SECTION 2: AN INTRODUCTION TO FLOATING POINT ARITHMETIC AND RATES OF CONVERGENCE

Consider the number 12. Of course, we can think of the number as

$$12 = 1 \cdot 10^1 + 2 \cdot 10^0 = (12)_{10}$$

Or, consider the number 12.625. We can think of this number as

$$12.625 = 1 \cdot 10^1 + 2 \cdot 10^0 + 6 \cdot 10^{-1} + 2 \cdot 10^{-2} + 6 \cdot 10^{-3}$$

$$= (12.625)_{10} \cdot 10^0 = (.12625)_{10} \cdot 10^2$$

On the other hand, we can also think of the number 12 as

$$12 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = (1100)_2 \cdot 2^0$$

Similarly, we can think of 12.625 as

$$12.625 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

$$= (1100.101)_2 \cdot 2^0 = (.1100101)_2 \cdot 2^1$$

Or, we can think of 12.625 as

$$12.625 = 1 \cdot 8^1 + 4 \cdot 8^0 + 5 \cdot 8^{-1} = (14.5)_8 \cdot 8^0 = (.145)_8 \cdot 8^2$$

Floating Point Arithmetic

**Definition:** an n digit floating point number, in base \( \beta \), has the form

$$\pm(d_1d_2d_3\cdots d_n)_\beta \cdot \beta^e$$

where \( d_i = 0,1,2,\ldots,\beta-1 \) and \( \beta \) is an integer. Unless the number in question is zero, we always write the number so that \( d_i \neq 0 \) (i.e., we say the number is "normalized"). The term

$$(d_1d_2d_3\cdots d_n)_\beta$$

is called the mantissa; \( e \), the exponent, is an integer. The number \( n \) is, of course, finite and is often called the "precision". The size of \( n \) depends upon the word length of the computer in question and, of course, this value varies considerably. The exponent, \( e \), is an integer and is limited to a range, denoted by

$$e_{\text{min}} \leq e \leq e_{\text{max}}$$

For instance, the largest number that can be represented on a \((\beta,n)\) machine is
\[(\beta - 1)(\beta - 1) \cdots (\beta - 1))_\beta \cdot \beta^{e_{\text{max}}} = (1 - \beta^{-n})\beta^{e_{\text{max}}}\]

What is the smallest positive, normalized, number that can be represented?

The representation of decimal fractions, using a \((\beta, n)\) machine, is the source of "round-off" error.

Ex. \((\beta = 10, n = 2)\) Let \(x = 2/3\). Then, using "rounding", the floating point representation of 2/3 is

\[f_l(x) = (0.67)_{10} \cdot 10^0\]

and using another method, called "chopping", we have

\[f_l(x) = (0.66)_{10} \cdot 10^0\]

or, for \(x = -838\), we have

\[f_l(x) = -(0.84)_{10} \cdot 10^3\]

and

\[f_l(x) = -(0.83)_{10} \cdot 10^3\]

respectively.

**Definition:** The value, using a \((\beta, n)\) machine, \(|f_l(x) - x|\) is called the round-off error.

For \(x \neq 0\), the value

\[\frac{|f_l(x) - x|}{|x|}\]

is called the relative error (when the relative error is multiplied by 100, we have the percentage error).

Ex: Let \(x_T = 10^{-5}\) denote the true value of \(x\). Let \(f_l(x_T) = 10^{-4}\). Then

\[|f_l(x_T) - x_T| = 9 \cdot 10^{-5}\]

but

\[\frac{|f_l(x_T) - x_T|}{|x_T|} = \frac{9 \cdot 10^{-5}}{10^{-5}} = 9.0\]

a percentage error of 900%.
We may also write $f(x)$ as $f(x) = x(1 + \delta_x)$ and, in this notation, we see that

$$|\delta_x| = \frac{|f(x) - x|}{|x|}$$

That is, $|\delta_x|$ is the relative error ($x \neq 0$).

