Scientific Computing Hw 1

Runze Fang

September 21, 2020

1 Question 1

1.1 Overflow test

When the precision is double, the upper bundary is 2 to the 1023. If the number keep growing, there will be an overflow.

When the precision is single, the upper bundary is 2 to the 127. If the number keep growing, there will be an overflow.

1.2 Division test

10 divided by 0 is infinity.
10 divided by infinity is 0.
0 divided by 0 is NaN.

1.3 NaN

10 divided by NaN is NaN.
NaN divide 10 is NaN.
10 add NaN is NaN.
2 to the NaN exponential is NaN.

1.4 Comparison between Nan, number and infinity

Number and infinity cannot compared with NaN. Only can use function IsNaN to compare between NaN and non-nan variables.
Number and infinity is comparable. Number is always smaller than infinity.

1.5 Signed Zeros

There is no negative zero in matlab.
If we try to mannually generate a -0 by using 1/-inf, the output is 0.
By the way, 0 is represented a positive number in matlab.

2 Question 2

2.1 Source of error

For both double and single precision, I use a for loop to iterate 30 times to see when the number will converge.
As shown in the Figure 1, in double precision, the number begin not to close to π in the 19th iteration. The last calculated number is 2.7911 in the 25th position.
And the number starting equal to 0 from the 26th iteration.
In single precision, the number begin not to close to π in the 6th iteration. The last calculated number is 3.0411 in the 11th position.
And the number starting equal to 0 from 12th iteration.

