

計算物理学 第一単元レポート

61908697 佐々木良輔

2-A.

プログラムについて

図 1 に PAD 図を示す. PAD 図は PadTools (<https://naoblo.net/misc/padtools/>) を用いて作図した. このプログラムでは double precision 型の OnePlusEpsilon 変数に $1 + 2^{-n}$ を順次代入し, これが 1 と一致したときにループを抜ける. このとき $2^{-(n-1)}$ が計算機イプシロンになる. ソースコード A.1 に fortran によるソースコードを示す.

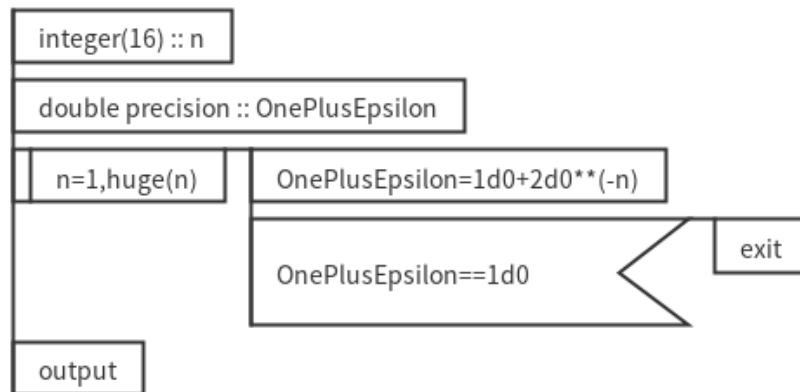


図 1 2-A. の PAD 図

出力結果

実行すると $n = 52$, $\varepsilon = 2.2 \times 10^{-16}$ という結果を得た. また OnePlusEpsilon を real 型に変え
ると $n = 23$, $\varepsilon = 1.2 \times 10^{-7}$ となった. これらは倍精度の指数部が 52 bit, 単精度の指数部
が 23 bit であることとも整合し, プログラムが正常に動作していると考えられる.[1]

参考文献

- [1] 電気通信大学情報基盤センター. 浮動小数点数の表現と誤差 - floating point number. <http://www.edu.cc.uec.ac.jp/~ta113003/htsecure/ref/FloatingPoint.html>. (Accessed on 10/10/2021).

付録 A ソースコード

ソースコード A.1 2-A のソースコード

```
1 program main
2   integer(16) :: n
3   double precision :: OnePlusEpsilon
4   !real :: OnePlusEpsilon
5   do n = 1, huge(n)
6     OnePlusEpsilon = 1d0 + 2d0**(-n)
7     if (OnePlusEpsilon == 1d0) exit
8   end do
9
10  write(*, '(a)', advance='no') "n = "
11  write(*, '(i0)', advance='no') (n - 1)
12  write(*, '(a)', advance='no') ", epsilon = "
13  write(*, '(E8.2)') (2d0**(-(n - 1)))
14
15 end
```
