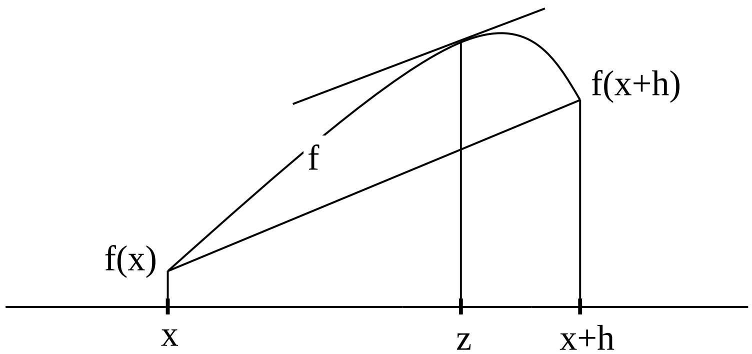
The Mean Value Theorem

The **mean value theorem** (**MVT**): Let $f(x)$ be differentiable. For **some** z in $[x, x+h]$:

$$f'(z) = \frac{f(x+h) - f(x)}{h}.$$

This is *intuitively* clear from:



We can **rewrite** the MV formula as:

$$f(x+h) = f(x) + hf'(z).$$

A generalization if f is **twice** differentiable is

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(z),$$

for some z in $[x, x+h]$.

Taylor Series

Taylor's Theorem: Let f have continuous derivative of order $0, 1, \ldots, (n+1)$, then

$$f(x+h) = \sum_{k=0}^{n} \frac{f^{(k)}(x)}{k!} h^k + E_{n+1},$$

where the error (remainder) $E_{n+1} = \frac{f^{(n+1)}(z)}{(n+1)!} h^{n+1}$, $z \in [x, x+h]$.

Taylor series. If f has **infinitely** many derivatives,

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!} f''(x) + \frac{h^3}{3!} f'''(x) + \cdots,$$

where we assume $\lim_{n \to \infty} E_{n+1} = 0$.

Example. $f(x) = \sin x$:

$$\begin{aligned}\sin(x+h) &= \sin(x) + h\sin'(x) + \frac{h^2}{2!}\sin''(x) \\ &+ \frac{h^3}{3!}\sin'''(x) + \frac{h^4}{4!}\sin''''(x) + \cdots,\end{aligned}$$

Letting $x = 0$, we get

$$\sin(h) = h - \frac{h^3}{3!} + \frac{h^5}{5!} - \cdots.$$

Numerical Approximation to $f'(x)$

If h is **small**, $f'(x)$ is nearly the **slope** of the line through $(x, f(x))$ and $(x+h, f(x+h))$. We write

$$f'(x) \approx \frac{f(x+h) - f(x)}{h},$$

the **forward difference** approximation.
How **good** is this approximation? Using the **truncated Taylor series**:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(z),$$

we get

$$\frac{f(x+h) - f(x)}{h} - f'(x) = \frac{h}{2}f''(z),$$

the **discretization error**: the difference between what we want and our approximation. Using the **discretization size** h. We say the discretization error is $O(h)$.

Computing the Approximation to f'(x)

```
main()  /* diff1.c: approx. deriv. */
{int n; double x,h,approx,exact,error;
 x=1.0; h=1.0; n=0;
 printf("\n h exact approx error");
 while(n<20) {
  n++;
  h=h/10;      /* h=10^(-n) */
  approx=(sin(x+h)-sin(x))/h;/*app.deriv.*/
  exact=cos(x);   /*exact derivative */
  error=approx - exact; /*disczn error */
  printf("\n ... ,h,exact,apprx,err);
 }
}
```

We want to approximate the deriv of $f(x) = \sin(x)$ at $x = 1$. Knowing $\sin'(x) = \cos(x)$, we can compute the **exact** discretization error. The program uses **double precision**, and displays

$$(\sin(x+h) - \sin(x))/h,$$

with the **error**, for h from 0.1 to 10^{-20}.

Convergence of Approximation

```
h       exact        approx       error
e-03 5.403023e-01 5.398815e-01 -4.20825e-04
e-04 5.403023e-01 5.402602e-01 -4.20744e-05
e-05 5.403023e-01 5.402981e-01 -4.20736e-06
e-06 5.403023e-01 5.403019e-01 -4.20746e-07
e-07 5.403023e-01 5.403023e-01 -4.18276e-08
e-08 5.403023e-01 5.403023e-01 -2.96988e-09
e-09 5.403023e-01 5.403024e-01  5.25412e-08
e-10 5.403023e-01 5.403022e-01 -5.84810e-08
e-11 5.403023e-01 5.403011e-01 -1.16870e-06
e-12 5.403023e-01 5.403455e-01  4.32402e-05
e-13 5.403023e-01 5.395684e-01 -7.33915e-04
e-14 5.403023e-01 5.440093e-01  3.70697e-03
e-15 5.403023e-01 5.551115e-01  1.48092e-02
e-16 5.403023e-01 0.000000e+00 -5.40302e-01
```

Discretization error reduced by ~ 10 when h is reduced by 10, so the error **is** $O(h)$. But when h gets **too** small, the approximation starts to get **worse!**

Q: Why?