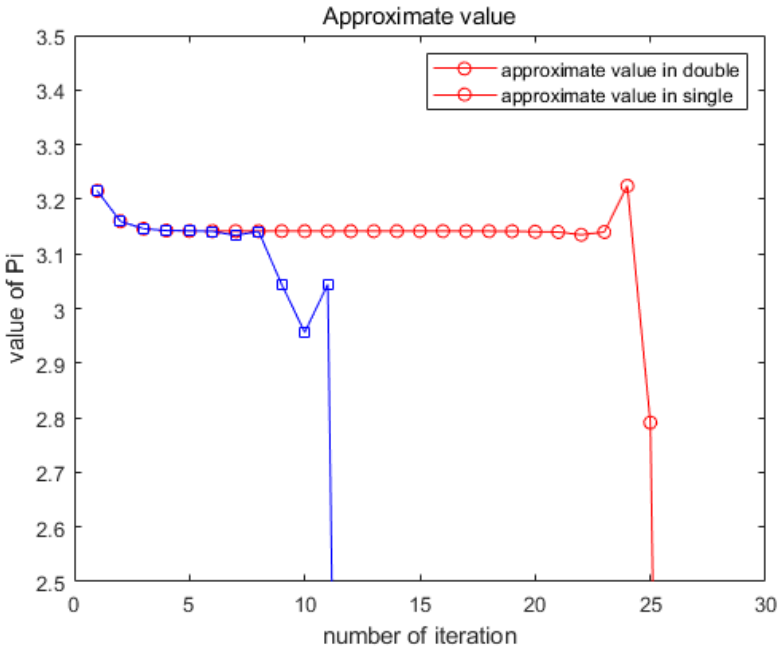

Figure 1: estimated pi in double and single precision

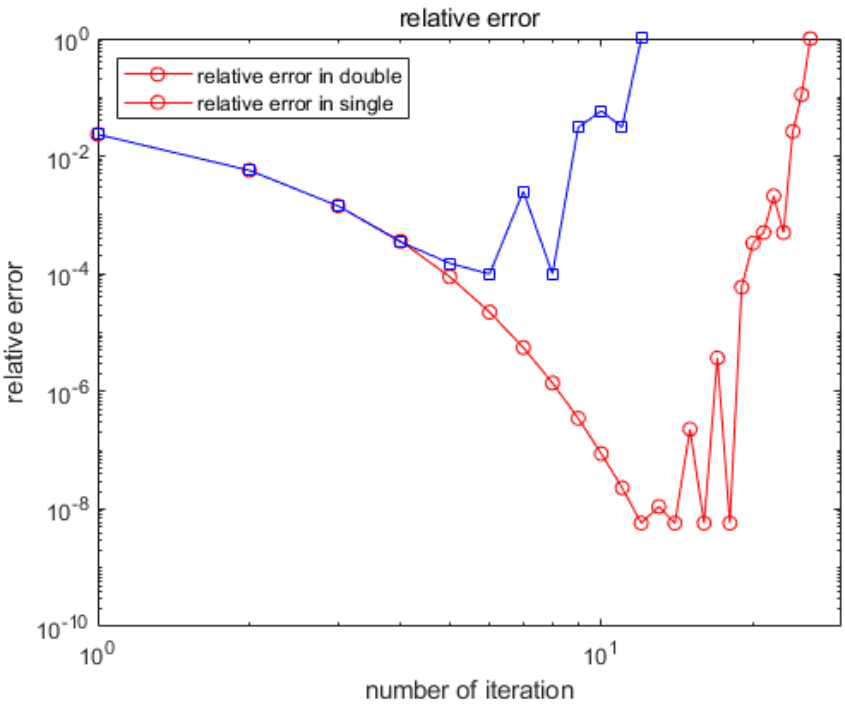

Figure 2: relative error in double and single precision

As shown in Figure 2, in single precision, the relative error goes bigger after 5th iteration, and get us about 3 digits accuracy.
In double precision, the ralative error goes bigger after 12th iteration, and get us about 8 digits accuracy.
When i is large, the roundoff error caused by catastrophic cancellation may cause the error.
By subtraction t^2 from 1, since 1 and t^2 has widely gap in magnitude, the result lost most of digits and equals 1.
There is also roundoff error when matlab store the value of π.

2.2 The Fix

Rewrite the equation $t_{i+1} = \frac{\sqrt{1+{t_i}^2}-1}{ti}$ to $t_{i+1} = \frac{1}{\sqrt{1+\frac{1}{t_i}^2}+\frac{1}{ti}}$
By doing so, we can avoid the catastrophic cancellation caused by subtracting a small number from 1.

