Maths for ML Specialisation Course 2

Harry Baines

2022

Week 1 : Univariate Calculus

Functions

Input / Output

$f(x) = x^2 + 3$ (function of x)

variables $\rightarrow$ $f(x) = g(x) + h(x-a)$

$f(x, g, h, a)$ (context is important!)

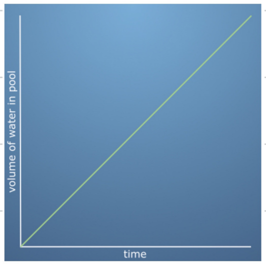
Select candidate function / hypothesis to model the world

Calculus : study of how functions change w.r.t. input variables

Practice Quiz

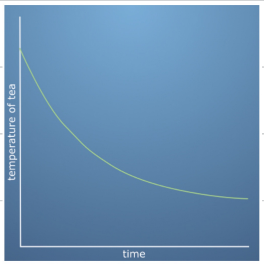
Imagine that you place one end of a water hose into a swimming pool and turn the tap on at the other end. Water then pours into the pool *at a constant rate*, causing the volume of water in the pool to increase at a constant rate**.**

While the swimming pool is still filling up with water, what would we expect the plot of the function of volume of water in the pool with respect to time to look like?



The tea is left to cool down. The speed of cooling depends on the temperature of the tea: when it is hot it cools down quickly and as it gets colder it cools down *more and more slowly*, until it approaches room temperature.

Which of the following graphs could represent the temperature of that cup of tea with time?

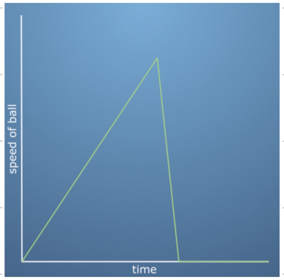


exponential function

Newton's law of cooling

Rahul drops a ball from the top of a ladder into a pit of sand.

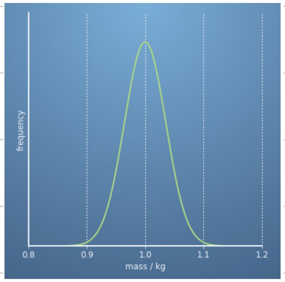
When the ball is released it begins to accelerate towards the ground, getting faster and faster until it hits the sand and quickly becomes stationary again. What would a plot of the speed of the ball against time look like?



ball speeds up at constant rate of acceleration, then decelerates and finishes with 0 speed

Bags of flour labelled 1 kg from a supermarket are weighed. Most of the weights measured are very close to 1 kg, with some a little more and others a little less. Those which are further away from 1 kg are found less and less often, with almost no bags more than 100 g out.

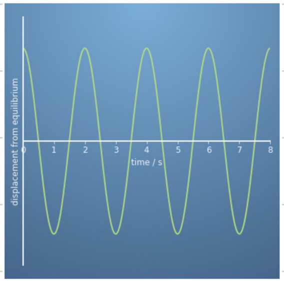
What might we expect the plot of frequency (i.e. how often a type of bag is found) against mass to look like?



weights can be approximated by a bell-curve (Normal distribution)

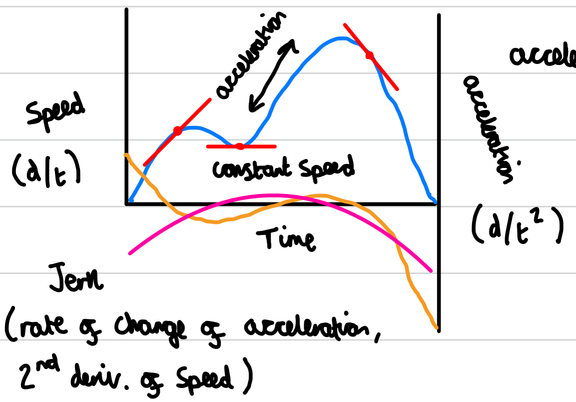
A mass is attached to a string and hung from the ceiling. It is then pulled away from its natural hanging position (called equilibrium) and released, so that it swings backwards and forwards. Let's assume there is no air resistance, so that when the mass swings back it returns all the way back to where it was originally released. It completes a full swing, away and back, every 2 seconds.

What is a reasonable plot for the displacement of the mass from equilibrium with respect to time?



This is called a simple harmonic oscillator - that is, we model the movement of the pendulum through time as a simple sine wave, with some amplitude (determined by the maximum distance of the pendulum to the equilibrium point) and some frequency (determined by the period of the swing).

The pendulum takes 2 seconds to complete a full revolution, which can also be described as swinging at a frequency of 0.5 Hz.



acceleration = local gradient of speed-time graph

take continuous function and described slope at every point by constructing a new function, which is it's derivative take baseline speed function and trying to figure out what function this would have been the gradient of

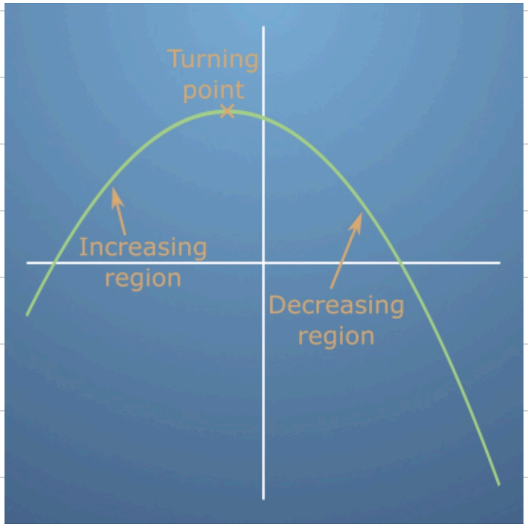
↳ i.e. inverse procedure of previous = anti-derivative = related to integral

↳ integral of speed-time graph = distance of car from starting position

↳ change of distance w.r.t. time (slope of distance-time graph), how much distance covered per unit time = speed

Practice Quiz

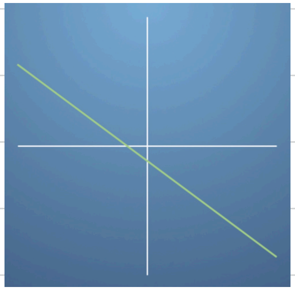
1. Gradient = rise over run (draw tangent line at point)

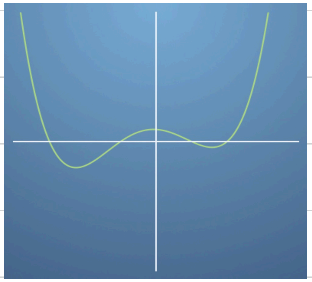


Smooth function gradient: increasing region, turning point, decreasing region

↑ +ve derivative ↑ 0 gradient ↑ -ve derivative

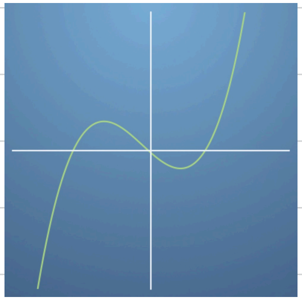
2. Derivative plot:



3. 

