Санкт-Петербургский Государственный Электротехнический Университет "ЛЭТИ"

кафедра физики

Задание №2 по дисциплине

"Физические основы информационных технологий"

Название: Искажение сигналы в волноводе

Фамилия И.О.: Виноградов К.А.

группа: 6304

Преподаватель: Альтмарк А.М.

Итоговый балл:

Санкт-Петербург 2018

**Цель работы.**

Исследование искажения сигнала в волноводе.

**Задание.**

Радиоимпульсы (рис.1), предназначенные для последовательной передачи двоичного кода, распространяются через цилиндрический диэлектрический волновод (рис.2). Оценить дистанцию прохождения последовательности радиоимпульсов в цилиндрическом диэлектрическом волноводе, определяемую искажением сигнала из-за волноводной дисперсии. При расчетах учитывать только первую моду. Исходными данными являются параметры волновода (*R*- радиус волновода, -диэлектрическая проницаемость), параметры последовательности радиоимпульсов (,,) и порог, разделяющий уровень "0" и "1". Порог определяется следующей формулой:

,

где *S* - уровень сигнала между импульсами,  - максимальное значение сигнала внутри импульса





Рисунок.1



Рисунок.2

**Ход работы.**

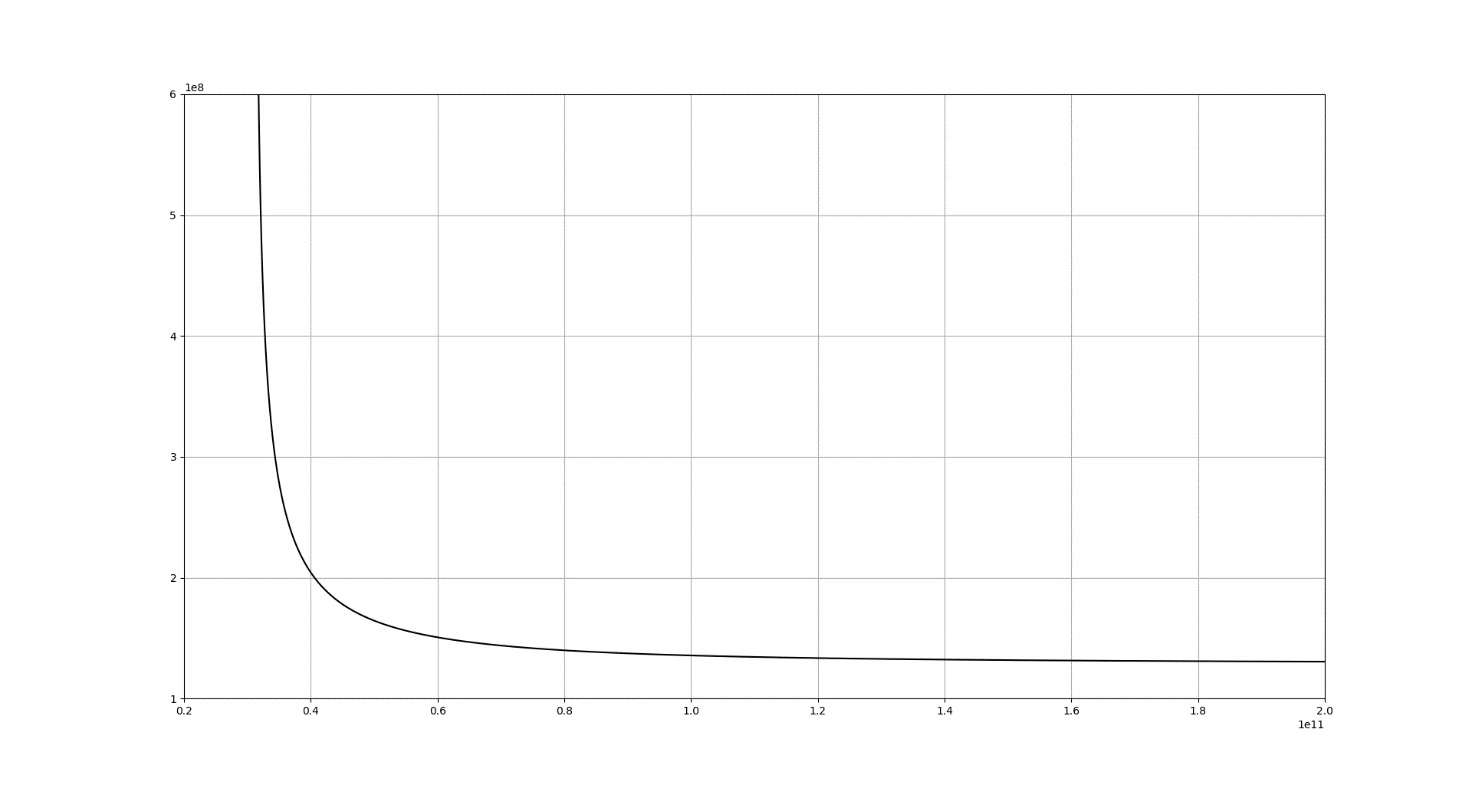
****

Рисунок 1 – График функции фазовой скорости от частоты

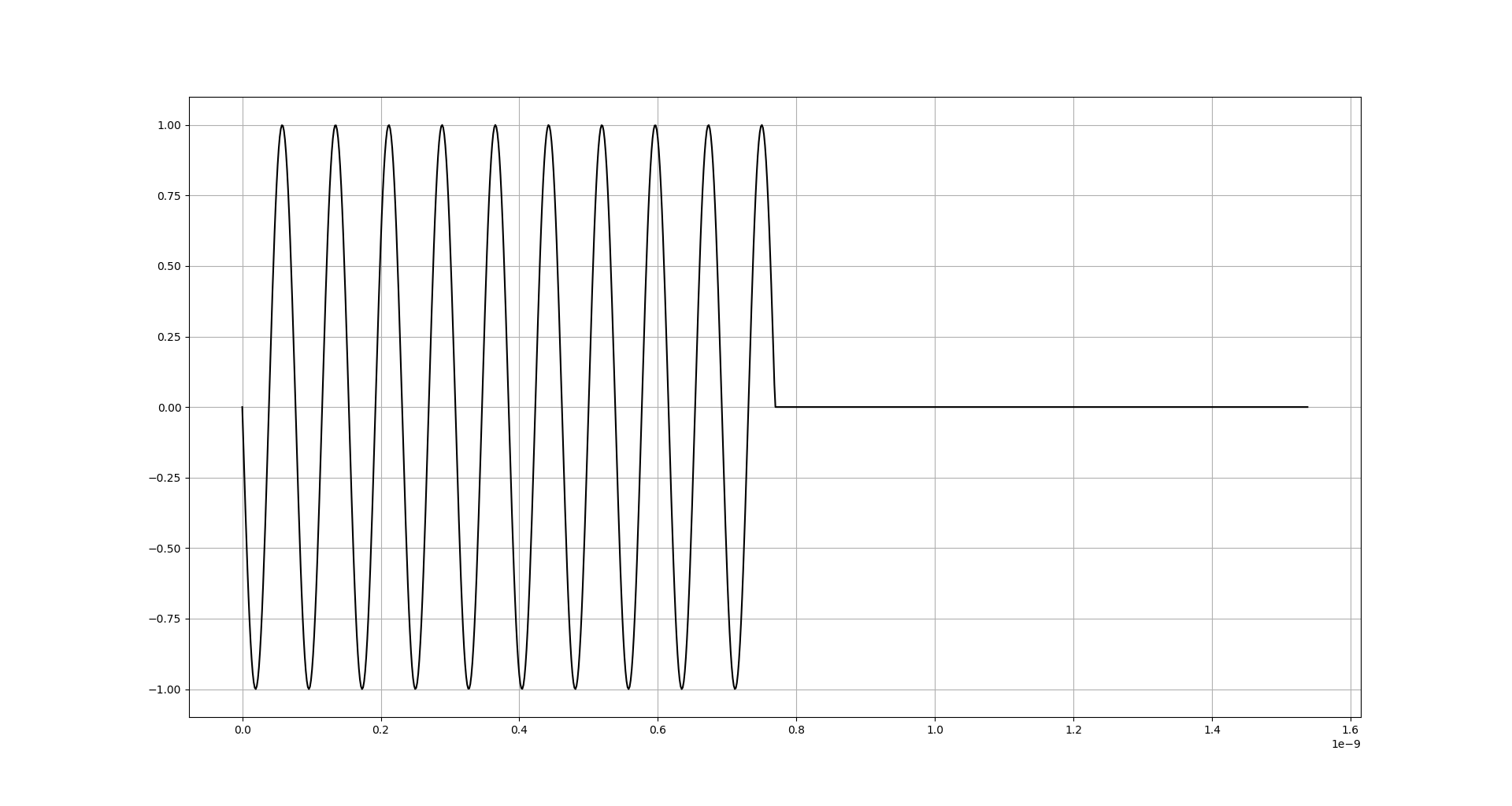
****

Рисунок 2 – Прямоугольный радиоимпульс

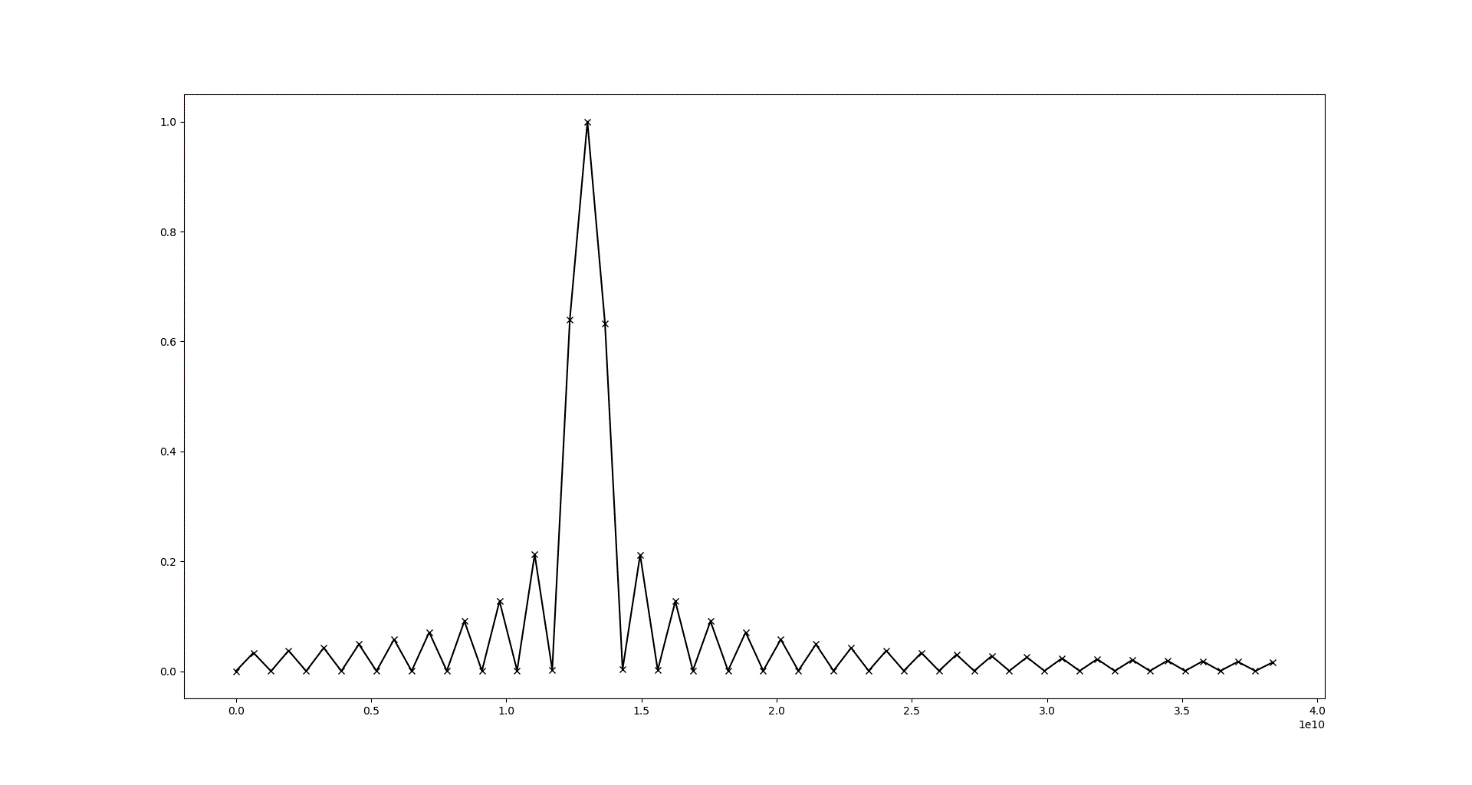


Рисунок 3 – Спектр

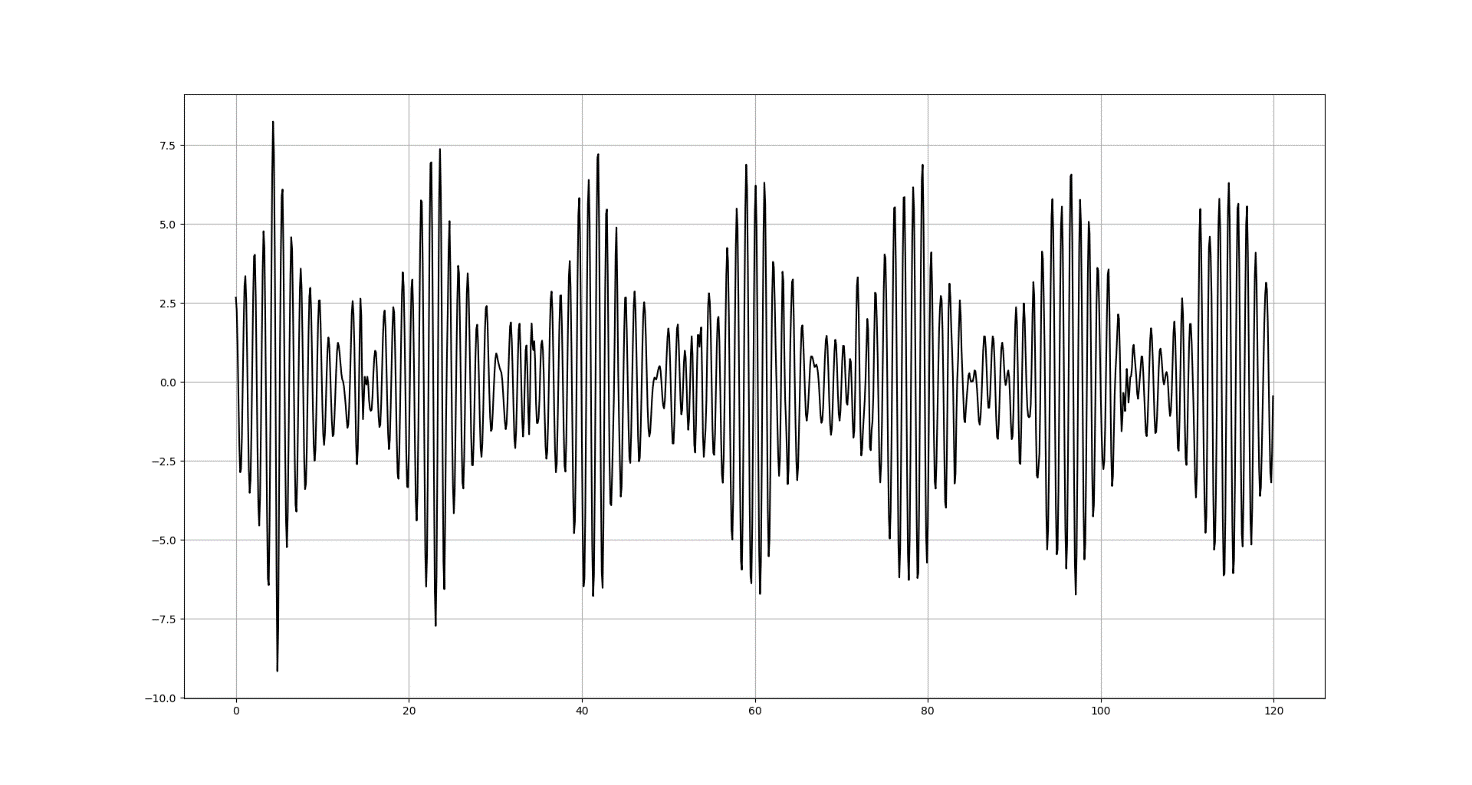


Рисунок 4 – Зависимость сигнала от расстояния

