Sprawozdanie
Obliczenia naukowe - lista 1

Kamil Król

1 Zadanie 1

1.1 MachEps

Epsilonem maszynowym macheps (ang. machine epsilon) nazywamy najmniejszą liczbę macheps większą od 0 taką, że fl(1.0 + macheps)>1.0.

W celu wyznaczenia metodą iteracyjną *macheps* zgodnego z powyższą definicją napisałem program, a jego wyniki zamieściłem w poniższej tabeli. Zgodnie z treścią zadania program uruchomiłem dla typów Float16, Float32 oraz Float64 i porównałem z wartościami zwracanymi przez funkcję *eps* dla każdego z typów.

Iteracyjne wyznaczanie macheps

	obliczony *macheps*	eps(type)	wartość z pliku float.h
Float16	0.000977	0.000977	xd
Float32	1.1920929e-7	1.1920929e-7	xd
Float64	2.220446049250313e-16	2.220446049250313e-16	xd

Okazało się, że wartości *macheps* wyznaczone przeze mnie są równe wartościom zwracanym przez wbudowaną w język Julia funkcją *eps*.

W treści zadania pojawia się pytanie: jaki związek ma liczba *macheps* z precyzją arytmetyki (oznaczaną na wykładzie przez ϵ)? W celu odpowiedzi na to pytanie przytoczę najpierw definicję precyzji arytmetyki - ϵ. Jest to największy błąd względny reprezentacji liczby jaki możemy popełnić i dla liczb reprezentowanych zgodnie ze standardem IEEE-754 wyraża się on wzorem: 2^{-t}. Podstawiając do wzoru dla arytmetyki Float32 mamy:

$$\epsilon = 2^{-24} = 0.5 \cdot 2^{-23} = \frac{1}{2} \cdot macheps$$

Wartość *macheps* dla Float32 w tabeli tj. 1.1920929e-7 jest zaokrąglona. Jej dokładna wartość wynosi: 1.1920928955078125e-7 co jest równe 2^{-23} (stąd równość). *Macheps* jest w komputerze przechowywany dokładnie. Wykonując to rozumowanie dla wszystkich typów widzimy zgodność i otrzymujemy zależność: $macheps = 2\epsilon$.

1.2 Eta

Kolejnym zadaniem jest wyznaczenie liczby *eta* takiej, że $eta > 0.0$ dla wszystkich typów zmiennopozycyjnych Float16, Float32, Float64. Wyniki napisanego przeze mnie programu, który iteracyjnie wyznacza te liczby, umieściłem w poniższej tabeli. Ponadto wartości otrzymanych liczb *eta* porównałem z wartościami zwracanymi przez funkcje: *nextfloat*(Float16(0.0)), *nextfloat*(Float32(0.0)), *nextfloat*(Float64(0.0)

Iteracyjne wyznaczanie eta

	obliczona *eta*	nextfloat(type)
Float16	6.0e-8	6.0e-8
Float32	1.0e-45	1.0e-45
Float64	5.0e-324	5.0e-324

Wartości obliczone przeze mnie okazały się takie same jak zwrócone przez funkcje wbudowane w język Julia. Kolejnym pytaniem jest: Jaki związek ma liczba *eta* z liczbą MIN_{sub}? TO DO
Innym pytaniem z treści zadania jest: co zwracają funkcje *floatmin*(Float32) i *floatmin*(Float64) i jaki jest związek zwracanych wartości z liczbą MIN_nor? TO DO

1.3 Liczba MAX

Kolejnym zadaniem do zrobienia było wyznaczenie (iteracyjnie) liczby *MAX* dla wszystkich typów zmiennopozycyjnych Float16, Float32, Float64 i porównanie wyników z wartościami zwracanymi przez funkcje: *floatmax*(Float16), *floatmax*(Float32), *floatmax*(Float64) oraz z danymi zawartymi w pliku nagłówkowym float.h dowolnej instalacji języka C. Liczbę *MAX* interpretuję jako największą wartość jaką można przechować w danym typie zmiennoprzecinkowym. Przy wyznaczaniu tej wartości musiałem pamiętać aby mantysa była wypełniona jedynkami. By to uzyskać postanowiłem wziąć liczbę *zaraz przed* liczbą 2.0 czyli 2.0 - *macheps*. Ten rezultat mogłem uzyskać też biorąc liczbę *zaraz przed* 1.0, wtedy byłoby to $1.0 - \frac{macheps}{2}$.

Iteracyjne wyznaczanie liczby *MAX*

	Obliczony *MAX*	maxfloat(type)	wartość z pliku float.h
Float16	6.55e4	6.55e4	xd
Float32	3.4028235e38	3.4028235e38	xd
Float64	1.7976931348623157e308	1.7976931348623157e308	xd

Ponownie wartości wyznaczone przeze mnie okazały się takie same jak te wyznaczone przez funkcje z języka Julia.

2 Second Section

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisissem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi necante...