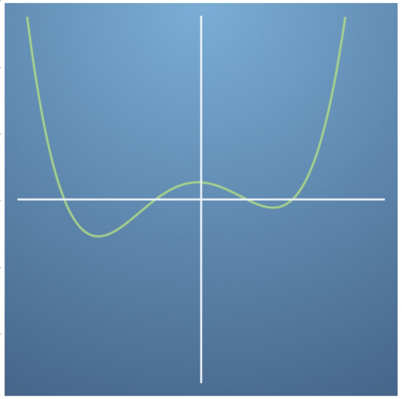
Derivative plot:

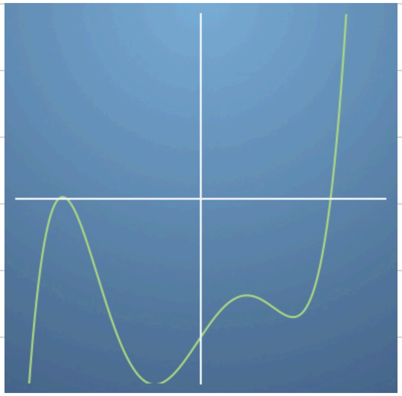
(how many times is gradient 0)

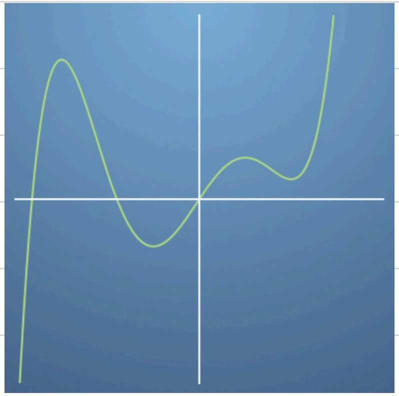


(gradient is 0 3 times)

4. Same ① as before but shifted up, so same ② (if one function is a vertical shift of another function they have the same differential)

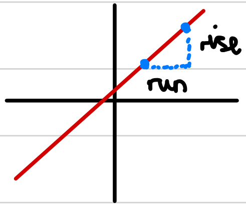
5. 

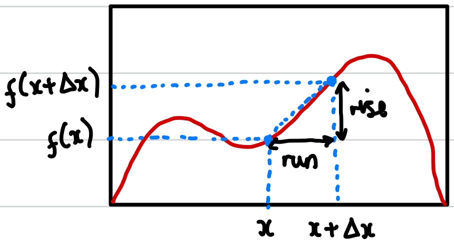




Which of the following diagram(s) could the above plot be the **derivative** of? Choose **all** correct answers. (Hint 1: How many times is the curve above equal to zero (i.e. crosses the horizontal axis). Hint 2: There are two graphs below that could both equally be correct, so select them both!)?

Derivatives





$$\text{gradient @ } x \approx \frac{\text{rise}}{\text{run}} = \frac{f(x+\Delta x) - f(x)}{(x+\Delta x) - x} = \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

for smooth, continuous functions: as Δx gets smaller, line connecting 2 points becomes better approximation of actual gradient @ x

$$\text{Gradient @ } x = \lim_{\Delta x \to 0} \left(\frac{f(x+\Delta x) - f(x)}{\Delta x} \right) = f'(x) = \frac{df}{dx}$$

as Δx goes to 0 (can't divide by 0, only extremely close)

Gradient of linear function is constant

e.g. $f(x) = 3x+2$, $f'(x) = \lim_{\Delta x \to 0} \left(\frac{3(x+\Delta x)+2-(3x+2)}{\Delta x} \right) = \lim_{\Delta x \to 0} \left(\frac{3x+3\Delta x+2-3x-2}{\Delta x} \right) = \lim_{\Delta x \to 0} \left(\frac{3\Delta x}{\Delta x} \right) = \lim_{\Delta x \to 0} (3) = 3$

Sum rule:

$$\frac{d}{dx}(f(x)+g(x)) = \frac{df(x)}{dx} + \frac{dg(x)}{dx}$$

e.g. $f(x) = 5x^2$, $f'(x) = \lim_{\Delta x \to 0} \left(\frac{5(x+\Delta x)^2 - 5x^2}{\Delta x} \right) = \lim_{\Delta x \to 0} \left(\frac{5(x^2+2x\Delta x+\Delta x^2) - 5x^2}{\Delta x} \right) = \lim_{\Delta x \to 0} \left(\frac{5x^2+10x\Delta x+5\Delta x^2-5x^2}{\Delta x} \right)$

$$= \lim_{\Delta x \to 0} (10x + \underbrace{5\Delta x}_{\text{becomes very small, so can ignore}}) = 10x$$

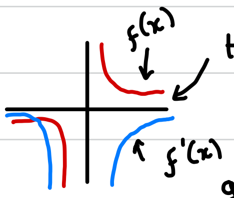
Power rule:

if $f(x) = ax^b$

then $f'(x) = abx^{b-1}$

Special Cases

$f(x) = \frac{1}{x}$



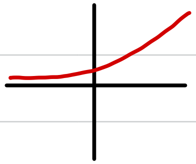
turn upwards = discontinuity

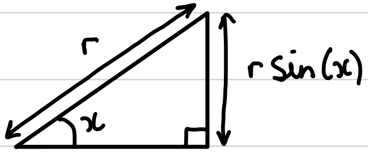
gradient = -ve everywhere, $\div 0$ is undefined

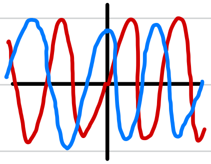
$$f'(x) = \lim_{\Delta x \to 0} \left(\frac{\frac{1}{x+\Delta x} - \frac{1}{x}}{\Delta x} \right) = \lim_{\Delta x \to 0} \left(\frac{\frac{x}{x(x+\Delta x)} - \frac{x+\Delta x}{x(x+\Delta x)}}{\Delta x} \right) = \lim_{\Delta x \to 0} \left(\frac{\frac{-\Delta x}{x(x+\Delta x)}}{\Delta x} \right) = \lim_{\Delta x \to 0} \left(\frac{-1}{x^2 + \underbrace{x\Delta x}_{\text{can be ignored}}} \right) = \frac{-1}{x^2}$$

Euler's no. = 2.71828...

$f(x) = e^x, \quad f'(x) = e^x, \quad f''(x) = e^x, \ldots$







$f(x) = \sin(x)$
$f'(x) = \cos(x)$
$f''(x) = -\sin(x)$
$f^{(3)}(x) = -\cos(x)$
$f^{(4)}(x) = \sin(x)$

$\sin(x) = \dfrac{e^{ix} - e^{-ix}}{2i}$

Practice Quiz

1. $f(x) = x^{173}, \quad f'(x) = 173x^{172}$

2. $f(x) = x^2 + 7 + \frac{1}{x}, \quad f'(x) = 2x - \frac{1}{x^2}$

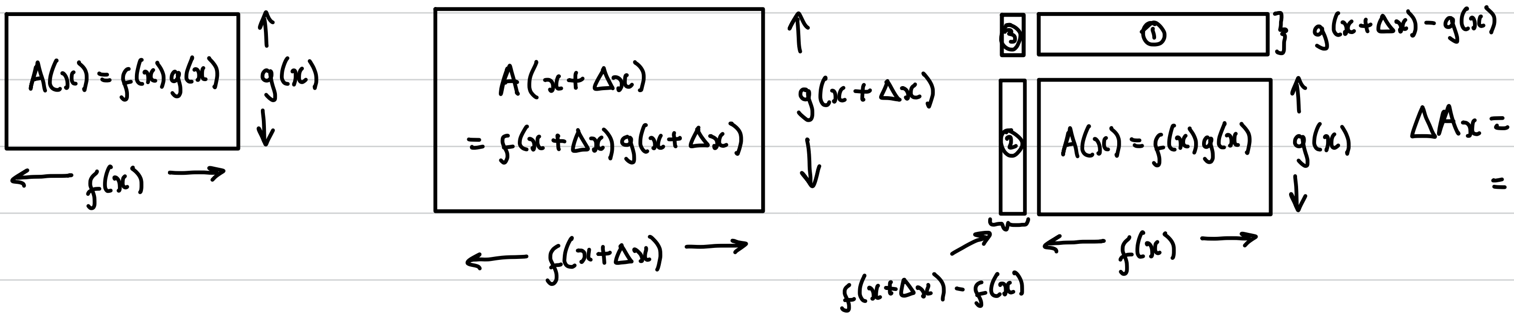
3. $f(x) = e^x + 2\sin(x) + x^3, \quad f'(x) = e^x + 2\cos(x) + 3x^2, \quad f''(x) = e^x - 2\sin(x) + 6x$

