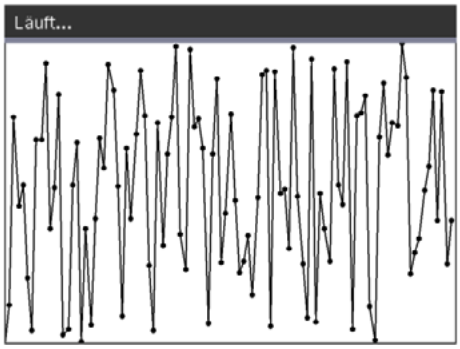
Spektralanalyse von Zufallszahlen

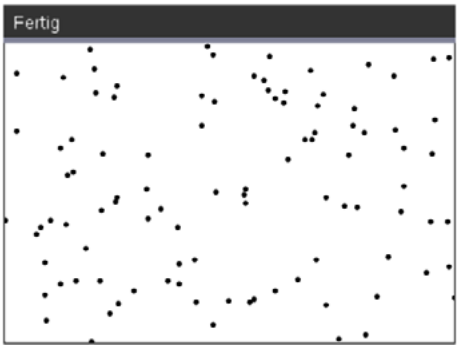
spektral.py, spektral1.py

Jetzt wollen wir aber die Qualität unseres Lehmer-Zufallszahlengenerators testen. Bei der Spektralanalyse wird der Generator verwendet, um Punkte in zwei, drei oder mehr Dimensionen zu erzeugen. Wie du siehst, das Programm ist sehr kurz:

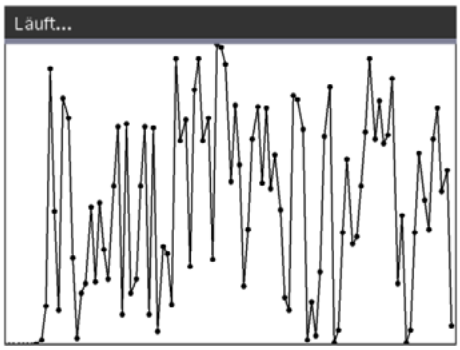
Nach dem Grafikmodul importieren wir auch unser in 2.2.5 hergestelltes Modul zufall. Nachdem in diesem die Prozedur rnd() aufgerufen und damit eine Folge von 100 Zufallszahlen gezeichnet wurde, hängen wir noch die Spektralanalyse an. Zuerst wird das Grafikfenster angepasst und anschließend 100 zufällige Koordinatenpaare erzeugt mit $(x,y) = (rnd(), rnd())$. plt.plot lässt zum Schluss auch diese Punkte zeichnen.

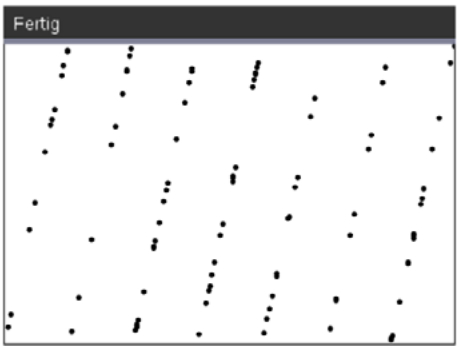
Für den MINST Zufallsgenerator nehmen wir die Werte $m = 2^{31} - 1$ und $a = 16807$ (2.2.5). Das führt zu den folgenden Ergebnissen, die deutlich machen, dass in beiden Fällen (ein- und zweidimensional) die Punkte zufällig erscheinen.





Dann ändern wir im Modul zufall (2.2.5) für a den Wert auf $a = 7$ in rnd(). Jetzt erscheinen die Punkte im ersten Bild noch immer zufällig, aber die 2D Spektralanalyse liefert deutlich eine zu große Gesetzmäßigkeit.





Der RANDU Generator von IBM hatte ein analoges Problem: die 3D Spektralanalyse ergab zu viel Struktur (Gesetzmäßigkeit).

Versuche auch eine Spektralanalyse der Zufallszahlen, die auf dem TI-Nspire™ CX II-TCAS (in der Calculator App) generiert werden, durchzuführen. (spektral1.py)