Family Name: Du

Given Name: Min

Student Number: 1002602230

1.

Compute the results of the following five arithmetic operations on each computer system:

In first computer systems that uses fixed-point decimal arithmetic:

(260 + 1.27) + 0.1

= 261.27 + 0.1

= 261.37

260 + (1.27 + 0.1)

= 260 + 1.37

= 261.37

3.08 × 0.1

= 0.308

Rounded to 0.31

3.08 × 0.01

= 0.0308

Rounded to 0.03

−900 – 100

= -1000

underflow

Rounded to -999.99

In second computer systems that uses floating-point decimal arithmetic:

(260 + 1.27) + 0.1

=261.27 + 0.1

=261 + 0.1

=261.1

=261

= 0.261×

260 + (1.27 + 0.1)

=260 + 1.37

=261.37

=261

=0.261 ×

3.08 × 0.1

=0.308

=0.308×

3.08 × 0.01

=0.0308

=0.308 ×

−900 – 100

= -1000

= -0.100 ×

What is the range of representable numbers in each of the two systems?

In first computer systems that uses fixed-point decimal arithmetic:

[-999.99, 999.99]

In second computer systems that uses floating-point decimal arithmetic:

[-0.999 × , -0.100× ] {0} [0.100× , 0.999 ×

What are the ranges of numbers for which underflow happens in each of the two systems?

In first computer systems that uses fixed-point decimal arithmetic: (-,-999.99)

In second computer systems that uses floating-point decimal arithmetic:

(-0.100× , 0 .100× )

2.

(a) Find the condition number of , and study for what values of x in IR the function f (x) is ill-conditioned:

Condition Number: = ||= ||=

When |x| :

We have , (by de l’Hospital’s rule),

Then , which is small, well-conditioned.

When x

We have ,

Then the condition number is very large, ill-conditioned.

When x :

We have ,

Then the condition number is very large, ill-conditioned.

Therefore, f(x) is ill-conditioned, when x = and

(b) f(x) =

Notice that we subtract two equal numbers( catastrophic cancellation), which implies that the numeric expression is unstable.

Alternative expression:

is the equivalent mathematical expression, it avoids subtracting two completely equal number. This expression is more stable for x close to 0.

(c)

Matlab script:

See attachment for code and output.

Comments:

For i = 2,3,4,8,13, the relative error is relative larger to the others, especially at i = 13, the expression produces a poor approximation.

For i = 1 and i > 5 (except the i value above) the expression produces a very accurate approximation, the relative error is very small.

3.

(a) f(x) = cos(x) = about point 0

= 1

The remainder theorem: . Let k = 5; a = 0.

Then , where c lies between 0 and x.

(b) = 1 (the approximate value of cos(1))

, where c lies between 0 and x = 1.

We do not know the exact value of c, bus it could get value at x =0, where cos(0) = 1 is the maximum value of cosine function, then cos(c) ≤ 1.

Therefore, the upper bound for the error of the approximation to cos(1) is = 0.00139.

(c) f(x) = cos(x) = about point 0

1 =

Note that all the odd powers of x in the Taylor polynomial will disappear as their coefficient is 0 due to the property of cos function.

, where c lies between 0 and x.

Note that since every time when 2n+2 is even, .

We do not know the exact value of c, but since it will depend on x, no matter what c is, it will be that cos(c) ≤ 1.

Then , the upper bound for the error of the approximation to cos x is .

(d)

(i) when x =2: 1 =

, where c note that .

Then cos(c) ≤ 1, and .

We would like , then .

Then we find the least n to let the remainder below is n = 11.

1 = is the efficient and accurate approximation to cos(2).

(ii) when x = 4, 1 =

, where c lies between 0 and 4, note that .

Then cos(c) ≤ 1, .

We would like , then .

Then we find the least n to let the remainder below is n = 15.

1 = is the efficient and accurate approximation to cos(4).

(iii) when x =6, 1 =

, where c lies between 0 and 6, Note that .

Then cos(c) ≤ 1, .

We would like , then .

Then we find the least n to let the remainder below is n = 19.

1 = is the efficient and accurate approximation to cos(6).

(iv) when x is large, 1 = .

, where c lies between 0 and x.

Since x is large, let x , value is a number belongs to , by cyclic property of cos. cos(x) = cos().

Then, we would like to calculate cos(value) now.

Then = .

Then c lies between 0 and value, cos(0) = 1 is the maximum value of cosine function by property of cos.

Then by cyclic property of cos, the least term of n at x is the same as the least term of n at value.

Then we get n.

Then is the efficient and accurate approximation to cos(x). value .

(v) when x is negative, 1 = .

According to the property of cos: cos(x) = cos(-x).(x is negative, -x is positive)

Let -x , cos(-x) = cos( value + ), by cyclic property of cos.

We would like to calculate cos(value) now.

Then .

Then c lies between 0 and value, cos(0) is the maximum value.

Then , by cyclic property of cos, the least term of n at x is the same as the least term of n at value.

Then we get n.

Then is the efficient and accurate approximation to cos(x), value .

Generalization, pseudo-code:

Approx\_to\_cos(x):

If x == 0 then

Return 1

Else if x == or x == then

Return 0

Else if x ==  then

Return -1

Else if x and x != 0, , , then

n = 9+2(x-1)

return

else if x then

return

else (x )

return

end if

(e) pseudo-code for

Mycos(x, n):

Sum = 1

for i in 0, 1,…, n:

for a in 0, 1, 2…2i:

numerator \*= x

denominator \*= a

if i % 2 == 0 then

sum += numerator / denominator

else

sum += - (numerator / denominator)

end if

return sum

4.

Let A = be a strictly column diagonally dominant matrix, to find the determinant of A, we would like to reduce matrix A to an upper triangular matrix.

First reduce the entry to 0 by Gauss Elimination: , , , …, .

Let be the coefficient after reducing the entry , j > 1.

(\*)

Let’s compute . (a)

Since A is strictly column diagonally dominant matrix, then

(a)

Combining (a) and (b), we get

Then by (\*),

Since , then .

Then the determinant of reduced matrix is the product of diagonal elements.

Moreover, since then det(A) = ….

Therefore, strictly column diagonally dominant matrix is non-singular.