4. $f'(x) = x^4 - \sin(x) - 3e^x, \quad f(x) = \frac{1}{5}x^5 + \cos(x) - 3e^x$ (+/- a constant)
   (anti-derivative)

5. $\dfrac{d}{dx}(\sqrt{x}) = \dfrac{d}{dx}(x^{1/2}) = \dfrac{1}{2}x^{-1/2} = \dfrac{1}{2x^{1/2}} = \dfrac{1}{2\sqrt{x}}$

   $x^{-a} = \dfrac{1}{x^a}$

Product Rule



$$\Delta A_x = ① + ② + ③$$
$$= f(x)\big(g(x+\Delta x) - g(x)\big) + g(x)\big(f(x+\Delta x) - f(x)\big) + \big(f(x+\Delta x) - f(x)\big)\big(g(x+\Delta x) - g(x)\big)$$

as Δx goes to 0, $$\lim_{\Delta x \to 0} \big(\Delta A(x)\big) = \lim_{\Delta x \to 0} \Big(f(x)\big(g(x+\Delta x) - g(x)\big) + g(x)\big(f(x+\Delta x) - f(x)\big) + \big(f(x+\Delta x) - f(x)\big)\big(g(x+\Delta x) - g(x)\big)\Big)$$

Smallest rectangle shrinks the fastest (∴ can leave out of expression)

$$\lim_{\Delta x \to 0} \big(\Delta A(x)\big) = \lim_{\Delta x \to 0} \Big(f(x)\big(g(x+\Delta x) - g(x)\big) + g(x)\big(f(x+\Delta x) - f(x)\big)\Big)$$

$$\lim_{\Delta x \to 0} \left(\frac{\Delta A(x)}{\Delta x} \right) = \lim_{\Delta x \to 0} \left(\frac{f(x)\big(g(x+\Delta x) - g(x)\big) + g(x)\big(f(x+\Delta x) - f(x)\big)}{\Delta x} \right)$$

$$= \lim_{\Delta x \to 0} \left(f(x) \frac{\big(g(x+\Delta x) - g(x)\big)}{\Delta x} + g(x) \frac{\big(f(x+\Delta x) - f(x)\big)}{\Delta x} \right)$$

(if $A(x) = f(x)g(x)$) $$= A'(x) = f(x)g'(x) + g(x)f'(x)$$

Practice Quiz

1. $\dfrac{dA(x)}{dx} = \dfrac{df(x)}{dx} g(x) + f(x) \dfrac{dg(x)}{dx}$

2. $A(x) = (x+2)(3x-3)$, $A'(x) = 1 \times (3x-3) + (x+2) \times 3 = 3x - 3 + 3x + 6 = 6x + 3$

   $f(x) = x+2$, $g(x) = 3x-3$ $f'(x) = 1$, $g'(x) = 3$

3. $f(x) = x^3 \sin(x)$, $f'(x) = \sin(x) \times 3x^2 + x^3 \cos(x) = 3x^2 \sin(x) + x^3 \cos(x)$

   $u = x^3$, $v = \sin(x)$

   $\dfrac{du}{dx} = 3x^2$, $\dfrac{dv}{dx} = \cos(x)$

4. $f(x) = \dfrac{e^x}{x} = e^x \dfrac{1}{x}$, $f'(x) = \dfrac{1}{x} \times e^x + e^x \times -\dfrac{1}{x^2} = \dfrac{e^x}{x} - \dfrac{e^x}{x^2} = e^x \left(\dfrac{1}{x} - \dfrac{1}{x^2} \right)$

   $u = e^x$, $v = \dfrac{1}{x}$

   $\dfrac{du}{dx} = e^x$, $\dfrac{dv}{dx} = -\dfrac{1}{x^2}$

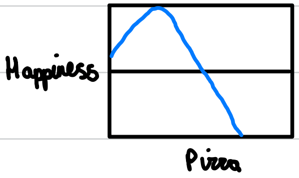
5. $U(x) = f(x) g(x) h(x)$, $A(x) = f(x) g(x)$, $A'(x) = f'(x) g(x) + f(x) g'(x)$

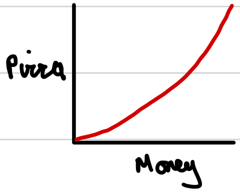
   $= A(x) h(x)$, $U'(x) = A'(x) h(x) + A(x) h'(x) = (f'(x) g(x) + f(x) g'(x)) h(x) + (f(x) g(x)) h'(x)$

6. $f(x) = x e^x \cos(x) = 1 \times e^x \times \cos(x) + x e^x \cos(x) + x e^x (-\sin(x)) = e^x \cos(x) + x e^x \cos(x) - x e^x \sin(x) = e^x (\cos(x) + x \cos(x) - x \sin(x))$

   $= e^x ((1+x) \cos(x) - x \sin(x))$

Chain Rule





$h(p(m))$, e.g. $h(p) = -\frac{1}{3}p^2 + p + \frac{1}{5}$

$p(m) = e^m - 1$

↳ $h(p(m)) = -\frac{1}{3}(e^m - 1)^2 + e^m - 1 + \frac{1}{5}$

$\frac{dh}{dm} = \frac{1}{3}e^m(5 - 2e^m)$

• : $h(p(m))$

• : derivative of

if $h = h(p)$ and $p = p(m)$

then $\frac{dh}{dm} = \frac{dh}{dp} \times \frac{dp}{dm}$

$\frac{dh}{dp} = 1 - \frac{2}{3}p$, $\frac{dp}{dm} = e^m$, $\frac{dh}{dp} \times \frac{dp}{dm} = (1 - \frac{2}{3}p)e^m = (1 - \frac{2}{3}(e^m - 1))e^m = \frac{1}{3}e^m(5 - 2e^m)$

can use quotient rule instead

example: $f(x) = \frac{\sin(2x^5 + 3x)}{e^{7x}} = \underbrace{e^{-7x}}_{h(x)}\left(\underbrace{\sin(2x^5 + 3x)}_{g(x)}\right)$

$h(v) = e^v$, $h'(v) = e^v$

$v(x) = -7x$, $v'(x) = -7$

$\frac{dh}{dv} \times \frac{dv}{dx} = -7e^{-7x}$

$g(u) = \sin(u)$, $g'(u) = \cos(u)$

$u(x) = 2x^5 + 3x$, $u'(x) = 10x^4 + 3$

$\frac{dg}{du} \times \frac{du}{dx} = \cos(u) \times (10x^4 + 3) = \cos(2x^5 + 3x)(10x^4 + 3)$

$\frac{df}{dx} = h\frac{dg}{dx} + g\frac{dh}{dx} = \dots = \frac{(10x^4 + 3)\cos(2x^5 + 3x)}{e^{7x}} - 7f(x)$

"premature optimisation is the root of all evil"

↳ don't spend time tidying up until you're sure you've finished making a mess!