Ex: Let's assume a $(\beta = 2, n)$ machine. Let

$$x^* = f \cdot 2^e, \quad \frac{1}{2} \leq f < 1$$

denote an actual machine number. $f \geq 1/2$ holds since we're assuming a normalized machine number and $\beta = 2$. By adding a "1" to the least significant place (see the end of this Section for the general case) of the floating point mantissa, the difference between $x^*$ and the next floating point number

$$y^* = x^* + \Delta$$

is given by

$$\Delta = 2^{-n} \cdot 2^e = 2^{-n+e}$$

Therefore, the round-off error in approximating a number

$$x \in (x^*, x^* + \Delta)$$

is at most

$$\frac{1}{2} \Delta = 2^{-n+e-1}$$

Observe, since we may assume that $\frac{1}{2} \leq f < 1$, we have that

$$|x| = |f| \cdot 2^e = 2 \cdot f \cdot 2^{e-1} \geq 2^{e-1}$$

Therefore, let $x$ be a number whose computer representation is $x^*$. Then

$$|x - x^*| \leq \Delta / 2 = 2^{e-n-1} \leq 2^{-n} \cdot 2^{e-1} \leq 2^{-n} |x|$$

and, therefore,

$$\frac{|x - x^*|}{|x|} \leq 2^{-n}$$
We can replicate the above argument for a general \((\beta, n)\) machine. In particular, let
\[ x = f \cdot \beta^e, \beta^{-1} \leq f < 1 \]
since we assume the number is a normalized one. Then, as before,
\[ |x| = |f| \beta^e = \beta |f| \beta^{e-1} \geq \beta^{e-1} \]
By adding to the most insignificant digit of the floating point mantissa (see the end of this Section for the general case), we see that the difference between a machine number, \(x^*\), and the next machine number,
\[ y^* = x^* + \Delta \]
is given by
\[ \Delta = \beta^{-n} \beta^e = \beta^{e-n} \]
Therefore, for any number \(x\) which has \(x^*\) as its computer representation, we have
\[ |x - x^*| \leq \Delta / 2 = \frac{1}{2} \beta^{e-n} = \frac{1}{2} \beta^{1-n} \beta^{e-1} \leq \frac{1}{2} \beta^{1-n} |x| \]
Hence, for \(x \neq 0\),
\[ \frac{|x - x^*|}{|x|} \leq \frac{1}{2} \beta^{1-n} \]
The right-hand-side of the above inequality is often called "machine epsilon" and the latter term is often abbreviated to "macheps". We'll denote "machine epsilon" by \(\varepsilon_M\) and we have
\[ \varepsilon_M = \frac{1}{2} \beta^{1-n} \]
and, therefore, for \(x \neq 0\), we have
\[ |x - x^*| \leq \varepsilon_M |x| \]
Essentially, \(1 + 2 \cdot \varepsilon_M\) is the smallest number greater than 1 which the \((\beta, n)\) machine is able to distinguish from 1.

**Exercise:** Write a single precision code (in 'C', the type is float) to subtract the two numbers \(a = 0.1234567\) and \(b = 0.1234566\). You may be surprised to see a relative error of over 4\%. What is the reason for this?
In fact, Turbo Pascal, for the above exercise, returns
\[ fl(a - b) = 1.043081E - 07 \]
where, of course, \( a - b = 1.0E - 07 \).

This phenomenon is called "subtractive cancellation" since the most significant digits are being canceled because of the subtraction of almost equal numbers. Indeed, in the above answer to the exercise, we see that the answer is accurate to only 2 significant digits while single precision is supposed to be "accurate" to 7 or 8 significant digits.

In fact, subtractive cancellation is what led to our earlier problem where we used the quadratic formula
\[
\frac{-b \pm \sqrt{b^2 - 4ac}}{2}
\]
When using the "+" root, if
\[ b^2 >> 4ac \]
then the numerator may suffer from subtractive cancellation and there is a loss of significant digits. This is precisely the reason we had trouble numerically estimating the derivative in some previous examples. In particular, as \( h \) gets small, the numerator of the term
\[
\frac{f(\bar{x} + h) - f(\bar{x})}{h}
\]
is the difference of nearly equal numbers and a loss of significant digits occurs. Indeed, to add insult to injury, the error due to that loss is then multiplied by a large number, \( 1/h \).

**Significant Digits:**
This topic is sometimes confusing; however, for the purposes of this course, we'll take a simplified and "intuitive" approach. The following definition is not necessarily a standard definition. However, as previously stated, the definition is sufficient for this course.
**Definition:** We say that \( \overline{x} \) approximates \( x \) to \( r \) significant digits if the relative error is less than one, AND

\[
\frac{|x - \overline{x}|}{|x|} \leq 5 \cdot 10^{-r}
\]

and \( r \) is the largest such integer.

