# TECHNISCHE UNIVERSITÄT BERLIN

Fakultät V – Verkehrs- und Maschinensysteme – Institut für Mechanik

FG Systemdynamik
und Reibungsphysik

Dr.-Ing. Markus Heß
M.Sc. Justus Benad
Josefine Wilhayn

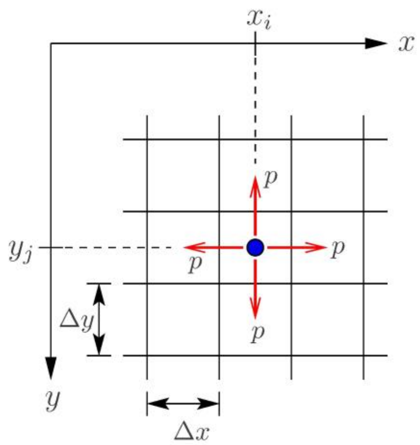
Numerische Simulationsverfahren im Ingenieurwesen

**Hausaufgabenblatt 3** WS 18/19

## Thema: Diffusion mit Block-CA

### Theorieaufgabe (6 Punkte): Modelle zur Abbildung von 2D-Diffusionsprozessen

Zur Simulation eines zweidimensionalen Diffusionsprozesses kann das gezeigte Modell genutzt werden. Es besteht aus einem quadratischen Gitter, dessen Zellen (binär) entweder durch ein Teilchen besetzt oder nicht besetzt sind. In jedem Zeitschritt können die Teilchen gleichwahrscheinlich in 4 Richtungen (Von-Neumann-Nachbarschaft) springen und dadurch ihre Position ändern.




a) Stellen Sie die Mastergleichung für die Wahrscheinlichkeitsdichte an einer beliebigen Position $(x_i, y_j)$ zur Zeit $t_k$ auf. Beachten Sie dabei, dass $p = 1/4$ gilt.

b) Entwickeln Sie die in Aufgabenteil a) aufgestellte Mastergleichung nach den kleinen Größen $\Delta x$ und $\Delta t$. Beachten Sie, dass $\Delta y = \Delta x$ gilt.

c) Vergleichen Sie Ihr Ergebnis mit der zweidimensionalen Diffusionsgleichung

$$\frac{\partial w(x,y,t)}{\partial t} = D\left(\frac{\partial^2 w(x,y,t)}{\partial x^2} + \frac{\partial^2 w(x,y,t)}{\partial y^2}\right)$$

und geben Sie an, durch welche numerischen Größen der Diffusionskoeffizient bestimmt wird.

**d) Zur Programmieraufgabe:**
Für den mittleren quadratischen Abstand bei einem zweidimensionalen Diffusionsprozess (unendlich ausgedehntes Gebiet vorausgesetzt) gilt

$$\left\langle r^2 \right\rangle = 4Dt\,.$$

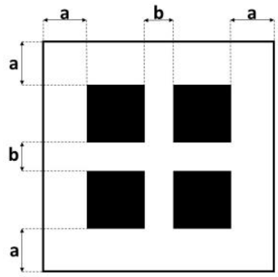
Vergleichen Sie diesen Zusammenhang mit dem Ergebnis aus Ihrer Simulation und geben Sie den Diffusionskoeffizienten an.

Abweichend von der Deadline für die Programmieraufgabe (03.12.) wird die **Abgabefrist für den theoretischen Teil** der Hausaufgabe bis **Donnerstag, den 06.12.2018 um $12^{15}$ Uhr** verlängert.

## Programmieraufgabe (14 Punkte): Diffusion

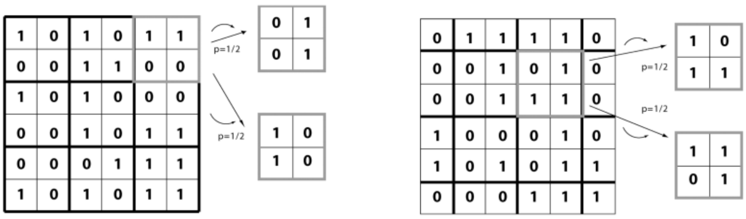
Schreiben Sie eine Funktion `NumSimHA3(n,J,k)`, welche bei $k = 1$ ein Gitter aus $n \times n$ Kästchen in Figure 1 ausgibt. Ein Kästchen kann dabei den Zustand 0 (weiß) oder 1 (schwarz) annehmen. Der Eingabeparameter $n$ soll gerade sein.

