes op een grid lijken te zitten, dat komt gedeeltelijk doordat we ook op een grid rekenen, door de deeltjes steeds een stapje van lengte 1 te laten maken.

Maar bij nader bestuderen kunnen we ook vaststellen dat deeltjes alleen op punten  $(i, j)$  zitten met  $i$  even en  $j$  even, of beide oneven. Hoe kan dit? Stel dat een deeltje  $K_n = k$  keer vrij is geweest, dan bevindt het zich als we alleen de stroming in beschouwing nemen, op  $(k, 0)$ . Voor de dispersie zal het deeltje dan nog  $k$  stappen moeten doen volgens figuur 3.1. Stel het deeltje gaat in een dispersiestap van  $(x_{k-1}, y_{k-1})$  naar  $(x_k, y_k)$ , dan is  $|(x_k - x_{k-1}) + (y_k - y_{k-1})| = 1$  en dus  $\sum_{\ell=1}^k |(x_\ell - x_{\ell-1}) + (y_\ell - y_{\ell-1})| = k$ . Tesamen met de stroming zorgt dit ervoor dat het deeltje zich op een plaats zal bevinden dat in  $2k$  stappen vanuit de oorsprong te bereiken is, dus op een plek  $(i, j)$  met  $i$  en  $j$  beide even of beide oneven.

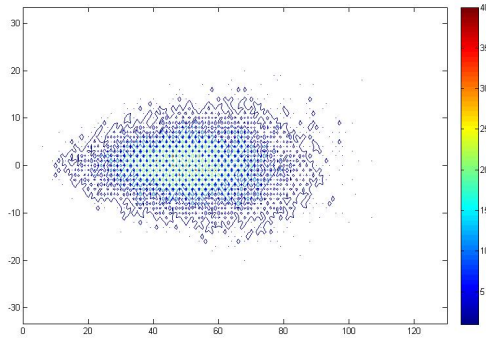
We zullen nu eens kijken wat er gebeurt als we  $a$ ,  $b$  en  $p_f(0)$  veranderen, aan de hand van wat contourplots.



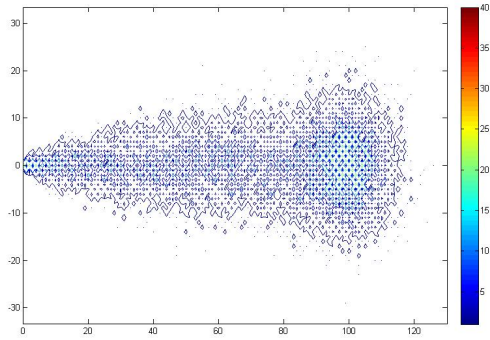
Figuur 3.4:  $a = 0.1$ ,  $b = 0.9$  en  $p_f(0) = 1$



Figuur 3.5:  $a = 0.9$ ,  $b = 0.1$  en  $p_f(0) = 1$



Figuur 3.6:  $a = 0.1$ ,  $b = 0.1$  en  $p_f(0) = 1$

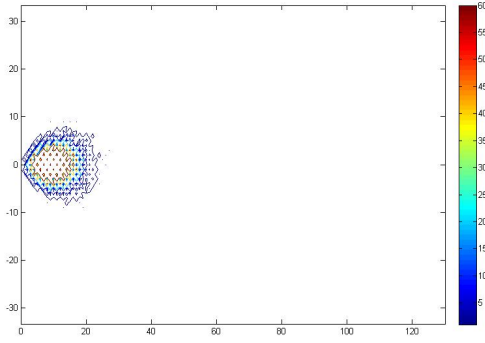


Figuur 3.7:  $a = 0.01$ ,  $b = 0.01$  en  $p_f(0) = 1$

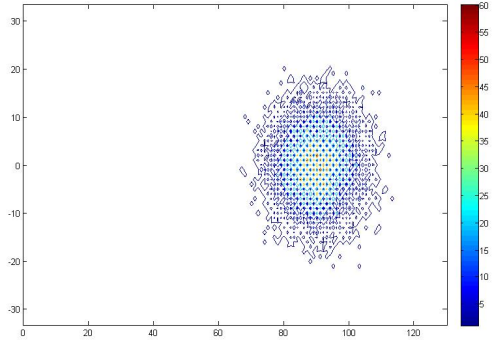
In de figuren 3.4 tot en met 3.7 is duidelijk te zien dat het veranderen van  $a$  en  $b$  veel effect heeft op de vervuilingswolk. Ten eerste natuurlijk in het punt waar omheen de vervuiling geconcentreerd is,  $(E[S_X], E[S_Y])$ . Omdat de deeltjes gemiddeld  $E[K_n] = n \frac{a}{a+b}$  keer vrij zijn (mits we het effect van de begintoestand mogen verwaarlozen), verwachten we door toedoen van de stroming, dat het centrum van de pluim rond  $x$ -positie  $\frac{na}{a+b}$  zit. Dit zien we mooi terug in figuren 3.4 en 3.5. Wat we hier ook zien is dat de verspreiding in zowel de  $x$ - als de  $y$ -richting veel kleiner is voor  $a = 0.1$  en  $b = 0.9$  dan voor  $a = 0.9$ ,  $b = 0.1$ . Ook dit is gemakkelijk te verklaren doordat deeltjes in het tweede geval veel meer vrij zijn geweest en dus ook meer stapjes dispersie hebben uitgevoerd. Hierdoor neemt de spreiding in alle richtingen toe.

Wat in figuren 3.6 en 3.7 gebeurt is echter veel interessanter. We zien hier duidelijk geen Gaussi-

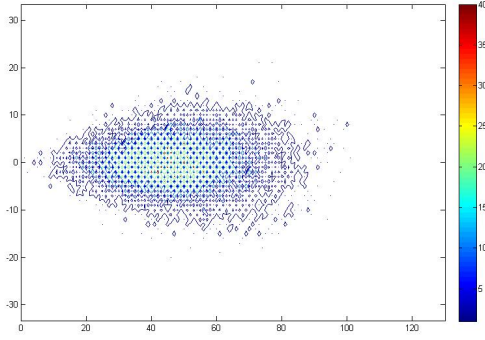
sche verdeling van de deeltjes rondom het punt  $(E[K_n], 0)$ , maar een soort druppelvorm met een hogere verspreiding in de  $y$ -richting voor hogere  $x$ -posities. Dit verschijnsel vormt de kern van artikel [3] en tevens van dit Bachelorproject. De verklaring voor deze druppelvorm is dat voorin de pluim vooral deeltjes zitten die een groot deel van de tijd vrij zijn geweest, terwijl achterin de pluim vooral deeltjes zijn gelocaliseerd die veel geadsorbeerd zijn. Deeltjes die veel vrij zijn geweest voeren meer dispersie uit en hebben dus een grotere spreiding hebben dan deeltjes die vooral geadsorbeerd zijn geweest. Hierdoor is de spreiding van de deeltjes voor hogere  $x$ -posities groter dan voor lagere  $x$ -posities.



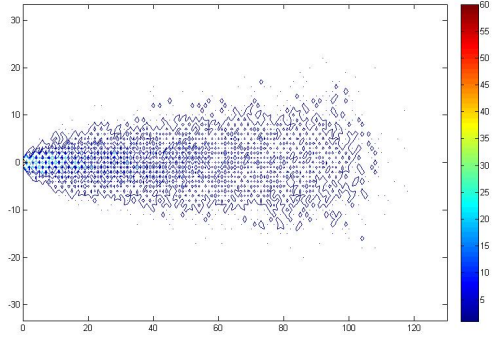
Figuur 3.8:  $a = 0.1$ ,  $b = 0.9$  en  $p_f(0) = 0$



Figuur 3.9:  $a = 0.9$ ,  $b = 0.1$  en  $p_f(0) = 0$



Figuur 3.10:  $a = 0.1$ ,  $b = 0.1$  en  $p_f(0) = 0$



Figuur 3.11:  $a = 0.01$ ,  $b = 0.01$  en  $p_f(0) = 0$

Tenslotte vergelijken we de pluimen waarin initi el alle deeltjes vrij zijn,  $p_f(0) = 1$  met de situatie waarin alle deeltjes in het begin geadsorbeerd zijn,  $p_f(0) = 0$ .

Voor de eerste twee gevallen zien we geen verschil in de uitkomsten. Voor  $a = 0.1$ ,  $b = 0.9$  in figuur 3.8 komt dit doordat voor het geval met  $p_f(0) = 1$  na  en tijdstap alweer 90% van de deeltjes geadsorbeerd is geworden, wat gelijk is aan de verdeling na  en tijdstap voor  $p_f(0) = 0$ ; de uitkomsten verschillen dus gemiddeld maar  en stapje. Voor  $a = 0.9$  en  $b = 0.1$  geldt precies hetzelfde, hier is in beide gevallen na  en tijdstapje 90% van de deeltjes vrij.

Voor  $a = b = 0.1$  in figuur 3.10 zien we alleen wat verandering in het aantal deeltjes dicht bij 0, maar voor  $a = b = 0.01$  zien we wel degelijk een groot verschil. In dit geval lijken de deeltjes voor  $p_f(0) = 1$  vooral geconcentreerd rond het punt  $(100, 0)$ , terwijl voor  $p_f(0) = 0$  de deeltjes meer in de buurt van  $(0, 0)$  lijken te blijven. Dit komt doordat in dit geval de kans dat een deeltje wisselt van vrij naar geadsorbeerd of omgekeerd, heel erg klein is. Figuur 3.7 ziet er dus uit als een wolk deeltjes die altijd vrij zijn, en elke stap wat deeltjes “verliest”, terwijl figuur 3.11 eruit ziet als een wolk deeltjes op  $(0, 0)$  die elke tijdstap wat deeltjes “los laat”.



## Hoofdstuk 4

# Twee modellen voor de verspreiding

We hebben nu een eerste model gemaakt om de vervuiling te simuleren, maar we kunnen nog een aanpassing maken waardoor het model de werkelijkheid nog wat beter beschrijft.

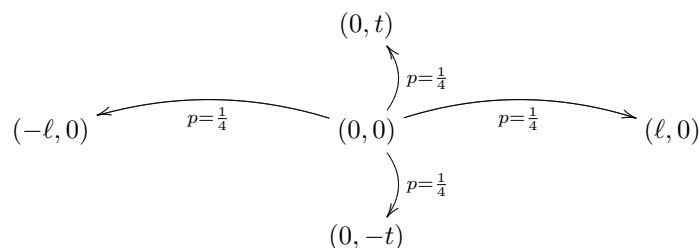
Hiertoe willen we dat de verspreiding in de x-richting een factor  $\gamma$  keer zo groot wordt als de verspreiding in de y-richting. De waarde  $\gamma = \sqrt{10}$  blijkt het meest overeen te komen met de werkelijkheid, hier zullen we dan ook vanuit gaan.

We merken op dat we dit in principe op twee verschillende manieren kunnen modelleren. De eerste is wellicht het meest voor de hand liggend, de kansen dat een deeltje een stapje in een van de vier richtingen doet, is even groot, maar de grootte van de stapjes zijn in verschillend in de x- en y-richting.

De tweede mogelijkheid is om de grootte van de stapjes gelijk te houden, maar de kansen aan te passen, zodat de varianties in de twee richtingen overeen komen met onze gestelde eis. Omdat deze methode wellicht iets minder voor de hand ligt, zullen we beginnen met de eerste methode.

### 4.1 Model I: Variatie in stapgrootte

We kijken naar een model, waarbij een deeltje dat op tijdstip  $t_n$  vrij is, met kansen  $p = \frac{1}{4}$  dispersie uitvoert in de positieve of negatieve x- of y-richting. We willen dat de variantie in de x-richting  $\sqrt{10}$  keer zo groot is als in de y-richting, ofwel  $\text{Var}(X_k) = 10 \cdot \text{Var}(Y_k)$ , aangezien de variantie kwadratisch afhangt van de afstand. Om dit te bereiken moeten een stapje in de x-richting dus  $\sqrt{10}$  keer zo groot maken als een stapje in de y-richting. Noemen we de longitudinale verplaatsing (in de x-richting) voor 1 dispersiestap even  $\ell$  en de transversale verplaatsing (in de y-richting)  $t$ , dan geldt dus dat  $\ell = \sqrt{10} \cdot t$ .

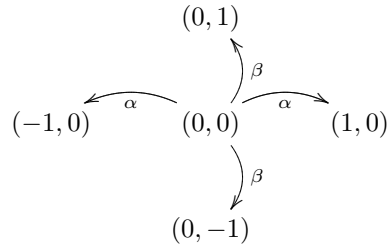


We kunnen nu voor dit model de varianties in de x- en y-richting opstellen;

$$\begin{aligned}\text{Var}(X_k) &= \text{E}[X_k^2] = \frac{1}{4}\ell^2 + \frac{1}{4}\ell^2 = \frac{1}{2}\ell^2 = 5t^2 \\ \text{Var}(Y_k) &= \text{E}[Y_k^2] = \frac{1}{4}t^2 + \frac{1}{4}t^2 = \frac{1}{2}t^2.\end{aligned}\tag{4.1}$$

## 4.2 Model II: Variatie in kansen

We zullen nu kijken naar een tweede model, waarin we de stapgrootte in de 4 richtingen gelijk houden, maar de kansen variëren. Dit model blijkt bij latere berekeningen handiger te zijn. We noemen in dit model de kans dat een deeltje dispersie uitvoert in de x-richting even  $\alpha$  en de kans voor de y-richting  $\beta$ . We hebben dan te maken met onderstaande situatie.



In dit model zijn de varianties in de x- en y-richting gelijk aan

$$\begin{aligned}\text{Var}(\tilde{X}_k) &= \text{E}[\tilde{X}_k^2] = \alpha \cdot 1 + \alpha \cdot 1 = 2\alpha \\ \text{Var}(\tilde{Y}_k) &= \text{E}[\tilde{Y}_k^2] = \beta \cdot 1 + \beta \cdot 1 = 2\beta.\end{aligned}\tag{4.2}$$

Om  $\alpha$  en  $\beta$  te bepalen, vergelijken we dit model met Model I, we willen namelijk dat de variantie in de x- en y-richting in Model II gelijk zijn aan die van Model 1, er moet dus er gelden dat

$$\text{Var}(\tilde{X}_k) = \text{Var}(X_k) \quad \text{en} \quad \text{Var}(\tilde{Y}_k) = \text{Var}(Y_k).\tag{4.3}$$

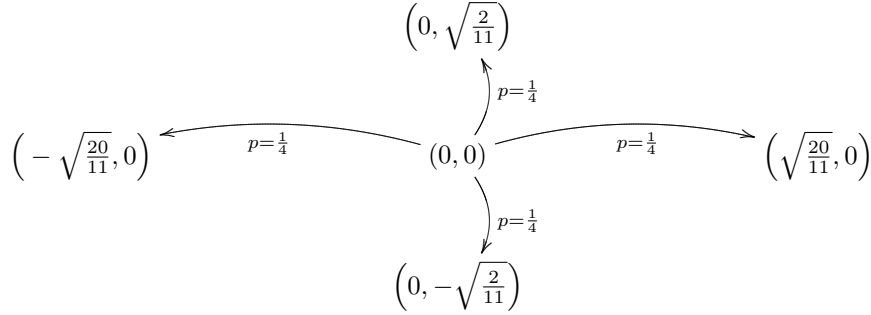
Substitutie van (4.1) en (4.2) in (4.3) geeft

$$\begin{aligned}2\alpha &= \frac{1}{2}\ell^2 = 5t^2 \\ 2\beta &= \frac{1}{2}t^2\end{aligned}$$

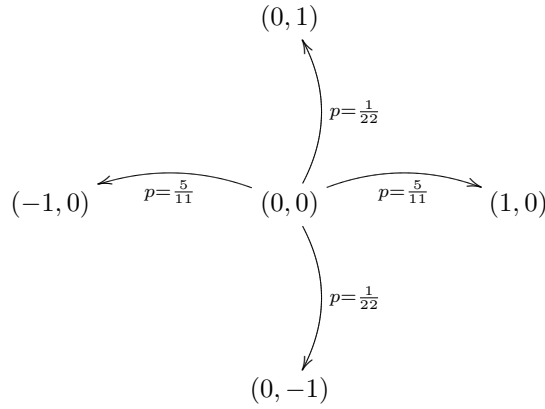
Tesamen met de voorwaarde dat  $2\alpha + 2\beta = 1$  volgt nu dat

$$\begin{aligned}2\alpha = 5t^2 = 20\beta &\Rightarrow \alpha = \frac{5}{11} \quad \text{en} \quad \beta = \frac{1}{22} \\ 5\frac{1}{2}t^2 = 1 &\Rightarrow t = \sqrt{\frac{2}{11}} \quad \text{en} \quad \ell = \sqrt{\frac{20}{11}}.\end{aligned}$$

Merk op dat nu beide modellen eenduidig vastliggen, we hebben dus de twee onderstaande modellen voor de dispersie.



Figuur 4.1: Model I: Variatie in de verplaatsing



Figuur 4.2: Model II: Variatie in de kansen

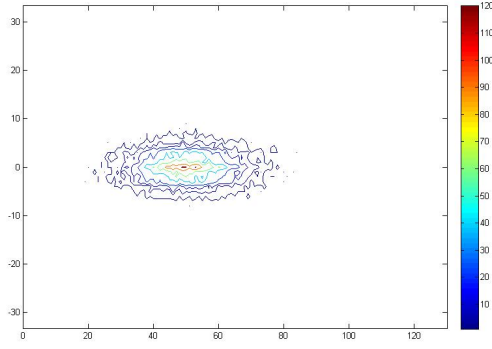
### 4.3 Simulaties met $\gamma = \sqrt{10}$ en $p_f(0) = \frac{a}{a+b}$

Voordat we aan het betreffende model gaan rekenen, willen we graag eerst nog wat simulaties doen met de twee modellen die hier zijn beschreven. We kiezen er hierbij voor om de initiële verdeling van de deeltjes gelijk te nemen aan de evenwichtsverdeling, ofwel  $p_f(0) = \frac{a}{a+b}$ , dit blijkt handig te zijn in onze latere berekeningen. We hebben daarbij in de figuren 3.4 t/m 3.11 gezien dat er voor  $a + b = 1$  weinig verschil zit in de pluimen met verschillende begincondities, dus voor deze gevallen is de aanname zeker gerechtvaardigd.

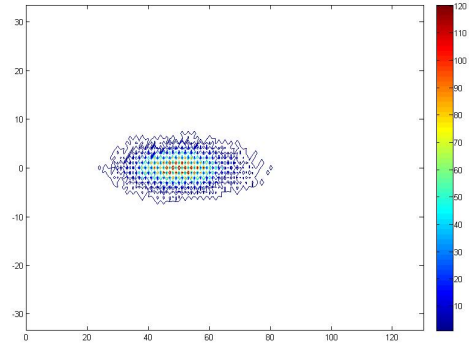
Als we naar de uitkomsten van de twee modellen kijken in figuur 4.3 tot en met 4.12 zien we twee verschillen. Ten eerste dat de contourplot van Model I bestaan uit lijnen en bij Model II uit stippen. Dit is iets dat te wijten is aan Matlab, in Model II zitten de deeltjes alleen op punten  $(i, j)$  met  $i$  en  $j$  beide even of beide oneven. Er vallen dus “gaten” tussen deze posities, waardoor Matlab de contourplot anders maakt. Bij Model I zitten de deeltjes op een ander raster, wat blijkbaar zorgt voor een nette contourplot.

Wat verder opvalt is dat voor figuren 4.5 en 4.11 de deeltjes ook op een negatieve x-positie kunnen zitten. Dit komt doordat een dispersiestapje naar links groter is dan een advectionstapje naar rechts, zo kan een vrij deeltje zich netto naar links verplaatsen. Bij Model II is dit niet het geval en zal een deeltje qua x-positie altijd op zijn plaats blijven, of naar rechts bewegen.

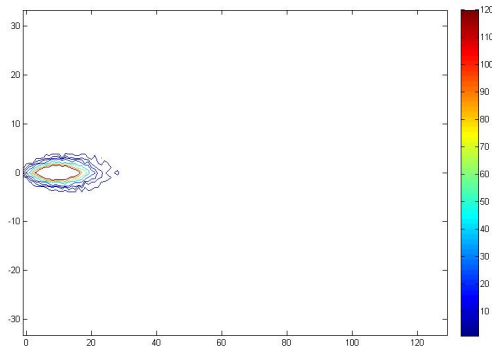
Afgezien van deze twee punten lijken de uitkomsten van de twee verschillende modellen identiek te zijn. We hebben dus nu een geloofwaardig model van de verspreiding van een vervuilingpluim in de aardbodem. In de volgende hoofdstukken zullen we verder in gaan op een kwalitatieve vaststelling van de vorm van deze vervuilingpluimen.



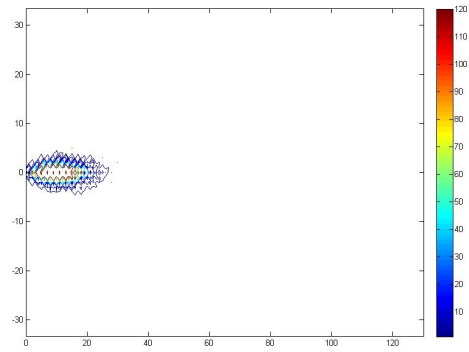
Figuur 4.3: Model I:  $a = 0.5$ ,  $b = 0.5$



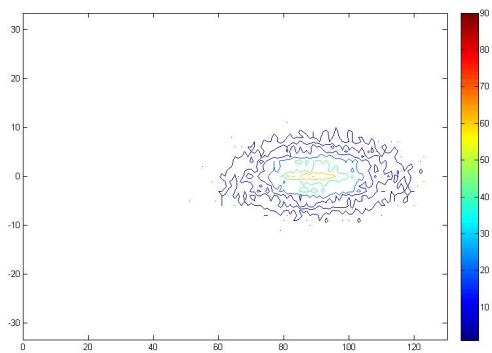
Figuur 4.4: Model II:  $a = 0.5$ ,  $b = 0.5$



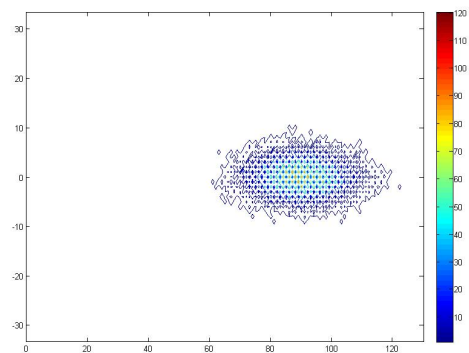
Figuur 4.5: Model I:  $a = 0.1$ ,  $b = 0.9$



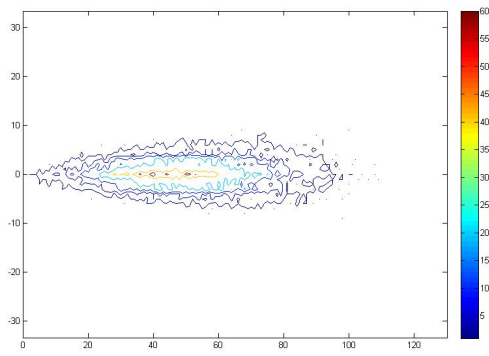
Figuur 4.6: Model II:  $a = 0.1$ ,  $b = 0.9$



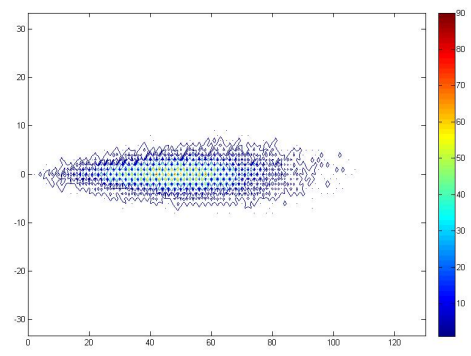
Figuur 4.7: Model I:  $a = 0.9$ ,  $b = 0.1$



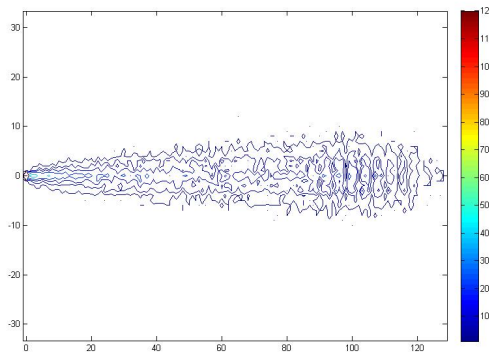
Figuur 4.8: Model II:  $a = 0.9$ ,  $b = 0.1$



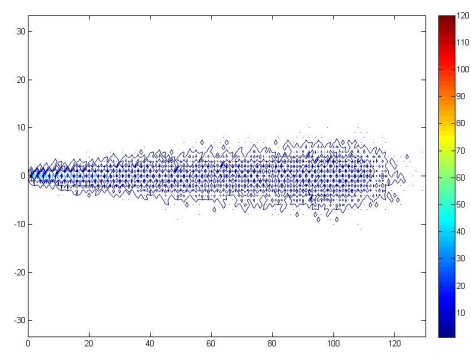
Figuur 4.9: Model I:  $a = 0.1$ ,  $b = 0.1$



Figuur 4.10: Model II:  $a = 0.1$ ,  $b = 0.1$



Figuur 4.11: Model I:  $a = 0.01$ ,  $b = 0.01$



Figuur 4.12: Model II:  $a = 0.01$ ,  $b = 0.01$



## Hoofdstuk 5

# Een maat voor de verspreiding van vervuilingsdeeltjes

We hebben in de voorgaande simulaties gezien dat de vervuilingspluimen er lang niet altijd uitzien als een Gaussisch patroon, doordat deeltjes die langer vrij zijn verder in de x-richting komen, maar ook meer dispersie kunnen uitvoeren dan deeltjes die weinig vrij zijn geweest. Vooral de pluimen met  $a = b = 0.1$  en  $a = b = 0.01$  laten dit goed zien. We zullen er nu naar kijken hoe we deze verspreiding kwalitatief zouden kunnen vaststellen.

### 5.1 Variantie van de pluim

We hebben al even kort gekeken naar de posities van de deeltjes. We noteren de positie van deeltje  $i$  ( $i = 1, 2, \dots, N$ ) met  $S_i = (S_{X,i}(n), S_{Y,i}(n))$ , het centrum van de pluim is dan gelijk aan

$$Z_N(n) = \left( \frac{1}{N} \sum_{i=1}^N S_{X,i}, \frac{1}{N} \sum_{i=1}^N S_{Y,i} \right) \approx (E[S_X], E[S_Y]),$$

waarbij de laatste afleiding geldt wegens de wet van de grote aantallen. Deze mogen we hier toepassen vanwege de onafhankelijkheid en de gelijke verdeling van de afzonderlijke deeltjes. Wat we graag willen is een maat voor de verspreiding van de deeltjes rond dit centrum. Dit is echter niet makkelijk als we aannemen dat de verdeling van de deeltjes geen Gaussisch patroon heeft, aangezien de verspreiding naar links dan niet hetzelfde zal zijn als de verspreiding naar rechts.

