Zadanie 1

W tym zadaniu wyliczam przybliżone pochodne:

(a) $D_h f(x) \equiv \frac{f(x+h)-f(x)}{h}$,

(b) $D_h f(x) \equiv \frac{f(x+h)-f(x-h)}{2h}$.

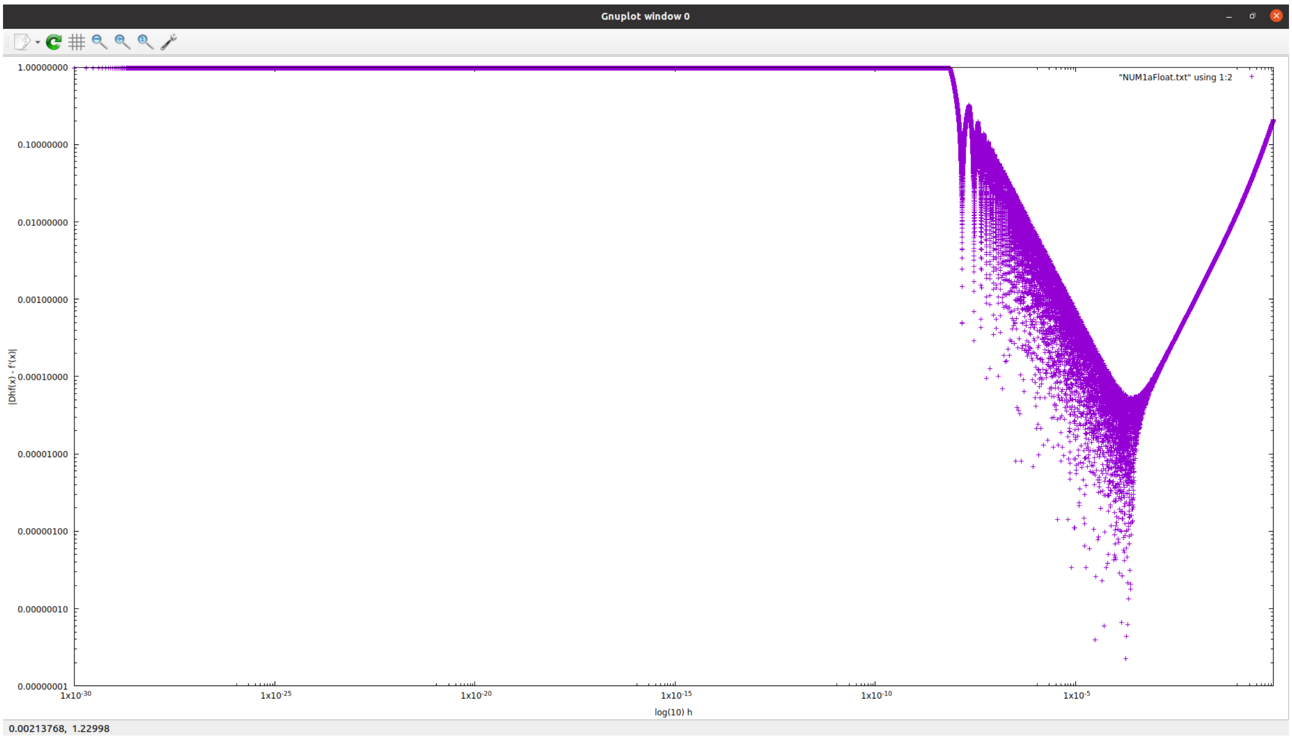
Dla f(x) = sin(x), x = 0.2 i dla zmiennego parametru h, używam także różnych typów float i double. Wyniki przybliżonych pochodnych znajdują się w plikach:

-przyblizonaPochodnaNUM1aFloat.txt
-przyblizonaPochodnaNUM1aDouble.txt
-przyblizonaPochodnaNUM1bFloat.txt
-przyblizonaPochodnaNUM1bDouble.txt

gdzie pierwsza kolumna to wartości h, a druga to przybliżone wartości dla funkcji sin(x).