Ex: \( x = 100, \overline{x} = 100.01 \). Then

\[
\frac{|x - \overline{x}|}{|x|} = 10^{-4} \leq 5 \cdot 10^{-4}
\]

and, by the definition above, we say that \( \overline{x} \) approximates \( x \) to \( r = 4 \) significant digits. Of course, by just looking at these two numbers we may have arrived at the same conclusion.

Ex: \( x = 100, \overline{x} = 100.06 \). Then

\[
\frac{|x - \overline{x}|}{|x|} = 6 \cdot 10^{-4} \leq 5 \cdot 10^{-3}
\]

and we say that \( \overline{x} \) approximates \( x \) to \( r = 3 \) significant digits.

Ex: \( x = 0.29467968, \overline{x} = 0.29467825 \). Then

\[
\frac{|x - \overline{x}|}{|x|} \approx 4.85277 \cdot 10^{-6} \leq 5 \cdot 10^{-6}
\]

and we say that \( \overline{x} \) approximates \( x \) to \( r = 6 \) significant digits. Inspection of the two numbers, in this case, shows that our working definition of "significant digits" may lead to an "extra digit" of significance compared to what our intuitive guess might be. For instance, consider an alternative calculation of “significant digits”:

\[
\text{round}_i(-\log_{10} \left( \frac{|x - \overline{x}|}{|x|} \right)) = \text{round}_i(-\log_{10} (4.85277 \cdot 10^{-6})) = \text{round}_i(5.315...) = 5
\]
Exercise: Use the following simple Fortran code to add 1.0e-03 to 100. Then run the
code again to add 1.0e-04 to 100; ...., then run the code again to add 1.0e-06 to 100.
What do you observe? Why? If you wish to write a "C" or Pascal code to execute the
same operations, make sure you use "single" precision.

```fortran
  e=100.0
  write(6,*), ' the number to be added = ?'
  read(5,*), add
  write(6,*), ' The Sum = ', e+add
  end
```

Consider an infinite sequence \{z_k\} where \(k = 1,2,3,...\) or, at times, \(k = 0,1,2,3,...\)
Sometimes we write \(z_k\) in place of \{z_k\} if the context is clear.

Definition: We say the sequence \{z_k\} converges to \(\bar{z}\) (written as \(z_k \to \bar{z}\), or \(\lim_{k \to \infty} z_k = \bar{z}\))
if "eventually all but a finite number of elements of the sequence are arbitrarily close
to \(\bar{z}\). To be precise, \(z_k \to \bar{z}\) if \(\forall \varepsilon > 0 \\exists K_\varepsilon \ni \forall k \geq K_\varepsilon\) we have \(|z_k - \bar{z}| < \varepsilon\).

Ex: Let \(z_k = 1/k, k = 1,2,3,...\) Then, of course, \(\bar{z} = 0\). In this case, given \(\varepsilon > 0\), we see
that \(K_\varepsilon > 1/\varepsilon\) suffices.

Ex: Let \(x\) be such that \(|x| < 1\). Then the geometric series

\[
\sum_{j=0}^{\infty} x^j = \frac{1}{1-x}
\]
as we know. But what is meant by the statement "the infinite series converges"? Let the
sequence \{z_k\} be defined by, for \(k = 0,1,2,3,...\),

\[
z_k = \sum_{j=0}^{k} x^j
\]
By "the series converges" is meant \(z_k \to \bar{z} = 1/(1-x)\).

In this course we'll be more concerned with properties of a convergent sequence than we
will be with whether or not the sequence converges.
**Big O and little o notation:**

Consider the two sequences, for $k \geq 1$,

$$z_k = 1/k^2 \quad \text{and} \quad \beta_k = 1/k$$

Note that both converge to zero and that

$$\frac{z_k}{\beta_k} = \frac{1}{k} \to 0$$

That is, the sequence $\{z_k\}$ converges to zero faster than $\{\beta_k\}$ converges to zero. In such a case we say that

$$z_k \text{ is } o(\beta_k)$$

This is read as $z_k$ is "little oh of $\beta_k"."