Practice Quiz

1. $f(x) = g(h(x))$, $\frac{dg}{dx} = \frac{dg}{dh} \times \frac{dh}{dx} = g'(h(x)) \times h'(x)$

2. $f(x) = e^{x^2-3}$, $g(h) = e^h$, $h(x) = x^2 - 3$

   $f(x) = g(h(x))$, $f'(x) = \frac{dg}{dx} = \frac{dg}{dh} \times \frac{dh}{dx} = e^h \times 2x = 2xe^{x^2-3}$

3. $f(x) = \sin^3(x) = (\sin(x))^3$, $f'(x) = 3\sin^2(x)\cos(x)$

4. $f(x) = \tan(x) = \frac{\sin(x)}{\cos(x)} = \underbrace{(\cos(x))^{-1}}_{u}\underbrace{\sin(x)}_{v}$, $f'(x) = (-1(\cos(x))^{-2} \times -\sin(x)) \times \sin(x) + (\cos(x))^{-1}\cos(x)$

   $= \frac{\sin^2(x)}{\cos^2(x)} + 1 = \tan^2(x) + 1$

5. $f(g(h(x))) = \frac{df}{dx} = \frac{df}{dg}\frac{dg}{dh}\frac{dh}{dx}$

   $f(x) = e^{\sin(x^2)}$ $\qquad$ $f'(x) = e^g \times \cos(h) \times 2x = 2xe^{\sin(x^2)}\cos(x^2)$

   $f(g) = e^g$, $f'(g) = e^g$

   $g(h) = \sin(h)$, $g'(h) = \cos(h)$

   $h(x) = x^2$, $h'(x) = 2x$

Assessment Quiz

1. $f(x) = x^{3/2} + \pi x^2 + \sqrt{7}$, $f'(x) = \frac{3}{2}x^{\frac{1}{2}} + 2\pi x = \frac{3}{2}\sqrt{x} + 2\pi x$, @ $x=2$: $f'(2) = \frac{3}{2}\sqrt{2} + 4\pi$

2. $f(x) = x^3\cos(x)e^x$, $u(x) = f(x)g(x)h(x)$, $A(x) = f(x)g(x)$, $A'(x) = g(x)f'(x) + f(x)g'(x)$

$$= \cos(x)3x^2 + x^3(-\sin(x))$$
$$= 3x^2\cos(x) - x^3\sin(x)$$

$$u'(x) = A(x)h'(x) + h(x)A'(x) = e^x x^3\cos(x) + e^x(3x^2\cos(x) - x^3\sin(x))$$
$$= x^3e^x\cos(x) + 3x^2e^x\cos(x) - e^x x^3\sin(x)$$
$$= e^x x^2(x\cos(x) + 3\cos(x) - x\sin(x))$$

3. $f(x) = e^{[(x+1)^2]}$

$f(u) = e^{u^2}$ $\quad f'(u) = 2ue^{u^2} = 2(x+1)e^{(x+1)^2}$

$u(x) = (x+1)$, $u'(x) = 1$

$$\frac{df}{dx} = \frac{df}{du} \times \frac{du}{dx} = 2(x+1)e^{(x+1)^2}$$

4. $f(x) = \underbrace{x^2}_{h(x)}\underbrace{\cos(x^3)}_{g(x)}$,

(chain rule): $g(u(x)) \rightarrow \frac{dg}{dx} = \frac{dg}{du} \times \frac{du}{dx} = -3x^2\sin(x^3)$

$g(u) = \cos(u)$, $g'(u) = -\sin(u)$

$u(x) = x^3$, $u'(x) = 3x^2$ $\quad$ (product rule): $f'(x) = h(x)g'(x) + g(x)h'(x) = -3x^4\sin(x^3) + 2x\cos(x^3)$

5. $f(x) = \overbrace{\sin(x)}^{g(x)}\,\overbrace{e^{\cos(x)}}^{h(x)}$

$g(x) = \sin(x)$, $g'(x) = \cos(x)$

$h(x) = e^{\cos(x)}$

$h(u) = e^{u}$, $h'(u) = e^{u} = e^{\cos(x)}$

$u(x) = \cos(x)$, $u'(x) = -\sin(x)$

(chain rule on $h(u(x))$):

, $\dfrac{dh}{dx} = \dfrac{dh}{du} \times \dfrac{du}{dx} = -e^{\cos(x)}\sin(x)$

(product rule): $\dfrac{df}{dx} = e^{\cos(x)}\cos(x) - e^{\cos(x)}\sin^2(x)$

@ $x = \pi$: $f'(\pi) = e^{-1}(-1) - e^{-1} = -e^{-1} = \dfrac{-1}{e}$

Week 2: Multivariate Calculus

y $\qquad$ x

Dependent / Independent variables

Take functions w/ variables and differentiate y w.r.t. x

e.g. $F = ma + dv^2$ $\qquad$ (depends on context, could be other way around)

y $\quad$ x $\quad$ x ← variable

constants

"Parameters" $\qquad$ ↙ mass of can

e.g. $m = 2\pi r^2 t\rho + 2\pi r h t \rho$

partial derivative → $\dfrac{\partial m}{\partial h} = 2\pi r t \rho$ (no longer has h: mass will vary linearly with height when all else is constant)

(differentiating fn. with >1 variable) $\quad \dfrac{\partial m}{\partial r} = 4\pi r t\rho + 2\pi h t \rho$, $\quad \dfrac{\partial m}{\partial t} = 2\pi r^2 \rho + 2\pi r h \rho$, $\quad \dfrac{\partial m}{\partial \rho} = 2\pi r^2 t + 2\pi r h t$ $\quad$ (consider each variable separately)

e.g. $f(x,y,z) = \underbrace{\sin(x)}\, e^{yz^2}$ ↖ becomes constant as no y

$\dfrac{\partial f}{\partial x} = \cos(x) e^{yz^2}$, $\quad \dfrac{\partial f}{\partial y} = \sin(x) e^{yz^2} z^2$, $\quad \dfrac{\partial f}{\partial z} = \sin(x) e^{yz^2} 2yz$

(total derivative)

$\therefore \dfrac{df(x,y,z)}{dt} = \dfrac{\partial f}{\partial x}\dfrac{dx}{dt} + \dfrac{\partial f}{\partial y}\dfrac{dy}{dt} + \dfrac{\partial f}{\partial z}\dfrac{dz}{dt}$

can get complex!