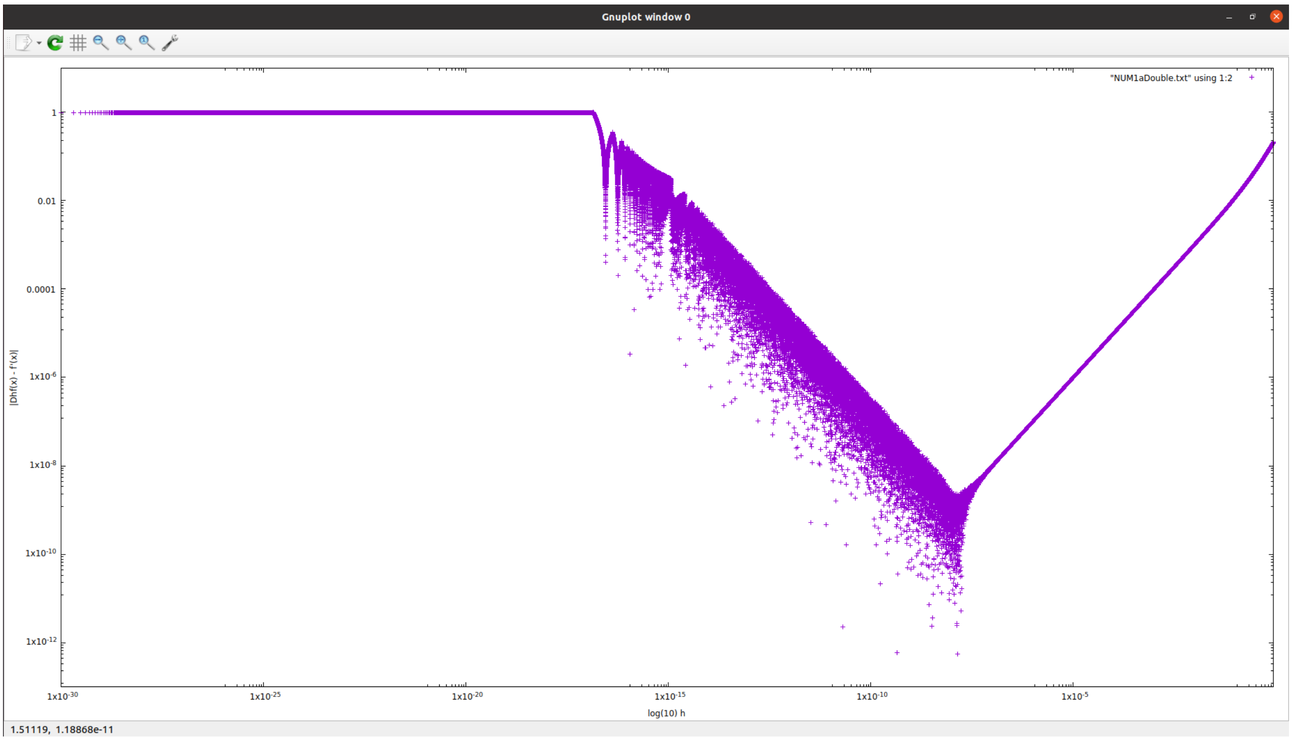
Następnie rysuję wykresy(w Gnuplot) błędów $|D_h f(x) – f'(x)|$. Kilka przykładowych wyników:

1. Dla podpunktu a i typu float:



Sposób uruchomienia: ***make aFloat***

2. Dla podpunktu a i typu double:

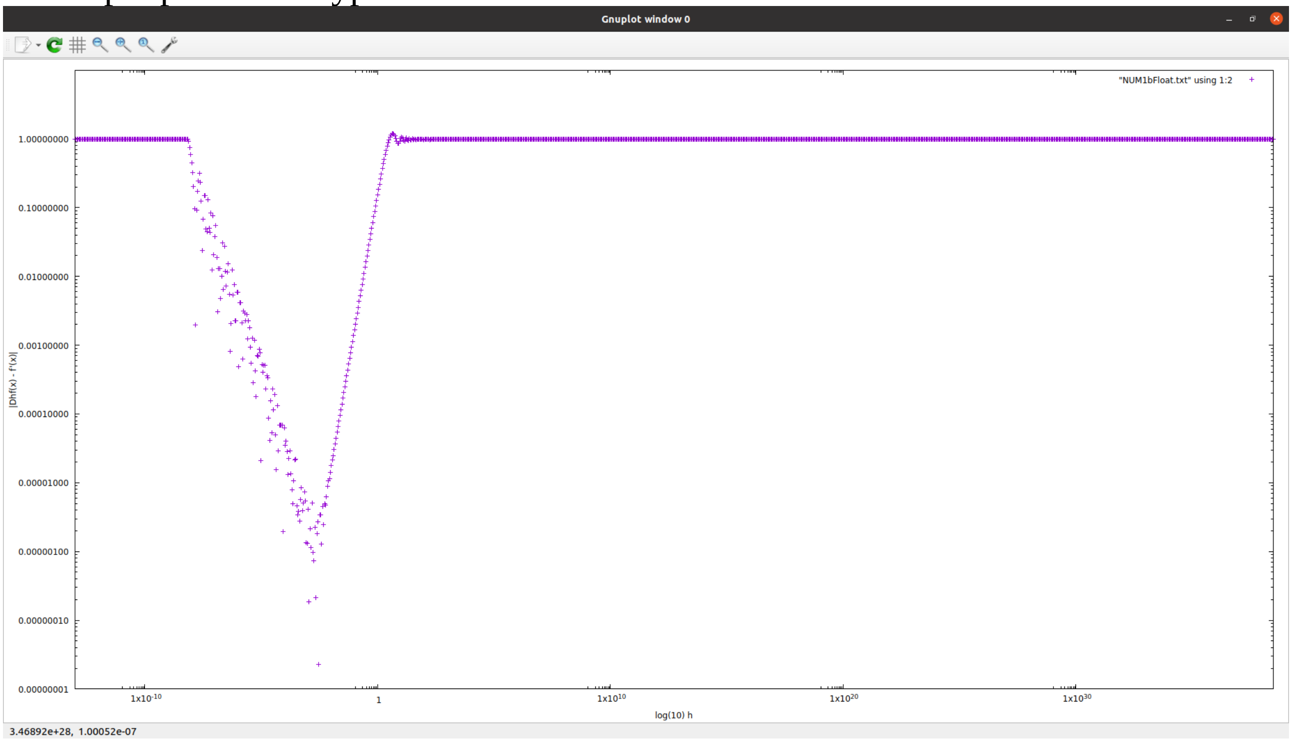


Sposób uruchomienia: ***make aDouble***
Jak widać dla obu podpunktów różnica między wykresami powstaje, ponieważ float jest precyzyjny do około 10^{-7}, a double do około 10^{-16}. Szum numeryczny po lewej stronie powstaje, ponieważ nie jesteśmy w stanie dokładnie reprezentować w komputerze wszystkich liczb(mamy ograniczoną pamięć), jest to tak zwany błąd zaokrąglenia .Te błędy dobrze widać po obliczeniu przybliżonych pochodnych:

```
0.000000007611094865467293857364 1.918794870376586914062500000000
0.000000007603484064588883484248 1.920715570449829101562500000000
0.000000007595880369137830712134 1.922638177871704101562500000000
0.000000007588284667292555241147 1.924562811851501464843750000000
0.000000007580696070874637371162 1.926489353179931640625000000000
0.000000007573115468062496802304 1.928417682647705078125000000000
0.000000007565541970677713834448 1.930348157882690429687500000000
0.000000007557976466898708167719 1.932280421257019042968750000000
0.000000007550418068547060101992 1.934214711189270019531250000000
0.000000007542867663801189337391 1.936150908470153808593750000000
0.000000007535324364482676173793 1.938089132308959960937500000000
0.000000007527789058769940311322 1.940029144287109375000000000000
0.000000007520260858484562049853 1.941971182823181152343750000000
0.000000007512740651804961089510 1.943915128707885742187500000000
0.000000007505227550552717730170 1.945861101150512695312500000000
0.000000007497722442906251671957 1.947808861732482910156250000000
0.000000007490224440687143214745 1.949758648872375488281250000000
0.000000007482734432073812058661 1.951710343360900878906250000000
0.000000007475251528887838503579 1.953664064407348632812500000000
0.000000007467776619307642249623 1.955619573593139648437500000000
0.000000007460308815154803596670 1.957577109336853027343750000000
0.000000007452848116429322544718 1.959536790847778320312500000000
0.000000007445395411309618793894 0.000000000000000000000000000000
0.000000007437949811617272644071 0.000000000000000000000000000000
0.000000007430511761441493945313 0.000000000000000000000000000000
0.000000007423081260782282697619 0.000000000000000000000000000000
0.000000007415658309639638900990 0.000000000000000000000000000000
0.000000007408242463924352705362 0.000000000000000000000000000000
0.000000007400834167725633960799 0.000000000000000000000000000000
0.000000007393433421043482667301 0.000000000000000000000000000000
0.000000007386039779788688974804 0.000000000000000000000000000000
0.000000007378653688050462733372 0.000000000000000000000000000000
0.000000007371275145828803943004 0.000000000000000000000000000000
0.000000007363903709034502753639 0.000000000000000000000000000000
0.000000007356539821756769015337 0.000000000000000000000000000000
0.000000007349183483995602728100 0.000000000000000000000000000000
0.000000007341834251661794041866 0.000000000000000000000000000000
0.000000007334492568844552806695 0.000000000000000000000000000000
0.000000007327157991454669172526 0.000000000000000000000000000000
0.000000007319830963581352989422 0.000000000000000000000000000000
0.000000007312511041135394407320 0.000000000000000000000000000000
```