Wat we het beste kunnen doen is conditioneren op de x-positie van een deeltje, we kijken dan naar de y-positie van een deeltje, gegeven dat de x-positie van het deeltje gelijk is aan  $x$ . We kunnen dan naar het centrum van de pluim kijken onder  $S_X(n) = x$ , deze conditionering noteren we even met een  $x$  boven de grootheid. We hebben dan

$$Z_N^x(n) = \left( \frac{1}{N_x} \sum_{i \in \mathcal{N}_x} x, \frac{1}{N_x} \sum_{i \in \mathcal{N}_x} S_{Y,i} \right) = \left( x, \frac{1}{N_x} \sum_{i \in \mathcal{N}_x} S_{Y,i} \right)$$

voor  $\mathcal{N}_x$  de verzameling van deeltjes  $i$  met  $S_{X,i} = x$  en  $N_x$  de cardinaliteit van  $\mathcal{N}_x$ .

Maar  $Z_N^x(n)$  is ongeveer gelijk aan  $(x, E[S_Y(n)|S_X(n) = x]) = (x, 0)$ . Om een maat te verkrijgen voor de spreiding dwars op de advecctie, van de deeltjes rondom  $Z_N^x(n)$ , kunnen we kijken naar het kwadraat van de y-positie. Dit noemen we  $V_N^x(n)$ , waarbij we nog steeds conditioneren naar  $S_X(n) = x$ . Dit is gelijk aan

$$V_N^x(n) = \frac{1}{N_x} \sum_{i \in \mathcal{N}_x} S_{Y,i}^2(n) \approx E[S_Y^2(n)|S_X(n) = x] = \text{Var}(S_Y(n)|S_X(n) = x)$$

wegens  $E[S_Y(n)] = 0$ . We kunnen dus kijken naar de variantie in de y-richting van vervuilingsdeeltjes die zich op x-positie  $x$  bevinden. Als onze aanname klopt, namelijk dat de vervuilingspluim geen Gaussisch patroon aanneemt, zal deze variantie aan de rechter kant van het centrum groter zijn dan aan de linker kant.

## 5.2 Bepalen van $\text{Var}(S_Y(n)|S_X(n) = x)$

We zijn dus geïnteresseerd in de uitdrukking  $\text{Var}(S_Y(n)|S_X(n) = x)$ . Om hiervoor een uitdrukking te vinden, zullen we deze variantie eerst moeten opsplitsen in verschillende kansen, die we onafhankelijk van elkaar kunnen bepalen. We beginnen met

$$\text{Var}(S_Y(n)|S_X(n) = x) = E[S_Y^2(n)|S_X(n) = x] - (E[S_Y(n)|S_X(n) = x])^2.$$

Maar we hadden al gezien dat  $E[S_Y(n)|S_X(n) = x] = 0$ , zodat

$$\text{Var}(S_Y(n)|S_X(n) = x) = E[S_Y^2(n)|S_X(n) = x].$$

Dit kunnen we nu uitschrijven tot de volgende vergelijking:

$$\text{Var}(S_Y(n)|S_X(n) = x) = \sum_y y^2 P(S_Y(n) = y|S_X(n) = x). \quad (5.1)$$

De voorwaardelijke kans  $P(S_Y(n) = y|S_X(n) = x)$  is echter niet zo makkelijk te bepalen, omdat we voor een deeltje dat zich op x-positie  $x$  bevindt, verder niks weten. We willen  $P(S_Y(n) = y|S_X(n) = x)$  dus graag verder uitsplitsen, zodat we weten wat de deeltjes tijdens hun reis hebben gedaan. De factor die hier informatie over geeft, is het aantal keren dat het deeltje vrij is geweest,  $K_n$ .

We hebben nu drie onderdelen waarnaar we de kans  $P(S_Y(n) = y|S_X(n) = x)$  willen uitsplitsen:  $S_X(n) = x$ ,  $S_Y(n) = y$  en  $K_n = k$ . Voor de verduidelijking noemen we deze even respectievelijk  $A$ ,  $B$  en  $C_k$ , met  $k = 0, 1, \dots, n$ . Wat we dus uiteindelijk willen weten is  $P(B|A)$ .

We weten dat

$$P(A \cap B) = \sum_k P(A \cap B \cap C_k)$$

Maar omdat  $P(A \cap B \cap C_k) = P(A \cap B|C_k) P(C_k)$ , kunnen we dit verder uitschrijven tot

$$P(A \cap B) = \sum_k P(A \cap B|C_k) P(C_k).$$

Voor de voorwaardelijke kans geldt dus

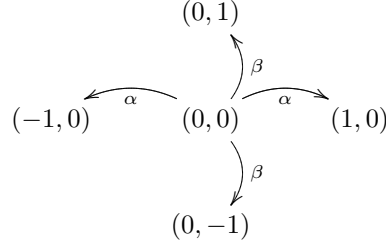
$$P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{\sum_k P(A \cap B|C_k) P(C_k)}{P(A)} = \frac{\sum_k P(A \cap B|C_k) P(C_k)}{\sum_k P(A|C_k) P(C_k)}.$$

Invullen van  $A$ ,  $B$  en  $C_k$  geeft ons dan een bruikbare uitdrukking voor  $P(S_Y(n) = y|S_X(n) = x)$ :

$$P(S_Y(n) = y|S_X(n) = x) = \frac{\sum_k P(S_Y(n) = y, S_X(n) = x|K_n = k) P(K_n = k)}{\sum_k P(S_X(n) = x|K_n = k) P(K_n = k)}. \quad (5.2)$$

We moeten dus drie kansen afleiden om de betreffende voorwaardelijke variantie te kunnen bepalen. Hiervoor is het erg belangrijk om te weten wat de verschillende deeltjes hebben gedaan in de tijden dat ze vrij zijn geweest. Het model dat ik hiervoor heb gebruikt, is het model waarin de deeltjes

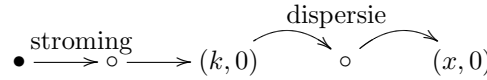
als ze vrij zijn met kansen  $\alpha$  en  $\beta$  een stapje doen in de x- respectievelijk de y-richting, met gemiddelde verplaatsing 0.



Daarbij komt nog dat vrije deeltjes per tijdstap ook precies 1 stapje meestromen met het grondwater, in de x-richting. We hebben nu genoeg vastgelegd om de verschillende kansen uit te werken.

### 5.2.1 Bepalen van $P(S_Y(n) = y, S_X(n) = x | K_n = k)$

Eerst bepalen we  $P(S_Y(n) = y, S_X(n) = x | K_n = k)$ , de kans dat een deeltje dat  $k$  keer vrij is geweest, precies op y-positie  $y$  en x-positie  $x$  terecht komt. We onderscheiden hiervoor 2 gevallen:  $x > k$  en  $x \leq k$ . Dit is nodig, omdat een deeltje dat  $k < x$  keer vrij is geweest, bovenop de  $k$  advectionstapjes nog  $x - k$  stapjes naar rechts nodig heeft om op positie  $x$  terecht te komen, en dan nog maar  $k - (x - k)$  stapjes in de x- of y-richting kan doen. Daarentegen zal een deeltje dat  $k \geq x$  keer vrij is geweest, en zich op positie  $x$  bevindt, altijd nog  $k - (k - x) = x$  stapjes in de x- of y-richting kunnen doen.

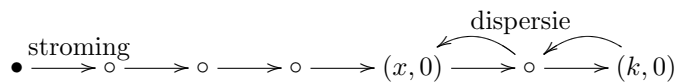


Figuur 5.1: Beweging van een deeltje met  $x > k$

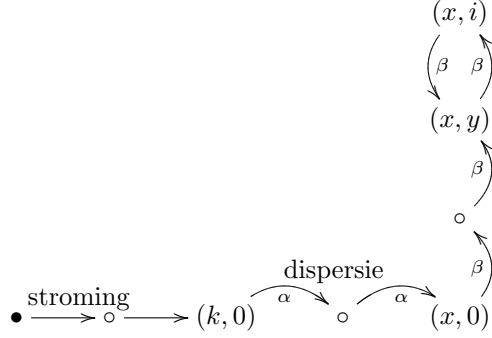
#### Het geval $x > k$ .

We bekijken eerst het geval  $x > k$ . Zo'n deeltje zal in ieder geval  $x - k$  keer een stapje naar rechts hebben moeten maken, in de  $k$  keren dat hij vrij is geweest. Dit levert een term  $\alpha^{x-k}$ . Daarnaast moet het deeltje in  $S_Y(n) = y$  uitkomen, wat een term  $\beta^y$  oplevert voor  $y \geq 0$  en  $\beta^{-y}$  voor  $y < 0$ . We kunnen echter makkelijk inzien dat de pluim symmetrisch is om  $y = 0$ , dus in plaats van  $y$  positief en negatief te nemen, kunnen we ook alleen de positieve  $y$  beschouwen, en het totaal met 2 vermenigvuldigen. We zullen er voor het gemak vanuit gaan dat  $y$  voortaan positief is. Ook merken we op dat voor  $x > k$  moet gelden dat  $y \leq k - (x - k)$ , want meer dispersiestapjes zijn gewoonweg niet mogelijk.

Als  $y = k - (x - k)$  zijn we klaar, maar stel  $y < k - (x - k)$ , wat deed het deeltje dan de rest van het aantal keren dat hij vrij is geweest? De rest van de tijd moet hij bewegingen hebben uitgevoerd die niks bijdragen aan zijn uiteindelijke positie, ofwel bewegingen in de x- of y-richting die tesamen op 0 uitkomen. Hiervoor moeten we een nieuw som-teken introduceren, bijvoorbeeld de som over het aantal keren dat een vrij deeltje een beweging naar boven uitvoert,  $i$ . We krijgen dan bovenop de stappen die we al hadden twee termen,  $\beta^i$  het aantal stappen naar boven en  $\beta^i$  het aantal stappen



Figuur 5.2: Beweging van een deeltje met  $x \leq k$



naar beneden. Houden we dan nog vrije tijdstappen over, dan voert het deeltje stappen in de x-richting uit, waarvan de helft naar links en de helft naar rechts:  $(\alpha^{\frac{k-2i+y-(x-k)}{2}})^2 = \alpha^{k-2i+y-(x-k)}$ . Alles tesamen geeft dit een term

$$\sum_i \beta^i \beta^{i-y} \alpha^{x-k} \alpha^{k-2i+y-(x-k)} = \sum_i \beta^{2i-y} \alpha^{k-2i+y}.$$

Nu hoeven we alleen nog de verschillende combinaties toe te voegen, om de uitdrukking voor  $P(S_Y(n) = y, S_X(n) = x | K_n = k)$  te krijgen. Ten eerste hebben we het aantal keren dat het deeltje in de  $k$  keren dat hij vrij is een beweging in de y-richting uitvoert,  $\binom{k}{2i-y}$ . Maar een deeltje voert in de y-richting bewegingen naar boven en naar beneden uit, daarvoor hebben we ook een combinatie nodig, bijvoorbeeld  $\binom{2i-y}{i}$  het aantal keren dat het deeltje naar boven beweegt. Tenslotte krijgen we nog een combinatie voor de bewegingen in de x-richting,  $\binom{k-2i+y}{\frac{k-2i+y-(x-k)}{2}}$  het aantal keren dat het deeltje naar rechts beweegt. Alles tesamen krijgen we dan de volgende uitdrukking:

$$P(S_Y(n) = y, S_X(n) = x | K_n = k) = \sum_i \beta^{2i-y} \alpha^{k-2i+y} \binom{k}{2i-y} \binom{2i-y}{i} \binom{k-2i+y}{\frac{k-2i+y-(x-k)}{2}}.$$

De som over  $i$  loopt vanaf  $y$ , omdat  $i$  het aantal stapjes is, dat het deeltje naar boven maakt, en dit moet in ieder geval  $y$  zijn. De bovengrens is zodanig dat het deeltje nog genoeg vrije stappen over heeft om in  $(x, y)$  te komen, dus de grootste mogelijke  $i$  is  $i = y + \frac{k-(x-k)-y}{2} = k - \frac{x}{2} + \frac{y}{2}$ . In dat geval zal een deeltje  $x-k$  stapjes naar rechts maken,  $y + \frac{k-(x-k)-y}{2}$  naar boven en  $\frac{k-(x-k)-y}{2}$  stapjes naar beneden. Alles tesamen volgt dat

$$P(S_Y(n) = y, S_X(n) = x | K_n = k) = \sum_{i=y}^{k-\frac{x}{2}+\frac{y}{2}} \beta^{2i-y} \alpha^{k-2i+y} \binom{k}{2i-y} \binom{2i-y}{i} \binom{k-2i+y}{\frac{k-2i+y-(x-k)}{2}} \quad (5.3)$$

**Voor  $x \leq k$ .**

We kunnen nu gaan kijken naar het geval  $x \leq k$ . Een aantal dingen blijven hetzelfde als bij het geval  $x > k$ , zoals  $\beta^i$ , de stappen naar boven en  $\beta^{i-y}$ , de stappen naar beneden. Het deeltje zal nu echter geen stappen naar rechts moeten maken om op x-positie  $x$  uit te komen, maar stappen naar links, wat leidt tot een term  $\alpha^{k-x}$ . Ook worden wederom de rest van de stappen gebruikt voor bewegingen heen en weer in de x-richting:  $\alpha^{k-2i+y-(k-x)}$ . Alles bij elkaar:

$$\begin{aligned} P(S_Y(n) = y, S_X(n) = x | K_n = k) &= \sum_i \beta^i \beta^{i-y} \alpha^{k-x} \alpha^{k-2i+y-(k-x)} \binom{k}{2i-y} \binom{2i-y}{i} \binom{k-2i+y}{\frac{k-2i+y-(k-x)}{2}} \\ &= \sum_i \beta^{2i-y} \alpha^{k-2i+y} \binom{k}{2i-y} \binom{2i-y}{i} \binom{k-2i+y}{\frac{k-2i+y-(k-x)}{2}} \end{aligned}$$

De bovengrens voor  $i$  is nu ook iets anders, namelijk  $i = y + \frac{k-(k-x)-y}{2} = \frac{x}{2} + \frac{y}{2}$ . De gehele uitdrukking wordt dan:

$$P(S_Y(n) = y, S_X(n) = x | K_n = k) = \sum_{i=y}^{\frac{x}{2} + \frac{y}{2}} \beta^{2i-y} \alpha^{k-2i+y} \binom{k}{2i-y} \binom{2i-y}{i} \binom{k-2i+y}{\frac{k-2i+y-(k-x)}{2}} \quad (5.4)$$

We kunnen nu vergelijkingen (5.3) en (5.4), de twee verschillende gevallen  $x > k$  en  $x \leq k$  op een slimme manier samennemen tot de volgende uitdrukking:

$$P(S_Y(n) = y, S_X(n) = x | K_n = k) = \sum_{i=y}^{\frac{x}{2} + \frac{y}{2} - \max(x-k, 0)} \beta^{2i-y} \alpha^{k-2i+y} \binom{k}{2i-y} \binom{2i-y}{i} \binom{k-2i+y}{\frac{k-2i+y-|x-k|}{2}} \quad (5.5)$$

### 5.2.2 Bepalen van $P(K_n = k)$

De tweede term die we moeten bepalen is  $P(K_n = k)$ , de kans dat een deeltje in  $n$  tijdstappen precies  $k$  keer vrij is geweest. Dit is een lastige term, omdat we wat willen weten over het aantal keren dat het deeltje wisselt van een geadsorbeerde toestand naar een vrije toestand en weer terug, maar hier weten we niks over.

We doen daarom voorlopig de concessie dat  $a + b = 1$  en  $p_f(0) = \frac{a}{a+b}$ . Dit maakt het veel makkelijker om  $P(K_n = k)$  te bepalen, omdat nu geldt dat de deeltjes beginnen in de evenwichtstoestand en er geldt dat  $1 - a = b$ . Een deeltje is dan op elk tijdstip met kans  $a$  vrij en met kans  $b$  geadsorbeerd, onafhankelijk van de vorige tijdstappen.  $K_n$  is daardoor een binomiale verdeling geworden, dus

$$P(K_n = k) = a^k (1 - a)^{n-k} \binom{n}{k}. \quad (5.6)$$

### 5.2.3 Bepalen van $P(S_X(n) = x | K_n = k)$

Tenslotte willen we nog weten wat de kans is dat een deeltje in  $S_X(n) = x$  belandt, gegeven dat hij precies  $k$  keer vrij is geweest. Wederom krijgen we te maken met  $|x - k|$  stapjes naar rechts of links (afhankelijk van of  $x > k$  of  $x \leq k$ ) om het deeltje op x-positie  $x$  te laten belanden:  $\alpha^{|x-k|}$ . Wat doet het deeltje de rest van de  $k$  stappen dat hij vrij is? Bewegingen maken in y-richting, met kans  $2\beta$  (met kans  $\beta$  naar boven en met kans  $\beta$  naar beneden), of in de x-richting met som 0. Hiervoor introduceren we een som over  $j$ , voor het aantal bewegingen naar rechts, dat gecompenseerd wordt door een beweging naar links. Tensamen hebben we dan

$$\sum_j \alpha^{|x-k|} \alpha^j \alpha^j (2\beta)^{k-2j-|x-k|} = \sum_j \alpha^{2j+|x-k|} (2\beta)^{k-2j-|x-k|}.$$

Nu moeten we weer de verschillende combinaties toevoegen. Ten eerste het aantal keren dat in  $k$  vrije stapjes  $k - 2j - |x - k|$  keer in de y-richting wordt bewogen,  $\binom{k}{k-2j-|x-k|}$ . Voor de bewegingen in de x-richting hebben we nog een term nodig die bepaalt in welke volgorde er naar links en naar rechts wordt gegaan, bijvoorbeeld  $\binom{2j+|x-k|}{j}$ . Voor de y-richting hebben we dit niet nodig, omdat het voor de x-positie niet uitmaakt of een deeltje naar boven of naar beneden beweegt.

$$P(S_X(n) = x | K_n = k) = \sum_j \alpha^{2j+|x-k|} (2\beta)^{k-2j-|x-k|} \binom{k}{k-2j-|x-k|} \binom{2j+|x-k|}{j}$$

Maar wat zijn de grenzen voor  $j$ ? De ondergrens is  $j = 0$ , immers het deeltje hoeft niet heen-en-weer te gaan in de x-richting, het kan ook  $k - |x - k|$  keer in de y-richting bewegen. De bovengrens

moet zodanig zijn dat het deeltje nog genoeg vrije stappen overhoudt om weer in  $x$  uit te komen, dus de bovengrens is  $j = \frac{k-|x-k|}{2}$  voor  $x$  even en  $j = \frac{k-|x-k|-1}{2}$  voor  $x$  oneven. Tesamen is de bovengrens voor  $j$  dan  $j = \lfloor \frac{k-|x-k|}{2} \rfloor$ , met  $\lfloor \cdot \rfloor$  afronding naar beneden. We hebben nu de complete uitdrukking voor  $P(S_X(n) = x | K_n = k)$ :

$$P(S_X(n) = x | K_n = k) = \sum_{j=0}^{\lfloor \frac{k-|x-k|}{2} \rfloor} \alpha^{2j+|x-k|} (2\beta)^{k-2j-|x-k|} \binom{k}{k-2j-|x-k|} \binom{2j+|x-k|}{j}. \quad (5.7)$$

### 5.2.4 Bepalen van grenzen

Nu kunnen we vergelijkingen (5.5), (5.6) en (5.7) in vergelijking (5.2) voegen tot de gezochte variantie:

$$\begin{aligned} \text{Var}(S_Y(n) | S_X(n) = x) &= \sum_y y^2 \frac{\sum_k P(S_Y(n) = y, S_X(n) = x | K_n = k) P(K_n = k)}{\sum_k P(S_X(n) = x | K_n = k) P(K_n = k)} \\ &= \frac{2 \sum_{y \geq 0} y^2 \sum_k \sum_{i=y}^{\frac{x}{2} + \frac{y}{2} - \max(x-k, 0)} \beta^{2i-y} \alpha^{k-2i+y} \binom{k}{2i-y} \binom{2i-y}{i} \binom{k-2i+y}{\frac{k-2i+y-|x-k|}{2}} a^k (1-a)^{n-k} \binom{n}{k}}{\sum_k \sum_{j=0}^{\lfloor \frac{k-|x-k|}{2} \rfloor} \alpha^{2j+|x-k|} (2\beta)^{k-2j-|x-k|} \binom{k}{k-2j-|x-k|} \binom{2j+|x-k|}{j} a^k (1-a)^{n-k} \binom{n}{k}} \end{aligned}$$

Wat ons nu nog rest is het bepalen van de grenzen voor  $k$  en  $y$ . Laten we beginnen met  $y$ . We hadden afgesproken  $y$  alleen positief te nemen en het geheel met 2 te vermenigvuldigen. De ondergrens voor  $y$  is dus 0. Voor de bovengrens moeten we kijken op welke  $y$ -positie een deeltje maximaal kan komen. Voor  $x > k$  is dit  $y = k - (x - k)$ , voor  $x \leq k$  is de bovengrens  $y = k - (k - x) = x$ . Dit kunnen we handig samenvoegen tot de bovengrens  $y = k - |x - k|$ . Waar we echter nog wel aan moeten denken is dat voor  $x$  even, een deeltje alleen op een even  $y$ -positie kan komen en evenzo kan een deeltje op een oneven  $x$  ook alleen op een oneven  $y$ -positie terecht komen. Ook moeten we opmerken dat de som over  $y$  nu na de som over  $k$  komt te staan, omdat de grenzen van  $y$  afhangen van  $k$ .

We hoeven nu alleen nog de grenzen voor  $k$  te bepalen, het aantal keren dat een deeltje vrij is. Intuïtief lijkt  $k=0$  de ondergrens te zijn, maar dit is niet altijd het geval. We kijken namelijk op een positie  $x$  en een deeltje dat nooit vrij is geweest, kan hier niet komen voor  $x > 0$ . De ondergrens is van de vorm  $k = \frac{x}{2}$ , een deeltje dat precies  $\frac{x}{2}$  keer vrij is geweest, is namelijk meegestroomd tot  $\frac{x}{2}$  en kan dan nog precies  $\frac{x}{2}$  stapjes naar rechts maken, om op  $x$  uit te komen. Voor  $x$  oneven moeten we afronden naar boven, zodat  $k = \frac{x+1}{2}$  de ondergrens is. De bovengrens is natuurlijk  $n$ , het totaal aantal tijdstapjes.

We hebben nu alles bepaald om de gezochte variantie op te kunnen schrijven, waarin we weer op een slimme manier  $x$  even en  $x$  oneven in één formule samen kunnen nemen.

$$\begin{aligned}
\text{Var}(S_Y(n)|S_X(n) = x) = & \\
& 2 \sum_{k=\lceil \frac{x}{2} \rceil}^n \sum_{y=x \bmod 2, +2, +4, \dots}^{k-|x-k|} y^2 \sum_{i=y}^{\frac{x}{2} + \frac{y}{2} - \max(x-k, 0)} \beta^{2i-y} \alpha^{k-2i+y} \binom{k}{2i-y} \binom{2i-y}{i} \binom{k-2i+y}{\frac{k-2i+y-|x-k|}{2}} a^k (1-a)^{n-k} \binom{n}{k} \\
& \hline
& \sum_{k=\lceil \frac{x}{2} \rceil}^n \sum_{j=0}^{\lfloor \frac{k-|x-k|}{2} \rfloor} \alpha^{2j+|x-k|} (2\beta)^{k-2j-|x-k|} \binom{k}{k-2j-|x-k|} \binom{2j+|x-k|}{j} a^k (1-a)^{n-k} \binom{n}{k}
\end{aligned} \tag{5.8}$$

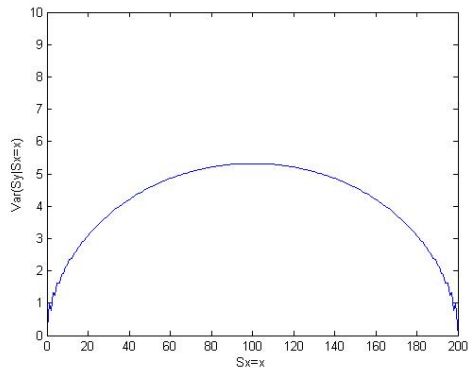
### 5.3 Uitkomsten met behulp van Matlab

Het bepalen van  $\text{Var}(S_Y(n)|S_X(n) = x)$  aan de hand van formule (5.8) zou een geestdodend karwei worden, vandaar dat we opnieuw de hulp inschakelen van Matlab. We kunnen de desbetreffende formule voor gegeven  $n, \alpha, \beta, a$  en  $b$  gemakkelijk programmeren in Matlab en de variantie uitrekenen voor verschillende  $x$ . De  $x$  waarin we geïnteresseerd zijn, lopen van  $x = 0$  tot  $x = 2n$ . Immers, na  $n$  tijdstapjes zullen de deeltjes zich op dit domein bevinden. Deeltjes die altijd geadsorbeerd zijn geweest, of steeds als ze vrij zijn geweest een dispersiestap naar links hebben uitgevoerd zullen zich op  $x = 0$  bevinden. Het andere uiterste zijn deeltjes die alle  $n$  tijdstappen vrij zijn geweest en steeds een dispersiestap naar rechts hebben uitgevoerd, zij zullen zich op  $x = 2n$  bevinden. In de volgende figuren staan voor de 3 gevallen met  $a + b = 1$  de berekende variantie van  $x = 0$  tot  $x = 2n$ . Hierbij is  $\alpha = \frac{5}{11}$  en  $\beta = \frac{1}{22}$ , zoals we in Hoofdstuk 4 hebben bepaald. Ter vergelijking is ook een vierde mogelijkheid toegevoegd waarbij alle deeltjes altijd vrij zijn ( $a = 1, b = 0$ ). Dit geeft de variantie voor het Gaussische geval.

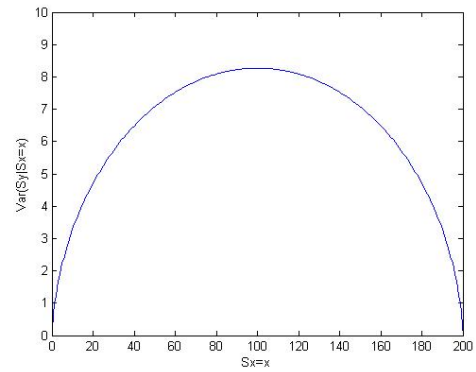
We zien ons vermoeden bevestigd, dat de variantie aan de linkerkant van het centrum van de pluim, kleiner is dan aan de rechterkant. De vervuilingspluimen vormen dus geen Gaussische krommen, want dan zou de  $\text{Var}(S_Y(n)|S_X(n) = x)$  een piek bij het centrum van de pluim moeten hebben, zoals in figuur 5.8.