Total derivative

x, y, z are themselves fn. of parameter t: $x = t-1$, $y = t^2$, $z = \dfrac{1}{t}$ → $f(t) = \sin(t-1) e^{t^2 \left(\frac{1}{t}\right)^2} = \sin(t-1) e$, $\quad \dfrac{df(t)}{dt} = \cos(t-1) e$

Practice Quiz

Treat every parameter and variable you aren't differentiating as a constant

1. $f(x,y) = \pi x^3 + xy^2 + my^4$, $\frac{\partial f}{\partial x} = 3\pi x^2 + y^2$, $\frac{\partial f}{\partial y} = 2xy + 4my^3$

2. $f(x,y,z) = x^2y + y^2z + z^2x$, $\frac{\partial f}{\partial x} = 2xy + z^2$, $\frac{\partial f}{\partial y} = x^2 + 2yz$, $\frac{\partial f}{\partial z} = y^2 + 2zx$

3. $f(x,y,z) = e^{2x}\sin(y)z^2 + \cos(z)e^xe^y$

   $\frac{\partial f}{\partial x} = 2e^{2x}\sin(y)z^2 + \cos(z)e^xe^y$, $\frac{\partial f}{\partial y} = e^{2x}\cos(y)z^2 + \cos(z)e^xe^y$, $\frac{\partial f}{\partial z} = 2ze^{2x}\sin(y) - \sin(z)e^xe^y$

4. $f(x,y)$, $x = x(t)$, $y = y(t)$, $\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt}$ (total derivative)

   $f(x,y) = \frac{\sqrt{x}}{y}$, $x(t) = t$, $y(t) = \sin(t)$, $\frac{dx}{dt} = 1$, $\frac{dy}{dt} = \cos(t)$, $\frac{\partial f}{\partial x} = \frac{1}{2}\frac{1}{y}x^{-1/2} = \frac{1}{2y\sqrt{x}}$, $\frac{\partial f}{\partial y} = -y^{-2}\sqrt{x} = -\frac{\sqrt{x}}{y^2}$

   $\left(\frac{1}{y}(x^{1/2})\right)$

   $$\frac{df}{dt} = \frac{1}{2y\sqrt{x}} - \frac{\sqrt{x}}{y^2}\cos(x) = \frac{1}{2\sin(t)\sqrt{t}} - \frac{\sqrt{t}}{\sin^2(t)}\cos(t)$$

5. $f(x,y,z)$, $x=x(t)$, $y=y(t)$, $z=z(t)$, $\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} + \frac{\partial f}{\partial z}\frac{dz}{dt}$

$f(x,y,z) = \cos(x)\sin(y)e^{2z}$, $x(t) = t+1$, $y(t) = t-1$, $z(t) = t^2$

$\frac{\partial f}{\partial x} = -\sin(x)\sin(y)e^{2z}$, $\frac{\partial f}{\partial y} = \cos(x)\cos(y)e^{2z}$, $\frac{\partial f}{\partial z} = \cos(x)\sin(y)2e^{2z}$, $\frac{dx}{dt} = 1$, $\frac{dy}{dt} = 1$, $\frac{dz}{dt} = 2t$

$$\frac{df}{dt} = -\sin(x)\sin(y)e^{2z} + \cos(x)\cos(y)e^{2z} + \cos(x)\sin(y)2e^{2z}(2t)$$

$$= -\sin(t+1)\sin(t-1)e^{2t^2} + \cos(t+1)\cos(t-1)e^{2t^2} + \cos(t+1)\sin(t-1)4te^{2t^2}$$

$$= e^{2t^2}\left[-\sin(t+1)\sin(t-1) + \cos(t+1)\cos(t-1) + \cos(t+1)\sin(t-1)4t\right]$$

The Jacobian

row vector of partial derivatives w.r.t. each variable

$f(x_1, x_2, x_3, \ldots)$

$J = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}, \ldots \right]$

tells us direction and magnitude of gradient at each point in space

for specific x, y, z coordinate, will return vector pointing in direction of steepest uphill slope of f

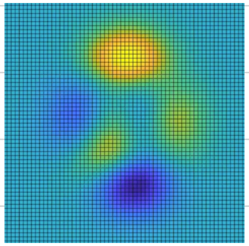
$f(x, y, z) = x^2 y + 3z$

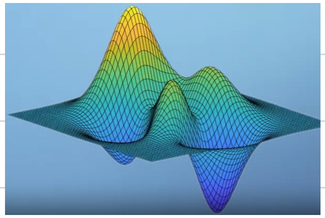
$\frac{\partial f}{\partial x} = 2xy, \quad \frac{\partial f}{\partial y} = x^2, \quad \frac{\partial f}{\partial z} = 3, \quad J = [2xy, x^2, 3]$

constant contribution of z, doesn't depend on location selected

e.g. $J(0,0,0) = [0,0,3]$

e.g. $z(x,y) = 3(1-x)^2 e^{-x^2-(y+1)^2} - 10\left(\frac{x}{5} - x^3 - y^5\right) e^{-x^2-y^2} - \frac{1}{3} e^{-(x+1)^2-y^2}$

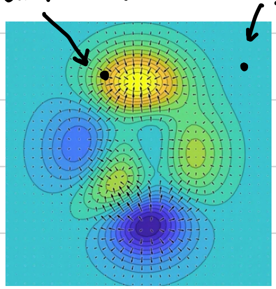




contour lines along same z

large Jacobian

small Jacobian



(contour plot)

Practice Quiz

1. $f(x,y) = x^2 y + \frac{3}{4}xy + 10, \quad \frac{\partial f}{\partial x} = 2xy + \frac{3}{4}y, \quad \frac{\partial f}{\partial y} = x^2 + \frac{3}{4}x, \quad J = \left[2xy + \frac{3}{4}y, x^2 + \frac{3}{4}x\right]$

2. $f(x,y) = e^x \cos(y) + xe^{3y} - 2, \quad J = \left[e^x \cos(y) + e^{3y}, -e^x \sin(y) + 3xe^{3y}\right]$

3. $f(x,y,z) = e^x \cos(y) + x^2 y^2 z^2, \quad J = \left[e^x \cos(y) + 2xy^2 z^2, -e^x \sin(y) + 2x^2 y z^2, e^x \cos(y) + 2x^2 y^2 z\right]$

4. $f(x,y,z) = x^2 + 3e^y e^z + \cos(x)\sin(z)$, $J = [2x - \sin(x)\sin(z),\ x^2 + 3e^y e^z + \cos(x)\sin(z),\ x^2 + 3e^y e^z + \cos(x)\cos(z)]$

↳ $J(0,0,0) = [0,3,4]$

5. $f(x,y,z) = xe^y\cos(z) + 5x^2\sin(y)e^z$, $J = [e^y\cos(z) + 10x\sin(y)e^z,\ xe^y\cos(z) + 5x^2\cos(y)e^z,\ -xe^y\sin(z) + 5x^2\sin(y)e^z]$

↳ $J(0,0,0) = [1,0,0]$

Applied Jacobian

$f(x,y) = e^{-(x^2+y^2)}$

$J = [-2xe^{-(x^2+y^2)},\ -2ye^{-(x^2+y^2)}]$

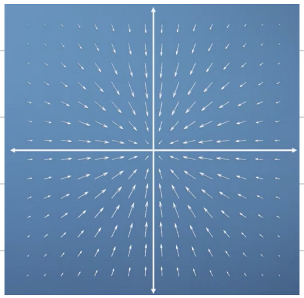
e.g. $J(-1,1) = [0.27, -0.27]$

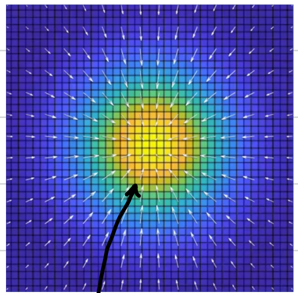
$J(2,2) = [-0.001, -0.001]$

$J(0,0) = [0,0]$ (zero vector)

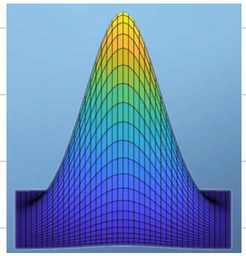
max or min or saddle (must be max given right picture)