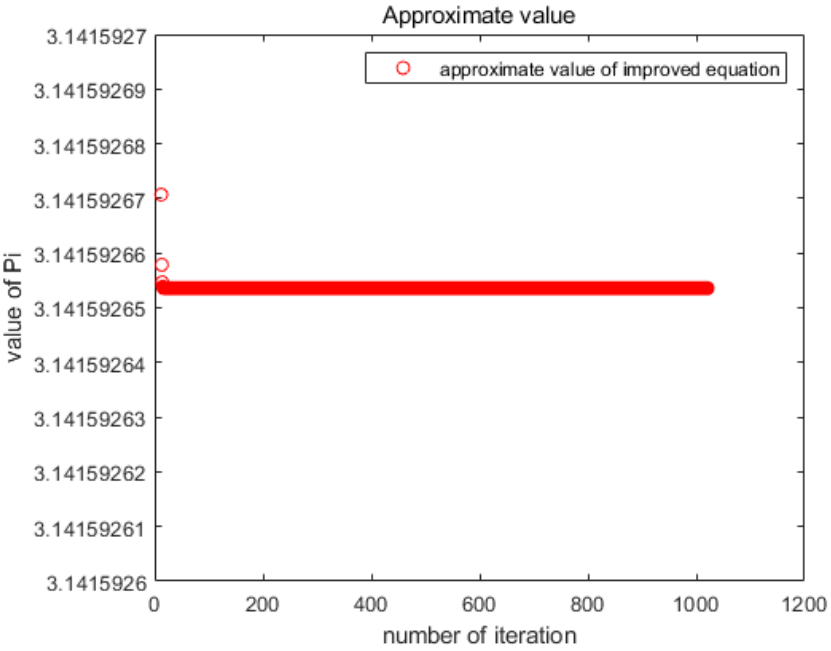

Figure 3: new pi by improved equation

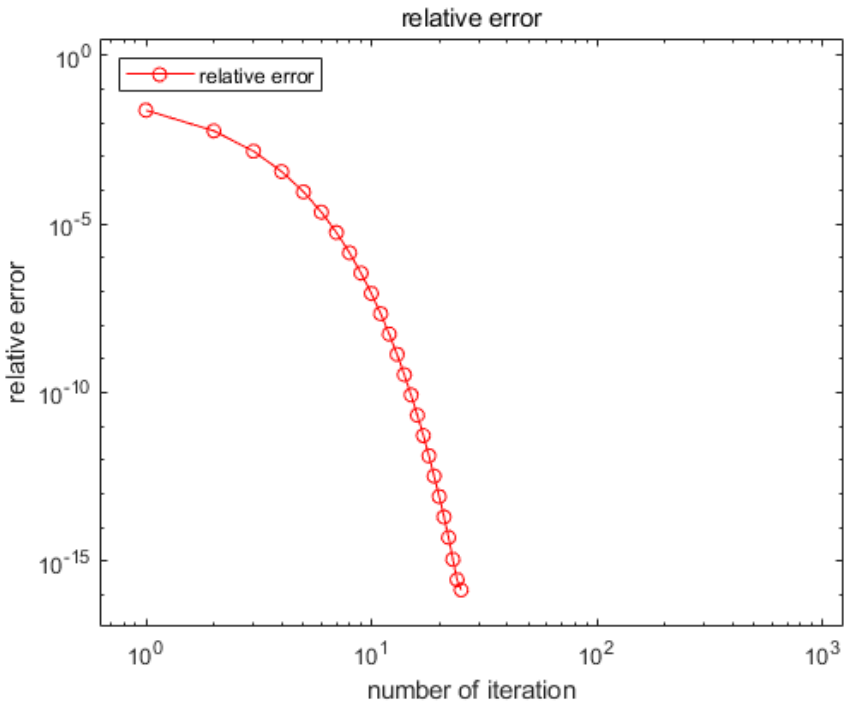

Figure 4: new relative error by improved equation

As shown in Figure 3 and Figure 4, we can almost get the accurate value of π using the improved equation.
We can get 16 digits of accuracy, which is touching the limit of double.
This time we are gonna have the roundoff error not because of the part of t in the equation, but because of the $6 \cdot 2^i$ part.
After iterating 1022 times, this part is going to overflow double.

3 Question 3

3.1 Numerical Troubles

I chose x=$2.5\times10^{-10} \times n$ as my iteration equation. The initial number is small, and will grow to the next magnitude every 20 iteration.
I did the iteration 100 times.