Explanation of Accuracy Loss

If $x = 1$, and $h < \epsilon/2 \approx 10^{-16}$ the **machine epsilon**, $x + h$ has the **same numerical value** as x, and so $f(x+h)$ and $f(x)$ **cancel** to give **0**: the answer has **no digits of precision**.

When h is a **little** bigger than ϵ, the values **partially cancel**. e.g. suppose that the first 10 digits of $f(x+h)$ and $f(x)$ are the same. Then, even though $\sin(x+h)$ and $\sin(x)$ are **accurate to 16 digits**, the **difference** has only **6 accurate digits**

In summary, using h **too big** means a big **discretization** error, while using h **too small** means a big **cancellation** error. For the function $f(x) = \sin(x)$, for example, at $x = 1$, the best choice of h is about 10^{-8}, or $\sim \sqrt{\epsilon}$.

Numerical cancellation, which results when subtraction of **nearly equal values** occurs, should always be avoided when possible.

Numerical Cancellation

One of the main causes for deterioration in precision is numerical cancellation.

Suppose x and y are exact values of two numbers and $x \neq y$. But in many situations we may only have the approximate values

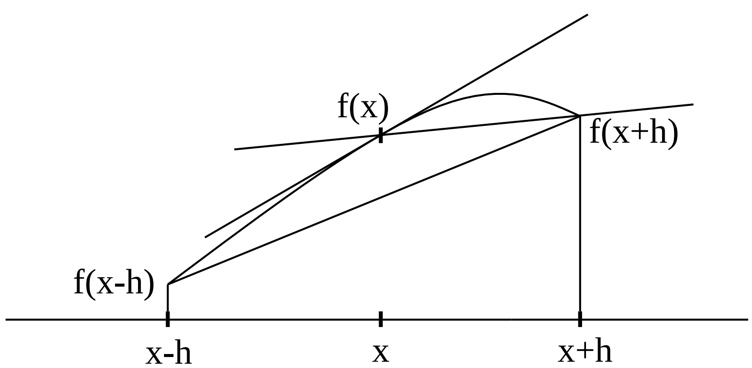
$$\hat{x} = x(1 + \delta_x), \qquad \hat{y} = y(1 + \delta_y),$$

where δ_x and δ_y are the ralative errors in $\hat{x}$ and $\hat{y}$, respectively. These values may be obtained from some computations or physical experiments. Suppose we want to compute $x - y$. But we can only compute $\hat{x} - \hat{y}$. Is it a good approximation of $x - y$? Let us look at the relative error:

$$\Big|\frac{(\hat{x} - \hat{y}) - (x - y)}{x - y}\Big| = \Big|\frac{x\delta_x - y\delta_y}{x - y}\Big| \leq \frac{|x|}{|x - y|}|\delta_x| + \frac{|y|}{|x - y|}|\delta_y| \leq \max\{|\delta_x|, |\delta_y|\}\frac{|x| + |y|}{|x - y|}.$$

This suggests when $|x - y| \ll |x| + |y|$, it is possible that the relative error may be very large. Here the rounding error in computing $\hat{x} - \hat{y}$ was not considered for simplicity.

More accurate numerical differentiation



As h **decreases**, the line through
$(x - h, f(x - h))$ and $(x + h, f(x + h))$ gives a **better approximation** to the **slope of the tangent** to f at x than the line through $(x, f(x))$ and $(x + h, f(x + h))$.
This observation leads to the approximation:

$$f'(x) \approx \frac{f(x + h) - f(x - h)}{2h},$$

the **central difference** formula.

Analyzing Central Difference Formula

From the **truncated Taylor series:**

$$f(x + h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(z_1),$$

with z_1 between x and $x + h$. But also

$$f(x - h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{6}f'''(z_2),$$

with z_2 between x and $x-h$.
Subtracting the 2nd from the 1st:

$$\frac{f(x+h)-f(x-h)}{2h} = f'(x) + \frac{h^2}{12}(f'''(z_1)+f'''(z_2))$$

So the **discretization error** is

$$\frac{h^2}{12}(f'''(z_1)+f'''(z_2)),$$

which is $O(h^2)$ <u>instead of $O(h)$</u>. When h is small enough, $O(h^2)$ will be (much) smaller than $O(h)$.