Jacobian vector field





high values of f

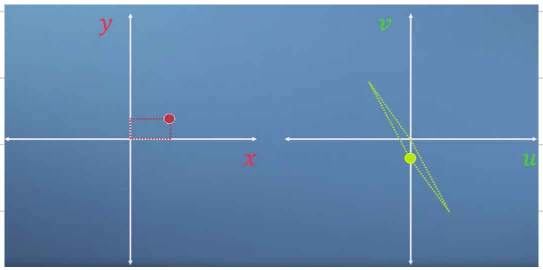


function in 3D

$u(x,y) = x - 2y$, $v(x,y) = 3y - 2x$

(2 vector spaces, 1 with coordinates in u,v and other in x,y)

↳ each point in x,y has corresponding location in u,v



$J_u = [\partial u/\partial x,\ \partial u/\partial y]$, $J_v = [\partial v/\partial x,\ \partial v/\partial y]$

$$J = \begin{bmatrix} \partial u/\partial x, & \partial u/\partial y \\ \partial v/\partial x, & \partial v/\partial y \end{bmatrix}$$ (Jacobian matrix for vector-valued functions)

↳ describes gradient of multi-variable system. For scalar valued multi-variable function: get row vector pointing in direction of greatest uphill slope with length proportional to local steepness

doesn't have variables : u and v are linear functions of x and y,
so gradient must be constant everywhere

(linear example)

$u(x,y) = x - 2y$, $v(x,y) = 3y - 2x$, $J = \begin{bmatrix} 1 & -2 \\ -2 & 3 \end{bmatrix}$

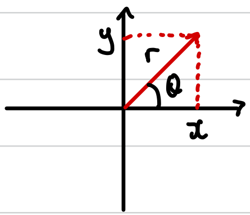
transformation matrix : linear transformation from x,y space to u,v space

e.g. $\begin{bmatrix} 1 & -2 \\ -2 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -4 \\ 5 \end{bmatrix}$

Non-linear functions may be smooth : each little region of space approximately linear when looking closely

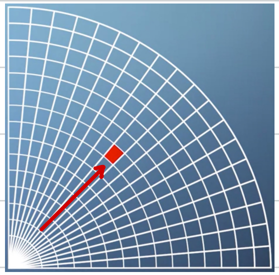
↳ by adding all contributions from Jacobian determinants at each point in space, can still calculate change in size of region after transformation

Cartesian to polar coordinates :



$x(r, \theta) = r\cos\theta$ (vector in terms of r and θ,

$y(r, \theta) = r\sin\theta$ but want in terms of x and y instead)



$$J = \begin{bmatrix} \partial x/\partial r & \partial x/\partial \theta \\ \partial y/\partial r & \partial y/\partial \theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{bmatrix}$$

$|J| = r(\cos^2\theta + \sin^2\theta) = r$

↳ in terms of r and not θ : as you move along r and away from origin, small regions of space will scale as a function of r

Practice Quiz

1. Jacobian matrix for vector valued function: $u(x,y) = x^2 - y^2$, $v(x,y) = 2xy$, $J = \begin{bmatrix} 2x & -2y \\ 2y & 2x \end{bmatrix}$

2. $u(x,y,z) = 2x + 3y$, $v(x,y,z) = \cos(x)\sin(z)$, $w(x,y,z) = e^x e^y e^z$, $J = \begin{bmatrix} 2 & 3 & 0 \\ -\sin(x)\sin(z) & 0 & \cos(x)\cos(z) \\ e^x e^y e^z & e^x e^y e^z & e^x e^y e^z \end{bmatrix}$

3. Linear equations: $u(x,y) = ax + by$, $v(x,y) = cx + dy$, $J = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$

4. $u(x,y,z) = 9x^2y^2 + ze^x$, $v(x,y,z) = xy + x^2y^3 + 2z$, $w(x,y,z) = \cos(x)\sin(z)e^y$, $J = \begin{bmatrix} 18x + ze^x & 18x^2y & e^x \\ y + 2xy^3 & x + 3x^2y^2 & 2 \\ -\sin(x)\sin(z)e^y & \cos(x)\sin(z)e^y & \cos(x)\cos(z)e^y \end{bmatrix}$

$J(0,0,0) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix}$

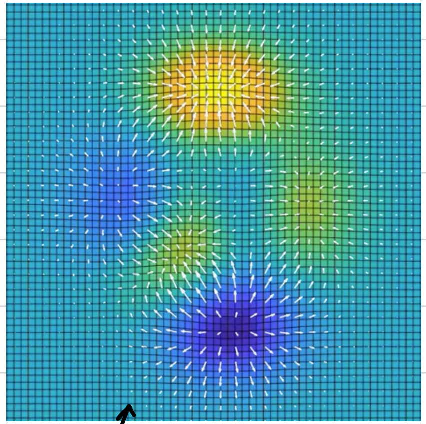
5. Jacobian of transformation from Spherical to 3D coordinates

$x(r,\theta,\phi) = r\cos(\theta)\sin(\phi)$, $y(r,\theta,\phi) = r\sin\theta\sin(\phi)$, $z(r,\theta,\phi) = r\cos(\phi)$

$$J = \begin{bmatrix} \cos(\theta)\sin(\phi) & -r\sin(\theta)\sin(\phi) & r\cos(\theta)\cos(\phi) \\ \sin(\theta)\sin(\phi) & r\cos(\theta)\sin(\phi) & r\sin(\theta)\cos(\phi) \\ \cos(\phi) & 0 & -r\sin(\phi) \end{bmatrix}$$

The Sandpit

Optimisation: finding input values to functions which correspond to max or min of a system



Jacobians point uphill but not necessarily tallest hill, can end up in local maximum

Can solve system analytically by finding Jacobian then find values of x and y which make it equal 0

↳ but can be tricky for complex functions (can find expression, but can have multiple locations of 0 gradient)

Peaks = maxima, tallest peak = global maximum, rest are local maxima

Troughs = minima, deepest point = global minimum, rest are local minima

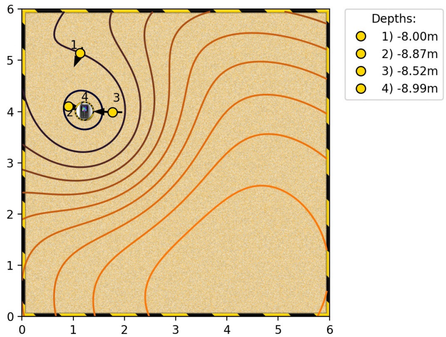
May be unable to find nice analytical expression so unable to plot entire function and look around

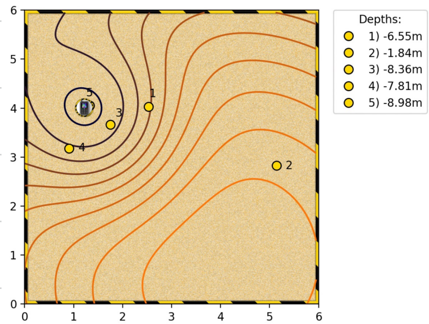
Can try function at different places, no need to evaluate everywhere inbetween

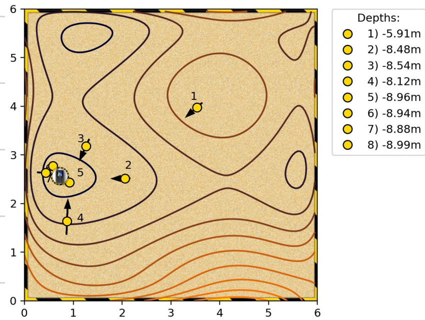
Calculation costs the same no matter how far points are

Multiple minima, noisy functions, narrow wells (can be computationally expensive!)