**Выводы.**

В результате выполнения данной лабораторной работы были построены графики фазовой скорости, спектра сигнала и зависимости сигнала от расстояния.

**ПРИЛОЖЕНИЕ А**

**КОД ПРОГРАММЫ**

from matplotlib import pyplot  
from matplotlib import mlab  
from scipy import special  
from scipy import optimize  
from scipy import constants  
import math  
import numpy  
  
  
R = 0.01  
eps = 5.4  
omg\_param = 2 \* math.pi \* 10 \*\* 9 \* 67  
frq = 13 \* 10 \*\* 9  
del\_t1 = 0.7692 \* 10 \*\* -9  
del\_t2 = 1.538 \* 10 \*\* -9  
del\_t3 = 0.07692 \* 10 \*\* -9  
P = 0.4  
N = 2 \*\* 10  
N1 = 60  
  
  
def get\_intervals(rel\_spd\_rng\_loc, bs\_rng\_loc, limit=0):  
 prev = bs\_rng\_loc[0]  
 intervals = []  
 for i in range(1, len(bs\_rng\_loc)):  
 if bs\_rng\_loc[i] \* prev < 0:  
 intervals.append([rel\_spd\_rng\_loc[i - 1], rel\_spd\_rng\_loc[i]])  
 if 0 < limit <= len(intervals):  
 break  
 prev = bs\_rng\_loc[i]  
 return intervals  
  
  
def phase\_spd\_func(omg\_loc, bs\_root\_loc):  
 temp = omg\_loc \*\* 2 \* eps / constants.speed\_of\_light \*\* 2 - bs\_root\_loc \*\* 2 / R \*\* 2  
 if temp <= 0:  
 return math.inf  
 else:  
 return omg\_loc / math.sqrt(temp)  
  
  
def group\_spd\_func(omg\_loc, phase\_spd\_loc):  
 if phase\_spd\_loc[0] \* phase\_spd\_loc[1] != 0:  
 temp = omg\_loc[1] / phase\_spd\_loc[1] - omg\_loc[0] / phase\_spd\_loc[0]  
 if temp != 0:  
 return (omg\_loc[1] - omg\_loc[0]) / temp  
 else:  
 return math.inf  
  
  
def kp\_func(rel\_spd\_loc, omg\_loc):  
 temp = eps - 1 / rel\_spd\_loc \*\* 2  
 if temp <= 0:  
 return math.inf  
 else:  
 return omg\_loc / constants.speed\_of\_light \* math.sqrt(temp)  
  
  
def bs\_func(rel\_spd\_loc, omg\_loc=omg\_param):  
 return special.j0(kp\_func(rel\_spd\_loc, omg\_loc) \* R)  
  
  
def check\_omg\_func():  
 i\_rng = range(0, 1000)  
 rel\_spd\_loc = [1 / math.sqrt(eps) + 4 / 1000 \* i for i in i\_rng]  
 bessel\_rng = [bs\_func(i, omg\_param) for i in rel\_spd\_loc]  
 root\_intervals = get\_intervals(rel\_spd\_loc, bessel\_rng)  
 rel\_spd\_roots\_loc = [optimize.bisect(bs\_func, i[0], i[1]) for i in root\_intervals]  
 return rel\_spd\_roots\_loc  
  