Er vallen verder nog een aantal dingen op aan de uitkomsten. Ten eerste dat de variantie niet alleen een waarde heeft voor punten waar de deeltjes zich daadwerkelijk bevinden, maar ook voor punten waar we helemaal geen vervuilingspluim zien! Als we terugkijken naar formule 5.2 zien we hoe dit kan. Voor punten waar de contourplot van de vervuilingswolk geen deeltjes meer laat zien, is  $P(S_Y(n) = y, S_X(n) = x | K_n = k)$  heel klein, maar  $P(S_X(n) = x | K_n = k)$  ook, waardoor een heel klein getal wordt gedeeld door een ander heel klein getal. Op deze manier krijgen we toch een grote waarde voor de variantie. Wat ook opvalt is dat de variantie symmetrisch is om  $x = n$ . Dit kunnen we niet met een simpel argument verklaren, dus hier zullen we later nog op terugkomen.

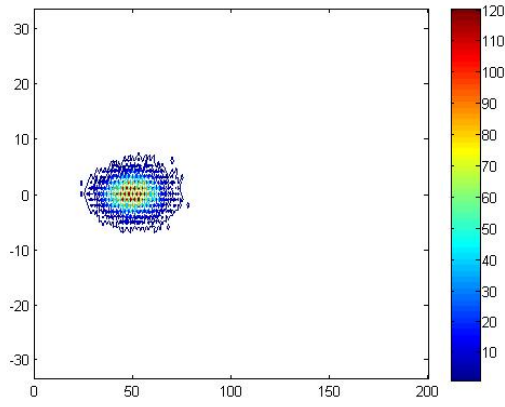
Iets anders opmerkelijks vindt vooral plaats bij het geval  $a = 0.1, b = 0.9$ , we zien hier een zigzag in de variantie. Hiervoor gaan we even terug naar onze bevinding dat deeltjes zich alleen op punten  $(i, j)$  kunnen bevinden met  $i$  en  $j$  beide even of beide oneven. Dit betekent dat op even  $x$ -posities, deeltjes zich kunnen bevinden op  $(x, 0)$ , maar op posities met  $x$  oneven, kan dit niet en zal het merendeel van de deeltjes zich bevinden op de punten  $(x, \pm 1)$ . Deze punten dragen echter bij aan de variantie, terwijl de punten  $(x, 0)$  dit niet doen. We zien dus telkens een piek in de variantie bij oneven  $x$ . Voor de andere 3 gevallen geldt dit fenomeen natuurlijk net zo goed, maar hier is dit minder sterk, doordat de spreiding in de  $y$ -richting groter is.



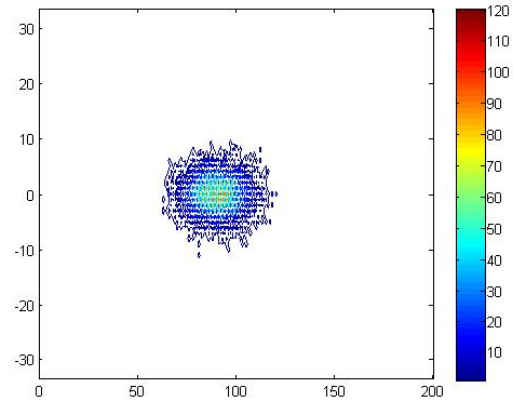
Figuur 5.3:  $\text{Var}(S_Y(n)|S_X(n) = x)$  voor  $a = 0.5$ ,  $b = 0.5$  en  $n = 100$



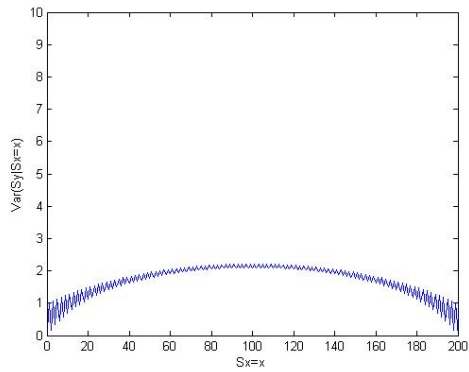
Figuur 5.4:  $\text{Var}(S_Y(n)|S_X(n) = x)$  voor  $a = 0.9$ ,  $b = 0.1$  en  $n = 100$



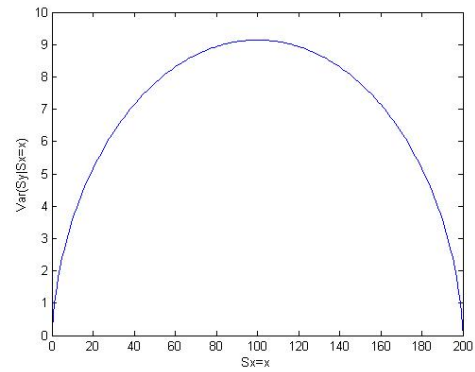
Figuur 5.5: De pluim van  $a = 0.5$ ,  $b = 0.5$ ,  $n = 100$  en  $N = 10,000$



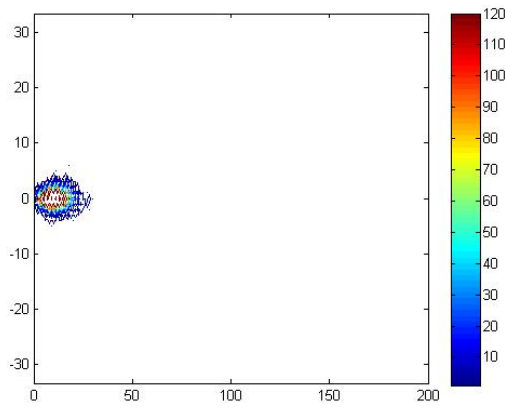
Figuur 5.6: De pluim van  $a = 0.9$ ,  $b = 0.1$ ,  $n = 100$  en  $N = 10,000$



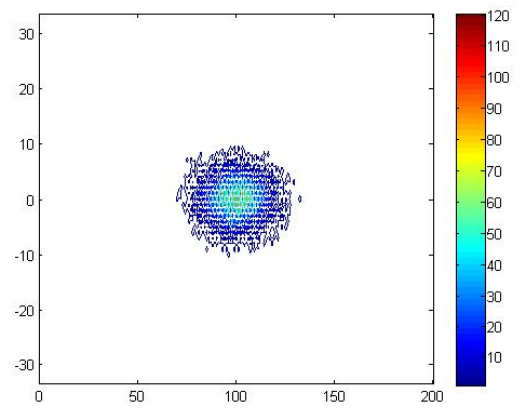
Figuur 5.7:  $\text{Var}(S_Y(n)|S_X(n) = x)$  voor  $a = 0.1$ ,  $b = 0.9$  en  $n = 100$



Figuur 5.8:  $\text{Var}(S_Y(n)|S_X(n) = x)$  voor  $a = 1$ ,  $b = 0$  en  $n = 100$



Figuur 5.9: De pluim van  $a = 0.1$ ,  $b = 0.9$ ,  $n = 100$  en  $N = 10,000$



Figuur 5.10: De pluim van  $a = 1$ ,  $b = 0$ ,  $n = 100$  en  $N = 10,000$

## 5.4 Bepalen van $P(K_n = k)$ voor $a + b \neq 1$

We willen nu ook graag  $\text{Var}(S_Y(n)|S_X(n) = x)$  bepalen voor de gevallen dat  $a + b \neq 1$ , maar hier komen we in de problemen met het bepalen van  $P(K_n = k)$ . Voor  $a + b = 1$  was het aantal keren dat een deeltje vrij is,  $K_n$  namelijk binomiaal verdeeld. Voor  $a + b \neq 1$  is  $K_n$  echter niet meer gewoon binomiaal verdeeld, maar zogenaamd matrix binomiaal verdeeld, met de overgangsmatrix uit hoofdstuk 1;

$$\begin{bmatrix} p_{ff} & p_{fa} \\ p_{af} & p_{aa} \end{bmatrix} = \begin{bmatrix} 1-b & b \\ a & 1-a \end{bmatrix}.$$

We kunnen  $P(K_n = k)$  dan bepalen met behulp van zijn kansgenererende functie, zoals gedaan wordt in [4]. Een makkelijkere manier, die betrekkelijk eenvoudig te implementeren is in Matlab, is in dit geval echter met recursie, zoals beschreven in [2]. Hiertoe splitsen we de deeltjes op tijdstip  $n$  op in twee soorten, naar of het deeltje in de laatste tijdstap vrij of geadsorbeerd was. We bekijken dus

$$p_n^i(k) = P(K_n = k, X_n = i), \quad i = 0, 1$$

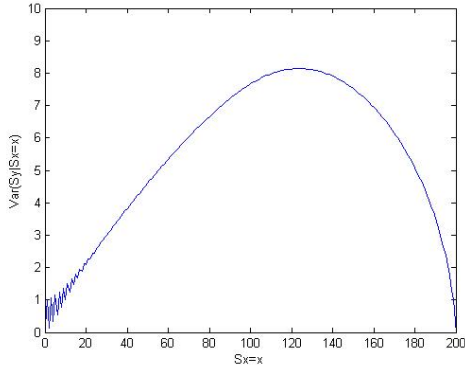
waarbij  $X_n = 0$  aangeeft dat dat deeltje tijdens tijdstap  $n$  geadsorbeerd was, terwijl  $X_n = 1$  aangeeft dat het deeltje vrij was. We merken op dat  $p_n(k) = p_n^0(k) + p_n^1(k)$  maar ook  $p_1^1(1) = \frac{a}{a+b}$ ,  $p_1^0(0) = \frac{b}{a+b}$ ,  $p_1^1(0) = 0$  en  $p_1^0(1) = 0$ . Immers, een deeltje dat op tijdstip 1 vrij was, kan niet in totaal 0 keer vrij zijn geweest, evenals een deeltje dat op tijdstip 1 geadsorbeerd was, niet in totaal 1 keer vrij geweest kan zijn. We kunnen  $p_n^i(k)$  dan recursief bepalen voor  $n \geq 1$  volgens

$$\begin{aligned} (i) \quad p_{n+1}^0(k) &= (1-a)p_n^0(k) + bp_n^1(k) \\ (ii) \quad p_{n+1}^1(k) &= ap_n^0(k-1) + (1-b)p_n^1(k-1). \end{aligned}$$

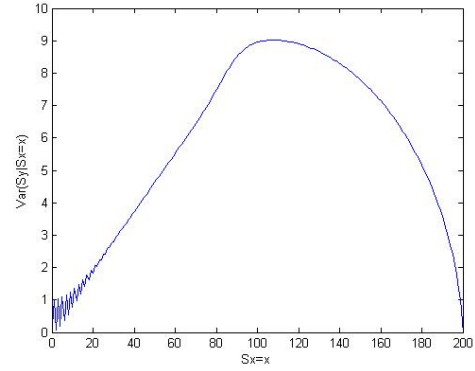
Met de hand zou dit wederom monnikenwerk zijn, maar voor Matlab is het een peulenschil. We kunnen op deze manier voor gegeven  $a$ ,  $b$  en  $n$  de kans  $P(K_n = k)$  betrekkelijk snel uitrekenen voor alle  $k$  en deze invullen in de formule voor  $\text{Var}(S_Y(n)|S_X(n) = x)$ . Op deze manier kunnen we de uitkomsten krijgen voor  $a = 0.1$ ,  $b = 0.1$  en  $a = 0.01$ ,  $b = 0.01$ , deze staan in figuren 5.11 en 5.12.

We zien dat de symmetrie in de voorwaardelijke variantie, die we zagen voor de gevallen waarbij  $a + b = 1$  hier niet opgaat, hoe dit komt zullen we bekijken in Hoofdstuk 6. Wat verder opvalt is dat de variantie voor  $x > 100$  erg lijkt op de variantie van het Gaussische geval  $a = 1$ ,  $b = 0$ . Dit kan verklaard worden door op te merken dat veel van de deeltjes die zo ver komen, (bijna) altijd vrij zullen zijn geweest, wat overeen komt met het  $a = 1$ ,  $b = 0$ . Tenslotte valt op dat we voor kleine  $x$  in beide gevallen weer een zigzag zien in de variantie. Dit hebben we al verklaard voor  $a = 0.1$ ,  $b = 0.9$ . Als we kijken naar de verdeling van het aantal keren dat de deeltjes vrij zijn geweest,  $P(K_n = k)$  zien we dat er op kleine  $x$  weinig deeltjes zitten, die volgens het plaatje van de pluim dicht bij elkaar zitten. De zigzag komt hier dus ook door deeltjes die op punten  $(i, j)$  zitten met  $i$  en  $j$  beide even of beide oneven.

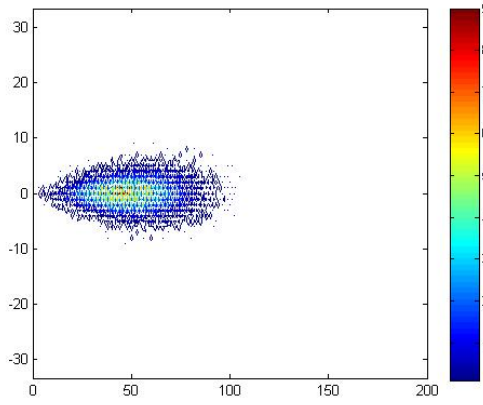
In dit hoofdstuk zijn we uitgegaan van een beginconditie in evenwichtstoestand, dus  $\frac{a}{a+b}$  fractie van de deeltjes waren op  $n = 0$  vrij en  $\frac{b}{a+b}$  gedeelte was geadsorbeerd. We hebben in hoofdstuk 1 gezien dat voor de gevallen met  $a + b = 1$  deze begintoestand weinig uitmaakt voor de uiteindelijke vervuilingspluim. Voor de twee gevallen met  $a + b \neq 1$  was er echter wel een groot verschil in de pluim, daarom zullen we daar ook nog even naar kijken. We kunnen deze beginconditie gemakkelijk veranderen in de recursieformules voor  $P(K_n = k)$ , zodat bijvoorbeeld  $p_1^1(1) = 0$ ,  $p_1^0(0) = 1$  waarop alle deeltjes voor  $n = 0$  geadsorbeerd zijn. Voor initiële vrije deeltjes krijgen we dan  $p_1^1(1) = 1$  en  $p_1^0(0) = 0$ . Zo krijgen we de uitkomsten voor  $a = b = 0.1$  in figuur 5.17 en  $a = b = 0.01$  in figuur 5.21.



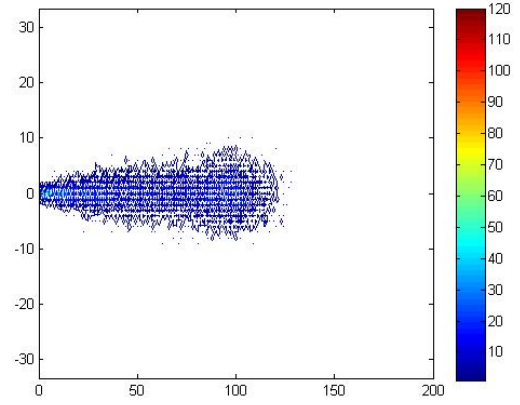
Figuur 5.11:  $\text{Var}(S_Y(n)|S_X(n) = x)$  voor  $a = 0.1$ ,  $b = 0.1$  en  $n = 100$



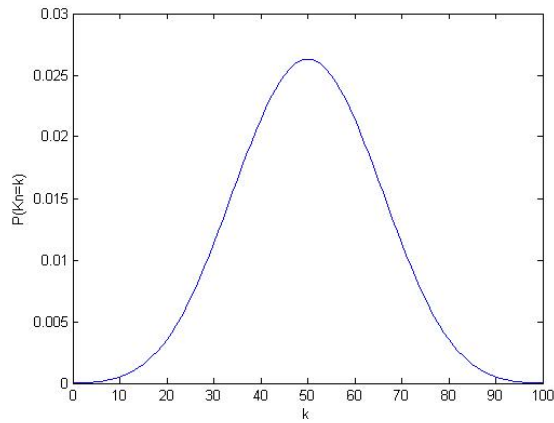
Figuur 5.12:  $\text{Var}(S_Y(n)|S_X(n) = x)$  voor  $a = 0.01$ ,  $b = 0.01$  en  $n = 100$



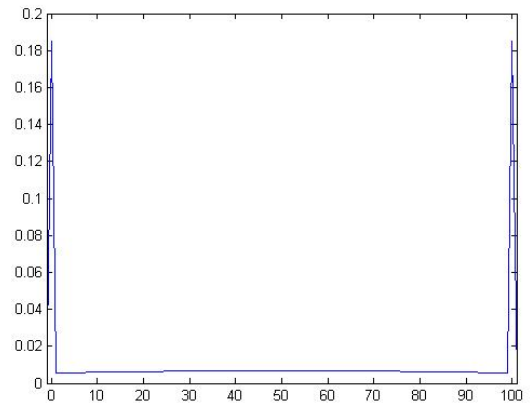
Figuur 5.13: De pluim van  $a = 0.1$ ,  $b = 0.1$  en  $n = 100$



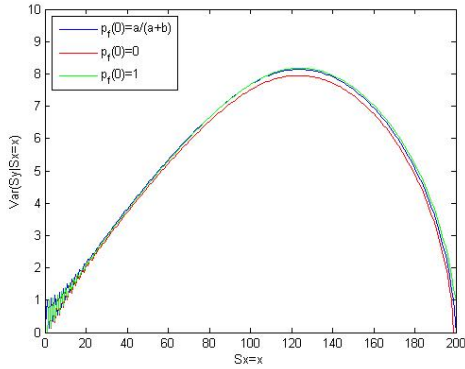
Figuur 5.14: De pluim van  $a = 0.01$ ,  $b = 0.01$  en  $n = 100$



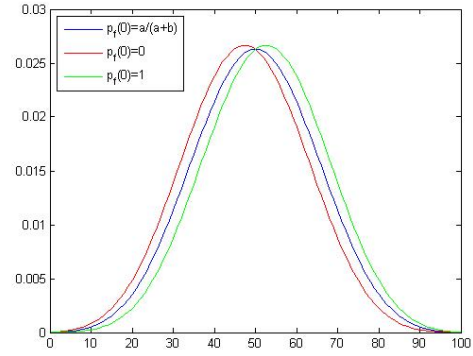
Figuur 5.15:  $P(K_n = k)$  voor  $a = 0.1$ ,  $b = 0.1$  en  $n = 100$



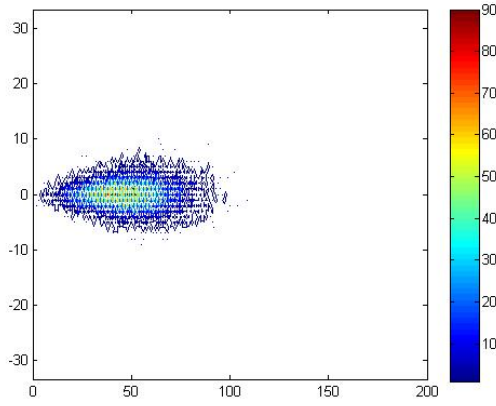
Figuur 5.16:  $P(K_n = k)$  voor  $a = 0.01$ ,  $b = 0.01$  en  $n = 100$



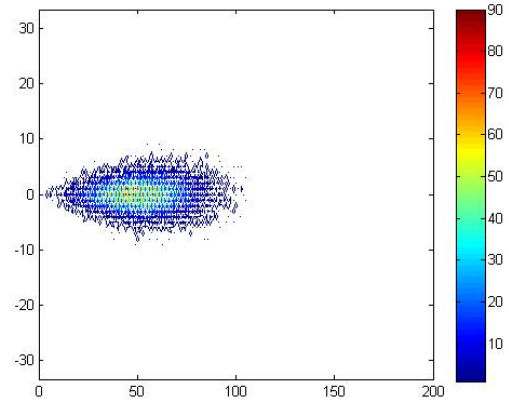
Figuur 5.17:  $\text{Var}(S_Y(n)|S_X(n)=x)$  voor  $a = 0.1$ ,  $b = 0.1$ ,  $n = 100$  en driemaal  $p_f(0)$



Figuur 5.18:  $P(K_n=k)$  voor  $a = 0.1$ ,  $b = 0.1$ ,  $n = 100$  en verschillende  $p_f(0)$



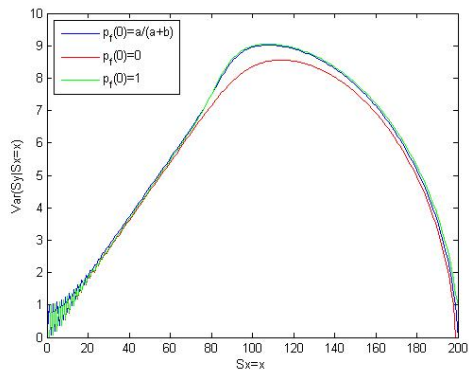
Figuur 5.19: De pluim van  $a = 0.1$ ,  $b = 0.1$ ,  $n = 100$  en  $p_f(0) = 0$



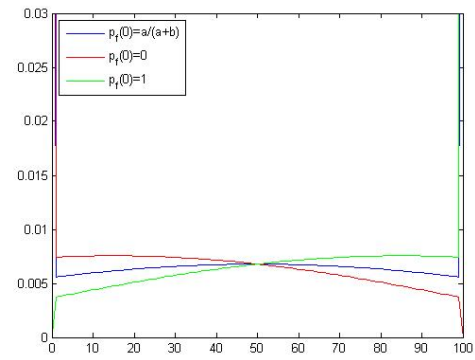
Figuur 5.20: De pluim van  $a = 0.1$ ,  $b = 0.1$ ,  $n = 100$  en  $p_f(0) = 1$

Voor het geval  $a = b = 0.1$  zien we niet erg veel verschil in zowel de verdeling van  $P(K_n = k)$  als in  $\text{Var}(S_Y(n)|S_X(n)=x)$ , in dit geval maakt de beginverdeling van de deeltjes over vrij en geadsorbeerd dus niet veel uit voor de uiteindelijke vorm en variantie van de pluim. Voor  $a = b = 0.01$  zien we echter wat meer verschil. Ten eerste in de verdeling van de deeltjes, voor  $p_f(0) = 0$  zien we een grote piek aan deeltjes bij  $k = 0$ , er zijn dus veel deeltjes die nooit vrij zijn geweest, voor  $p_f(0) = 1$  zien we juist veel deeltjes die altijd vrij zijn geweest. In lijn met de verwachtingen zien we daarop voor  $p_f(0) = \frac{a}{a+b}$  twee pieken, namelijk voor zowel  $k = 0$  als  $k = 100$ . Hoe komt het dan dat we in de variantie in dit geval geen groot verschil zien? Dat komt doordat deeltjes met  $k = 0$  altijd in de oorsprong zijn blijven zitten en dus ook niks bijdragen aan de variantie. Wel zien we dat de variantie voor  $p_f(0) = 0$  wat kleiner is dan voor  $p_f(0) = 1$ , dit valt te verklaren doordat de deeltjes in het eerste geval gemiddeld meer geadsorbeerd zijn geweest (ook als we de deeltjes die altijd geadsorbeerd zijn geweest weg zouden strepen), waardoor ze minder bijdragen aan de variantie. Dit fenomeen zien we voor  $a = b = 0.1$  ook lichtelijk, maar omdat daar de overgangskansen 10 keer zo groot zijn, is het verschil in de variantie minder sterk.

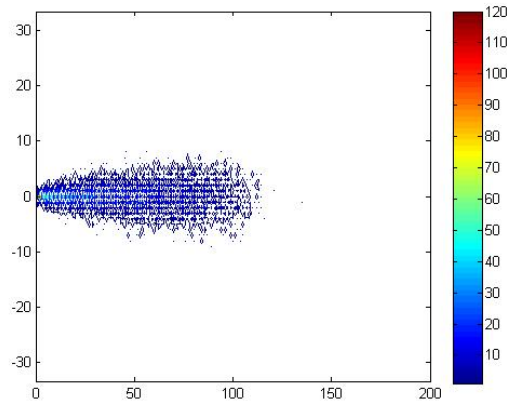
We hebben in dit hoofdstuk voor verschillende gevallen de verwachte variantie berekend voor de vervuiling pluimen. Eén van de dingen die hierbij opviel is een symmetrie in de variantie rond het punt  $x = n$  voor de gevallen met  $a + b = 1$ , hier zullen we in het volgende hoofdstuk dieper op ingaan.



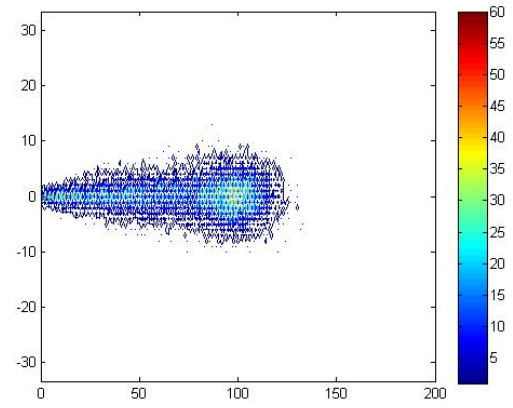
Figuur 5.21:  $\text{Var}(S_Y(n)|S_X(n) = x)$  voor  $a = 0.01$ ,  $b = 0.01$ ,  $n = 100$  en driemaal  $p_f(0)$



Figuur 5.22:  $P(K_n = k)$  voor  $a = 0.01$ ,  $b = 0.01$ , en verschillende  $p_f(0)$



Figuur 5.23: De pluim van  $a = 0.01$ ,  $b = 0.01$ ,  $n = 100$  en  $p_f(0) = 0$



Figuur 5.24: De pluim van  $a = 0.01$ ,  $b = 0.01$ ,  $n = 100$  en  $p_f(0) = 1$



## Hoofdstuk 6

# De symmetrie in de variantie rondom $n$

Na het bepalen van  $\text{Var}(S_Y(n)|S_X(n) = x)$  voor de gevallen met  $a + b = 1$  viel ons iets merkwaardigs op, namelijk een symmetrie in de variantie rond het punt  $x = n$ . In dit hoofdstuk gaan we op zoek naar een verklaring hiervoor en een bewijs van dit vermoeden. Een eerste idee omtrent deze symmetrie is wellicht dat er in punten  $n - x$  en  $n + x$  ( $x = 0, 1, \dots, n$ ) evenveel vervuilingsdeeltjes terecht komen, die zich over dezelfde y-posities zullen verdelen. Het is in ons model waar dat als je de vervuilingspluim op  $n - x$  vergelijkt met die op  $n + x$ , je op dezelfde y-posities deeltjes aan kunt treffen. Een deeltje dat alle  $n$  de keren vrij is geweest, zal op x-positie  $n - x$  namelijk op y-posities  $-x, -x + 2, \dots, x - 2, x$  terecht kunnen komen, maar ook op x-positie  $n + x$  zal dit deeltje op deze y-posities kunnen eindigen. Maar deeltjes die  $k < n$  keer vrij zijn geweest, kunnen nooit op verdere y-posities komen. Dit gegeven zou echter wellicht tot de foutieve conclusie kunnen leiden dat deeltjes gelijk verdeeld zijn over x-posities  $n - x$  en  $n + x$  met bijbehorende y-posities. Een blik op de vervuilingspluim van bijvoorbeeld het geval  $a = b = 0.5$ ,  $n = 100$  in figuur 3.3 doet inzien dat dit niet het geval kan zijn. We zien namelijk deeltjes op x-posities  $n - x$  voor  $x \in \{30, 31, \dots, 69, 70\}$ , maar op punten  $n + x$  zien we voor geen enkele  $x$  deeltjes zitten. Het eerste vermoeden dat de deeltjes eerlijk verdeeld zijn over  $n - x$  en  $n + x$  met bijbehorende y-posities, is dus foutief. Het is zelfs zo dat deeltjes op  $n - x$  kunnen komen door  $k = \frac{n-x}{2}, \frac{n-x}{2} + 1, \dots, n$  keer vrij te zijn, terwijl deeltjes die in  $n + x$  willen komen ten minste  $\frac{n+x}{2}$  keer vrij moeten zijn. De kans om in  $n - x$  te komen is dus veel groter dan de kans om in  $n + x$  te komen. Maar hoe kunnen we de symmetrie om x-positie  $n$  dan verklaren?