Measure negative of Jacobian, $-J = -\nabla f(x)$ to go downhill







(gradient descent)

Congratulations!

Well done, you found the phone.

The Hessian

allows us to check which feature we are standing on with point of 0 gradient

Jacobian : vector of first-order derivatives

Hessian : matrix of second-order derivatives for function of n variables

Can keep differentiating : w.r.t. x_1 keep others constant, then w.r.t. x_2 keep others constant

$$H = \begin{bmatrix} \partial^2 f/\partial x_1^2 & \partial^2 f/\partial x_1 x_2 & \cdots & \partial^2 f/\partial x_1 x_n \\ \partial^2 f/\partial x_2 x_1 & \partial^2 f/\partial x_2^2 & \cdots & \partial^2 f/\partial x_2 x_n \\ \vdots & \vdots & \ddots & \vdots \\ \partial^2 f/\partial x_n x_1 & \partial^2 f/\partial x_n x_2 & \cdots & \partial^2 f/\partial x_n^2 \end{bmatrix}$$

$$= \begin{bmatrix} \partial_{x_1 x_1} f & \partial_{x_1 x_2} f & \cdots & \partial_{x_1 x_n} f \\ \partial_{x_2 x_1} f & \partial_{x_2 x_2} f & \cdots & \partial_{x_2 x_n} f \\ \vdots & \vdots & \ddots & \vdots \\ \partial_{x_n x_1} f & \partial_{x_n x_2} f & \cdots & \partial_{x_n x_n} f \end{bmatrix}$$ nxn square matrix

e.g. $H = \begin{bmatrix} \partial_{x,x} f & \partial_{x,y} f & \partial_{x,z} f \\ \partial_{y,x} f & \partial_{y,y} f & \partial_{y,z} f \\ \partial_{z,x} f & \partial_{z,y} f & \partial_{z,z} f \end{bmatrix}$

Find Jacobian, then differentiate terms to get Hessian

$f(x,y,z) = x^2yz$

$J = [2xyz, x^2z, x^2y]$ (Jacobian row vector)

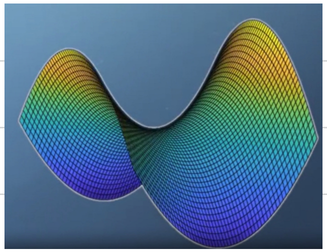
$H = \begin{bmatrix} 2yz & 2xz & 2xy \\ 2xz & 0 & x^2 \\ 2xy & x^2 & 0 \end{bmatrix}$ (partial derivative w.r.t. x, y, z for each component, 1^{st} row = 1^{st} component, 2^{nd} row = 2^{nd} component etc.)

↳ Symmetric across leading diagonal (true for continuous functions)

e.g. $f(x,y) = x^2+y^2$, $J = [2x, 2y]$, $H = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$, $|H| = 4$ ↙ determinant

↖ if +ve : maximum or minimum, -ve : not maximum or minimum

↳ look at top left = 2 (+ve : minimum, -ve : maximum)



$f(x,y) = x^2 - y^2$

gradient = 0 in saddle point in middle

Practice Quiz

1. $f(x,y) = x^3y + x + 2y$, $J = \begin{bmatrix} 3x^2y + 1, & x^3 + 2 \end{bmatrix}$, $H = \begin{bmatrix} 6xy & 3x^2 \\ 3x^2 & 0 \end{bmatrix}$

2. $f(x,y) = e^x\cos(y)$, $J = \begin{bmatrix} e^x\cos(y), & -e^x\sin(y) \end{bmatrix}$, $H = \begin{bmatrix} e^x\cos(y) & -e^x\sin(y) \\ -e^x\sin(y) & -e^x\cos(y) \end{bmatrix}$

3. $f(x,y) = \frac{x^2}{2} + xy + \frac{y^2}{2}$, $J = \begin{bmatrix} x+y, & x+y \end{bmatrix}$, $H = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

4. $f(x,y,z) = x^2e^{-y}\cos(z)$, $J = \begin{bmatrix} 2xe^{-y}\cos(z), & -x^2e^{-y}\cos(z), & -x^2e^{-y}\sin(z) \end{bmatrix}$

$$H = \begin{bmatrix} 2e^{-y}\cos(z) & -2xe^{-y}\cos(z) & -2xe^{-y}\sin(z) \\ -2xe^{-y}\cos(z) & x^2e^{-y}\cos(z) & -x^2e^{-y}\cos(z) \\ -2xe^{-y}\sin(z) & x^2e^{-y}\sin(z) & -x^2e^{-y}\cos(z) \end{bmatrix}$$

5. $f(x,y,z) = xe^y + y^2\cos(z)$, $J = \begin{bmatrix} e^y, & xe^y + 2y\cos(z), & -y^2\sin(z) \end{bmatrix}$

$$H = \begin{bmatrix} 0 & e^y & 0 \\ e^y & xe^y + 2\cos(z) & -2y\sin(z) \\ 0 & -2y\sin(z) & -y^2\cos(z) \end{bmatrix}$$

Numerical methods: if we don't have function to optimise, how to build Jacobian of partial derivatives?

↳ many problems with no nice explicit formula or do but solving directly takes too long

↳ can generate approximate solutions

$$J = \left[\frac{f(x+\Delta x, y) - f(x,y)}{\Delta x}, \frac{f(x, y+\Delta y) - f(x,y)}{\Delta y} \right]$$ (start from initial location, approximate each partial derivative in turn)

↳ how big of a step to take? too big: overshoot, too small: numerical issues

↳ computer evaluates function at a point: only stored to a no. of significant figures, so if point is too close, computer may not register any change at all

↳ noisy data: calculate gradient using different step sizes, and take an average

Practice Quiz

1. $f(x,y,z) = x^2\cos(y) + e^z\sin(y)$, $J = \left[2x\cos(y), -x^2\sin(y) + e^z\cos(y), e^z\sin(y)\right]$, @ $(\pi, \pi, 1)$. $J(\pi,\pi,1) = (-2\pi, -e, 0)$

2. Jacobian of vector valued functions: $u(x,y) = x^2y - \cos(x)\sin(y)$, $v(x,y) = e^{x+y}$

$$J = \begin{bmatrix} 2xy + \sin(x)\sin(y), & x^2 - \cos(x)\cos(y) \\ e^{x+y} & e^{x+y} \end{bmatrix}, @ (0,\pi): J(0,\pi) = \begin{bmatrix} 0 & 1 \\ e^\pi & e^\pi \end{bmatrix}$$

3. $f(x,y) = x^3\cos(y) - x\sin(y)$, $J = \left[3x^2\cos(y) - \sin(y), -x^3\sin(y) - x\cos(y)\right]$, $H = \begin{bmatrix} 6x\cos(y) & -3x^2\sin(y) - \cos(y) \\ -3x^2\sin(y) - \cos(y) & -x^3\cos(y) + x\sin(y) \end{bmatrix}$

4. $f(x,y,z) = xy + \sin(y)\sin(z) + z^3e^x$, $J = \left[y + z^3e^x,\ x + \cos(y)\sin(z),\ \sin(y)\cos(z) + 3z^2e^x\right]$

$$H = \begin{bmatrix} z^3e^x & 1 & 3z^2e^x \\ 1 & -\sin(y)\sin(z) & \cos(y)\cos(z) \\ 3z^2e^x & \cos(y)\cos(z) & -\sin(y)\sin(z) + 6ze^x \end{bmatrix}$$

