1 Numeric derivatives

`SymPy` returns symbolic derivatives. Up to choices of simplification, these answers match those that would be derived by hand. This is useful when comparing with known answers and for seeing the structure of the answer. However, there are times we just want to work with the answer numerically. For that we have other options.

1.0.1 Approximate derivatives

An approximate derivative can be used. By approximating the limit of the secant line with a value for a small, but positive, h, we get an approximation. That is

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

This is the forward-difference approximation. The central difference approximation looks both ways:

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

Though in general they are different, they are both approximations. The central difference is a bit more accurate for the same size h. However, both are susceptible to round-off errors. The numerator is a subtraction of like-size numbers - a perfect opportunity to lose precision. As such there is a balancing act: if h is too small the round-off errors are problematic, if h is too big, the approximation to the limit is not good. For the forward difference h values around 10^{-8} are good, for the central difference, values around 10^{-6} are good for most instances.

Example Let's verify that the forward difference isn't too far off.

```
f(x) = exp(-x^2/2)
c = 1
h = 1e-8
approx = (f(c+h) - f(c)) / h
```

```
-0.6065306479285937
```

We can compare to the actual with:

```
using CalculusWithJulia
@vars x
df = diff(f(x), x)
actual = N(df(c))
abs(actual - approx)
```

```
1.178403974334123392219821170344191813548718695568289215873505651941374433299215e-08
```

The error is about 1 part in 100 million.

The central difference is better here:

```
h = 1e-6
approx = (f(c+h) - f(c-h)) / (2h)
abs(actual - approx)
```

1.567568231457459370771602167922282298718695568289215873505651941374433299214873e-11

1.0.2 Automatic derivatives

There are some other ways to compute derivatives numerically that give much more accuracy at the expense of some increased computing time. Automatic differentiation is the general name for a few different approaches. These approaches promise less complexity - in some cases - than symbolic derivatives and more accuracy than approximate derivatives. In fact the accuracy is on the order of machine precision.

The `ForwardDiff` package provides one of several ways for `Julia` to compute automatic derivatives. This package is loaded with `CalculusWithJulia`, but its functions are not exported, so their usage requires qualification. To illustrate, to find the derivative of $f(x)$ at a *point* we have this syntax:

```
f(x) = exp(-x^2/2)
c = 1
ForwardDiff.derivative(f, c)    # derivative is qualified by a module name
```

-0.6065306597126334

The `CalculusWithJulia` package defines an operator `D` which goes from finding a derivative at a point with `ForwardDiff.derivative` to definin a function which evaluates the derivative at each point. It is defined along the lines of `D(f) = x -> ForwardDiff.derivative(f,x)` in parallel to how the derivative operation for a function is defined mathematically from the definition for its value at a point.

Here we see the error in estimating $f'(1)$ for the $f(x) = e^{-x^2/2}$.

```
approx = D(f)(c)         # D(f) is a function, D(f)(c) is the function called on c
abs(actual - approx)
```

6.593178415491414032952187053331255443171078412649434805862556670078512733295529e-19

In this case, it is exact.

The `D` operator is only defined for most functions, not all. (The `diff` operator of `SymPy` is somewhat similar in that respect.)

Example For $f(x) = \sqrt{1 + \sin(\cos(x))}$ compare the difference between the forward derivative with $h = 1e-8$ and that computed by `D` at $x = \pi/4$.

The forward derivative is found from:

```
f(x) = sqrt(1 + sin(cos(x)))
c, h = pi/4, 1e-8
fwd = (f(c+h) - f(c))/h
```

-0.20927346522370271

That given by `D` is:

```
ds_value = D(f)(c)
ds_value, fwd, ds_value - fwd
```

```
(-0.20927346371432803, -0.20927346522370271, 1.5093746807970376e-9)
```

Finally, `SymPy` gives an exact value we use to compare:

```
@vars x
actual = diff(f(x), x) |> subs(x, PI/4) |> N
actual - ds_value, actual - fwd
```

```
(-5.29413734954932810851790641707227270040811239385425461541883810861494031
49753e-17, 1.509374627855664131196135271205690010185476120918876061457453845
811618913850597e-09)
```

Convenient notation `Julia` allows the possibility of extending functions to different types. Out of the box, the `'` notation is not employed for functions, but is used for matrices. It is used in postfix position, as with `A'`. We can define it to do the same thing as `D` for functions and then, we can evaluate derivatives with the familiar `f'(x)`. This is done in `CalculusWithJulia` along the lines of `Base.adjoint(f::Function) = D(f)`.

Then, we have, for example:

```
f(x) = sin(x)
f'(pi), f''(pi)
```

```
(-1.0, -1.2246467991473532e-16)
```

Example Suppose our task is to find a zero of the second derivative of $f(x) = e^{-x^2/2}$ in $[0, 10]$, a known bracket. The `D` function takes a second argument to indicate the order of the derivative (e.g., `D(f,2)`), but we use the more familiar notation:

```
f(x) = exp(-x^2/2)
fzero(f'', 0, 10)
```

1.0

We pass in the function object, `f''`, and not the evaluated function.

1.1 Recap on derivatives in Julia

A quick summary for finding derivatives in `Julia`, as there are 3 different manners:

- Symbolic derivatives are found using `diff` from `SymPy`
- Automatic derivatives are found using the notation `f'` using `ForwardDiff.derivative`
- approximate derivatives at a point, `c`, for a given `h` are found with `(f(c+h)-f(c))/h`.

For example

```
f(x) = exp(-x)*sin(x)
c = pi
h = 0.1
@vars x

fp = diff(f(x),x)
fp, fp(c)
```

```
(-exp(-x)*sin(x) + exp(-x)*cos(x), -exp(-pi))
```

As compared to

```
f'(c), (f(c+h)-f(c))/h
```

```
(-0.043213918263772265, -0.03903643351818505)
```

1.2 Questions

Question Find the derivative using a forward difference approximation of $f(x) = x^x$ at the point $x = 2$ using `h=0.1`:

Using `D` or `f'` find the value using automatic differentiation

Question Mathematically, as the value of `h` in the forward difference gets smaller the forward difference approximation gets better. On the computer, this is thwarted by floating point representation issues (in particular the error in subtracting two like-sized numbers in forming $f(x+h) - f(x)$.)

For `1e-16` what is the error (in absolute value) in finding the forward difference approximation for the derivative of $\sin(x)$ at $x = 0$?

Repeat for $x = pi/4$:

⊛ Question
Let $f(x) = x^x$. Using `D`, find $f'(3)$.

⊛ Question
Let $f(x) = |1 - \sqrt{1 + x}|$. Using `D`, find $f'(3)$.

⊛ Question
Let $f(x) = e^{\sin(x)}$. Using `D`, find $f'(3)$.

⊛ Question
For `Julia`'s `airyai` function find a numeric derivative using the forward difference. For $c = 3$ and $h = 10^{-8}$ find the forward difference approximation to $f'(3)$ for the `airyai` function.

⊛ Question
Find the rate of change with respect to time of the function $f(t) = 64 - 16t^2$ at $t = 1$.

⊛ Question
Find the rate of change with respect to height, h, of $f(h) = 32h^3 - 62h + 12$ at $h = 2$.