### 6.1 Reductie tot symmetrie in de kansen

Laten we nog eens terugkijken naar hoe  $\text{Var}(S_Y(n)|S_X(n) = n - x)$  is opgebouwd:

$$\text{Var}(S_Y(n)|S_X(n) = n - x) = \sum_{y=-x}^x y^2 P(S_Y(n) = y|S_X(n) = n - x). \quad (6.1)$$

En voor  $\text{Var}(S_Y(n)|S_X(n) = n + x)$  geldt natuurlijk hetzelfde:

$$\text{Var}(S_Y(n)|S_X(n) = n + x) = \sum_{y=-x}^x y^2 P(S_Y(n) = y|S_X(n) = n + x). \quad (6.2)$$

Omdat formules (6.1) en (6.2) over dezelfde som lopen, geldt de gelijkheid in de voorwaardelijke varianties, als er gelijkheid is in de voorwaardelijke kansen, dus als  $P(S_Y(n) = y|S_X(n) = n - x) = P(S_Y(n) = y|S_X(n) = n + x)$ . We vermoeden dat er niet evenveel deeltjes op x-posities  $n - x$  en  $n + x$  komen, maar ze zich wel op dezelfde manier verdelen over de verschillende y-posities. Laten we

eerst eens kijken naar de x-posities 1 en  $2n-1$ . Op x-positie 1 kunnen deeltjes zich bevinden in de punten  $(1, -1)$  en  $(1, 1)$ . Wegens de symmetrie om de x-as, geldt dat  $P(S_Y(n) = -1 | S_X(n) = 1) = P(S_Y(n) = 1 | S_X(n) = 1)$ , dus de helft van de deeltjes op x-positie 1 zit op  $(1, -1)$  en de andere helft zit op  $(1, 1)$ . Derhalve

$$\text{Var}(S_Y(n) | S_X(n) = 1) = \sum_y y^2 P(S_Y(n) = y | S_X(n) = 1) = (-1)^2 \cdot \frac{1}{2} + 1^2 \cdot \frac{1}{2} = 1.$$

Maar op x-positie  $2n-1$  geldt hetzelfde. Een deeltje kan namelijk alleen op deze x-positie komen door alle  $n$  keer vrij te zijn en  $n-1$  keer dispersie uit te voeren naar rechts. Het deeltje kan dan nog precies één keer omhoog of naar beneden bewegen, met gelijke kansen. Dus ook

$$\text{Var}(S_Y(n) | S_X(n) = 2n-1) = \sum_y y^2 P(S_Y(n) = y | S_X(n) = 2n-1) = (-1)^2 \cdot \frac{1}{2} + 1^2 \cdot \frac{1}{2} = 1.$$

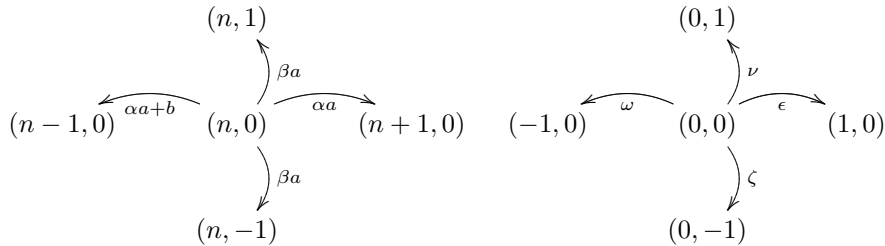
We zien dus dat hoewel de kansen dat een deeltje op x-positie 1 of  $2n-1$  komt, niet gelijk zijn, de voorwaardelijke kansen (en dus de varianties) wel gelijk zijn. Nu we een vermoeden hebben waar de symmetrie vandaan komt, kunnen we proberen dit te bewijzen.

## 6.2 Bewijs van de symmetrie<sup>1</sup>

Tot nu toe hebben we de situatie altijd bekeken met als oorsprong het punt waarin de deeltjes begonnen op  $n=0$ . We kunnen de situatie echter ook bekijken vanuit een meebewegend assenstelsel. Als we het assenstelsel laten meegaan met de advection, dus per tijdstapje één stapje in de x-richting, krijgen we een assenstelsel met oorsprong  $(n, 0)$ . In dit geval willen we dus bewijzen dat  $P(S_Y(n) = y | S_X(n) = -x) = P(S_Y(n) = y | S_X(n) = x)$ .

Vanuit dit meebewegende assenstelsel kan een deeltje per tijdstap 4 dingen doen. Ten eerste naar rechts bewegen met kans  $\alpha * a$ , het deeltje is dan vrij en voert dispersie uit naar rechts. Verder kan het deeltje vrij zijn en dispersie naar boven of beneden uitvoeren, elk met kans  $\beta * a$ . Naar links bewegen kan een deeltje op twee manieren doen, ten eerste door vrij te zijn en naar links te bewegen, de kans hierop is  $\alpha * a$ . Ten tweede kan het deeltje ook geadsorbeerd zijn, dan beweegt het niet, maar ten opzichte van ons bewegende assenstelsel zal het deeltje dan naar links bewegen. De kans hierop is  $b$ , we kijken hier immers naar de gevallen met  $a+b=1$ . We krijgen dan het linker schema. In dit model begint een deeltje dus op x-positie  $(n, 0)$  en voert het  $n$  stapjes uit volgens het linker schema. We hebben nu op een slimme manier de positie van de deeltjes losgekoppeld van  $K_n$ , het aantal keren dat het deeltje vrij is.

We kunnen ook kijken naar een algemener geval, waarbij we de kans om naar boven of beneden te gaan  $\nu$ , respectievelijk  $\zeta$  nemen, de kans om naar rechts te gaan  $\epsilon$  en de kans om naar links te gaan  $\omega$ . Dit is het rechter schema. We kunnen dan voor een algemene a-symmetrische, simpele random walk (d.w.z. met alleen stappen van lengte 1) die begint in  $(0, 0)$ , bewijzen dat  $\text{Var}(S_Y(n) | S_X(n) = x)$  symmetrisch is rondom  $x=0$ .



<sup>1</sup>Dit bewijs is met dank aan mijn begeleider, die met het idee kwam van een meebewegend assenstelsel en zo de symmetrie wist te bewijzen.

We willen nu aantonen dat voor deze rechter random walk geldt dat  $\text{Var}(S_Y(n)|S_X(n) = -x) = \text{Var}(S_Y(n)|S_X(n) = x)$ , door aan te tonen dat

$$P(S_X(n) = x, S_Y(n) = y) P(S_X(n) = -x) = P(S_X(n) = -x, S_Y(n) = y) P(S_X(n) = x).$$

Laten we eerst eens kijken naar  $P(S_X(n) = x, S_Y(n) = y)$ .

Een deeltje kan vanuit zijn beginpositie  $(0, 0)$  in het punt  $(x, y)$  komen door netto  $x$  stapjes naar rechts te maken en  $y$  stapjes naar boven. Maar het deeltje kan ook bewegingen maken in de x- en y-richting die worden gecompenseerd door een tegengestelde beweging. We hebben dus een som nodig, bijvoorbeeld een som over  $j$  voor de bewegingen in de x-richting die worden gecompenseerd door een tegenovergestelde beweging. De bewegingen in de y-richting volgen dan vanzelf uit het feit dat het aantal bewegingen in totaal  $n$  moet zijn. We krijgen dan de volgende vorm:

$$P(S_X(n) = x, S_Y(n) = y) = \sum_j c_j \epsilon^{x+j} \omega^j \nu^{y+\frac{1}{2}(n-x-y-2j)} \zeta^{\frac{1}{2}(n-x-y-2j)}.$$

Waarbij  $c_j$  een constante is die het aantal mogelijke combinaties aangeeft. De som over  $j$  loopt vanaf 0, immers, het deeltje is niet verplicht om heen en weer te bewegen in de x-richting. De bovengrens van  $j$  is zodanig dat er geen heen en weer bewegingen meer zijn in de y-richting, ofwel  $j = \frac{n-x-y}{2}$ . We krijgen dan

$$P(S_X(n) = x, S_Y(n) = y) = \sum_{j=0}^{\frac{n-x-y}{2}} c_j \epsilon^{x+j} \omega^j \nu^{\frac{1}{2}(n-x-y-2j)} \zeta^{\frac{1}{2}(n-x-y-2j)}. \quad (6.3)$$

Evenzo kunnen we de term voor  $P(S_X(n) = -x, S_Y(n) = y)$  bepalen. We moeten in dit geval de term  $\epsilon$  vervangen door de term  $\omega$  en omgekeerd, een deeltje moet om op x-positie  $-x$  te eindigen immers naar links bewegen, terwijl een deeltje dat in x-positie  $x$  moet eindigen naar rechts moet bewegen. We krijgen dan

$$P(S_X(n) = -x, S_Y(n) = y) = \sum_{j=0}^{\frac{n-x-y}{2}} c_j \omega^{x+j} \epsilon^j \nu^{\frac{1}{2}(n-x-y-2j)} \zeta^{\frac{1}{2}(n-x-y-2j)}. \quad (6.4)$$

De kans  $P(S_X(n) = x)$  kunnen we op een zelfde manier berekenen. We introduceren wederom een som over de bewegingen die een deeltje maakt in de x-richting, die worden gecompenseerd door een tegenovergestelde beweging. Deze stapjes noemen we  $i$  om deze som later niet in te verwarren met de som over  $j$ . Een deeltje moet natuurlijk netto  $x$  keer in de x-richting bewegen, maar hoe veel mag een deeltje bewegen in de y-richting? Net zo vaak als hij wil, want dat is voor ons hier niet interessant. We kunnen zelfs de beweging omhoog en naar beneden samen voegen, het deeltje beweegt dan met kans  $\nu + \zeta$  *niet in de x-richting*. De som over  $i$  loopt weer vanaf 0, want een deeltje is niet verplicht om heen en weer te bewegen in de x-richting, het kan ook alleen  $x$  keer naar rechts bewegen en de rest in de y-richting. Het maximum van  $i$  is  $\lfloor \frac{n-x}{2} \rfloor$ , het deeltje beweegt dan  $x + \lfloor \frac{n-x}{2} \rfloor$  keer naar rechts en  $\lfloor \frac{n-x}{2} \rfloor$  keer naar links. Alles tesamen krijgen we

$$P(S_X(n) = x) = \sum_{i=0}^{\lfloor \frac{n-x}{2} \rfloor} \tilde{c}_i \epsilon^{x+i} \omega^i (\nu + \zeta)^{n-x-2i}. \quad (6.5)$$

Hierbij is  $\tilde{c}_i$  wederom een constante die het aantal combinaties aangeeft.

We kunnen nu ook gemakkelijk  $P(S_X(n) = -x)$  bepalen door  $\epsilon$  weer te vervangen door  $\omega$  en omgekeerd. Zo krijgen we

$$P(S_X(n) = -x) = \sum_{i=0}^{\lfloor \frac{n-x}{2} \rfloor} \tilde{c}_i \omega^{x+i} \epsilon^i (\nu + \zeta)^{n-x-2i}. \quad (6.6)$$

Om aan te tonen dat

$$P(S_X(n) = x, S_Y(n) = y) P(S_X(n) = -x) = P(S_X(n) = -x, S_Y(n) = y) P(S_X(n) = x)$$

moeten we nu de formules (6.3) en (6.6) met elkaar vermenigvuldigen.

$$\begin{aligned} & P(S_X(n) = x, S_Y(n) = y) P(S_X(n) = -x) = \\ &= \left( \sum_{j=0}^{\frac{n-x-y}{2}} c_j \epsilon^{x+j} \omega^j \nu^{\frac{1}{2}(n-x+y-2j)} \zeta^{\frac{1}{2}(n-x-y-2j)} \right) \left( \sum_{i=0}^{\lfloor \frac{n-x}{2} \rfloor} \tilde{c}_i \omega^{x+i} \epsilon^i (2\beta a)^{n-x-2i} \right) \\ &= \sum_{i=0}^{\lfloor \frac{n-x}{2} \rfloor} \sum_{j=0}^{\frac{n-x-y}{2}} c_j \epsilon^{x+j} \omega^j \nu^{\frac{1}{2}(n-x+y-2j)} \zeta^{\frac{1}{2}(n-x-y-2j)} \tilde{c}_i \omega^{x+i} \epsilon^i (2\beta a)^{n-x-2i} \\ &= \sum_{i=0}^{\lfloor \frac{n-x}{2} \rfloor} \sum_{j=0}^{\frac{n-x-y}{2}} \tilde{c}_i c_j \epsilon^{x+i+j} \omega^{x+i+j} \nu^{\frac{1}{2}(n-x+y-2j)} \zeta^{\frac{1}{2}(n-x-y-2j)} (\nu + \zeta)^{n-x-2i} \end{aligned}$$

En als we formules (6.4) en (6.5) met elkaar vermenigvuldigen krijgen we

$$\begin{aligned} & P(S_X(n) = -x, S_Y(n) = y) P(S_X(n) = x) = \\ &= \left( \sum_{j=0}^{\frac{n-x-y}{2}} c_j \omega^{x+j} \epsilon^j \nu^{\frac{1}{2}(n-x+y-2j)} \zeta^{\frac{1}{2}(n-x-y-2j)} \right) \left( \sum_{i=0}^{\lfloor \frac{n-x}{2} \rfloor} \tilde{c}_i \epsilon^{x+i} \omega^i (2\beta a)^{n-x-2i} \right) \\ &= \sum_{i=0}^{\lfloor \frac{n-x}{2} \rfloor} \sum_{j=0}^{\frac{n-x-y}{2}} c_j \omega^{x+j} \epsilon^j \nu^{\frac{1}{2}(n-x+y-2j)} \zeta^{\frac{1}{2}(n-x-y-2j)} \tilde{c}_i \epsilon^{x+i} \omega^i (2\beta a)^{n-x-2i} \\ &= \sum_{i=0}^{\lfloor \frac{n-x}{2} \rfloor} \sum_{j=0}^{\frac{n-x-y}{2}} \tilde{c}_i c_j \epsilon^{x+i+j} \omega^{x+i+j} \nu^{\frac{1}{2}(n-x+y-2j)} \zeta^{\frac{1}{2}(n-x-y-2j)} (\nu + \zeta)^{n-x-2i}. \end{aligned}$$

Deze twee uitdrukkingen zijn dus precies gelijk aan elkaar! Blijkbaar is het zo dat er op x-posities  $-x$  en  $x$  niet evenveel deeltjes komen, maar de deeltjes die er komen zijn wel gelijk verdeeld over de mogelijke y-posities, waardoor  $\text{Var}(S_Y(n)|S_X(n) = x)$  op beide posities gelijk is.

Maar hoe zit het dan met het geval  $a + b \neq 1$ ? Geldt hier hetzelfde? Voor  $a + b = 1$  is de kans dat een deeltje op een bepaald tijdstip vrij is, onafhankelijk van of het deeltje het tijdstip ervoor ook vrij is geweest. De kans om vrij te zijn is in dit geval altijd  $a$  en de kans om geadsorbeerd te zijn is  $b$ . Maar voor  $a + b \neq 1$  is dit niet het geval. Voor  $a = b = 0.1$  bijvoorbeeld, is een deeltje dat op een bepaald tijdstip vrij is, met kans 0.9 de tijdstap daarna ook vrij. We hebben dan niet meer te maken met een random walk, want als een deeltje geadsorbeerd is tijdens tijdstap  $i$ , dan voert hij een stapje uit naar links (eigenlijk beweegt het deeltje niet, maar ons assenstelsel beweegt een stapje naar rechts), maar tijdens tijdstap  $i + 1$  zal hij dan ook met grote kans een stap naar links maken. Dit is geen random walk, want bij een random walk is er geen afhankelijkheid in de tijdstappen. Het hierboven geleverde bewijs gaat dan niet op.

# Hoofdstuk 7

## De empirische variantie

In Hoofdstuk 5 hebben we een formule afgeleid voor  $\text{Var}(S_Y(n)|S_X(n) = x)$ , maar we willen natuurlijk wel graag controleren of deze formule juist is. Daarom zullen we in dit hoofdstuk twee statistische methodes gebruiken om  $\text{Var}(S_Y(n)|S_X(n) = x)$  empirisch te bepalen ter controle van vergelijking (5.8).

### 7.1 Simple sampling

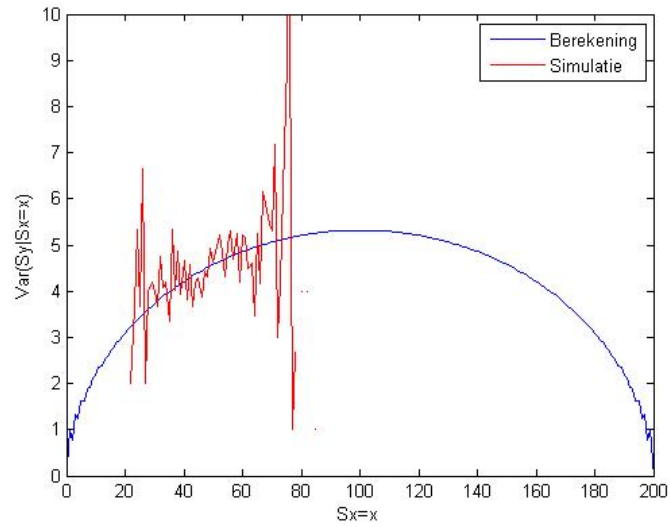
Laten we eerst eens beginnen met simple sampling. We hebben in Hoofdstuk 4 een aantal pluimen gesimuleerd, waaruit we  $\text{Var}(S_Y(n)|S_X(n) = x)$  kunnen bepalen. Hiertoe moeten we eerst conditioneren naar  $x$ , dus we moeten voor een bepaalde  $x$  alle deeltjes bepalen die zich na  $n$  tijdstappen op deze  $x$ -positie bevinden. Vervolgens willen we de variantie van deze deeltjes bepalen, volgens de steekproefvariantie

$$\frac{1}{N_x - 1} \sum_{i \in \mathcal{N}_x} (y_i - \bar{y})^2 \approx \frac{1}{N_x} \sum_{i \in \mathcal{N}_x} y_i^2. \quad (7.1)$$

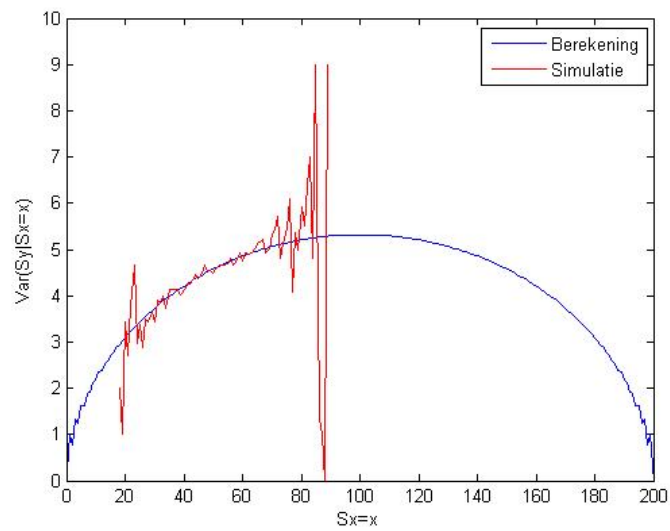
Waarbij de benadering geldt wegens de wet van de grote aantallen,  $\bar{y} \approx \text{E}[S_Y(n)|S_X(n) = x] = 0$ . We zullen eerst eens kijken naar een uitkomst voor  $a = b = 0.5$  met  $N = 10\,000$ .

Wat meteen opvalt aan de empirische variantie in figuur 7.1 is het grote aantal uitschieters, vooral tussen  $x = 20$  en  $x = 30$  en tussen  $x = 60$  en  $x = 80$ . We hebben eerder al vastgesteld dat het merendeel van de deeltjes tussen  $x = 40$  en  $x = 60$  zitten, hierbuiten zijn er blijkbaar te weinig deeltjes om een fatsoenlijke schatting te maken van de variantie. Wat we kunnen doen is het aantal deeltjes in de simulatie verhogen, we moeten echter niet overmoedig worden, want Matlab kan geen enorme vectoren aan. Voor  $N = 250\,000$  gaat het nog net goed en wordt de uitkomst al een stuk netter.

Figuur 7.2 geeft al een veel beter beeld van de kwaliteit van de schatting van  $\text{Var}(S_Y(n)|S_X(n) = x)$  en doet ons vermoeden dat de door ons berekende variantie juist is. Willen we echter ook voor de  $x$ -posities tussen 0 en 30 en tussen 70 en 100 een fatsoenlijke schatting, dan hebben we aan de hier gebruikte methode niet genoeg, aangezien hier in onze simulatie geen deeltjes terecht komen.



Figuur 7.1: De eerder berekende variantie en de empirische variantie van de simulatie, voor  $a = b = 0.5$  en  $N = 10\,000$



Figuur 7.2: De eerder berekende variantie en de empirische variantie van de simulatie, voor  $a = b = 0.5$  en  $N = 250\,000$

## 7.2 Importance sampling

Het probleem dat we hier hebben met het schatten van de voorwaardelijke variantie, leent zich uitstekend voor importance sampling. Het idee van importance sampling is dat je de verdeling verandert, zodat je meer metingen krijg in het gebied dat je het meest interesseert. Je moet dan natuurlijk achteraf weer compenseren voor het feit dat je eigenlijk de verkeerde verdeling hebt gebruikt. Het probleem dat we hier hebben is dat de kans dat er een deeltje tussen  $x = 0$  en  $x = 30$  of tussen  $x = 70$  en  $x = 100$  terechtkomt te klein is, waardoor we geen fatsoenlijke schatting kunnen maken van de variantie in de  $y$ -richting. Doormiddel van importance sampling willen we hier verandering in brengen.

### 7.2.1 Hoe importance sampling werkt

Importance sampling kan toegepast worden op Monte Carlo simulaties. Een Monte Carlo simulatie is opgebouwd uit allemaal kleine onafhankelijke simulaties die dezelfde verdeling hebben en tesamen de uitkomst vormen van de uiteindelijke simulatie. Wat wij doen met het simuleren van  $N$  onafhankelijke vervuulingsdeeltjes, die dezelfde verdeling hebben, is dus een Monte Carlo simulatie.

In [1] kunnen we precies lezen hoe importance sampling werkt, we lichten een deel nog even uit. Stel  $f$  is een kansdichtheidsfunctie die waarden aanneemt op  $\mathcal{S} \subset \mathbb{R}^d$  en we willen een schatting maken van

$$I = \int_{\mathcal{S}} k(\vec{x}) f(\vec{x}) d\vec{x} = \mathbb{E}[k(\vec{X})],$$

waarbij  $\vec{X} \sim f$ .

We kunnen dan een andere (strikt positieve) kansdichtheidsfunctie  $\phi$  nemen op  $\mathcal{S}$  en dan geldt dat

$$I = \int_{\mathcal{S}} \left( \frac{k(\vec{x}) f(\vec{x})}{\phi(\vec{x})} \right) \phi(\vec{x}) d\vec{x} = \mathbb{E} \left[ \frac{k(\vec{Y}) f(\vec{Y})}{\phi(\vec{Y})} \right],$$

waarbij  $\vec{Y}$  nu verdeeld is volgens  $\phi$ .

We kunnen op deze manier simuleren volgens de kansdichtheidsfunctie  $\phi$ , die we natuurlijk zo kiezen dat we meer trekkingen krijgen op plekken die voor ons interessant zijn, maar die door simulatie met kansdichtheidsfunctie  $f$  onderbelicht blijven. Echter, wat we dan als uitkomst krijgen, is  $\mathbb{E}[k(\vec{Y})]$ , dit moeten we dan nog wel vermenigvuldigen met de factor  $\frac{f(\vec{Y})}{\phi(\vec{Y})}$ , waarbij we voor  $f$  en  $\phi$  de theoretische kansdichtheidsfuncties gebruiken. Onze uiteindelijke uitkomst is dan weer  $I = \mathbb{E} \left[ \frac{k(\vec{Y}) f(\vec{Y})}{\phi(\vec{Y})} \right] = \mathbb{E}[k(\vec{X})]$ , maar met een andere variantie op de gebieden waarin we geïnteresseerd zijn. Wat we natuurlijk willen is dat deze variantie kleiner is dan de variantie van de oorspronkelijke simulatie.

### 7.2.2 Toepassen van importance sampling op de empirische variantie

Nu we weten hoe importance sampling werkt en wanneer we het kunnen toepassen, moeten we eerst bedenken hoe we dit wapen in gaan zetten. Importance sampling staat erom bekend dat het een efficiënte manier kan zijn om de variantie in schattingen te verkleinen, maar als het verkeerd gebruikt wordt, kan het juist een averechts effect hebben.

