Function Evaluation

In scientific computing we routinely evaluate functions from spaces over real (or complex) numbers into real (or complex) spaces. We write

$$f : D \to R$$

to say that f is a function with domain D and range R (each element of D is associated (through f) to exactly one element of R (as in $f(d) = r$)). You can think of D and R as subsets of the real numbers, but often they are finite dimensional vector spaces over real (or complex) numbers.

We are using a finite model of the reals (the floating point numbers) to approximate the domain and range spaces, so when we use the term "evaluate f at x", we really mean "evaluate f at our approximation of x", which will give us an approximation to f evaluated at our approximation to x.

Some notation might help here. Let's say we'd like to evaluate f at x, that is, we'd like to compute $y = f(x)$. If $\bar{x}$ is our approximation of $x \in D$, then (like-it-or-not) we are actually trying to compute $\tilde{y} = f(\bar{x})$. But because of rounding errors in this function evaluation, we instead compute $\bar{y}$ as our approximation to $\tilde{y}$. Our attempt to compute $y = f(x)$ returns instead $\bar{y} = \bar{f}(\bar{x})$, and we hope that $\bar{y} \approx \tilde{y} \approx y$.

In summary:
- $y = f(x)$ is what we want, but we have $\bar{x}$ instead of x,
- $\tilde{y} = f(\bar{x})$, is what we try to compute, but is subject to rounding errors, and
- $\bar{y} = \bar{f}(\bar{x})$ is what we have actually computed.

Attempting to predict or discover how much $\bar{y}$ and y might differ is a rather difficult question in general, and is an important part of numerical analysis. One way to simplify the question is to imagine we *can* compute $\tilde{y}$: When we speak of "the magic method" we imagine a method which returns the $\bar{y}$ that is the closest element of our model of R to $\tilde{y}$. This is the best we might do, and is a useful concept to keep in mind in any analysis of computing with real numbers. We'll see that how good (or poorly) the magic method performs is a measure of how difficult our problem is (condition numbers...).

Think about just how good an approximation the magic method might give for various functions (e.g. $y = f(x) = 1/x$, $y = f(x) = \sin(x)$, $y = f(A, b) = A^{-1}b$, etc.), or how you might evaluate $f(x)$ using only arithmetic operations, or how you might approximate the error $y - \bar{y}$.

You may think that "evaluate f at x" is a trivial task, but if you think about trying to solve *any* problem (which has a unique solution), then you come to see that this is exactly what we are trying to do: the solution is f(input data), for some function f which maps the input data to the solution. At the risk of getting too philosophical, I also remind you that this function that we are trying to evaluate is probably only an approximation of some (more complicated, or unknown, or unknowable) function.