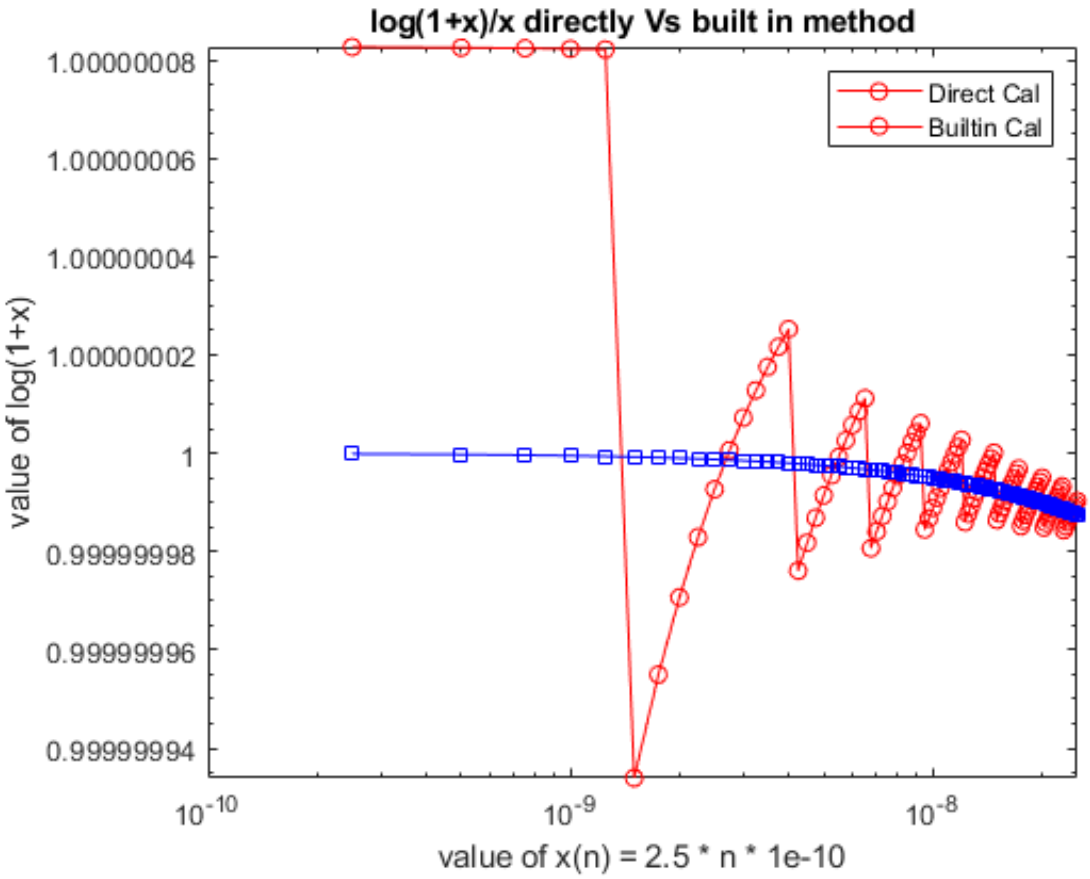

Figure 5: value by direct calculation and builtin method

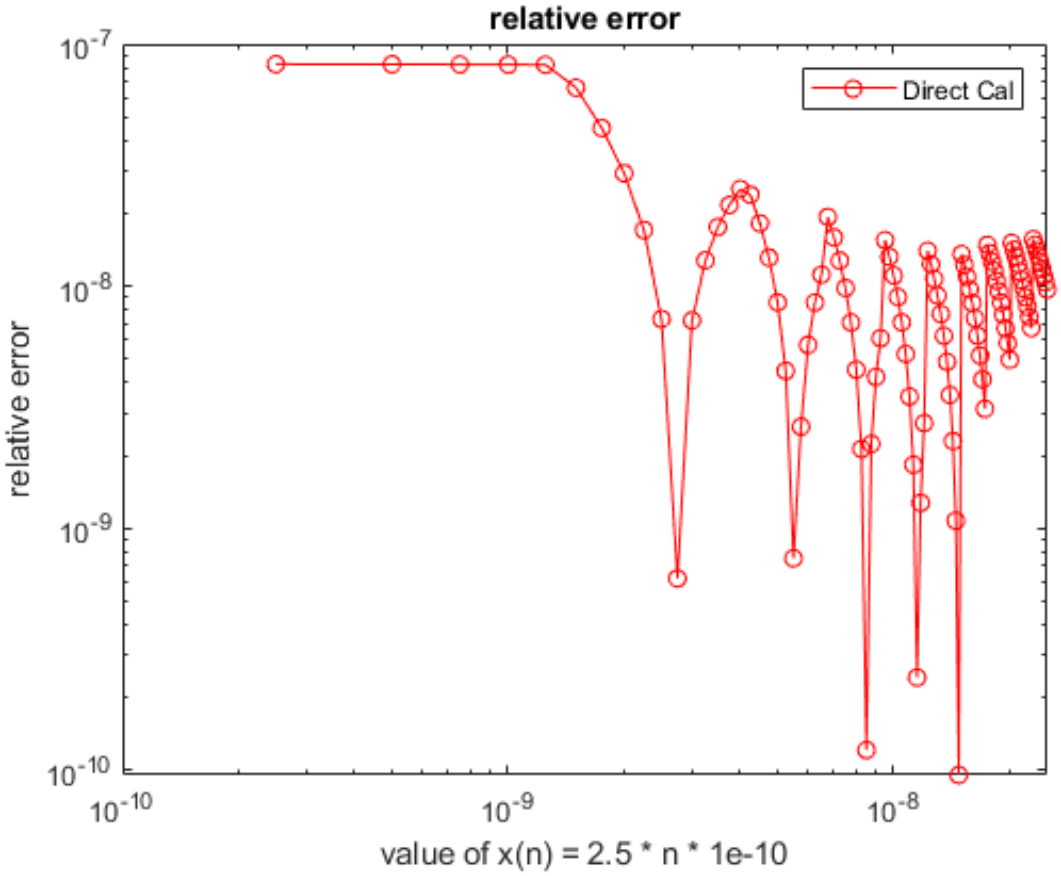

Figure 6: relative error of direct calculation

Compared with the built in log1p method, direct calculation is unstable. It can not compute after 10^{-8} magnitude.
The built in method is stable all the time.

3.2 Taylor Approximation

I do the calculation using the first 3 terms in Taylor series,
where $log(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3}$.