Laten we ons eerst eens concentreren op het gebied tussen  $x = 0$  en  $x = 30$ . Als we voor deze posities een (fatsoenlijke) schatting willen maken van  $\text{Var}(S_Y(n) | S_X(n) = x)$ , dan zullen we er voor moeten zorgen dat hier veel deeltjes terecht komen. De vraag is dus hoe we onze Monte Carlo simulatie zo zullen aanpassen dat dit het geval is. Een eerste mogelijkheid is om ons model zo aan te passen dat vrije deeltjes met een grotere kans een stap naar links maken en met kleinere

kans een stap naar rechts, op die manier zullen deeltjes minder ver komen. Een tweede mogelijkheid hebben we eigenlijk al gezien als we naar het geval  $a = 0.1$ ,  $b = 0.9$  kijken, als we zorgen dat deeltjes met een kleinere kans vrij zijn, zullen ze ook minder ver komen. Zijn beide mogelijkheden even goed? We moeten natuurlijk niet vergeten dat we na simulatie moeten compenseren voor het modelleren met de verkeerde kansdichtheidsfunctie door te vermenigvuldigen met de theoretische factor  $\frac{f(\vec{Y})}{\phi(\vec{Y})}$ . In het eerste geval is dit

$$\frac{f(\tilde{x}, y, k)}{\phi(\tilde{x}, y, k)} = \frac{P(S_Y(n) = y | S_X(n) = \tilde{x})}{P(S_Y(n) = y | \tilde{S}_X(n) = \tilde{x})},$$

waarbij  $\tilde{S}_X(n)$  verdeeld is volgens het model waarin een vrij deeltje een grotere kans heeft zich naar links te bewegen dan naar rechts. Maar  $P(S_Y(n) = y | S_X(n) = x)$  is juist de term waarvan we willen controleren of hij klopt! Dus naast dat we bij deze methode de term  $P(S_Y(n) = y | \tilde{S}_X(n) = x)$  nog moeten bepalen, schieten we er niks mee op voor de controle van vergelijking (5.8).

Zou het tweede idee, met de verandering van de verdeling van  $K_n$  beter werken? In dit geval gaan we over van een stochast  $\vec{X} = (S_X(n), S_Y(n), K_n)$  met  $K_n \sim \text{Bin}(n, a)$  naar de nieuwe stochast  $\vec{Y} = (S_X(n), S_Y(n), \tilde{K}_n)$ , waarbij  $\tilde{K}_n$  de nieuwe verdeling is van het aantal keren dat een deeltje vrij is. Dan geldt

$$\frac{f(x, y, \tilde{k})}{\phi(x, y, \tilde{k})} = \frac{P(K_n = \tilde{k})}{P(\tilde{K}_n = \tilde{k})},$$

waarbij  $\tilde{k} = 0, 1, \dots, n$ . Deze term kunnen we wel gewoon bepalen zonder gebruik te maken van de formule die we juist willen nagaan. We moeten wel opletten, want  $\text{Var}(S_Y(n) | S_X(n) = x)$  bestaat uit twee componenten, een teller en een noemer, die elk van  $K_n$  afhangen. We moeten dus allebei compenseren met de factor  $\frac{f(\vec{Y})}{\phi(\vec{Y})}$ . We hebben dan de teller waarbij  $k(\vec{Y})$  gedefinieerd is als

$$k(x, y, \tilde{k}) = \sum_k \sum_y y^2 P(S_Y(n) = y, S_X(n) = x | \tilde{K}_n = k) P(\tilde{K}_n = k)$$

en de noemer waarvoor

$$k(x, y, \tilde{k}) = \sum_k P(S_X(n) | \tilde{K}_n = k) P(\tilde{K}_n = k).$$

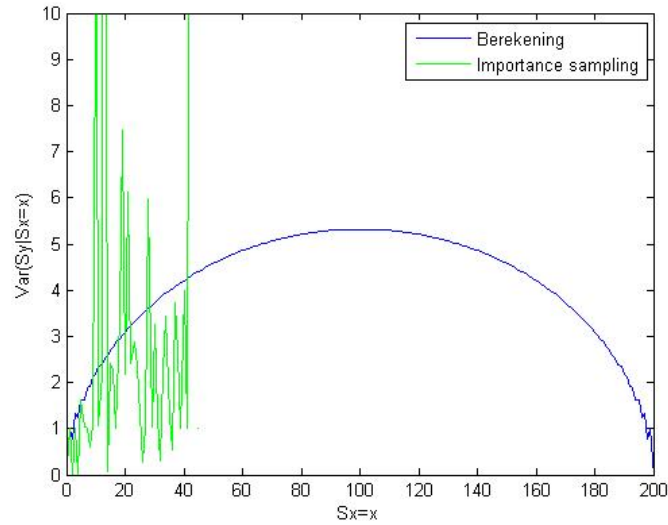
Als we voor  $\phi$  ook een binomiale verdeling nemen, maar met  $\tilde{a}$  zodanig dat de deeltjes terecht zullen komen tussen  $x = 0$  en  $x = 30$ , dan is  $\tilde{K}_n \sim \text{Bin}(n, \tilde{a})$  en

$$\frac{f(x, y, \tilde{k})}{\phi(x, y, \tilde{k})} = \frac{P(K_n = \tilde{k})}{P(\tilde{K}_n = \tilde{k})} = \frac{a^{\tilde{k}}(1-a)^{n-\tilde{k}} \binom{n}{\tilde{k}}}{\tilde{a}^{\tilde{k}}(1-\tilde{a})^{n-\tilde{k}} \binom{n}{\tilde{k}}} = \frac{a^{\tilde{k}}(1-a)^{n-\tilde{k}}}{\tilde{a}^{\tilde{k}}(1-\tilde{a})^{n-\tilde{k}}}.$$

We kunnen nu de deeltjes simuleren met de nieuwe  $\tilde{K}_n$ . Vervolgens moeten we  $P(\tilde{K}_n = \tilde{k})$  corrigeren met de factor  $\frac{P(K_n = \tilde{k})}{P(\tilde{K}_n = \tilde{k})}$ . We krijgen dan

$$\text{Var}(S_Y(n) | S_X(n) = x) = \frac{\sum_{\tilde{k}} \sum_y y^2 P(S_Y(n) = y, S_X(n) = x | \tilde{K}_n = \tilde{k}) P(\tilde{K}_n = \tilde{k}) \frac{a^{\tilde{k}}(1-a)^{n-\tilde{k}}}{\tilde{a}^{\tilde{k}}(1-\tilde{a})^{n-\tilde{k}}}}{\sum_{\tilde{k}} P(S_X(n) | \tilde{K}_n = \tilde{k}) P(\tilde{K}_n = \tilde{k}) \frac{a^{\tilde{k}}(1-a)^{n-\tilde{k}}}{\tilde{a}^{\tilde{k}}(1-\tilde{a})^{n-\tilde{k}}}}. \quad (7.2)$$

Als we nu  $\tilde{a} = 0.15$  nemen, dan ligt de verwachte x-positie van de deeltjes op 15, dus dan zouden we een goede schatting moeten kunnen krijgen voor deeltjes rond  $x = 15$ . We laten Matlab dus eerst voor alle deeltjes  $\tilde{K}_n$  bepalen, die binomiaal verdeeld is met succeskans  $\tilde{a} = 0.15$ . Vervolgens bepalen we met behulp van random waarden wat de deeltjes aan dispersie hebben uitgevoerd en we tellen de advection er bij. Nu we een wolk aan deeltjes hebben volgens de nieuwe verdeling, kunnen we hiervoor per x-positie  $\text{Var}(S_Y(n)|S_X(n) = x)$  bepalen waarbij we volgens formule (7.2) vermenigvuldigen met de factor  $\frac{P(K_n = \tilde{k})}{P(\tilde{K}_n = \tilde{k})}$ . We krijgen dan de uitkomst van figuur 7.3.



Figuur 7.3: De eerder berekende variantie en importance sampling voor  $a = b = 0.5$  met  $\tilde{a} = 0.15$  en  $N = 250\,000$ .

De uitkomst in figuur 7.3 ziet er niet zo goed uit, de empirische variantie is in het algemeen te laag en heeft enorme uitschieters. Dit kan twee dingen betekenen: de formule die we bepaald hebben voor  $\text{Var}(S_Y(n)|S_X(n) = x)$  is niet goed, of er gaat iets mis bij de importance sampling.

### 7.2.3 De variantie van importance sampling voor een binomiale verdeling

Importance sampling staat erom bekend dat het de variantie van een simulatie flink kan verbeteren, maar ook verslechteren. We gaan in een voorbeeld bekijken wat het effect kan zijn van het verschuiven van een kansverdeling op de variantie van een schatter voor een parameter. We bekijken een  $\text{Bin}(n, \tilde{a})$  verdeling  $\tilde{K}_n$  waarvan we de verwachting  $I = \mathbb{E}[\tilde{K}_n] = n\theta$  willen schatten. We beschikken over trekkingen uit een  $\text{Bin}(n, a)$  verdeling  $K_n$ . De vergelijkingen uit 7.2.1 komen er op neer dat we dan de verwachting van een stochast  $W$  schatten, gegeven door

$$W = V \left( \frac{\tilde{a}}{a} \right)^V \left( \frac{1-\tilde{a}}{1-a} \right)^{n-V}.$$

Ter controle rekenen we even  $\mathbb{E}[W]$  uit:

$$\begin{aligned} \sum_w w P(W = w) &= \sum_{v=0}^n v \left( \frac{\tilde{a}}{a} \right)^v \left( \frac{1-\tilde{a}}{1-a} \right)^{n-v} P(V = v) \\ &= \sum_{v=0}^n v \left( \frac{\tilde{a}}{a} \right)^v \left( \frac{1-\tilde{a}}{1-a} \right)^{n-v} a^v (1-a)^{n-v} \binom{n}{v} \\ &= \sum_{v=0}^n v \tilde{a}^v (1-\tilde{a})^{n-v} \binom{n}{v} \\ &= n\tilde{a} \end{aligned}$$

De kwaliteit van de schatter  $W$  wordt bepaald door zijn variantie  $\text{Var}(W) = \mathbb{E}[W^2] - (\mathbb{E}[W])^2$ .

$$\begin{aligned} \mathbb{E}[W^2] &= \mathbb{E} \left[ V^2 \left( \frac{\tilde{a}}{a} \right)^{2V} \left( \frac{1-\tilde{a}}{1-a} \right)^{2n-2V} \right] \\ &= \sum_{h=0}^n h^2 \left( \frac{\tilde{a}^2}{a^2} \right)^h \left( \frac{(1-\tilde{a})^2}{(1-a)^2} \right)^{n-h} P(V = h) \\ &= \sum_{h=0}^n h^2 \left( \frac{\tilde{a}^2}{a^2} \right)^h \left( \frac{(1-\tilde{a})^2}{(1-a)^2} \right)^{n-h} a^h (1-a)^{n-h} \binom{n}{h} \\ &= \sum_{h=0}^n h^2 \binom{n}{h} \left( \frac{\tilde{a}^2}{a} \right)^h \left( \frac{(1-\tilde{a})^2}{1-a} \right)^{n-h} \\ &:= \sum_{h=0}^n h^2 \binom{n}{h} p^h q^{n-h} \end{aligned}$$

Voor  $p = \frac{\tilde{a}^2}{a}$  en  $q = \frac{(1-\tilde{a})^2}{1-a}$ . Deze machtreeks is om te schrijven zoals beschreven in bijlage I tot

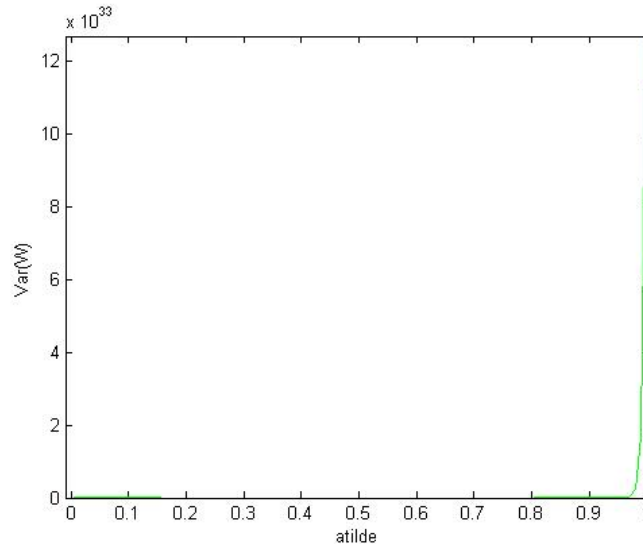
$$\mathbb{E}[W^2] = (p+q)^{n-2} (pqn + p^2n^2). \quad (7.3)$$

Omdat  $\mathbb{E}[W] = n\tilde{a}$ , is

$$\text{Var}(W) = (p+q)^{n-2} (pqn + p^2n^2) - (n\tilde{a})^2. \quad (7.4)$$

Als we bijvoorbeeld  $a = \frac{1}{2}$  nemen, dan is  $p = 2\tilde{a}^2$  en  $q = \frac{(1-\tilde{a})^2}{1-a} = 2(1-\tilde{a})^2$ . We krijgen dan de volgende uitdrukking voor  $\text{Var}(W)$ ;

$$\text{Var}(W) = 8(2\tilde{a}^2 + 1 - 2\tilde{a})^{n-2} \tilde{a}^2 (n - 2\tilde{a}n + \tilde{a}^2n + \tilde{a}^2n^2) - n^2\tilde{a}^2. \quad (7.5)$$



Figuur 7.4:  $\text{Var}(W)$  voor  $n = 100$  en  $a = 0.5$  met  $\tilde{a}$  van 0 tot 1.

Een plot van  $\text{Var}(W)$  voor  $n = 100$  is te zien in figuur 7.4.

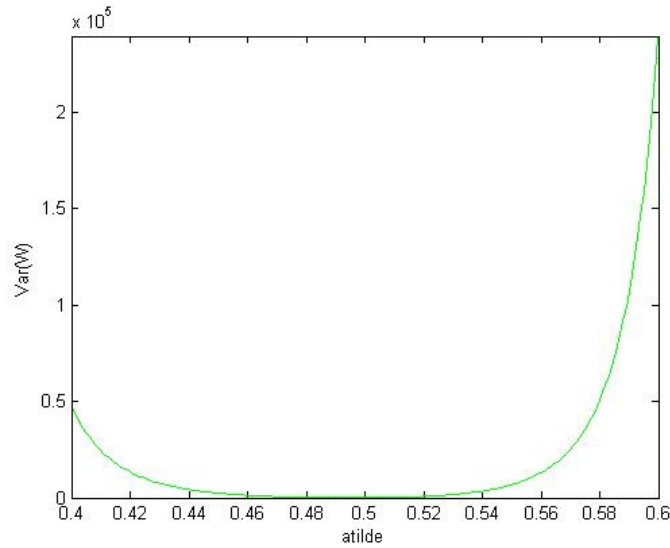
In de figuur kunnen we weinig zien van het verloop van de variantie, omdat er een piek zit bij  $\tilde{a} = 1$ . Wat echter wel opvalt is hoe enorm  $\text{Var}(W)$  kan zijn. Laten we ook even een plot bekijken voor  $\tilde{a}$  in de buurt van 0.5, deze is te vinden in figuur 7.5.

Uit figuur 7.5 worden we al wijzer. We zien dat de variantie enorm toeneemt rondom  $\tilde{a} = 0.5$ . We moeten echter wel er om denken dat dit de variantie is voor één realisatie, we kunnen dit dus nog delen door  $N$ , het aantal realisaties dat we gebruiken. Hoe groot de variantie van  $W$  is voor  $\tilde{a} = 0.15$  kunnen we berekenen door dit in te vullen in formule (7.5), we vinden dan dat  $\text{Var}(W) \approx 2.51 \cdot 10^{18}$ . Dit getal kunnen we nog delen door het aantal deeltjes dat we gebruikt hebben,  $N = 250\,000$ . We krijgen dan nog steeds  $\text{Var}(W) \approx 10^{13}$ . De variantie van ons antwoord in figuur 7.3 is dus enorm! Dit verklaart waarom we een realisatie kregen die helemaal niet in de buurt ligt van de berekende  $\text{Var}(S_Y(n)|S_X(n) = x)$ . Blijkbaar kunnen we alleen goede realisaties krijgen van de importance sampling als  $\tilde{a}$  niet te ver weg ligt van de verwachte x-positie van de originele pluim,  $E[S_X(n)]$ .

### 7.3 Verschillende simulaties

We hebben gezien dat we niet moeten verwachten dat we een importance sampling simulatie een schatting kunnen maken van  $\text{Var}(S_Y(n)|S_X(n) = x)$  voor alle  $x$  tussen 0 en 30. Dit omdat de importance sampling alleen een goede schatting geeft voor waarden rondom de verwachte x-positie,  $na$ . Daarbij is er nog een eigenschap van de importance sampling simulatie waarmee we te maken hebben. Er geldt namelijk dat importance sampling simulaties alleen een goede waarde geven dichtbij x-positie  $n\tilde{a}$ . Daarbuiten worden de schattingen snel slechter.

Wat we dus nog kunnen doen is de x-as opdelen in kleine intervalletjes en voor elk intervalletje een schatting maken van de variantie in de y-richting. We moeten dan zorgen dat de verwachte x-positie van de deeltjes middenin het intervalletje zit. Laten we bijvoorbeeld intervalletjes nemen van lengte 5, dan moeten we voor het eerste intervalletje (van 1 tot 5)  $\tilde{a}$  gelijk nemen aan  $\frac{3}{n} = \frac{3}{100}$ , zodat  $E[\tilde{S}_X(n)] = n\tilde{a} = 3$ . In figuur 7.6 is dit gedaan voor  $a = b = 0.5$ , met 100 000 deeltjes. Simulatie met 250 000 deeltjes zou wel mogelijk zijn, maar omdat we de importance sampling



Figuur 7.5:  $\text{Var}(W)$  voor  $n = 100$  en  $a = 0.5$  met  $\tilde{a}$  van 0.4 tot 0.6.

simulatie nu 20 keer moeten doen (voor elk interval een keer), zou dit erg veel tijd gaan kosten. We moeten echter niet verwachten dat we voor alle  $x$ -posities een goede uitkomst krijgen, want we hebben nog steeds te maken met de enorme variantie die we in de vorige paragraaf hebben afgeleid.

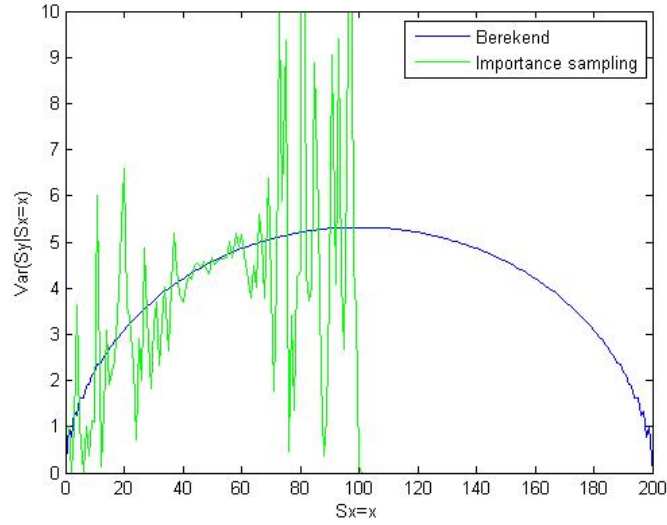
Het is nu wel gelukt om voor het hele interval  $0 \leq x \leq 100$  een simulatie te krijgen. We zien wel dat de importance sampling simulatie nog steeds erg veel uitschieters heeft, vooral voor  $x > 65$ . Dat de uitschieters aan de rechter kant van  $x = 50$  groter zijn dan aan de linker kant, hadden we al kunnen voorspellen als we naar figuur 7.5 kijken. Hier zien we namelijk dat de variantie van de importance sampling schatter  $W$  veel sneller groeit voor  $\tilde{a} > 0.5$  dan voor  $\tilde{a} < 0.5$ .

Al met al kunnen we zeggen dat de berekende formule voor  $\text{Var}(S_Y(n)|S_X(n) = x)$  niet zozeer wordt ondersteund door de importance sampling simulaties, maar er is ook niet aangetoond dat de formule onjuist is. We zullen nog even kijken naar de uitkomsten voor de andere combinaties van  $a$  en  $b$  die hebben behandeld, voor  $p_f(0) = \frac{a}{a+b}$ . Deze staan in de figuren 7.7 tot en met 7.12.

Voor de gevallen met  $a + b = 1$  is het voldoende om de variantie tot en met  $n$  te bepalen, omdat we al een symmetrie hebben bewezen die vervolgens de waarden voor  $n < x \leq 2n$  geeft. Dit komt goed uit, want we hadden afgeleid dat de variantie aan de rechter kant van het centrum van de pluim snel groter wordt. We zouden dus nooit een fatsoenlijke schatting krijgen voor zulke grote  $x$ .

Voor  $a = 0.1$ ,  $b = 0.9$  zien we dat de importance sampling schatting al snel een te grote afwijking geeft. Dit hebben we ook gezien voor  $a = b = 0.5$ , rechts van het centrum van de pluim bleek de importance sampling schatting erg snel op te blazen. Echter, voor de  $x$ -posities waar de pluim zich daadwerkelijk bevindt, kloppen zowel de simple sampling schatter als de importance sampling schatter heel goed. Dit is gedeeltelijk te danken aan het feit dat de deeltjes erg dicht bij elkaar liggen, waardoor er genoeg deeltjes op een bepaalde  $x$ -positie zijn om een goede schatting te maken.

Voor  $a = 0.9$ ,  $b = 0.1$  geeft de simple sampling schatting een goede waarde voor  $60 \leq x \leq 120$ , die hier de berekende variantie goed benadert. De importance sampling schatter geeft alleen een goede waarde voor  $x$  dicht bij 90, dus rond de kern van de pluim. Maar op dit punt is de importance sampling schatting eigenlijk een simple sampling schatter, want hier is  $\tilde{a} \approx a$ ! Hoe het kan dat de



Figuur 7.6: De berekende en gesimuleerde variantie met importance sampling op deelintervallen van lengte 5, voor  $a = b = 0.5$ .

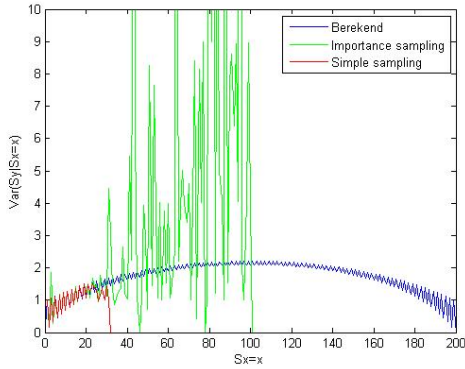
importance sampling voor de andere  $x$  zo'n slechte benadering geeft, kunnen we met onze kennis tot nu toe niet zeggen. Wat we wel weten is dat we voor de waarden ver weg van  $x = 90$  een grote variantie kunnen verwachten. Wellicht zorgt de grote verspreiding van de deeltjes voor een te grote afwijking.

Voor  $a = b = 0.1$  en  $a = b = 0.01$  zien we dat de simple sampling schatter in een veel groter gebied de variantie kan schatten dan voor de andere drie gevallen. Dit is omdat de deeltjes over een veel groter domein verspreid zijn, namelijk helemaal van  $x$ -positie 0 tot 100 (voor  $a = b = 0.01$  zelfs tot 130). Het goede nieuws is dat de simple sampling schatter erg mooi overeenkomt met de berekende curve. We kunnen hier tevens mooi zien dat de importance sampling schatter wel degelijk een goede schatting kan geven, want voor  $0 \leq x \leq 80$  komt deze erg mooi overeen met de simple sampling schatter. Maar hoe komt het dat we geen schattingen hebben voor hogere  $x$ ? We hebben hier niet te maken met een symmetrie die de variantie rechts van  $x = n$  kan bepalen, dus hiervan zullen we apart een schatting moeten maken. Voor de simple sampling schatter is de afwezigheid van een schatting voor hogere  $x$  omdat hier gewoonweg geen deeltjes zijn gekomen bij de simulatie. Maar voor de importance sampling schatter is er iets anders aan de hand. Hierbij hebben we de verdeling van  $K_n$  veranderd om op alle gebieden deeltjes te krijgen. Voor  $x > n$  zouden we de verdeling van  $K_n$  zodanig moeten maken dat deeltjes meer dan  $n$  keer vrij zijn. Dat op zich leidt niet tot problemen, maar als we willen corrigeren met de factor

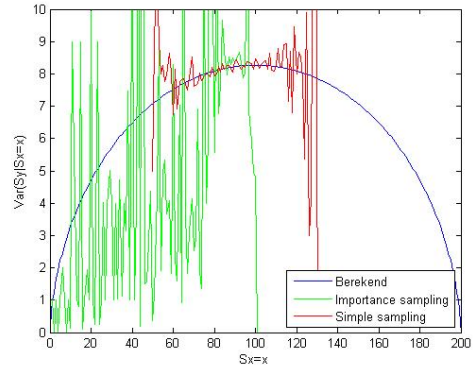
$$\frac{f(x, y, \tilde{k})}{\phi(x, y, \tilde{k})} = \frac{P(K_n = \tilde{k})}{P(\tilde{K}_n = \tilde{k})}$$

gaat het mis, aangezien  $\phi(\vec{Y}) = P(\tilde{K}_n = \tilde{k}) = 0$ . In ons oorspronkelijke model kan een deeltje namelijk niet meer dan  $n$  keer vrij zijn. We kunnen dus op deze manier geen schattingen maken voor  $x > n$ .

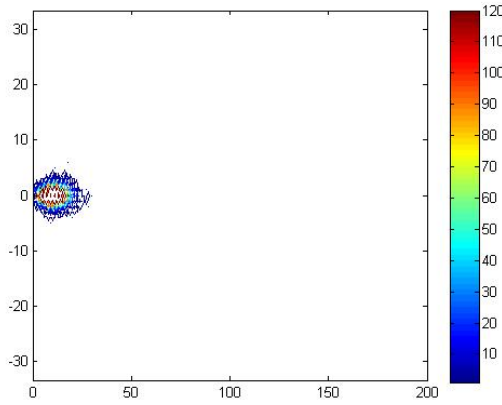
We kunnen wel kijken of de berekende variantie voor  $x > n$  klopt, door ons te realiseren dat deeltjes die zo ver komen (bijna) altijd vrij zijn geweest. De variantie van deze deeltjes komt in de buurt van de variantie voor  $a = 1, b = 0$ . Laten we dus de berekende varianties nog even



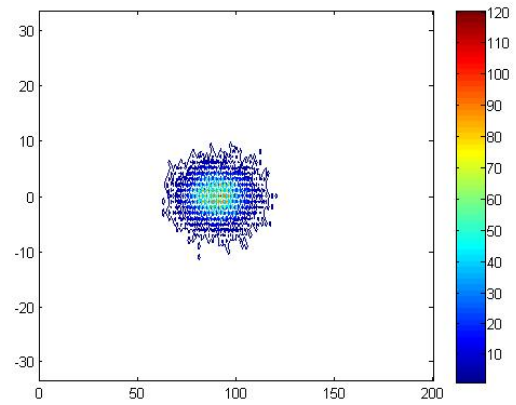
Figuur 7.7:  $\text{Var}(S_Y(n)|S_X(n) = x)$  voor  $a = 0.1$ ,  $b = 0.9$ ,  $n = 100$  en  $N = 250\,000$



Figuur 7.8:  $\text{Var}(S_Y(n)|S_X(n) = x)$  voor  $a = 0.9$ ,  $b = 0.1$ ,  $n = 100$  en  $N = 250\,000$



Figuur 7.9: De pluim van  $a = 0.1$ ,  $b = 0.9$ ,  $n = 100$  en  $N = 10\,000$



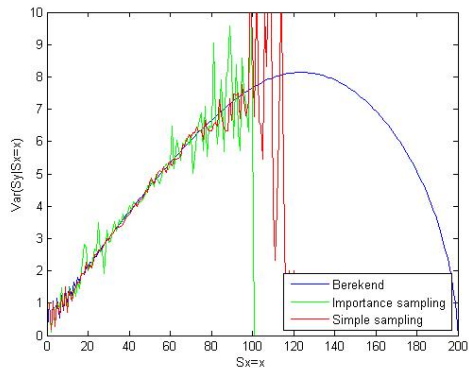
Figuur 7.10: De pluim van  $a = 0.9$ ,  $b = 0.1$ ,  $n = 100$  en  $N = 10\,000$

vergelijken met die voor de random walk. In figuren 7.15 en 7.16 staan ze beide.

