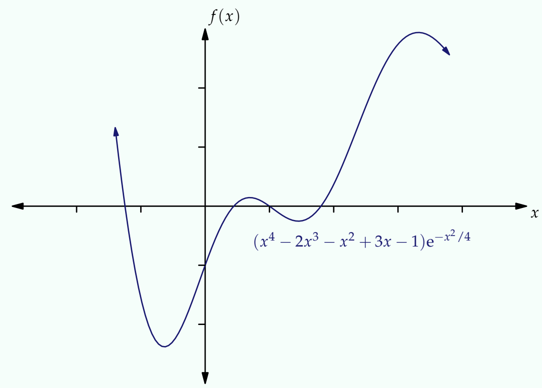
DATA 1010
In-class exercises
Samuel S. Watson
21 September 2018

Problem 1

Consider the function $f(x) = (x^4 - 2x^3 - x^2 + 3x - 1)e^{-x^2/4}$. Implement the gradient descent algorithm for finding the minimum of this function.

(i) If the learning rate is $\epsilon = 0.1$, which values of x_0 have the property that $f(x_n)$ is close to the global minimum of f when n is large?

(ii) Is there a starting value x_0 between -2 and -1 and a learning rate ϵ such that the gradient descent algorithm does not reach the global minimum of f? Use the graph for intuition.



Solution

The following is an implementation of gradient descent:

```
using LinearAlgebra

function graddescent(f,x₀,ϵ,threshold)
    df(x) = f([x 1; 0 x])[1,2] # auto diff
    x = x₀
    while abs(df(x)) > threshold
        x = x - ϵ*df(x)
    end
    x
end
f(t) = exp(-t^2/4)*(t^4 - 2t^3 - t^2 + 3t - I)
```

Trying various values of x_0, and looking at the graph, we conjecture that the global minimum is reached when the starting value x_0 is between the first two points where f has a local maximum (approximately -2.83 and 0.145). Between 0.145 and the next local maximum (approximately 2.94), the algorithm leads us to the local minimum around $x = 1.45$. Outside the interval from the first local maximum the last, the sequence of iterates appears to head off to $-\infty$ or $+\infty$.

Skipping over the global minimum to the local one requires choosing ϵ large enough that the first jump skips over the local maximum at 0.145. A little experimentation shows that $x = -1.5$ and $\epsilon = 0.25$ works (among many other possibilities).

Problem 2

Which of the following two lines of Julia code returns the larger value? You may take it as given that they do not return the same value. Explain.

```
sum(sqrt(k)^2 == k for k=1:100)
sum(sqrt(k^2) == k for k=1:100)
```

Solution

The second one returns 100. This is because k^2 is a perfect square for each value of k. Thus its square root is exactly representable as a `Float64`. The first line returns a value less than or equal to 100, and since 100 was ruled out in the problem statement, we can be sure it returns a value less than 100.

(Note that the argument applied to the second line cannot be applied to the first, since $\sqrt{k}$ is irrational for 90 of the values of k from 1 to 100, and it is therefore not representable as a `Float64`.)

Problem 3

Suppose that $f : [0,1] \to \mathbb{R}$ is a strictly increasing, continuous function such that $f(0) < 0 < f(1)$. The intermediate value theorem tells us that $f(x_0) = 0$ for exactly one value of x_0 between 0 and 1. Consider the following method for approximating x_0:

(i) Check the sign of $f(1/2)$.

(ii) Depending on the result of (i), check the sign of either $f(1/4)$ or $f(3/4)$.

(iii) Depending on the results of (i) and (ii), check the sign of $f(1/8)$ or $f(3/8)$ or $f(5/8)$ or $f(7/8)$.

(iv) Continue in this way, repeatedly narrowing down the interval which contains x_0, for some fixed number of iterations.

Answer the following questions about this algorithm.

(a) What would be the maximum possible number of iterations required to determine which two `Float64` values x_0 lies between?

(b) Implement this algorithm in Julia. Your function should pass the following tests.

```
findzero(x->x^2-1/2) == 1/sqrt(2)
findzero(x->x-1/2) == 1/2
```

Solution

We start with $x = \frac{1}{2}$ and move left or right depending on the sign of $f(x)$. The step size shrinks by a factor of 2 each time, so we perform that step inside the `for` loop.

If we're only going to use this function on `Float64` values, then we don't have to subdivide more than 1074 times, since the smallest increment between representable numbers in $[0,1]$ is $\frac{1}{2^{1074}}$. Therefore, we can cap the number of steps at 1074. Also, we should stop if we reach a value of x for which $f(x) = 0$.

```
function findzero(f)
    x = 1/2
    stepsize = 1/2
    for k=1:1074
        stepsize /= 2
        if f(x) > 0
            x -= stepsize
        elseif f(x) < 0
            x += stepsize
        else
            return x
        end
    end
    x
end
```

If we wanted to write generic code that would work for more precise number types, we could add a `precision` keyword argument and stop the loop when the stepsize gets smaller than that value.

Problem 4

The matrix $\begin{bmatrix} 1/2 & \sqrt{3}/2 \\ -\sqrt{3}/2 & 1/2 \end{bmatrix}$ represents an n° rotation about the origin.

(a) Check that the determinant of this matrix is compatible with the claim that it represents a rotation.

(b) Find n.

Solution

The determinant is equal to $\frac{3}{4} + \frac{1}{2} = +1$. This implies that the transformation preserves orientations (+) and areas (1). Thus the transformation is indeed a rotation.

The first column is $[1/2, \sqrt{3}/2]$, which means that the vector $[1, 0]$ is mapped to that vector. We can find the angle θ between those two vectors by calculating the dot product to find that $\cos\theta = \frac{1}{2}$.