  
def heaviside\_func(x\_loc):  
 if x\_loc >= 0:  
 return 1  
 else:  
 return 0  
  
  
def f2(t\_loc):  
 return math.sin(frq \* 2 \* math.pi \* t\_loc)\*(heaviside\_func(t\_loc - del\_t1) - heaviside\_func(t\_loc))  
  
  
def mod\_sin(x\_loc):  
 if x\_loc == 0:  
 return 1  
 else:  
 return math.sin(x\_loc)/x\_loc  
  
  
def spectrum\_fun(f\_loc):  
 return mod\_sin((f\_loc - frq) \* 2 \* math.pi \* del\_t1 / 2) \* del\_t1 \* 9 \*\* 10  
  
  
def signal(t\_loc, z\_loc, frqss\_loc, spectre):  
 sgn = 0  
 for i in range(len(frqss\_loc)):  
 omg = frqss\_loc[i] \* 2 \* math.pi  
 phs\_spd\_loc = phase\_spd\_func(omg, sup\_roots[0])  
 sgn += spectre[i] \* math.cos(omg \* t\_loc - z\_loc \* omg / phs\_spd\_loc)  
 return sgn  
  
  
def max\_dist(distance\_loc, fk\_rng\_loc, spec\_rng\_loc):  
 max\_amplitude = 0  
 while max\_amplitude < P \* max\_deviation:  
 for j in range(distance\_loc, distance\_loc + 16):  
 amplitude\_current = math.fabs(signal(del\_t1 / 2, j / 100, fk\_rng\_loc, spec\_rng\_loc))  
 if amplitude\_current > max\_amplitude:  
 max\_amplitude = amplitude\_current  
 distance\_loc += 17.5  
 print("Max distance is", distance\_loc/100, "m")  
  
  
sup\_range = mlab.frange(0.0, 5000.0, 0.1)  
sup\_results = [special.j0(i) for i in sup\_range]  
sup\_intervals = get\_intervals(sup\_range, sup\_results, 1)  
sup\_roots = [optimize.bisect(special.j0, i[0], i[1]) for i in sup\_intervals]  
omg\_rng = mlab.frange(omg\_param / 1000, omg\_param \* 2, omg\_param / 10000)  
spd\_rng = [phase\_spd\_func(omg, sup\_roots[0]) for omg in omg\_rng]  
  
  
pyplot.plot(omg\_rng, spd\_rng, color="black")  
pyplot.xlim(0.2 \* 10 \*\* 11, 2 \* 10 \*\* 11)  
pyplot.ylim(1 \* 10 \*\* 8, 6 \* 10 \*\* 8)  
pyplot.grid()  
pyplot.show()  
  
  
time\_moments = [del\_t2 \* i/(N-1) for i in range(N)]  
imp\_rng = [f2(t) for t in time\_moments]  
pyplot.plot(time\_moments, imp\_rng, color="black")  
pyplot.grid()  
pyplot.show()  
  
  
fk\_rng = [i/del\_t2 for i in range(N1)]  
spec\_rng\_abs = [numpy.abs(spectrum\_fun(i)) for i in fk\_rng]  
specter\_range\_norm = spec\_rng\_abs / numpy.max(spec\_rng\_abs)  
pyplot.plot(fk\_rng, specter\_range\_norm, marker='x', color="black")  
pyplot.show()  
  
  
distances = [i / 10 for i in range(1200)]  
sig = [signal(del\_t1 / 2, i / 100, fk\_rng, spec\_rng\_abs) for i in distances]  
max\_deviation = numpy.max(sig[:10])  
  
pyplot.plot(distances, sig, color="black")  
pyplot.grid()  
pyplot.show()  
  
max\_dist(10, fk\_rng, spec\_rng\_abs)