Als Startkonfiguration positionieren Sie in Figure 1 vier schwarze $\frac{n}{4} \times \frac{n}{4}$ Vierecke wie in der nebenstehenden Abbildung im Feld, wobei $a = \frac{3}{16}n$ und $b = \frac{n}{8}$ (runden Sie diese Angaben zur Darstellung auf dem $n \times n$ Gitter).




Die zeitliche Entwicklung mit insgesamt $J$ Zeitschritten soll folgenden Regeln unterliegen:

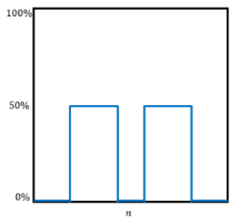
In einem Zeitschritt geschieht Folgendes: Die einzelnen $2 \times 2$ Blöcke des Gitters werden mit der Wahrscheinlichkeit $p = 0.5$ entweder nach links oder rechts gedreht, die Grenzen zwischen den Blöcken werden um eine Zeile und Spalte verschoben, und die sich ergebenen neuen Blöcke werden jeweils wieder mit der Wahrscheinlichkeit $p = 0.5$ entweder nach links oder rechts gedreht:




Verwenden Sie periodische Randbedingungen.

Nach jedem dieser Zeitschritte soll die sich ergebende Verteilung in der Figure 1 geplottet werden. Damit es fortlaufend zur Darstellung als Video kommt können Sie in Ihrer Zeitschleife z.B. einen Befehl wie `pause(10^-6)` verwenden. Bitte versuchen Sie ihren Code so zu optimieren, dass das Video mit dieser Art der Darstellung bei einem gegeben $n$ und $J$ so schnell wie möglich durchläuft.

Ist $k = 2$, soll ein anderes Video gezeigt werden:
Stellen Sie in Figure 1 für jeden Zeitschritt die relative Belegung der einzelnen Spalten der $n \times n$ Matrix (also die Summe aller belegten Zellen pro Spalte geteilt durch $n$) über der Spaltenposition dar. Bei der Anfangskonfiguration ergibt sich dann z.B. das folgende Bild (nicht geglättet):




Glätten Sie die Kurven etwas, indem Sie jedem Wert den Durchschnittswert seiner Nachbarschaft mit $r = 4$ zuweisen. (An den Rändern dieser Figure müssen Sie es nicht so genau nehmen, sie dürfen `xlim([10,n-10])` setzen um die Ränder nicht zu zeigen). Achten Sie jedoch auch in diesem Video auf eine effiziente Implementierung und wenden Sie ihr Wissen aus den vergangenen Hausaufgaben an.

Unabhängig von der Eingabe für $k$ soll nach Ablauf des jeweiligen Videos in Figure 1 auch noch eine weitere Figure 2 ausgegeben werden:

Stellen Sie in dieser den durchschnittlichen quadratischen Abstand aller schwarzen Teilchen von der Mitte des Gitters über der Zeit dar.

Allgemeines: **Verwenden Sie insgesamt nur eine Schleife** (Zeitschleife). Versuchen Sie, jede neue räumliche Verteilung innerhalb der Zeitschleife ohne weitere Schleifen, sondern ausschließlich mit Matrixoperationen zu berechnen. Versuchen Sie außerdem, jede neue räumliche Verteilung innerhalb der Zeitschleife ohne die Anwendung von Abfragen wie z.B. `if`, `else if`, `else` oder `switch`, `case`, `otherwise` zu erhalten. (Um die beiden Fälle $k = 1$ oder $k = 2$ für den Plot zu unterscheiden, darf natürlich eine Abfrage oder Ähnliches verwendet werden.)

*Hinweise zur Abgabe der Programmieraufgabe:*

Das Skript `NumSimHA3.m` bitte als Anhang einer E-Mail mit dem Betreff `NumSimHA3` an `j.benad@tu-berlin.de` senden.

Die Abgabedeadline ist der 3.12.2018 um $14^{15}$ Uhr.

Bitte in dem Skript die folgende Form verwenden:

```
% Nachname1    Matrikelnummer1   (Liste bitte alphabetisch nach Nachnamen ordnen)
% Nachname2    Matrikelnummer2
% Nachname3    Matrikelnummer3
% Nachname4    Matrikelnummer4

function NumSimHA3(n,J,k)

    % Hier den Code einfügen. Bitte gut kommentieren.

end
```