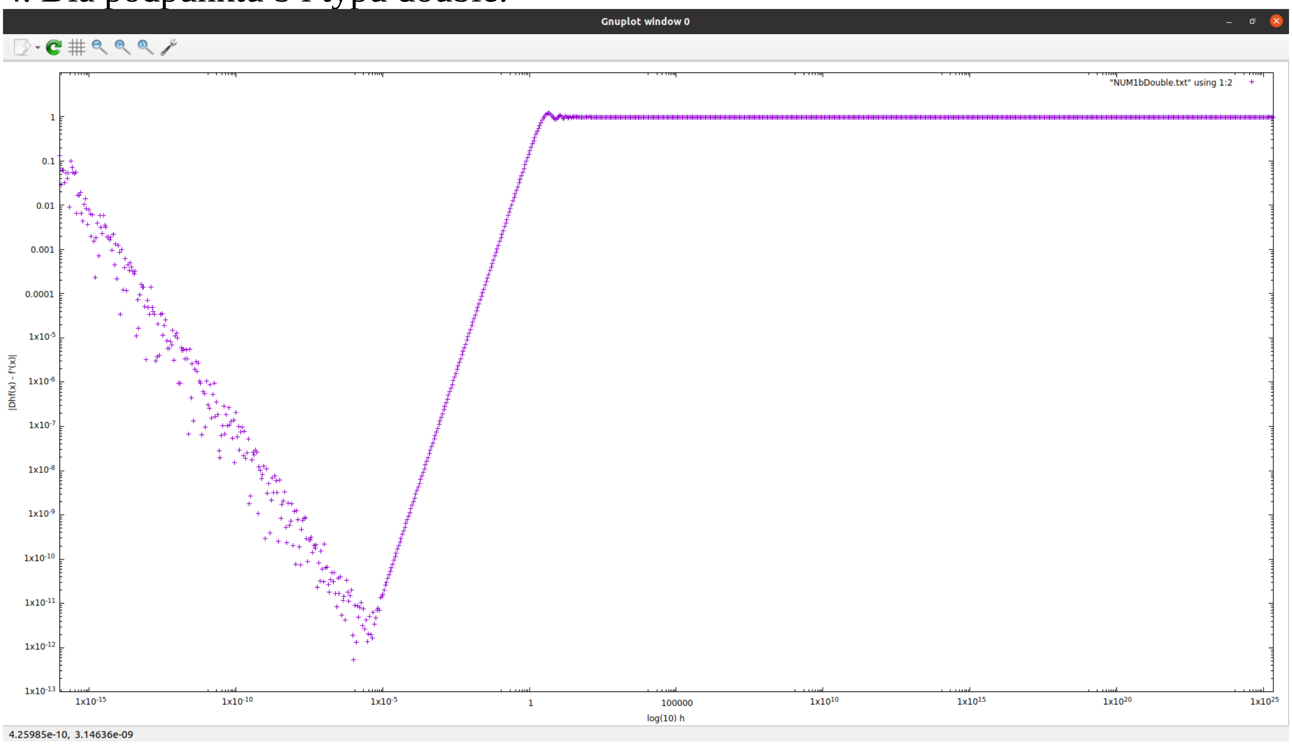
Jak wiadomo pochodną dla sin(x) jest cos(x). Zbiorem wartości dla funkcji cos(x) jest przedział <-1;1>, więc zdecydowanie wyszliśmy za zakres cosinusa. Widać także, że od pewnego momentu pojawiły się same 0, jest to właśnie koniec precyzji, w tym przypadku dla typu float.

3. Dla podpunktu b i typu float:



Sposób uruchomienia: ***make bFloat***

4. Dla podpunktu b i typu double:



Sposób uruchomienia: ***make bDouble***

Różnica pomiędzy tymi dwoma wynika z różnicy precyzji między typami float i double(tak jak w podpunkcie a). Szum numeryczny wynika także z błędu zaokrąglenia.