Physik am Komputer 27. Juni 2010

Übung 06

Michael Kopp

Aufgabe 1: Perkulation Die Aufgabe ist ralisiert in `perk04.cpp`; dazu wird `Spielfeld.h` benötigt. Ich habe gewisse Ansätze eingebgeneb, mit deren Hilfe man die Ausgabe auch mit `burnview.tcl` machen könnte (bspw ist die Funktion `Spielfeld::ausgabe(char)` schon so geschrieben, dass man mit dem Parameter `'t'` die Ausgabe mit `burnview.tcl` gestalten könnte) doch dies schien mir einfach nicht praktisch.

Aktuell gibt das Programm deshalb einfach auf Kommandozeile aus: Rauten `#` sind besetzte Felder und Ats `@` sind *brennende* Felder.

Die endgültige Ausgabe, ob eine Perkulation stattgefunden hat ist über `std::cerr` realisiert.

Die „normalen“ Zufallszahlen `rand()` sind hier wohl eigentlich ungeeignet weil sie wiederholende Muster in den Feldern erzeugen; ich verwende die Funktion `zufall()`, mit der man – wenn man will – einen anspruchsvolleren Generator verwenden kann um die Felder weniger gleichförmig zu verteilen.

Aufgabe 2: Perkulationsschwelle Verwendet man als Parameter fuer die Ausgabe `n`, so wird nichts außer 1 bzw 0 für Perkulation bzw. keine Perkulation ausgegeben.

Das Script `schwelle.sh` nutzt das aus und damit wurde Abb. 1 erzeugt. Die angefittete Funktion ist

$$f(x) = \frac{1}{1 + \exp(-106.239571482936 \cdot (x - 0.592395554506743))} \qquad (1)$$

womit man eindeutig sagen kann, dass die Perkulationsschwelle bei $0.5923\ldots$ liegt.

Aufgabe 3: Game of live Ich habe schon vorher das Programm `game01.cpp` geschrieben; es ist ein *Game of live* mit Ausgabe auf die Kommandozeile...

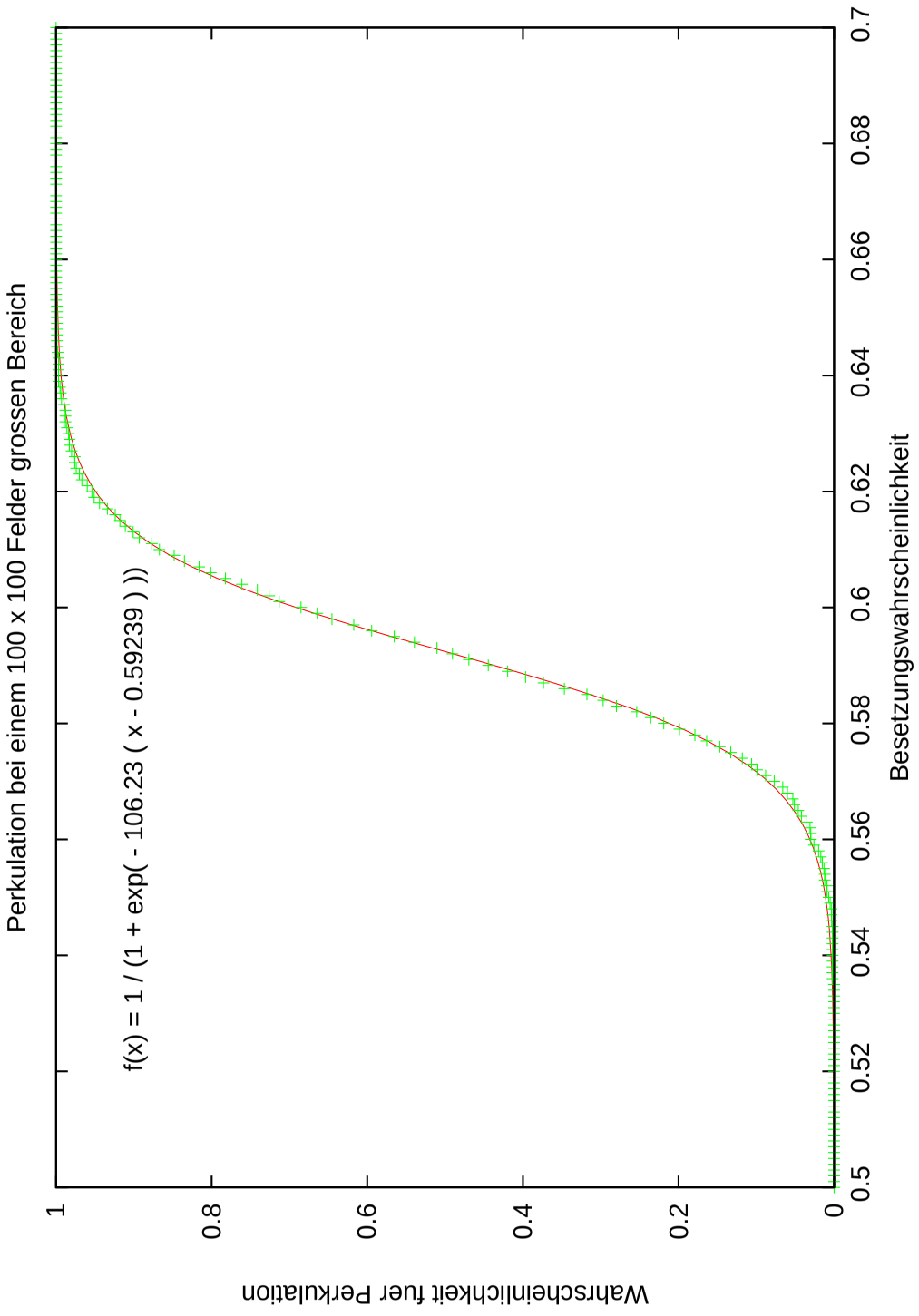

Abbildung 1: Perkulationswahrscheinlichkeit über Besetzungswahrscheinlichkeit