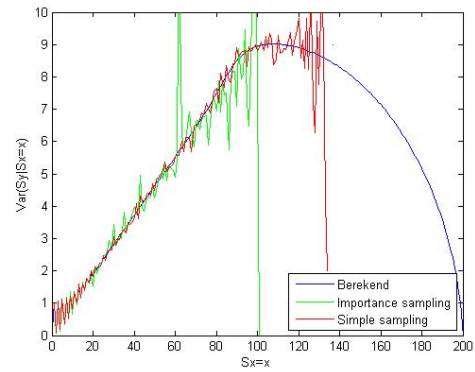
Vooral voor  $a = b = 0.01$  zien we een mooie overeenkomst voor  $x > n$ , dat komt doordat er veel deeltjes zijn die alle  $n$  keer vrij zijn geweest, zoals we ook kunnen zien in figuur 5.22. Deze deeltjes hebben precies dezelfde variantie als de deeltjes voor  $a = 1$ ,  $b = 0$ .

Voor  $a = b = 0.1$  zien we een groter verschil in de twee varianties. Dit is te verklaren doordat de deeltjes voor  $a = b = 0.1$  een grotere kans hebben om een keer geabsorbeerd te worden, dus er zullen minder deeltjes alle  $n$  de keren vrij zijn. En deeltjes die minder dan  $n$  keer vrij zijn geweest, hebben minder in de  $y$ -richting kunnen bewegen en zullen dus ook een kleinere bijdrage leveren aan de variantie.

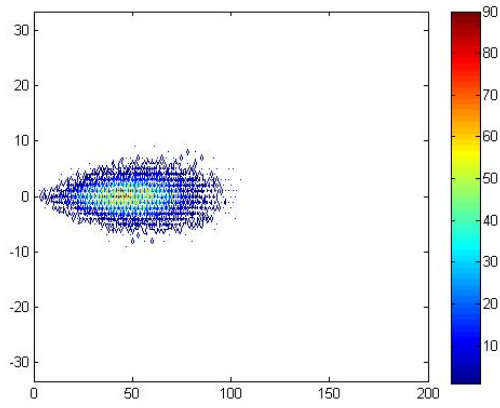
Al met al kunnen we toch wel stellen dat de berekende varianties lijken te kloppen. Niet alle uitkomsten van de statistische methodes ondersteunen de berekende varianties even goed, maar geen van allen verwerpen de berekening. Wel zou hier wellicht nog meer onderzoek naar kunnen worden gedaan.



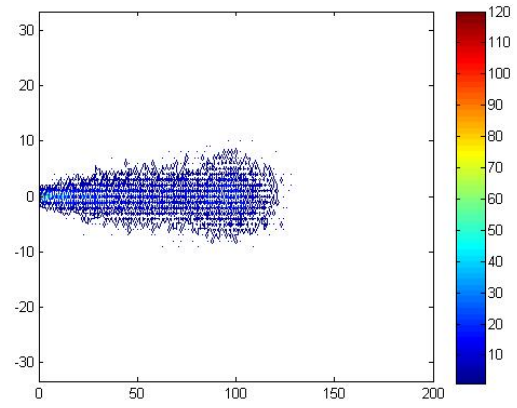
Figuur 7.11:  $\text{Var}(S_Y(n)|S_X(n) = x)$  voor  $a = 0.1$ ,  $b = 0.1$ ,  $n = 100$  en  $N = 250\,000$



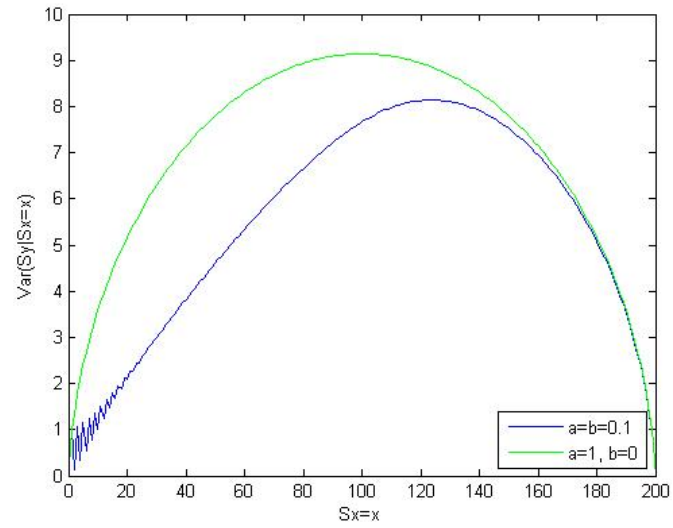
Figuur 7.12:  $\text{Var}(S_Y(n)|S_X(n) = x)$  voor  $a = 0.01$ ,  $b = 0.11$ ,  $n = 100$  en  $N = 250\,000$



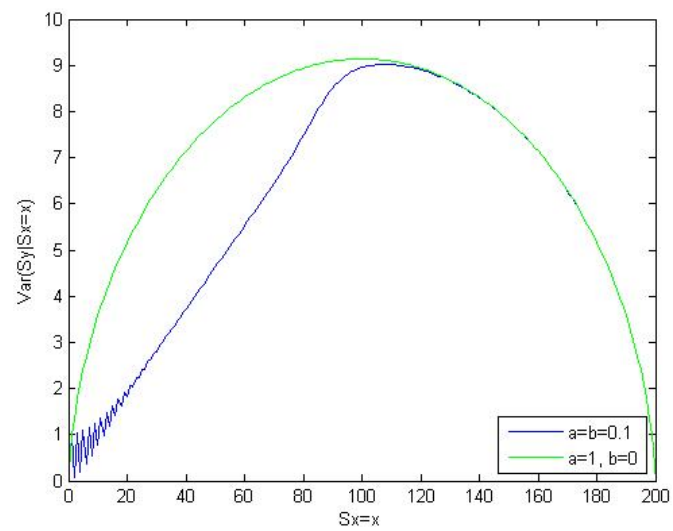
Figuur 7.13: De pluim van  $a = 0.1$ ,  $b = 0.1$ ,  $n = 100$  en  $N = 10\,000$



Figuur 7.14: De pluim van  $a = 0.01$ ,  $b = 0.01$ ,  $n = 100$  en  $N = 10\,000$



Figuur 7.15: De berekende varianties voor  $a = b = 0.1$  en  $a = 1, b = 0$ .



Figuur 7.16: De berekende varianties voor  $a = b = 0.01$  en  $a = 1, b = 0$ .

## Hoofdstuk 8

# Een andere maat voor de verspreiding van deeltjes

In hoofdstuk 5 hebben we iets vreemds opgemerkt aan  $\text{Var}(S_Y(n)|S_X(n) = x)$ , namelijk dat deze ook een significante waarde aanneemt op plaatsen waar zich in het algemeen helemaal geen vervuiliingsdeeltjes bevinden. We hebben al beredeneerd hoe dit komt, namelijk doordat we delen door de factor  $P(S_X(n) = x)$ , die op deze plekken heel klein is. In dit hoofdstuk zullen we een manier bekijken om hiervoor te compenseren.

### 8.1 Vermenigvuldiging met $P(S_X(n) = x)$

We willen dus graag  $\text{Var}(S_Y(n)|S_X(n) = x)$  compenseren voor het feit dat hij ook waarden aanneemt op punten waar geen deeltjes komen. Het doel hiervan is dat we graag een maat willen hebben voor de verspreiding van de vervuiliingsdeeltjes waaruit we de vorm van de pluim weer kunnen afleiden. Aangezien de totale variantie in de y-richting wordt gegeven door

$$\text{Var}(S_Y(n)) = \sum_x \text{Var}(S_Y(n)|S_X(n) = x) P(S_X(n) = x),$$

komen we op het idee om  $\text{Var}(S_Y(n)|S_X(n) = x)$  te vermenigvuldigen met de term  $P(S_X(n) = x)$ . We weten dan ook zeker dat de term geen grote waarde zal aannemen op plaatsen waar deeltjes slechts met een kleine kans zijn. Door vervolgens de wortel te nemen, krijgen we een uitdrukking voor de standaardafwijking, die hopelijk de vorm van de pluim benadert. Merk op dat

$$\begin{aligned} \text{Var}(S_Y(n)|S_X(n) = x) P(S_X(n) = x) &= \sum_y y^2 P(S_Y(n) = y|S_X(n) = x) P(S_X(n) = x) \\ &= \sum_y y^2 P(S_X(n) = x, S_Y(n) = y) \\ &= \sum_k \sum_y y^2 P(S_Y(n) = y, S_X(n) = x|K_n = k) P(K_n = k) \end{aligned}$$

Dit kunnen we voor  $a + b = 1$  uitrekenen met behulp van formules (5.5) en (5.6), we krijgen dan

$$\begin{aligned} \text{Var}(S_Y(n)|S_X(n) = x) P(S_X(n) = x) &= \\ 2 \sum_{k=\lceil \frac{x}{2} \rceil}^n \sum_{y=x \bmod 2, +2, +4, \dots}^{k-|x-k|} y^2 \sum_{i=y}^{k-\frac{x}{2}+\frac{y}{2}-\max(x-k,0)} \beta^{2i-y} \alpha^{k-2i+y} \binom{k}{2i-y} \binom{2i-y}{i} \binom{k-2i+y}{\frac{k-2i+y-|x-k|}{2}} a^k (1-a)^{n-k} \binom{n}{k}. \end{aligned} \quad (8.1)$$

Deze berekening kunnen we gemakkelijk uitvoeren met Matlab, in de figuren 8.1 tot en met 8.5 staat de standaardafwijking vermenigvuldigd met  $\sqrt{P(S_X(n) = x)}$ . In de figuren staat ter vergelijking ook  $P(S_X(n) = x)$ .

We zien dat  $SD_x = \sqrt{\text{Var}(S_Y(n)|S_X(n) = x)P(S_X(n) = x)}$  een veel betere indicatie geeft van de vorm van de pluim dan  $\text{Var}(S_Y(n)|S_X(n) = x)$ . Het is echter wel wat lastiger om te zien of  $SD_x$  symmetrisch is rondom het centrum van de pluim, hiervoor moeten we aandachtig kijken naar de x-posities waarvoor de figuur weer naar 0 gaat. We moeten daarbij wel opmerken dat de figuren welliswaar een betere benadering geven van de vorm van de pluim, maar we kunnen nog steeds de vorm niet afleiden uit de figuur. Zo zouden we naar aanleiding van figuur 8.1 misschien denken dat de pluim zich tussen  $x = 20$  en  $x = 90$  bevindt, maar dit is foutief. Ook in de figuren voor  $a = 0.9, b = 0.1$  en  $a = 0.1, b = 0.9$  zien we dat de term  $P(S_X(n) = x)$  eigenlijk niet sterk genoeg is, waardoor we nog geen goede indicatie krijgen van de vorm.

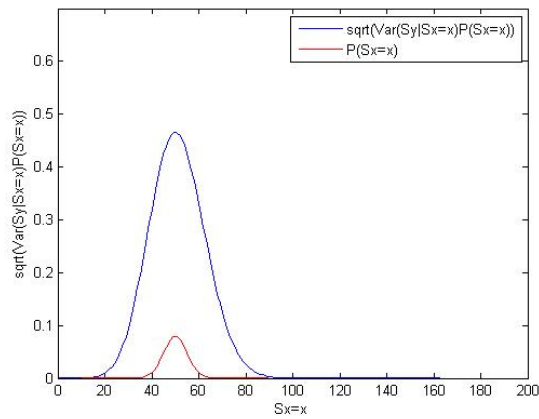
Voor  $a = b = 0.1$  en  $a = b = 0.01$  kunnen we  $SD_x$  ook bepalen, alleen moeten we dan de recursie gebruiken om  $P(K_n = k)$  te bepalen. De uitkomsten hiervan staat in de figuren 8.7 en 8.9.

Voor  $a = b = 0.1$  zien we al wat beter dat de blauwe lijn niet symmetrisch is om het midden van de pluim op  $x = 50$ . Ook zien we tussen  $x = 80$  en  $x = 100$  dat  $SD_x$  veel groter is dan  $P(S_X(n) = x)$ , doordat deeltjes die zo ver zijn gekomen veel vrij zijn geweest en dus veel bijdragen aan de variantie. We kunnen in dit geval de vorm van de pluim wel min of meer afleiden uit de vorm van  $SD_x$ , met x-posities van 0 tot 110 en de grootste spreiding in de y-richting rond x-positie 60.

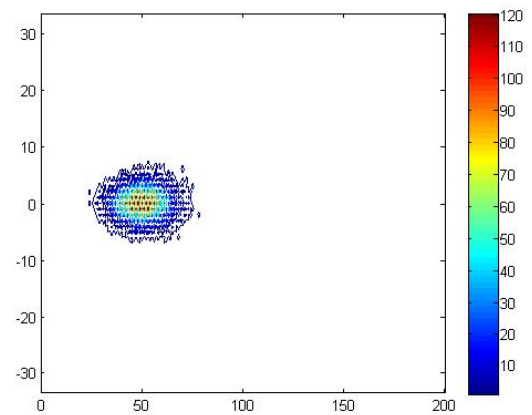
Voor  $a = b = 0.01$  zien we aan de vorm van  $SD_x$  nog wel het beste hoe de vervuulingspluim eruit ziet. We zien aan de grafiek van  $P(S_X(n) = x)$  dat er veel deeltjes in  $(0, 0)$  zijn gebleven, en aan de blauwe grafiek zien we dat er veel deeltjes rondom x-positie 100 zitten. Deze deeltjes hebben een grotere spreiding dan de deeltjes die minder vaak vrij zijn geweest. Ook de spreiding van de deeltjes tussen  $x = 0$  en  $x = 100$  lijkt aardig overeen te komen met de figuur.

Al met al kunnen we opmerken dat  $SD_x$  voor  $a = b = 0.1$  en  $a = b = 0.01$  een redelijke maat geeft voor de vorm van de vervuulingspluim, maar voor de drie gevallen met  $a + b = 1$  laat het nog wat te wensen over. In deze gevallen geeft

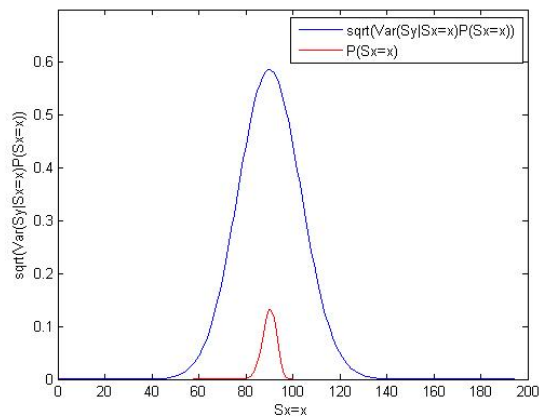
$SD_x$  nog een te grote waarde voor x-posities waar zich praktisch geen deeltjes bevinden.



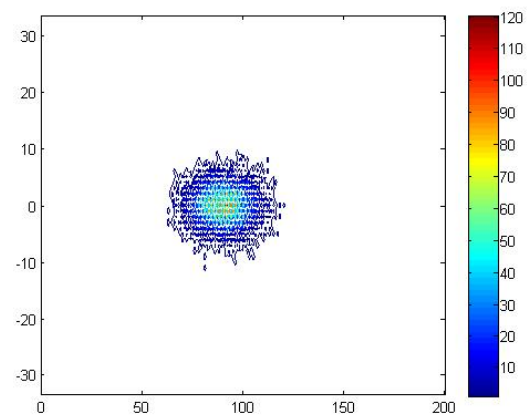
Figuur 8.1:  $SD_x$  en  $P(S_X(n) = x)$  voor  $a = 0.5$ ,  $b = 0.5$



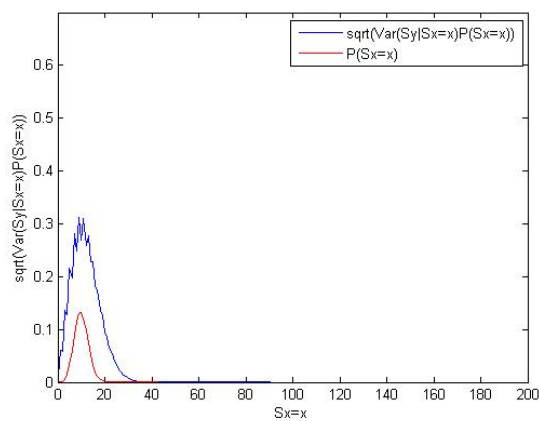
Figuur 8.2: De pluim van  $a = 0.5$ ,  $b = 0.5$



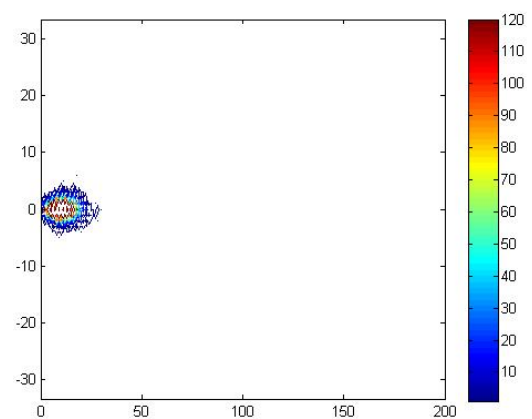
Figuur 8.3:  $SD_x$  en  $P(S_X(n) = x)$  voor  $a = 0.9$ ,  $b = 0.1$



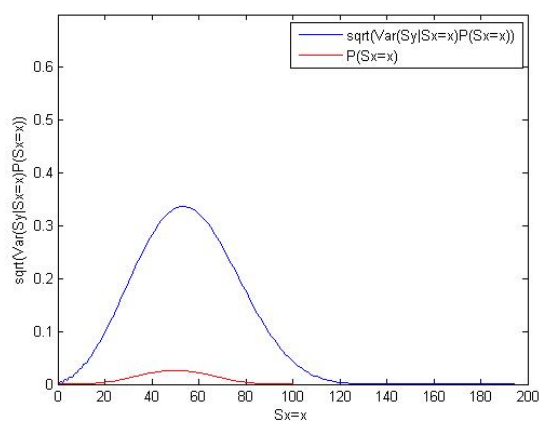
Figuur 8.4: De pluim van  $a = 0.9$ ,  $b = 0.1$



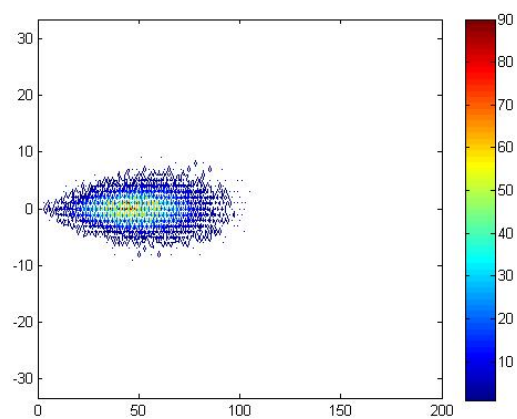
Figuur 8.5:  $SD_x$  en  $P(S_X(n) = x)$  voor  $a = 0.1$ ,  $b = 0.9$



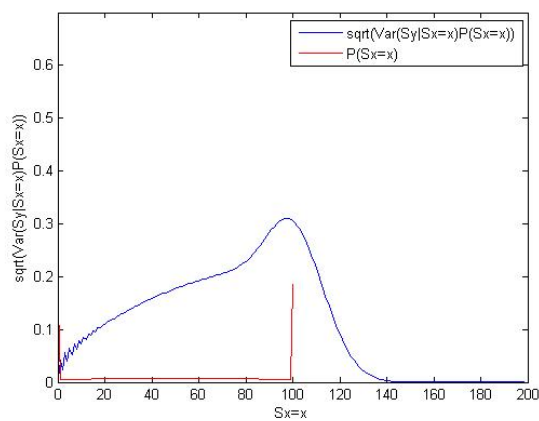
Figuur 8.6: De pluim van  $a = 0.1$ ,  $b = 0.9$



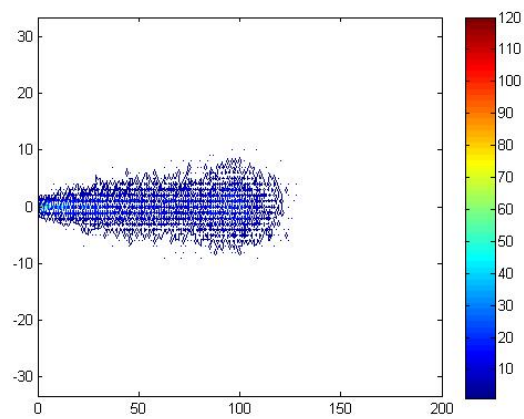
Figuur 8.7:  $SD_x$  en  $P(S_X(n) = x)$  voor  $a = 0.1$ ,  $b = 0.1$



Figuur 8.8: De pluim van  $a = 0.1$ ,  $b = 0.1$



Figuur 8.9:  $SD_x$  en  $P(S_X(n) = x)$  voor  $a = 0.01$ ,  $b = 0.01$



Figuur 8.10: De pluim van  $a = 0.01$ ,  $b = 0.01$

## Hoofdstuk 9

# Samenvatting

We simuleren een vervuilingswolk in de bodem met een groot aantal vervuilingsdeeltjes,  $N$  die onafhankelijk van elkaar op een bepaald tijdstip vrij zijn met kans  $a$  of geadsorbeerd met kans  $b$ . Dit simuleren we door discretisatie van de tijd met overgangskansen volgens een Markov keten. Een vrij deeltje stroomt per tijdstap één stapje mee met de x-richting en hij voert dispersie uit in de x- of y-richting. Hierbij is de standaardafwijking in de x-richting  $\sqrt{10}$  keer zo groot als de dispersie in de y-richting. Een geadsorbeerd deeltje daarentegen beweegt niet.

We kunnen de dispersie op twee manieren simuleren, namelijk door de deeltjes met kans  $\frac{1}{4}$  naar links, rechts, onder of boven de laten bewegen, waarbij de stapjes in de x-richting  $\sqrt{10}$  keer zo groot zijn als de stapjes in de y-richting. De tweede mogelijkheid is om de grootte van de stapjes gelijk te houden, maar de kansen op dispersie in de x- of y-richting te veranderen.

Simulatie van bovenstaande modellen met Matlab laat zien dat de vervuilingspluim er op het oog niet uitziet als een Gaussische patroon, we zien dat de verspreiding in de y-richting aan de rechterkant van de pluim vaak groter is dan aan de linkerkant van de pluim. Dit komt doordat deeltjes die op een verdere x-positie zitten meestal meer vrij zijn geweest dan deeltjes op een lagere x-positie. Hierdoor hebben ze ook meer vrije stapjes gehad om dispersie in de y-richting uit te voeren.

Wat we hebben gezien in de pluimen van de simulaties, willen we ook graag op een kwalitatieve manier vastleggen. Dit doen we door op een gegeven x-positie te kijken naar de variantie in de y-richting, ofwel  $\text{Var}(S_Y(n)|S_X(n) = x)$ . Door deze variantie op te splitsen naar  $\text{Var}(S_Y(n)|S_X(n) = x) = \sum_y y^2 P(S_Y(n) = y|S_X(n) = x)$  kunnen we deze variantie bepalen door de mogelijke bewegingen van een deeltje te volgen. We moeten dan echter wel conditioneren naar het aantal keren dat een deeltje vrij is geweest,  $K_n$ . Hiertoe moeten we ook weten wat  $P(K_n = k)$  is. Als  $a + b = 1$  is  $K_n$  binomiaal verdeeld. De variantie  $\text{Var}(S_Y(n)|S_X(n) = x)$  wordt uiteindelijk gegeven door

$$\frac{2 \sum_{k=\lceil \frac{x}{2} \rceil}^n \sum_{y=x \bmod 2, +2, +4, \dots}^{k-|x-k|} y^2 \sum_{i=y}^{\frac{x}{2} + \frac{y}{2} - \max(x-k, 0)} \beta^{2i-y} \alpha^{k-2i+y} \binom{k}{2i-y} \binom{2i-y}{i} \binom{k-2i+y}{\frac{k-2i+y-|x-k|}{2}} a^k (1-a)^{n-k} \binom{n}{k}}{\sum_{k=\lceil \frac{x}{2} \rceil}^n \sum_{j=0}^{\lfloor \frac{k-|x-k|}{2} \rfloor} \alpha^{2j+|x-k|} (2\beta)^{k-2j-|x-k|} \binom{k}{k-2j-|x-k|} \binom{2j+|x-k|}{j} a^k (1-a)^{n-k} \binom{n}{k}}$$

Echter, als  $a + b \neq 1$  is  $K_n$  Markov binomiaal verdeeld. We kunnen  $P(K_n = k)$  dan bepalen met recursie in  $n$ , door te conditioneren op de staat van het deeltje (vrij of geadsorbeerd) in tijdstap  $n$ .

Als we de resultaten voor  $\text{Var}(S_Y(n)|S_X(n) = x)$  berekenen met behulp van Matlab, valt voor  $a + b = 1$  een symmetrie op rondom x-positie  $n$ . Dit kunnen we verklaren door te kijken vanuit

een assenstelsel dat meebeweegt met de grondwaterstroom. De bewegingen zijn dan op te vatten als een simpele random walk rondom het punt  $(n, 0)$ , waarbij een geadsorbeerd deeltje een stapje naar links zet. Op deze manier krijgen we een random walk die a-symmetrisch is in de x-richting. Nu is aan te tonen dat  $P(S_Y(n) = y | S_X(n) = n - x) = P(S_Y(n) = y | S_X(n) = n + x)$ .