On the other hand, consider the sequence

$$z_k = \frac{10}{k^2} - \frac{40}{k} + e^{-k}$$

Then, $\frac{z_k}{\beta_k} \to 10$, not zero, and we say that

$$z_k \text{ is } O(\beta_k)$$

This is read as $z_k$ is "big oh of $\beta_k". While both sequences converge to zero, the ratio does not and we therefore do not say that $\{z_k\}$ converges to zero faster than $\{\beta_k\}$. However, we will state that $\{z_k\}$ converges at least as fast as $\{\beta_k\}$.

**Definition:** The sequence $\{z_k\}$ is said to be $o(\beta_k)$ if $z_k \to 0$, $\beta_k \to 0$, and

$$\lim_{k \to \infty} \frac{z_k}{\beta_k} = 0$$

We say $\{z_k\}$ is $O(\beta_k)$ if

$$\frac{|z_k|}{|\beta_k|} \leq M$$

for all $k$ sufficiently large and where $M > 0$. 

26
Often we also use the notation $O(h)$ and $o(h)$ to denote the same concept as above, except we consider $h$ be a "continuous parameter". However, if we have $h \to 0$, we can always choose a sequence of $h$'s, say $h_k$, so that $h_k \to 0$ as $k \to \infty$. That is, the two concepts are the same.

Ex: We've previously seen that

$$f(x + h) - f(x) = f'(x) + \frac{1}{2} f''(\xi) h = f'(x) + O(h)$$

That is, the term $\frac{1}{2} f''(\xi) h$ goes to zero at least as fast as $h$ goes to zero and, therefore, this term is $O(h)$. Hence,

$$\left| \frac{f(x + h) - f(x)}{h} - f'(x) \right| = O(h)$$

That is, the forward difference approximation converges at least as fast as $h$ converges to zero. Please keep in mind that the latter is not an impressive statement.

Ex: We've also previously seen that

$$f(x + h) - f(x - h) = f'(x) + \frac{1}{2} \left( \frac{1}{3!} f'''(\xi) h^2 + \frac{1}{3!} f'''(\psi) h^2 \right) = f'(x) + O(h^2)$$

Or,

$$\left| \frac{f(x + h) - f(x - h)}{2h} - f'(x) \right| = O(h^2)$$

That is, the central difference approximation to the derivative converges at least as fast as $h^2$ converges to zero. Intuitively, this is a much more impressive statement than that which was made for the forward difference approximation. Clearly,

$$h^2 \text{ is } o(h)$$

Note the following about $o(h)$ and $O(h)$:

- $o(h) + O(h) = O(h)$
- $o(h)O(h) = o(h)$
- $O(h) \pm O(h) = O(h)$
• for $ao(h) = o(h)$
• $o(O(h)) = o(h)$
• $O(o(h)) = o(h)$
• $O(O(h)) = O(h)$

Ex: $\sin(h)$ is $O(h)$ since $\lim_{h \to 0} \frac{\sin(h)}{h} = 1$. But what about $\sin(h^2)$? The next to last bullet implies that $\sin(h^2)$ is $o(h)$.

**Exercise:** Let $z_k = 1/k \ln k$. Is $z_k o(1/k)$? Is $z_k O(1/k^2)$?

**Rates of Convergence**

$o$ and $O$ concepts provide a means by which one may make comparisons between sequences: "one sequence converges faster than another", or "one sequence converges at least as fast as another". However, the "little oh" and "big oh" concepts say virtually nothing about how a given sequence behaves; i.e., the concept says nothing about a given sequence when no comparisons with another are made. For instance, let $z_k \to \bar{z}$. For a given $k$, let $|z_k - \bar{z}| = e_k$ and $|z_{k+1} - \bar{z}| = e_{k+1}$ denote the "errors" at iterations $k$ and $k=1$, respectively. That is, the errors are the measure of how far the current $z$ is from the limit point, $\bar{z}$. Since the sequence is assumed to converge, these errors are, for all sufficiently large $k$, smaller than unity (one). If one could say that, for all sufficiently large $k$, $e_{k+1}$ is considerably smaller than $e_k$, we would have some indication that the convergence rate, at the "tail of the sequence", is good. That is, the errors are being reduced rapidly. For instance, suppose one could say that

$$e_{k+1} = e_k^2$$

That is, if this were true for all sufficiently large $k$, we have that at iteration $k+1$ the error is the square of the previous error and, of course, for large $k$, the latter is less than one. To make matters specific, we provide the following definition for "rate of convergence".
Definition: Assume \( z_k \to \mathbb{Z} \), and assume

\[
\lim_{k \to \infty} \frac{e_{k+1}}{e_k^\alpha} = \lambda
\]

The limit, \( \lambda \geq 0 \), is called the asymptotic error constant (aec). The number \( \alpha > 0 \) is the largest such number and is called the rate of convergence of the sequence \( z_k \).