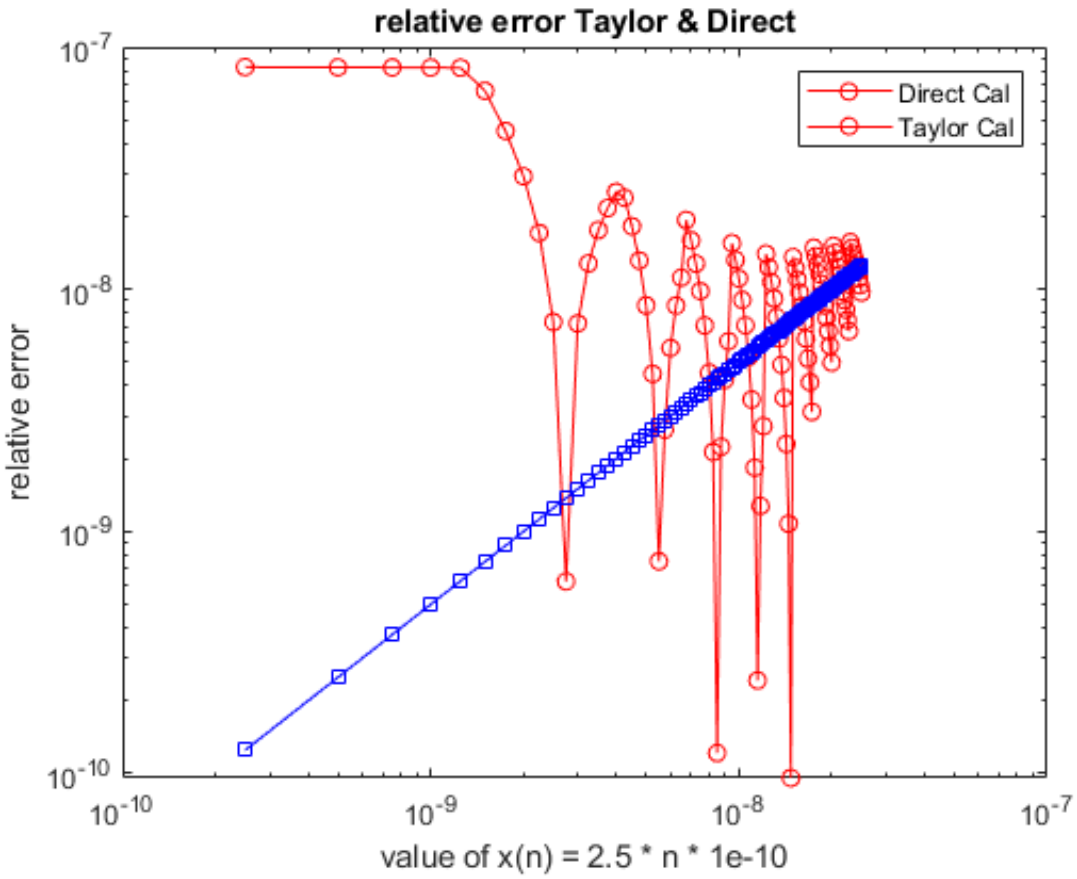

Figure 7: relative error in taylor approximation

As shown in Figure 7, we can see that after 10^{-7}, the direct computing method start to lose digits, while with the number is getting smaller, taylor method become more accurate.
Therefore, we can estimate that when x0 $<$ 1e-6, the taylor truncation is better, when x0 $>$ 1e-6, using direct computation is more convenient.

4 Question 4

4.1 Numerical Errors

Computing in double precision

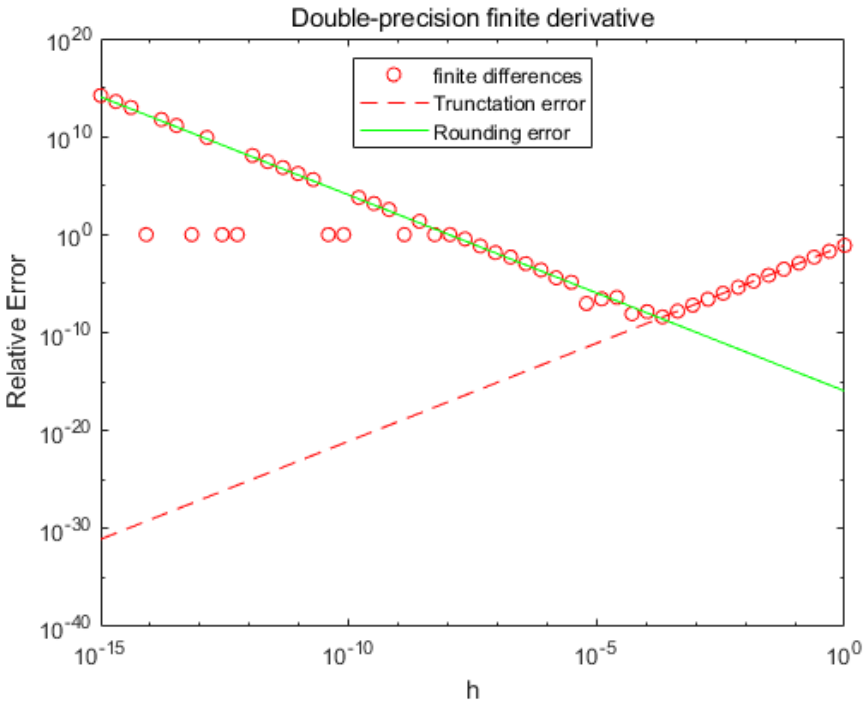

Figure 8: double precision

As shown in figure 11, we get the most accurate answer when h is around 10^{-4}, and we get about 9 digits of accuracy.
Computing in single precision