Vervolgens willen we graag de formule voor  $\text{Var}(S_Y(n) | S_X(n) = x)$  controleren door na simulatie de variantie te berekenen voor deeltjes die in x-positie  $x$  eindigen. Als we dit doen voor simple sampling, krijgen we niet over het gehele bereik een schatting van de variantie, hiertoe gaan we over op importance sampling. We veranderen de kans dat een deeltje vrij is, om ook schattingen te krijgen voor plekken waar meestal geen deeltjes komen. We krijgen dan echter wel te maken met uitkomsten met veel pieken. Deze simulaties kunnen onze berekening daarom niet echt ondersteunen, maar ook niet weerleggen. Om hier een goede uitspraak over te kunnen doen, is er daarom nog meer onderzoek nodig.

Tenslotte hebben we nog even gekeken naar een andere indicator van de spreiding,  $\sqrt{\text{Var}(S_Y(n) | S_X(n) = x) P(S_X(n) = x)}$ . Hierin kunnen we beter zien waar de vervuilingsdeeltjes zich bevinden, maar vooral voor de gevallen met  $a + b = 1$  zou dit nog wat beter mogen kunnen.

## Hoofdstuk 10

# Conclusie en aanbevelingen

We zijn begonnen met de wetenschap dat lang werd gedacht dat een vervuilingspluim in de bodem een Gaussische vorm zou aannemen. Na een paar eenvoudige simulaties zoals beschreven in [3] merken we op dat dit niet het geval lijkt te zijn. Het beste is dit te zien voor  $a = b = 0.01$ . De sigaarvorm van de pluim komt doordat deeltjes die veel vrij zijn, veel advection uitvoeren in de x-richting, maar ook veel dispersie kunnen uitvoeren in zowel de x- als de y-richting. Hierdoor wordt de spreiding in de y-richting groter voor hogere  $x$ .

Om deze aanname kwalitatief te onderzoeken, kijken we naar de variantie in de y-richting op een bepaalde x-positie,  $\text{Var}(S_Y(n)|S_X(n) = x)$ . We vinden dan krommes die niet alleen een waarde geven voor de variantie waarop de pluim deeltjes heeft zitten, maar ook op de punten daarbuiten! Dit komt doordat we naar de voorwaardelijke variantie kijken. We kunnen naar een punt kijken waar met een enorm kleine kans deeltjes komen, maar dat maakt voor de voorwaardelijke variantie niet uit. Het gaat in dit geval alleen om de variantie *gegeven* dat er deeltjes op de x-positie zitten. Ook heel opmerkelijk is dat  $\text{Var}(S_Y(n)|S_X(n) = x)$  een maximum aanneemt op x-positie  $n$ , maar dit laat wel zien dat onze aanname klopt. Deeltjes op verdere x-posities hebben een grotere variantie (behalve voor  $x > n$ , maar hier komen alleen deeltjes voor  $a = b = 0.01$ ). Verder hebben we gezien dat voor  $a + b = 1$  de variantie symmetrisch is rondom  $x = n$  en dit hebben we ook bewezen. Er is hier echter geen koppeling met de praktijk, want we hebben ook gezien dat deeltjes in deze gevallen niet voorbij het punt  $x = n$  komen.

Vervolgens hebben we met behulp van simple sampling en importance sampling aan de eerdere simulaties gekeken of de berekende varianties kloppen. De simple sampling uitkomsten ondersteunen de berekeningen voor x-posities waar in de simulaties voldoende deeltjes kwamen. Voor  $a = b = 0.1$  en  $a = b = 0.01$  hebben we de importance sampling schatter kunnen testen en deze blijkt goed te werken. We zien echter dat als we te ver weg komen van de echte pluim, de schattingen een te grote variantie geven. Al met al heeft de importance sampling schatter weinig kunnen bijdragen aan deze controle. Maar er is geen enkele aanleiding om te geloven dat onze berekeningen voor  $\text{Var}(S_Y(n)|S_X(n) = x)$  niet goed zijn, de importance sampling verworpt onze antwoorden niet.

Tenslotte hebben we gekeken naar  $\sqrt{\text{Var}(S_Y(n)|S_X(n) = x) P(S_X(n) = x)}$  als maat voor de spreiding van de deeltjes. Hierin kunnen we voor de gevallen  $a = b = 0.1$  en  $a = b = 0.01$  redelijk goed zien waar de deeltjes zich bevinden, maar voor de gevallen met  $a + b = 1$  kunnen we dit nog niet zo goed zien.

Voor verder onderzoek kan er nog gekeken worden naar een andere importance sampling schatter. We hebben toen nu toe de verdeling van  $K_n$  veranderd van een (Markov) binomiale verdeling, naar een binomiale verdeling met een veranderde succeskans. Hier stuitten we echter op een enorme variantie in de schatter, waardoor we geen fatsoenlijke antwoorden kregen voor x-posities die ver van de daadwerkelijke pluim liggen. Wellicht geeft een andere verdeling van de importance sam-

pling schatter  $\tilde{K}_n$  en betere uitkomst.

Ook kan er nog verder gezocht worden naar een betere maat voor de spreiding, waaruit we zowel de vorm van de pluim als de verspreiding van de deeltjes kunnen aflezen. Als we kijken naar  $\text{Var}(S_Y(n)|S_X(n) = x)$  weten we welliswaar dat 95.4% van de deeltjes binnen twee standaardafwijkingen van het gemiddelde zal zitten en 99.7% binnen drie standaardafwijkingen, maar waar de deeltjes precies zitten, weten we niet. In [3] is gekeken naar de spreiding van de deeltjes in de x-richting, hiermeer zouden we een waarschijnlijkheidsinterval kunnen opstellen voor de x-posities van de deeltjes. Maar het handigst zou natuurlijk zijn om dit allemaal samen te kunnen nemen.

Tenslotte zou er nog een verbetering kunnen worden gedaan aan het model. In het echt is het niet zo dat deeltjes naar hartelust geadsorbeerd kunnen zijn, vaak heeft de bodem een maximum aantal deeltjes dat het op een bepaalde plek kan adsorberen. Hierdoor zal het model echter wel een stuk ingewikkelder worden. De staat van een deeltje wordt dan immers niet meer alleen bepaald door de overgangsmatrix.

## Hoofdstuk 11

## Bijlagen

## Bijlage I: Afleiding van $\sum_{h=0}^n \binom{n}{h} h^2 p^h q^{n-h} = (p+q)^{n-2} (pqn + p^2n^2)$

We willen een eenvoudige uitdrukking afleiden voor

$$\sum_{h=0}^n h^2 \binom{n}{h} p^h q^{n-h}.$$

Voor de uitdrukking zonder de term  $h^2$  weten we een eenvoudige afleiding, immers

$$\sum_{h=0}^n \binom{n}{h} p^h q^{n-h} = (p+q)^n. \quad (11.1)$$

De term  $h^2$  kunnen we krijgen door twee keer te differentiëren. Formule (11.1) één keer afleiden naar  $p$  geeft

$$\begin{aligned} \frac{d}{dp} \left( \sum_{h=0}^n \binom{n}{h} p^h q^{n-h} \right) &= \frac{d}{dp} (p+q)^n \\ \sum_{h=0}^n \binom{n}{h} h p^{h-1} q^{n-h} &= n(p+q)^{n-1}, \end{aligned}$$

Volgens de somregel van het differentiëren. Nogmaals afleiden naar  $p$  geeft dan

$$\begin{aligned} \frac{d}{dp} \left( \sum_{h=0}^n \binom{n}{h} h p^{h-1} q^{n-h} \right) &= \frac{d}{dp} (n(p+q)^{n-1}) \\ \sum_{h=0}^n \binom{n}{h} h(h-1) p^{h-2} q^{n-h} &= n(n-1)(p+q)^{n-2}. \end{aligned}$$

Dan werken we de haakjes van de linkerzijde weg:

$$\sum_{h=0}^n \binom{n}{h} h(h-1) p^{h-2} q^{n-h} = \sum_{h=0}^n \binom{n}{h} h^2 p^{h-2} q^{n-h} - \sum_{h=0}^n \binom{n}{h} h p^{h-2} q^{n-h}.$$

Voor de tweede term hadden we op een factor  $p$  na al een uitdrukking gevonden na één keer differentiëren naar  $p$ :

$$\sum_{h=0}^n \binom{n}{h} h p^{h-2} q^{n-h} = \frac{n}{p} (p+q)^{n-1},$$

zodat

$$\sum_{h=0}^n \binom{n}{h} h^2 p^{h-2} q^{n-h} - \frac{n}{p} (p+q)^{n-1} = n(n-1)(p+q)^{n-2}$$

Vermenigvuldigen met  $p^2$ , herschikken en haakjes uitwerken geeft dan

$$\begin{aligned} \sum_{h=0}^n \binom{n}{h} h^2 p^h q^{n-h} &= p^2 \left( \frac{n}{p} (p+q)^{n-1} + n(n-1)(p+q)^{n-2} \right) \\ &= pn(p+q)^{n-1} + p^2 n^2 (p+q)^{n-2} - p^2 n(p+q)^{n-2} \\ &= (p+q)^{n-2} (p^2 n + pqn + p^2 n^2 - p^2 n) \\ &= (p+q)^{n-2} (pqn + p^2 n^2). \end{aligned}$$

## Bijlage II: Matlab model voor het simuleren van een pluim met $\text{Var}(S_X(n)) = \text{Var}(S_Y(n) = y)$

```
%Dit programma maakt een verdeling van de vervuilingswolk met behulp van
%"stapjes" op een grid.
close all
clear all
%t geeft de tijd in dagen, die met stapjes van 1 wordt doorlopen.
%N geeft het aantal deeltjes dat wordt geïnjecteerd
%a is p_{af}, de kans dat een geadsorbeerd deeltje vrij wordt op een
%bepaald tijdstip. b is p_{fa}, de kans dat een vrij deeltje geadsorbeerd
%wordt op een bepaald tijdstip.
%pin geeft p_f(0), de kans waarmee een deeltje initieel vrij is.
t=100;
N=10000;
a=0.5;
b=0.5;
pin=0;
%x en y zijn vectoren die voor elk deeltje zijn x- resp. y-positie bijhoudt
%na een bepaald aantal tijdstappen.
%dis en ads houden voor elk deeltje bij in welke tijdstappen deze dispersie
%danwel advection uitvoeren.
x=zeros(N,1);
y=zeros(N,1);
dis=zeros(N,t+1);
ads=zeros(N,t+1);
%Eerst wordt de initiele conditie van de deeltjes bepaald mbv een u(0,1)
%verdeling. Vervolgens wordt op elke tijdstap mbv een u(0,1) verdeling
%bepaald of het deeltje dispersie of advection uitvoert.
%De x- en y-coördinaten worden daarna veranderd aan de hand van dis en ads.
%In dit programma geldt dat de kans dat het deeltje bij dispersie 1 stapje
%naar links, danwel rechts, danwel naar voren danwel naar achteren gaat
%allemaal 1/4e is. Ook dit wordt bepaald met een u(0,1) verdeling.
dis(:,1)=(rand(N,1)<=pin).*ones(N,1);
ads(:,1)=ones(N,1)-dis(:,1);
r=rand(N,2,t);
for i=1:t
    ads(:,i+1)=(r(:,1,i)<=(1-a)).*ads(:,i)+(r(:,1,i)<=b).*dis(:,i);
    dis(:,i+1)=ones(N,1)-ads(:,i+1);
    x=x+dis(:,i+1).*(1+(r(:,2,i)<=0.25)-((r(:,2,i)>0.25).*(r(:,2,i)<=0.50))));
    y=y+dis(:,i+1).*(((r(:,2,i)>0.50).*(r(:,2,i)<=0.75))-(r(:,2,i)>0.75));
end
%Vervolgens definiëren we dx en dy, de reikwijdte van x en y waarover we de
%contourplot willen maken. Deze zijn alleen afhankelijk van t, en dus
%hetzelfde voor verschillende a, b en pin.
%We maken eerst een plot met voor elk deeltje een . op zijn eindplek.
dx=0:1:ceil(1.3*t);
dy=floor(-t/3):1:ceil(t/3);
figure(1);
plot(x,y,'.')
AXIS([0 1.3*t -t/3 t/3])
%Vervolgens maken we een matrix met grootte dy+1 bij dx+1, die we opslaan
%als sparse om het programma sneller te maken. Voor elk deeltje bekijken we
```

```

%zijn plaats in de matrix en tellen hier 1 bijop. Zo krijgen we de
%concentratie op alle punten.
z=zeros(2*ceil(t/3)+1,ceil(1.3*t)+1);
Z=sparse(z);
for i=1:N
    Z((y(i)+ceil(t/3+1)),(x(i)+1))=Z((y(i)+ceil(t/3)+1),(x(i)+1))+1;
end
%v zorgt ervoor dat we op de contourplot ook een hoogtelijn krijgen op 1,
%dus de plek waar de wolk zich bevindt. Daarbij zijn ook voor verschillende
%a,b en pin de hoogtelijnen bij dezelfde concentratie.
%Tenslotte plotten we de concentratie op dy+1 bij dx+1.
v=[1;5;10;20;40;60];
figure(2);
contour(dx,dy,Z,v);colorbar;
AXIS([0 1.3*t -t/3 t/3])

```

### Bijlage III: Matlab model voor het simuleren van een pluim met $\text{Var}(S_X(n)) = 10 \text{Var}(S_Y(n) = y)$ d.m.v. variatie in stapgrootte

```
%Dit programma maakt een verdeling van de vervuilingswolk met behulp van
%"stapjes" op een grid. Hierbij is de variantie in de x-richting 10 keer
%zo groot als de variantie in de y-richting. In dit model variëren we de
%stapgrootte volgens Model I.
close all
clear all
%t geeft de tijd in dagen, die met stapjes van 1 wordt doorlopen.
%N geeft het aantal deeltjes dat wordt geïnjecteerd
%a is  $p_{\{af\}}$ , de kans dat een geadsorbeerd deeltje vrij wordt op een
%bepaald tijdstip. b is  $p_{\{fa\}}$ , de kans dat een vrij deeltje geadsorbeerd
%wordt op een bepaald tijdstip.
%pin geeft  $p_f(0)$ , de kans waarmee een deeltje initieel vrij is.
t=100;
N=10000;
a=0.1;
b=0.1;
pin=a/(a+b);
%x en y zijn vectoren die voor elk deeltje zijn x- resp. y-positie bijhoudt
%na een bepaald aantal tijdstappen.
%dis en ads houden voor elk deeltje bij in welke tijdstappen deze dispersie
%danwel advection uitvoeren.
x=zeros(N,1);
y=zeros(N,1);
dis=zeros(N,t+1);
ads=zeros(N,t+1);
%Eerst wordt de initiele conditie van de deeltjes bepaald mbv een u(0,1)
%verdeling. Vervolgens wordt op elke tijdstap mbv een u(0,1) verdeling
%bepaald of het deeltje dispersie of advection uitvoert.
%De x- en y-coördinaten worden daarna veranderd aan de hand van dis en ads.
%In dit programma geldt dat de kans dat het deeltje bij dispersie
% $\sqrt{2/11}$  stapje naar links, danwel rechts, of  $\sqrt{20/11}$  stapje naar
%voren danwel naar achteren gaat, allemaal  $1/4e$  is. Ook dit wordt bepaald
%met een u(0,1) verdeling.
dis(:,1)=(rand(N,1)<=pin).*ones(N,1);
ads(:,1)=ones(N,1)-dis(:,1);
r=rand(N,2,t);
for i=1:t
    ads(:,i+1)=(r(:,1,i)<=(1-a)).*ads(:,i)+(r(:,1,i)<=b).*dis(:,i);
    dis(:,i+1)=ones(N,1)-ads(:,i+1);
    x=x+dis(:,i+1).*(1+sqrt(20/11)*(r(:,2,i)<=0.25)-sqrt(20/11)*((r(:,2,i)>0.25)
        .*(r(:,2,i)<=0.50)))));
    y=y+dis(:,i+1).*(sqrt(2/11)*((r(:,2,i)>0.50).*(r(:,2,i)<=0.75))-sqrt(2/11)
        .*(r(:,2,i)>0.75)));
end
%Omdat hier de stapjes in de x- en y-richting geen gehele getallen zijn,
%moeten we de uiteindelijke posities van de deeltjes afronden om deze te
%kunnen plotten.
y=round(y);
x=round(x);
```

```

%Vervolgens definiëren we dx en dy, de reikwijdte van x en y waarover we de
%contourplot willen maken. Deze zijn alleen afhankelijk van t, en dus
%hetzelfde voor verschillende a, b en pin. Omdat de x-coördinaat van een
%deeltje soms negatief blijkt te zijn, laten we dx lopen van het minimum
%van 0 en de kleinste x-coördinaat van de deeltjes.
%We maken eerst een plot met voor elk deeltje een . op zijn eindplek.
minx=min(min(x,0));
dx=minx:1:ceil(1.3*t)+minx+2;
dy=floor(-t/3):1:ceil(t/3);
figure(1);
plot(x,y,'.')
AXIS([minx 1.3*t+minx -t/3 t/3])
%Vervolgens maken we een matrix met grootte dy+1 bij dx+1, die we opslaan
%als sparse om het programma sneller te maken. Voor elk deeltje bekijken we
%zijn plaats in de matrix en tellen hier 1 bijop. Zo krijgen we de
%concentratie op alle punten.
z=zeros(2*ceil(t/3)+1,ceil(1.3*t)+3);
Z=sparse(z);
for i=1:N
    Z((y(i)+floor(t/3+1)),(x(i)+1-minx))=Z((y(i)+floor(t/3)+1),(x(i)+1-minx))+1;
end
%v zorgt ervoor dat we op de contourplot ook een hoogtelijn krijgen op 1,
%dus de plek waar de wolk zich bevindt. Daarbij zijn ook voor verschillende
%a,b en pin de hoogtelijnen bij dezelfde concentratie.
%Tenslotte plotten we de concentratie op dy+1 bij dx+1.
v=[1;5;10;20;40;60;90;120];
figure(2);
contour(dx,dy,Z,v);colorbar;
AXIS([minx 1.3*t+minx -t/3 t/3])

```

## Bijlage IV: Matlab model voor het simuleren van een pluim met $\text{Var}(S_X(n)) = 10 \text{Var}(S_Y(n) = y)$ d.m.v. variatie in kansen

```
%Dit programma maakt een verdeling van de vervuilingswolk met behulp van
%"stapjes" op een grid. Hierbij is de variantie in de x-richting 10 keer
%zo groot als de variantie in de y-richting. In dit model variëren we de
%kansen, volgens Model II.
tic
close all
clear all
%t geeft de tijd in dagen, die met stapjes van 1 wordt doorlopen.
%N geeft het aantal deeltjes dat wordt geïnjecteerd
%a is  $p_{\{af\}}$ , de kans dat een geadsorbeerd deeltje vrij wordt op een
%bepaald tijdstip. b is  $p_{\{fa\}}$ , de kans dat een vrij deeltje geadsorbeerd
%wordt op een bepaald tijdstip.
%pin geeft  $p_f(0)$ , de kans waarmee een deeltje initieel vrij is.
t=100;
N=10000;
a=0.01;
b=0.01;
pin=0;%a/(a+b);
%x en y zijn vectoren die voor elk deeltje zijn x- resp. y-positie bijhoudt
%na een bepaald aantal tijdstappen.
%dis en ads houden voor elk deeltje bij in welke tijdstappen deze dispersie
%danwel advection uitvoeren.
x=zeros(N,1);
y=zeros(N,1);
dis=zeros(N,t+1);
ads=zeros(N,t+1);
%Eerst wordt de initiele conditie van de deeltjes bepaald mbv een u(0,1)
%verdeling. Vervolgens wordt op elke tijdstap mbv een u(0,1) verdeling
%bepaald of het deeltje dispersie of advection uitvoert.
%De x- en y-coördinaten worden daarna veranderd aan de hand van dis en ads.
%In dit programma geldt dat de kans dat het deeltje bij dispersie een stapje
%naar links, danwel rechts gaat 1/22e is en de kans dat een deeltje een
%stapje naar achteren of naar voren gaat 5/11e is.
%Ook dit wordt bepaald met een u(0,1) verdeling.
dis(:,1)=(rand(N,1)<=pin).*ones(N,1);
ads(:,1)=ones(N,1)-dis(:,1);
r=rand(N,2,t);
for i=1:t
    ads(:,i+1)=(r(:,1,i)<=(1-a)).*ads(:,i)+(r(:,1,i)<=b).*dis(:,i);
    dis(:,i+1)=ones(N,1)-ads(:,i+1);
    x=x+dis(:,i+1).*(1+(r(:,2,i)<=5/11)-((r(:,2,i)>5/11).*(r(:,2,i)<=10/11)));
    y=y+dis(:,i+1).*((r(:,2,i)>10/11).*(r(:,2,i)<=21/22))-(r(:,2,i)>21/22));
end
%Vervolgens definiëren we dx en dy, de reikwijdte van x en y waarover we de
%contourplot willen maken. Deze zijn alleen afhankelijk van t, en dus
%hetzelfde voor verschillende a, b en pin.
%We maken eerst een plot met voor elk deeltje een . op zijn eindplek.
minx=0;
dx=minx:1:ceil(1.7*t);
```

```

dy=floor(-t/3):1:ceil(t/3);
figure(1);
plot(x,y,'.')
AXIS([minx 1.3*t+minx -t/3 t/3])
%Vervolgens maken we een matrix met grootte dy+1 bij dx+1, die we opslaan
%als sparse om het programma sneller te maken. Voor elk deeltje bekijken we
%zijn plaats in de matrix en tellen hier 1 bijop. Zo krijgen we de
%concentratie op alle punten.
z=zeros(2*ceil(t/3)+1,ceil(1.7*t)+1-minx);
Z=sparse(z);
for i=1:N
    Z((y(i)+ceil(t/3)+1),(x(i)+1-minx))=Z((y(i)+ceil(t/3)+1),(x(i)+1-minx))+1;
end
%v zorgt ervoor dat we op de contourplot ook een hoogtelijn krijgen op 1,
%dus de plek waar de wolk zich bevindt. Daarbij zijn ook voor verschillende
%a,b en pin de hoogtelijnen bij dezelfde concentratie.
%Tenslotte plotten we de concentratie op dy+1 bij dx+1.
v=[1;5;10;20;40;60;90;120];
figure(2);
contour(dx,dy,Z,v);colorbar;
AXIS([0 200 -t/3 t/3])
toc

```

## Bijlage V: Matlab model voor het uitrekenen van de variantie voor $a + b = 1$ volgens formule (5.8)

### varygxstukjes.m

```
%Dit programma bepaalt de variantie van Sy voor een gegeven Sx, door middel
%van de berekende formule, voor a+b=1.
tic
%clear all
warning off
%x=10;
%n geeft het aantal tijdsstappen, p0 t/m p3 zijn de kansen dat een vrij
%deeltje een stapje naar resp. rechters, onder, links, boven maakt. Ook wel
%alpha en beta, de kansen voor resp. naar links/rechts en naar boven/onder
n=100;
a=1;
b=0;
p0=5/11;
p1=1/22;
p2=5/11;
p3=1/22;
alpha=p0;
beta=p1;
sumpx=0;
%We berekenen de variantie per x-positie mbv een som over x, volgens de afgeleide formule
%Hierbij staan de verschillende kansen in aparte Matlab files, om het overzichtelijk te
%houden.
%De disp zorgt ervoor dat tijdens de berekening in het Matlab scherm kan worden gezien
%hoe ver de berekening al is.
for x=0:1:2*n
    if ceil(x/10)==floor(x/10)
        disp([num2str(ceil(100*x/200)) '% done.']);
    end
    teller=0;
    noemer=0;
    for k=ceil(x/2):1:n %k is het aantal keren dat het deeltje vrij is geweest
        tellerk=0;
        for y=mod(x,2):2:k-abs(x-k) %y is de y-positie van een deeltje
            stukjespyexgk %roept de file aan die P(S_Y(n)=y,S_X(n)=x|K_n=k) berekent
            tellerk=tellerk+y^2*pygxgk;
        end
        teller=teller+2*tellerk*a^k*(1-a)^(n-k)*nchoosek(n,k);

        stukjespx %roept de file aan die P(S_X(n)=x) berekent
        noemer=noemer+px;
    end
    vary(x+1)=teller/noemer;
end
X=(0:1:x); %X zorgt ervoor dat de waarden van vary op de goede x-positie
plot(X,vary) %worden geplot
AXIS([0 200 0 10])
YLABEL('Var(Sy|Sx=x)')
XLABEL('Sx=x')
```

toc

### stukjespyexgk.m

```
% Dit programma bepaalt  $P(S_Y(n)=y, S_X(n)=x | K_n=k)$  aan de hand van formule 5.5
pygxgk=0;
for i=y:1:y/2+x/2-max(x-k,0)
    pygxgk=pygxgk+beta^(2*i-y)*alpha^(k-2*i+y)*nchoosek(k,2*i-y)*nchoosek(2*i-y,i)
    *nchoosek(k-2*i+y,(k-2*i+y-abs(k-x))/2);
end
```

### stukjespx.m

```
%Dit programma bepaalt  $P(S_X(n)=x)$  aan de hand van formule 5.6 en 5.7
pxgk=0;
for j=0:1:floor((k-abs(x-k))/2)
    pxgk=pxgk+alpha^(2*j+abs(x-k))*(2*beta)^(k-2*j-abs(x-k))
    *nchoosek(k,k-2*j-abs(x-k))*nchoosek(2*j+abs(x-k),j);
end
%Omdat dit programmaatje zowel voor de gevallen met  $a+b=1$  als met  $a+b \neq 1$  wordt
%gebruikt, maken we hier onderscheid tussen deze twee gevallen. Voor  $a+b=1$ 
%gebruiken we de berekening uit de formule, voor  $a+b \neq 1$  gebruiken we de vector
%pkmatrixbin die bepaald wordt door de file stukjespkmatrixbin.m
if a+b==1
    px=pxgk*a^k*(1-a)^(n-k)*nchoosek(n,k);
else
    px=pxgk*pkmatrixbin(k+1);
end
```