Intuitively, for large \( k \), we have

\[
e_{k+1} \approx \lambda e_k^\alpha
\]

Ex: Let \( z_k = \frac{1}{k} \). Then

\[
\frac{1}{k+1} \frac{1}{k} = \frac{k^\alpha}{k+1}
\]

and we see that \( \alpha = 1 \) is the largest \( \alpha \) that allows the above term to have a limit. Of course, in this case we have

\[
k / k + 1 \to \lambda = 1
\]

Whenever \( \alpha = 1 \) we say the convergence is linear or geometric. (Note that when the convergence rate is linear, it must be the case that \( \lambda \leq 1 \) since, otherwise, the sequence \( z_k \) would not be a convergent sequence.) For this example, we see that the "aec" is \( \lambda = 1 \).

Since \( e_{k+1} \approx \lambda e_k^\alpha \), this sequence has the property that, for large \( k \), the error at iteration \( k+1 \) is almost that of iteration \( k \). That is, since \( \alpha = 1 = \lambda \), the convergence of this sequence to its limit is certainly not rapid, i.e., \( e_{k+1} \approx e_k \). Indeed, for this sequence, we see that

\[
\frac{|z_{k+1} - z_k|}{|z_k - z_{k-1}|} \leq \frac{k-1}{k+1}
\]

and therefore

\[
|z_{k+1} - z_k| = \frac{k-1}{k+1} |z_k - z_{k-1}|
\]

which, of course, shows that the "rate of change" in the sequence \( \{z_k\} \), at iteration \( k+1 \), is approximately equal to the "rate of change" at the previous iteration. Hence the term "linear convergence".
Ex: Let \( z_k = 1/k^2 \). We already know that \( z_k = o(1/k) \); that is, \( z_k \) approaches zero faster than does the sequence \( \{1/k\} \). In fact, simple calculations show that

\[
z_k = o(1/k \ln k)
\]

and

\[
1/k \ln k = o(1/k)
\]

That is, \( z_k \) converges faster than does \( 1/k \ln k \) and the latter, in turn, converges faster than does \( 1/k \). Now,

\[
\frac{e_{k+1}}{e_k^\alpha} = \frac{k^{2\alpha}}{(k+1)^2}
\]

and we therefore see that this sequence converges if, and only if, \( \alpha \leq 1 \) (recall, we only consider \( \alpha > 0 \)); hence, the rate of convergence is linear (\( \alpha = 1 \)) and the aec is \( \lambda = 1 \). Of course one may ask why a sequence that is faster than another (recall, this \( z_k = o(1/k) \)) ends up with the same aec and rate of convergence (\( \lambda = 1, \alpha = 1 \)) as the "slower" sequence. To see why, in this case, we proceed as follows:

\[
\frac{|z_{k+1} - z_k|}{|z_k - z_{k-1}|} = \frac{(k-1)^2 (2k+1)}{(k+1)^2 |2k+1|} \approx \left(\frac{k-1}{k+1}\right)^2
\]

and, therefore,

\[
|z_{k+1} - z_k| \approx \left(\frac{k-1}{k+1}\right)^2 |z_k - z_{k-1}|
\]

and, again, we see that the "rate of change" of the sequence \( \{z_k\} \), for large \( k \), at iteration \( k+1 \) is approximately equal to the "rate of change" at the previous iteration, \( k \). It is within this context that we see that two sequences, one that converges "faster" than the other, can end up with the same rate of convergence parameters, \( \alpha \) and \( \lambda \).

Ex: Let \( z_k = (1/k)^k \). Then \( z_k \to 0 \). Also, show that \( z_k = o(1/k^2) \) (i.e., the sequence \( z_k \) converges faster than the sequence \( 1/k^2 \) converges. Some simple algebra shows that

\[
\frac{e_{k+1}}{e_k^\alpha} = \frac{k^{\alpha k}}{(k+1)^{k+1}} = \frac{k^{\alpha k}}{(k+1)(k+1)^k}
\]

and
When $\alpha = 1$ we see that this term converges to zero; when $\alpha > 1$ we see that this term becomes arbitrarily large. Hence, the rate of convergence is linear ($\alpha = 1$) and the "aec" is $\lambda = 0$. That is, for this sequence we see that

$$\lim_{k \to \infty} \frac{e_{k+1}}{e_k} = 0$$

Whenever the rate of convergence is linear and the aec is given by $\lambda = 0$, we say that the rate of convergence is superlinear.