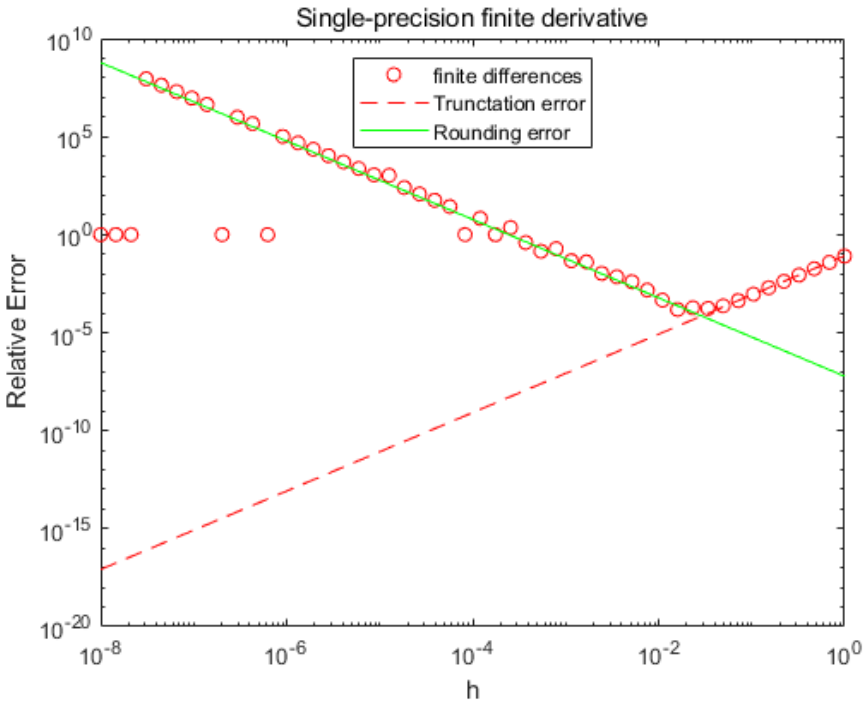

Figure 9: single precision

As shown in figure 12, we get the most accurate answer when h is around 10^{-2}, and we get about 4 digits of accuracy.

4.2 Optimal h

For the truncatin error, I use the next-order term in Taylor series expansion.
let $f(x_0+h) = f(x_0) + hf'(x_0) + \frac{h^2}{2}f''(x_0) + \frac{h^3}{6}f^{(3)}(x_0) + \frac{h^4}{24}f^{(4)}(x_0)$
$f(x_0-h) = f(x_0) - hf'(x_0) + \frac{h^2}{2}f''(x_0) - \frac{h^3}{6}f^{(3)}(x_0) + \frac{h^4}{24}f^{(4)}(x_0)$
let $f(x_0) = f''(x_0) = f''(x_0) + \frac{h^4}{12}f^{(4)}(x_0)$
Now we can estimate the truncation error as: $\epsilon_t = \frac{|f(x_0)-f''(x_0)|}{h^2 f''(x_0)} \approx \frac{h^2}{12}$.
For roundoff error, we should estimate the relative error in the numberator and in the denominator and then add them up.
Denominator: the only error comes from rounding to the nearest floating point numbner, so the relative error in the dem=nominator is on the order of u.
Numerator is computed with absolute error of about u, where u is the machine

precision or roundoff unit.
Te actual value of the numerator is close to $h^2 f''(x = x_0)$, so the magnitude of the relativfe error in the numerator is on the order of $\epsilon_r \approx \left|\frac{u}{h^2 f''(x=x_0)}\right| \approx \frac{u}{h^2}$.
The magnitude of the overall relative error is approximately the sum of the truncation and roundoff errors, $\epsilon \approx \epsilon_t + \epsilon_r = \frac{h^2}{12} + \frac{u}{h^2}$.
The minimum error is achieved when $\epsilon_t = \epsilon_r$.
In the condition of double precision, $h \approx \epsilon^{\frac{1}{4}} \approx (12 \times 10^{-16})^{\frac{1}{4}} = 1.9105e - 04$.
In the condition of single precision, $h \approx \epsilon^{\frac{1}{4}} \approx (12 \times 6 \times 10^{-8})^{\frac{1}{4}} = 0.0291$.
Compared with figures in problem 4.1, the result is very close: in double precision figure, we get best result when $h \approx 10^{-4}$,
and in single precision figure, we get the best result when $h \approx 10^{-2}$.