## Bijlage VI: Matlab model voor het uitrekenen van de variantie voor $a + b \neq 1$

### varygxstukjesaenbniet1.m

```
%Dit programma bepaalt de variantie van Sy voor een gegeven Sx, door middel
%van de berekende formule voor a+b!=1.
tic
%clear all
warning off
%x=10;
%n geeft het aantal tijdsstappen, p0 t/m p3 zijn de kansen dat een vrij
%deeltje een stapje naar resp. rechters, onder, links, boven maakt. Ook wel
%alpha en beta, de kansen voor resp. naar links/rechts en naar boven/onder
n=100;
a=0.01;
b=0.01;
p0=5/11;
p1=1/22;
p2=5/11;
p3=1/22;
alpha=p0;
beta=p1;
sumpx=0;
%We berekenen de variantie per x-positie mbv een som over x, volgens de afgeleide
%formule. Hierbij staan de verschillende kansen in aparte Matlab files, om het
%overzichtelijk te houden.
%De disp zorgt ervoor dat tijdens de berekening in het Matlab scherm kan worden
%gezien hoe ver de berekening al is.
%Het programma stukjespkmatrixbin geeft de vector met P(Kn=k) voor alle k.
stukjespkmatrixbin
for x=0:1:2*n
    if ceil(x/10)==floor(x/10)
        disp([num2str(ceil(100*x/200)) '% done.']);
    end
    teller=0;
    noemer=0;
    for k=ceil(x/2):1:n %k het aantal keren dat het deeltje vrij is geweest
        tellerk=0;
        for y=mod(x,2):2:k-abs(x-k) %y de y-positie van een deeltje
            stukjespyexgk
            tellerk=tellerk+y^2*pygxgk;
        end
        teller=teller+2*tellerk*pkmatrixbin(k+1);

        stukjespx %roept de file aan die P(S_X(n)=x) berekent
        noemer=noemer+px;
    end
    vary(x+1)=teller/noemer;
end
X=(0:1:x); %X zorgt ervoor dat de waarden van vary op de goede x-positie
plot(X,vary) %worden geplot
YLABEL('Var(Sy|Sx=x)')
```

```
XLABEL('Sx=x')
toc
```

### stukjespkmatrixbin.m

```
%Dit programma bepaalt  $P(K_n=k)$  voor  $a+b!=1$ . pvrij en pads houden voor
%elke m en k bij hoe groot de kans is dat  $K_m=k$ , onderverdeeld naar of
%een deeltje het laatste tijdstapje vrij (pvrij) of geadsorbeerd (pads)
%was. We vullen deze matrices in mbv de recursie in paragraaf 5.4
pvrij=zeros(n+1,n+1);
pads=zeros(n+1,n+1);
pads(2,1)=b/(a+b);
pvrij(2,2)=a/(a+b);
for k=0:1:n
    for m=1:1:n-1
        pads(m+2,k+1)=(1-a)*pads(m+1,k+1)+b*pvrij(m+1,k+1);
        if k>=1
            pvrij(m+2,k+1)=a*pads(m+1,k)+(1-b)*pvrij(m+1,k);
        else
            pvrij(m+2,k+1)=0;
        end
    end
end
end
% $P(K_n=k)$  is dan gelijk aan pads(n+1,k)+pvrij(n+1,k). pkmatrixbin is een
%vector die voor elke k  $P(K_n=k)$  teruggeeft.
pkmatrixbin=pads(n+1,:)+pvrij(n+1,:);
```

## Bijlage VII: Matlab model voor het berekenen van de simple sampling variantie

```
%Dit programma is voor het simuleren van de simple sampling variantie.
%Dit programma maakt eerst een verdeling van de vervuilingswolk met behulp van
%"stapjes" op een grid. Vervolgens filteren we de eindposities van de
%deeltjes naar x-positie en bepalen we per x-positie de variantie van de
%deeltjes in de y-richting.
tic
close all
clear all
%t geeft de tijd in dagen, die met stapjes van 1 wordt doorlopen.
%N geeft het aantal deeltjes dat wordt geïnjecteerd
%a is  $p_{af}$ , de kans dat een geadsorbeerd deeltje vrij wordt op een
%bepaald tijdstip. b is  $p_{fa}$ , de kans dat een vrij deeltje geadsorbeerd
%wordt op een bepaald tijdstip.
%pin geeft  $p_f(0)$ , de kans waarmee een deeltje initieel vrij is.
n=100;
N=250000;
a=0.1;
b=0.1;
pin=a/(a+b);
p0=5/11;
p1=1/22;
p2=5/11;
p3=1/22;
%x en y zijn vectoren die voor elk deeltje zijn x- resp. y-positie bijhoudt
%na een bepaald aantal tijdstappen.
%dis en ads houden voor elk deeltje bij in welke tijdstappen deze dispersie
%danwel advection uitvoeren.
x=zeros(N,1);
y=zeros(N,1);
dis=zeros(N,n+1);
ads=zeros(N,n+1);
%Eerst wordt de initiele conditie van de deeltjes bepaald mbv een u(0,1)
%verdeling. Vervolgens wordt op elke tijdstap mbv een u(0,1) verdeling
%bepaald of het deeltje dispersie of advection uitvoert.
%De x- en y-coördinaten worden daarna veranderd aan de hand van dis en ads.
%In dit programma geldt dat de kans dat het deeltje bij dispersie een stapje
%naar links, danwel rechts gaat 1/22e is en de kans dat een deeltje een
%stapje naar achteren of naar voren gaat 5/11e is.
%Ook dit wordt bepaald met een u(0,1) verdeling.
dis(:,1)=(rand(N,1)<=pin).*ones(N,1);
ads(:,1)=ones(N,1)-dis(:,1);
r=rand(N,2,n);
for i=1:n
    ads(:,i+1)=(r(:,1,i)<=(1-a)).*ads(:,i)+(r(:,1,i)<=b).*dis(:,i);
    dis(:,i+1)=ones(N,1)-ads(:,i+1);
    x=x+dis(:,i+1).*(1+(r(:,2,i)<=p2)-((r(:,2,i)>p2).*(r(:,2,i)<=(p0+p2))));
    y=y+dis(:,i+1).*((r(:,2,i)>(p0+p2)).*(r(:,2,i)<=(1-p3)))-(r(:,2,i)>(1-p3)));
end
%Vervolgens definiëren we dx en dy, de reikwijdte van x en y waarover we de
%contourplot willen maken. Deze zijn alleen afhankelijk van t, en dus
```

```

%hetzelfde voor verschillende a, b en pin. minx geeft de kleinst mogelijke
%x-positie van de deeltjes, maar dit is voor dit model gelijk aan 0.
%We maken eerst een plot met voor elk deeltje een . op zijn eindplek.
minx=0;
dx=minx:1:2*n;
dy=floor(-n/3):1:ceil(n/3);
figure(1);
plot(x,y,'.')
%Vervolgens maken we een matrix met grootte dy+1 bij dx+1, die we opslaan
%als sparse om het programma sneller te maken. Voor elk deeltje bekijken we
%zijn plaats in de matrix en tellen hier 1 bijop. Zo krijgen we de
%concentratie op alle punten.
z=zeros(2*ceil(n/3)+1,2*n+1);
Z=sparse(z);
for i=1:N
    Z((y(i)+ceil(n/3)+1),(x(i)+1-minx))=Z((y(i)+ceil(n/3)+1),(x(i)+1-minx))+1;
end
%v zorgt ervoor dat we op de contourplot ook een hoogtelijn krijgen op 1,
%dus de plek waar de wolk zich bevindt. Daarbij zijn ook voor verschillende
%a,b en pin de hoogtelijnen bij dezelfde concentratie.
%Tenslotte plotten we de concentratie op dy+1 bij dx+1.
v=[1;5;10;20;40;60;90;120];
figure(2);
contour(dx,dy,Z,v);colorbar;
teller=zeros(2*n,1);
tellernul=zeros(2*n,1);
%We gaan de deeltjes nu sorteren naar x-positie. We kijken daartoe op
%x-positie j, gaan alle deeltjes af en als een deeltje zich op
%x-positie j bevindt, verhogen we de teller ypos met zijn y-positie in
%het kwadraat. Ook tellen we een op bij de teller, omdat we het
%aantal deeltjes per x-positie moeten bijhouden.
%Dit lijkt een hele domme manier om deze berekening te maken, omdat we
%alle deeltjes 2n keer doorlopen, maar het blijkt sneller te zijn dan
%een vector te maken met lengte 2n en alle deeltjes 1 keer af te gaan.
%Matlab kan namelijk niet goed werken met grote vectoren.
for j=1:1:2*n
    ypos=0;
    for p=1:1:N
        if x(p)==j
            teller(j)=teller(j)+1;
            ypos(teller(j))=y(p);
            if y(p)==0
                tellernul(j)=tellernul(j)+1;
            end
        end
    end
    totvar(j)=1/teller(j)*sum(ypos.^2);
end
%We plotten de simple sampling variantie en om een idee te krijgen van
%de kwaliteit van deze schatter, ook het aantal deeltjes per x-positie.
%Om een idee te krijgen hoeveel deeltjes zich op y-positie 0 bevinden
%(dus niet bijdragen aan de variantie), hebben we tellernul, die bevat
%per x-positie het aantal deeltjes op y-positie 0.
figure(3);

```

```
plot(totvar,'r')
axis([0 200 0 10])
figure(4);
plot(teller)
hold on
plot(tellernul,'r')
hold off
toc
```

## Bijlage VIII: Matlab model voor het berekenen van één importance sampling realisatie voor de variantie als $a + b = 1$

```
%In dit programma maken we een importance sampling schatter van de
%voorwaardelijke variantie. Hiertoe maken we een vervuilingsswolk
%volgens de nieuwe verdeling, bin(n,q) (q is in het verslag atilde.
%Vervolgens berekenen we per x-positie de empirische variantie en
%corrigeren we met de factor f/phi.
%In dit programma is a+b=1;
tic
close all
clear all
%t geeft de tijd in dagen, die met stapjes van 1 wordt doorlopen.
%N geeft het aantal deeltjes dat wordt geïnjecteerd
%a is  $p_{af}$ , de kans dat een geadsorbeerd deeltje vrij wordt op een
%bepaald tijdstip. b is  $p_{fa}$ , de kans dat een vrij deeltje geadsorbeerd
%wordt op een bepaald tijdstip.
%pin geeft  $p_f(0)$ , de kans waarmee een deeltje initieel vrij is.
N=250000;
t=100;
a=0.5;
b=0.5;
p0=5/11;
p1=1/22;
p2=5/11;
p3=1/22;
q=0.15;
%We simuleren hier op een andere manier, om een beter inzicht te krijgen
%dat dit een Monte Carlo methode is. Hiertoe introduceren we de vector k
%die voor elk deeltje bijhoudt, hoe vaak het vrij is en de vectoren links,
%rechts, op en neer die bijhouden voor vaak een deeltje de betreffende
%bewegingen uitvoert.
k=zeros(N,1);
links=zeros(N,1);
rechts=zeros(N,1);
op=zeros(N,1);
neer=zeros(N,1);
x=zeros(N,1);
y=zeros(N,1);
%pk houdt bij hoeveel deeltjes k keer vrij zijn geweest, varia is de
%simple sampling variantie per x-positie. Teller houdt bij hoeveel deeltjes
%er per x-positie zijn.
varia=zeros(201,1);
teller=zeros(201,1);
pk=sparse(zeros(101,1));
for n=1:N
    if ceil(n/25000)==floor(n/25000) %Laat zien hoe ver we zijn
        disp([num2str(ceil(100*n/N)) '% done.']);
    end
    k(n)=binornd(t,q);
    for i=1:k(n)
        r=rand(1);
        if r<p0
```

```

        links(n)=links(n)+1;
    elseif r<(p0+p1)
        op(n)=op(n)+1;
    elseif r<(p0+p1+p2)
        rechts(n)=rechts(n)+1;
    else
        neer(n)=neer(n)+1;
    end
    %De uiteindelijke x- en y-posities worden bepaalt door k,
    %links, rechts, op en neer. Ze worden hier berekent.
    x(n)=k(n)+rechts(n)-links(n);
    y(n)=op(n)-neer(n);
    pk(k(n)+1)=pk(k(n)+1)+1;
end
varia(x(n)+1)=varia(x(n)+1)+(y(n))^2;
teller(x(n)+1)=teller(x(n)+1)+1;
end
disp('deeltjes zijn verspreid'); %Laat zien dat de simulatie erop zit
%We maken een plaatje van de vervuilingswolk en de variantie van deze
%simulatie.
plot(x,y,'.')
varia=varia./teller;
figure(2)
plot(varia)
%Omdat de variantie uit een teller en een noemer bestaat, die beide
%gecorrigeerd moeten worden, houden we dit bij met twee vectoren,
%boven (teller) en onder (noemer). In een for-loop corrigeren we
%per x-positie. Hiertoe gaan we per x-positie alle deeltjes af,
%maar dat gaat betrekkelijk snel.
boven=zeros(2*t+1,1);
onder=zeros(2*t+1,1);
var2=zeros(2*t+1,1);
for xpos=0:200
    for n=1:N
        if x(n)==xpos
            if k(n)<t;
                boven(x(n)+1)=boven(x(n)+1)+(y(n))^2*(a/q)^(k(n))
                    *((1-a)/(1-q))^(t-k(n));
                onder(x(n)+1)=onder(x(n)+1)+1*(a/q)^(k(n))*((1-a)
                    /(1-q))^(t-k(n));
            end
        end
    end
end
end
%De importance sampling variantie is dan boven/onder, deze plotte we.
var2=boven./onder;
figure(3)
plot(var2)
toc

```

## Bijlage IX: Matlab model voor het berekenen van meerdere importance sampling realisaties voor de variantie, met $a + b = 1$

```
%In dit programma maken we een meerdere importance sampling schatters
%van de voorwaardelijke variantie. Hiertoe maken we een vervuilingswolk
%volgens de nieuwe verdeling, bin(n,q) (q is in het verslag atilde.
%Vervolgens berekenen we per x-positie de empirische variantie en
%corrigeren we met de factor f/phi.
%In dit programma is a+b=1;
tic
close all
clear all
%t geeft de tijd in dagen, die met stapjes van 1 wordt doorlopen.
%N geeft het aantal deeltjes dat wordt geïnjecteerd
%a is  $p_{af}$ , de kans dat een geadsorbeerd deeltje vrij wordt op een
%bepaald tijdstip. b is  $p_{fa}$ , de kans dat een vrij deeltje geadsorbeerd
%wordt op een bepaald tijdstip.
%pin geeft  $p_f(0)$ , de kans waarmee een deeltje initieel vrij is.
N=100000;
t=100;
a=0.5;
b=0.5;
p0=5/11;
p1=1/22;
p2=5/11;
p3=1/22;
%We doen de simulatie nu voor elk intervalletje met lengte 5.
uiteinvar=zeros(201,1);
for pos=3:5:100
    q=pos/100;
    %We simuleren hier op een andere manier, om een beter inzicht te krijgen
    %dat dit een Monte Carlo methode is. Hiertoe introduceren we de vector k
    %die voor elk deeltje bijhoudt, hoe vaak het vrij is en de vectoren links,
    %rechts, op en neer die bijhouden voor vaak een deeltje de betreffende
    %bewegingen uitvoert.
    k=zeros(N,1);
    links=zeros(N,1);
    rechts=zeros(N,1);
    op=zeros(N,1);
    neer=zeros(N,1);
    x=zeros(N,1);
    y=zeros(N,1);
    %pk houdt bij hoeveel deeltjes k keer vrij zijn geweest, varia is de
    %simple sampling variantie per x-positie. Teller houdt bij hoeveel deeltjes
    %er per x-positie zijn.
    varia=zeros(201,1);
    teller=zeros(201,1);
    pk=sparse(zeros(101,1));
    for n=1:N
        if ceil(n/25000)==floor(n/25000) %Laat zien hoe ver we zijn
            disp([num2str(ceil(100*n/N)) '% done.']);
        end
    end
end
```

```

k(n)=binornd(t,q);
for i=1:k(n)
    r=rand(1);
    if r<p0
        links(n)=links(n)+1;
    elseif r<(p0+p1)
        op(n)=op(n)+1;
    elseif r<(p0+p1+p2)
        rechts(n)=rechts(n)+1;
    else
        neer(n)=neer(n)+1;
    end
    %De uiteindelijke x- en y-posities worden bepaalt door k,
    %links, rechts, op en neer. Ze worden hier berekent.
    x(n)=k(n)+rechts(n)-links(n);
    y(n)=op(n)-neer(n);
    pk(k(n)+1)=pk(k(n)+1)+1;
end
varia(x(n)+1)=varia(x(n)+1)+(y(n))^2;
teller(x(n)+1)=teller(x(n)+1)+1;
end
disp('deeltjes zijn verspreid'); %Laat zien dat de simulatie erop zit
%We maken een plaatje van de vervuilingswolk en de variantie van deze
%simulatie.
plot(x,y,'.')
varia=varia./teller;
figure(2)
plot(varia)
%Omdat de variantie uit een teller en een noemer bestaat, die beide
%gecorrigeerd moeten worden, houden we dit bij met twee vectoren,
%boven (teller) en onder (noemer). In een for-loop corrigeren we
%per x-positie. Hiertoe gaan we per x-positie alle deeltjes af,
%maar dat gaat betrekkelijk snel.
boven=zeros(2*t+1,1);
onder=zeros(2*t+1,1);
var2=zeros(2*t+1,1);
for xpos=0:200
    for n=1:N
        if x(n)==xpos
            if k(n)<t;
                boven(x(n)+1)=boven(x(n)+1)+(y(n))^2*(a/q)^(k(n))
                    *((1-a)/(1-q))^(t-k(n));
                onder(x(n)+1)=onder(x(n)+1)+1*(a/q)^(k(n))*((1-a)
                    /(1-q))^(t-k(n));
            end
        end
    end
end
end
%De importance sampling variantie is dan boven/onder, deze geldt
%als schatting op het interval. De uiteindelijke variantie plotten
%we.
var2=boven./onder;
uiteinvar(pos-2:pos+2)=var2(pos-2:pos+2);
end

```

```
plot(uiteinvar)  
toc
```

## Bijlage X: Matlab model voor het berekenen van meerdere importance sampling realisaties voor de variantie, met $a + b \neq 1$

```
%In dit programma maken we een meerdere importance sampling schatters
%van de voorwaardelijke variantie. Hiertoe maken we een vervuilingswolk
%volgens de nieuwe verdeling, bin(n,q) (q is in het verslag atilde.
%Vervolgens berekenen we per x-positie de empirische variantie en
%corrigeren we met de factor f/phi.
%Dit programma werkt ook voor a+b!=1.
tic
close all
clear all
%t geeft de tijd in dagen, die met stapjes van 1 wordt doorlopen.
%N geeft het aantal deeltjes dat wordt geïnjecteerd
%a is p_{af}, de kans dat een geadsorbeerd deeltje vrij wordt op een
%bepaald tijdstip. b is p_{fa}, de kans dat een vrij deeltje geadsorbeerd
%wordt op een bepaald tijdstip.
%pin geeft p_f(0), de kans waarmee een deeltje initieel vrij is.
N=100000;
t=100;
a=0.5;
b=0.5;
p0=5/11;
p1=1/22;
p2=5/11;
p3=1/22;
stukjespkmatrixbin %geeft P(K_n=k)
%We doen de simulatie nu voor elk intervalletje met lengte 5.
uiteinvar=zeros(201,1);
for pos=3:5:100
    q=pos/100;
    %We simuleren hier op een andere manier, om een beter inzicht te krijgen
    %dat dit een Monte Carlo methode is. Hiertoe introduceren we de vector k
    %die voor elk deeltje bijhoudt, hoe vaak het vrij is en de vectoren links,
    %rechts, op en neer die bijhouden voor vaak een deeltje de betreffende
    %bewegingen uitvoert.
    k=zeros(N,1);
    links=zeros(N,1);
    rechts=zeros(N,1);
    op=zeros(N,1);
    neer=zeros(N,1);
    x=zeros(N,1);
    y=zeros(N,1);
    %pk houdt bij hoeveel deeltjes k keer vrij zijn geweest, varia is de
    %simple sampling variantie per x-positie. Teller houdt bij hoeveel deeltjes
    %er per x-positie zijn.
    varia=zeros(201,1);
    teller=zeros(201,1);
    pk=sparse(zeros(101,1));
    for n=1:N
        if ceil(n/25000)==floor(n/25000) %Laat zien hoe ver we zijn
            disp([num2str(ceil(100*n/N)) '% done.']);
        end
    end
end
```

```

k(n)=binornd(t,q);
for i=1:k(n)
    r=rand(1);
    if r<p0
        links(n)=links(n)+1;
    elseif r<(p0+p1)
        op(n)=op(n)+1;
    elseif r<(p0+p1+p2)
        rechts(n)=rechts(n)+1;
    else
        neer(n)=neer(n)+1;
    end
    %De uiteindelijke x- en y-posities worden bepaalt door k,
    %links, rechts, op en neer. Ze worden hier berekent.
    x(n)=k(n)+rechts(n)-links(n);
    y(n)=op(n)-neer(n);
    pk(k(n)+1)=pk(k(n)+1)+1;
end
varia(x(n)+1)=varia(x(n)+1)+(y(n))^2;
teller(x(n)+1)=teller(x(n)+1)+1;
end
disp('deeltjes zijn verspreid'); %Laat zien dat de simulatie erop zit
%We maken een plaatje van de vervuilingswolk en de variantie van deze
%simulatie.
plot(x,y,'.')
varia=varia./teller;
figure(2)
plot(varia)
%Omdat de variantie uit een teller en een noemer bestaat, die beide
%gecorrigeerd moeten worden, houden we dit bij met twee vectoren,
%boven (teller) en onder (noemer). In een for-loop corrigeren we
%per x-positie. Hiertoe gaan we per x-positie alle deeltjes af,
%maar dat gaat betrekkelijk snel.
boven=zeros(2*t+1,1);
onder=zeros(2*t+1,1);
var2=zeros(2*t+1,1);
for xpos=0:200
    for n=1:N
        if x(n)==xpos
            if k(n)<t;
                boven(x(n))=boven(x(n))+(y(n))^2*pkmatrixbin(k(n)+1)*(1/q)^(k(n))
                    *(1/(1-q))^(100-k(n))*1/(nchoosek(100,k(n)));
                onder(x(n))=onder(x(n))+pkmatrixbin(k(n)+1)*(1/q)^(k(n))
                    *(1/(1-q))^(100-k(n))*1/(nchoosek(100,k(n)))
            end
        end
    end
end
end
%De importance sampling variantie is dan boven/onder, deze geldt
%als schatting op het interval. De uiteindelijke variantie plotten
%we.
var2=boven./onder;
uiteinvar(pos-2:pos+2)=var2(pos-2:pos+2);
end

```

```
plot(uiteinvar)
toc
```

## Bijlage XI: Matlab model voor het uitrekenen van $\text{Var}(S_Y(n)|S_X(n) = x) P(S_X(n) = x)$ voor $a + b = 1$

```
%Dit programma bepaalt de variantie van Sy voor een gegeven Sx,
%vermenigvuldigd met de kans dat Sx=x, door middel van de
%berekende formule, voor a+b=1.
tic
%clear all
warning off
%x=10;
%n geeft het aantal tijdstappen, p0 t/m p3 zijn de kansen dat een vrij
%deeltje een stapje naar resp. rechters, onder, links, boven maakt. Ook wel
%alpha en beta, de kansen voor resp. naar links/rechts en naar boven/onder
n=100;
a=0.5;
b=0.5;
p0=5/11;
p1=1/22;
p2=5/11;
p3=1/22;
alpha=p0;
beta=p1;
sumpx=0;
%We berekenen de variantie per x-positie mbv een som over x, volgens de
%afgeleide formule. Hierbij staan de verschillende kansen in aparte Matlab
%files, om het overzichtelijk te houden.
%De disp zorgt ervoor dat tijdens de berekening in het Matlab scherm kan
%worden gezien hoe ver de berekening al is.
for x=0:1:2*n
    if ceil(x/10)==floor(x/10)
        disp([num2str(ceil(100*x/200)) '% done.']);
    end
    teller=0;
    noemer=0;
    for k=ceil(x/2):1:n %k is het aantal keren dat het deeltje vrij is geweest
        tellerk=0;
        for y=mod(x,2):2:k-abs(x-k) %y is de y-positie van een deeltje
            %stukjespyexgk roept de file aan die P(S_Y(n)=y,S_X(n)=x|K_n=k) berekent
            stukjespyexgk
            tellerk=tellerk+y^2*pygxgk;
        end
        teller=teller+2*tellerk*a^k*(1-a)^(n-k)*nchoosek(n,k);
    end
    noemer=1;
end
vary(x+1)=teller/noemer;
end
X=(0:1:x); %X zorgt ervoor dat de waarden van vary op de goede x-positie
plot(X,vary) %worden geplot
AXIS([0 200 0 10])
YLABEL('Var(Sy|Sx=x)')
XLABEL('Sx=x')
toc
```

## Bijlage XII: Matlab model voor het uitrekenen van $\text{Var}(S_Y(n)|S_X(n) = x) P(S_X(n) = x)$ voor $a + b \neq 1$

```
%Dit programma bepaalt de variantie van Sy voor een gegeven Sx
%vermenigvuldigd met de kans dat Sx=x, door middel van de berekende
%formule voor a+b!=1.
tic
%clear all
warning off
%x=10;
%n geeft het aantal tijdstappen, p0 t/m p3 zijn de kansen dat een vrij
%deeltje een stapje naar resp. rechters, onder, links, boven maakt. Ook wel
%alpha en beta, de kansen voor resp. naar links/rechts en naar boven/onder
n=100;
a=0.01;
b=0.01;
p0=5/11;
p1=1/22;
p2=5/11;
p3=1/22;
alpha=p0;
beta=p1;
sumpx=0;
%We berekenen de variantie per x-positie mbv een som over x, volgens de
%afgeleidde formule. Hierbij staan de verschillende kansen in aparte Matlab
%files, om het overzichtelijk te houden.
%De disp zorgt ervoor dat tijdens de berekening in het Matlab scherm kan
%worden gezien hoe ver de berekening al is.
%Het programma stukjespkmatrixbin geeft de vector met P(Kn=k) voor alle k.
stukjespkmatrixbin
for x=0:1:2*n
    if ceil(x/10)==floor(x/10)
        disp([num2str(ceil(100*x/200)) '% done.']);
    end
    teller=0;
    noemer=0;
    for k=ceil(x/2):1:n      %k het aantal keren dat het deeltje vrij is geweest
        tellerk=0;
        for y=mod(x,2):2:k-abs(x-k)      %y de y-positie van een deeltje
            stukjespyexgk
            tellerk=tellerk+y^2*pygxgk;
        end
        teller=teller+2*tellerk*pkmatrixbin(k+1);
    end
    noemer=1;
    end
    vary(x+1)=teller/noemer;
end
X=(0:1:x); %X zorgt ervoor dat de waarden van vary op de goede x-positie
plot(X,vary) %worden geplott
YLABEL('Var(Sy|Sx=x)')
XLABEL('Sx=x')
toc
```

# Bibliografie

- [1] N. Madras. Lectures on Monte Carlo Methods. 29-42, 2002.
- [2] E. Omeij, J. Santos, S. Van Gulck. A Markov Binomial Distribution. *Applicable Analysis and Discrete Mathematics*, 38-42, 2008.
- [3] G. Uffink, A. Elfeki, M. Dekking, J. Bruining en C. Kraaikamp. Advection, Dispersion and Kinetic Adsorption in a Homogeneous Aquifer. Preprint 2008.
- [4] R. Viveros et al. Binomial and Negative Binomial Analogues Under Correlated Bernoulli Trials. *The American Statistician*, 1984.