Ex: Let the number $a \in (0,1)$, and let $z_k = a^{2^k}$. Then $z_k \to 0$. Some simple algebra shows that

$$\frac{e_{k+1}}{e_k} = a^{(2-\alpha)2^k}$$

For $\alpha = 2$, we see that the above term is equal to 1 for all $k$ and, hence, the limit is equal to 1. For $\alpha > 2$, we see that the above term becomes arbitrarily large as $k \to \infty$. Therefore, in this case we have

$$\lim_{k \to \infty} \frac{e_{k+1}}{e_k}^2 = 1$$

Note that we have, in this case, $e_{k+1} = e_k^2$. That is, the error at iteration $k+1$ is the square of the error at the previous iteration, $k$. Since, in this example, the error at iteration $k$ is less than 1, the error at the next iteration (the square of the error at iteration $k$) is a considerable improvement over the error at iteration $k$.

**Definition:** When $\alpha = 2$ we say the rate of convergence is quadratic. Note that when the rate of convergence is quadratic, the "aec" may be larger than 1 (unlike the linear case).

**Exercise:** Assume that $|x| < 1$. Find the rate of convergence of the sequence of partial sums that comprise the geometric series. In particular, first show that

$$z_k = \sum_{j=0}^{k} x^j = \frac{1 - x^{k+1}}{1 - x}$$
and, of course, $\frac{1}{x} = 1/(1 - x)$.

Is the convergence linear? Is the convergence superlinear?
Is the convergence quadratic? Is the sequence \( \{z_k - \frac{1}{x}\} = o(1/k) \)? What about \( o(1/k^2) \)?

**Geometric Sums**

\[ x \neq 0: \]

\[
\sum_{k=0}^{n} x^k = \frac{1 - x^{n+1}}{1 - x}
\]

\[
\sum_{k=1}^{n} x^k = x \left( \frac{1 - x^n}{1 - x} \right)
\]

\[
\sum_{k=j+1}^{n} x^k = x^{j+1} \left( \frac{1 - x^{n-j}}{1 - x} \right), \quad 1 \leq j < n
\]

\[
\therefore \text{ for } x \neq 0 \text{ and } |x| < 1
\]

\[
\sum_{k=0}^{\infty} x^k = \lim_{n \to \infty} \sum_{k=0}^{n} x^k = \frac{1}{1 - x}
\]

**Back to Floating Point Arithmetic**

In the first part of this Section, it was shown that the difference between a floating point number, \( x^* \), and the next floating point number, \( y^* \), is given by \( \Delta = 2^{-n}2^e \). However, the argument given there assumed that the \( n^{th} \) digit in the mantissa (for the \( \beta = 2 \) case) was zero. We now consider cases where the \( n^{th} \) digit is not zero, but is equal to one.

Assume that all the digits are equal to one. That is,

\[
x^* = (.111\cdots1)2^e = 2^e \sum_{k=1}^{n} 2^{-k} = 2^e \left(1 - 2^{-n}\right)
\]

The next highest number is, therefore,

\[
y^* = (.100\cdots0)2^{e+1} = 2^e
\]

Then,

\[
\Delta = y^* - x^* = 2^{-n}2^e
\]
There is one remaining case: The last zero of the mantissa is in the jth location, where 
1 < j < n (therefore, all subsequent mantissa values are equal to one). That is,
\[ x^* = (.1 1 \cdots 1 0 1 0 1 \cdots 1)2^e \]

Then, the next machine number is given by
\[ y^* = (.1 1 \cdots 1 0 1 1 0 \cdots 0)2^e \]

Therefore, we can represent \( \Delta \) by the amount that was added to \( x^* \), minus the amount 
that was taken away, in order to reach \( y^* \). That is,
\[ \Delta = 2^{-j}2^e - 2^e \sum_{k=j+1}^{n} 2^{-k} = 2^{-n}2^e \]