5. $f(x,y,z) = xy\cos(z) - \sin(x)e^yz^3$, $J = \left[y\cos(z) - \cos(x)e^yz^3,\ x\cos(z) - \sin(x)e^yz^3,\ -xy\sin(z) - 3z^2\sin(x)e^y\right]$

$$H = \begin{bmatrix} \sin(x)e^yz^3 & \cos(z) - \cos(x)e^yz^3 & -y\sin(z) - 3z^2\cos(x)e^y \\ \cos(z) - \cos(x)e^yz^3 & -\sin(x)e^yz^3 & -x\sin(z) - 3z^2\sin(x)e^y \\ -xy\cos(z) - 6z\sin(x)e^y & -x\sin(z) - 3z^2\sin(x)e^y & -xy\cos(z) - 6z\sin(x)e^y \end{bmatrix}$$

$@(0,0,0)$: $H(0,0,0) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Week 3 : Multivariate Chain Rule

Total derivative : e.g. $f(x,y,z) = \sin(x)e^{yz^2}$, $x = t-1$; $y = t^2$; $z = \frac{1}{t}$

$$\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{dx}{\partial t} + \frac{\partial f}{\partial y}\frac{dy}{dt} + \frac{\partial f}{\partial z}\frac{dz}{dt} = \cos(t-1)e$$ (piecewise manner - quick for computers)

n-dimensional vector

$f(x_1, x_2, x_3, \dots, x_n) = f(\mathbf{x})$

$x_1(t) = \dots$

$x_2(t) = \dots$

$\vdots$

$x_n(t) = \dots$

$f(\mathbf{x}(t))$

$\frac{df}{dt} = ?$

$$\frac{\partial f}{\partial \mathbf{x}} = \begin{bmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \\ \vdots \\ \partial f / \partial x_n \end{bmatrix} \qquad \frac{d\mathbf{x}}{dt} = \begin{bmatrix} dx_1 / dt \\ dx_2 / dt \\ \vdots \\ dx_n / dt \end{bmatrix}$$

column vector $\cdot\ (J_f)^T$

$$\frac{df}{dt} = \underbrace{\frac{\partial f}{\partial \mathbf{x}} \cdot \frac{d\mathbf{x}}{dt}} = J_f \frac{d\mathbf{x}}{dt}$$ (generalized form of multivariate chain rule)

dot product of 2 column vectors is same as multiplying row vector by column vector

e.g. $f(x) = 5x$, $x(u) = 1-u$, $u(t) = t^2$

$(f(t) = 5 - 5t^2)$

↳ $\frac{df}{dt} = -10t$

works for chains of univariate functions (works for multivariate, but more details)

↳ chain rule : $\frac{df}{dt} = \frac{df}{dx}\frac{dx}{du}\frac{du}{dt} = (5)(-1)(2t) = -10t$

(still relating scalar input t to scalar output f through 2 intermediary vector valued functions)

$$f(x(u(t)))$$

$$f(x) = f(x_1, x_2), \quad x(u) = \begin{bmatrix} x_1(u_1, u_2) \\ x_2(u_1, u_2) \end{bmatrix}, \quad u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

vector-valued function

2 inputs, 2 outputs

$$\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial u}\frac{du}{dt} = \underbrace{\begin{bmatrix} \frac{\partial f}{\partial x_1}, & \frac{\partial f}{\partial x_2} \end{bmatrix}}_{\text{Jacobian of } f} \underbrace{\begin{bmatrix} \frac{\partial x_1}{\partial u_1} & \frac{\partial x_1}{\partial u_2} \\ \frac{\partial x_2}{\partial u_1} & \frac{\partial x_2}{\partial u_2} \end{bmatrix}}_{\text{Jacobian of } x} \underbrace{\begin{bmatrix} \frac{du_1}{dt} \\ \frac{du_2}{dt} \end{bmatrix}}_{\text{Derivative vector of } u}$$

$$(1\times1) = (1\times2) \times (2\times2) \times (2\times1)$$

Practice Quiz

$x = (x_1, x_2)$

1. $f(x) = f(x_1, x_2) = x_1^2 x_2^2 + x_1 x_2$

$x_1(t) = 1 - t^2$

$x_2(t) = 1 + t^2$

$$\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} = \begin{bmatrix} 2x_1x_2^2 + x_2, & 2x_1^2x_2 + x_1 \end{bmatrix}\begin{bmatrix} -2t \\ 2t \end{bmatrix}$$

$x = (x_1, x_2, x_3)$

2. $f(x) = f(x_1, x_2, x_3) = x_1^3 \cos(x_2) e^{x_3}$

$x_1(t) = 2t$

$x_2(t) = 1 - t^2$

$x_3(t) = e^t$

$$\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} = \begin{bmatrix} 3x_1^2\cos(x_2)e^{x_3}, & -x_1^3\sin(x_2)e^{x_3}, & x_1^3\cos(x_2)e^{x_3} \end{bmatrix}\begin{bmatrix} 2 \\ -2t \\ e^t \end{bmatrix}$$

3. $f(x) = f(x_1, x_2) = x_1^2 - x_2^2$

$x_1(u_1, u_2) = 2u_1 + 3u_2$

$x_2(u_1, u_2) = 2u_1 - 3u_2$

$u_1(t) = \cos(t/2)$

$u_2(t) = \sin(2t)$

$$\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial u}\frac{du}{dt} = \begin{bmatrix} 2x_1, & -2x_2 \end{bmatrix}\begin{bmatrix} 2 & 3 \\ 2 & -3 \end{bmatrix}\begin{bmatrix} -\frac{1}{2}\sin(\frac{t}{2}) \\ 2\cos(2t) \end{bmatrix}$$

4. $f(x) = f(x_1, x_2) = \cos(x_1)\sin(x_2)$ $\quad (x = (x_1, x_2),\ u = (u_1, u_2))$

$x_1(u_1, u_2) = 2u_1^2 + 3u_2^2 - u_2$

$x_2(u_1, u_2) = 2u_1 - 5u_2^3$

$u_1(t) = e^{t/2}$

$u_2(t) = e^{-2t}$

$$\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial u}\frac{du}{dt} = \begin{bmatrix} -\sin(x_1)\sin(x_2), & \cos(x_1)\cos(x_2) \end{bmatrix}\begin{bmatrix} 4u_1 & 6u_2 - 1 \\ 2 & -15u_2^2 \end{bmatrix}\begin{bmatrix} \frac{1}{2}e^{t/2} \\ -2e^{-2t} \end{bmatrix}$$

5. $f(x) = f(x_1, x_2, x_3) = \sin(x_1)\cos(x_2)e^{x_3}$ $\quad (x = (x_1, x_2),\ u = (u_1, u_2))$

$x_1(u_1, u_2) = \sin(u_1) + \cos(u_2)$

$x_2(u_1, u_2) = \cos(u_1) - \sin(u_2)$

$x_3(u_1, u_2) = e^{u_1 + u_2}$

$u_1(t) = 1 + t/2$

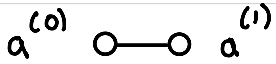
$u_2(t) = 1 - t/2$

$$\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial u}\frac{du}{dt} = \begin{bmatrix} \cos(x_1)\cos(x_2)e^{x_3}, & -\sin(x_1)\sin(x_2)e^{x_3}, & \sin(x_1)\cos(x_2)e^{x_3} \end{bmatrix} \times$$

$$\begin{bmatrix} \cos(u_1) & -\sin(u_2) \\ -\sin(u_1) & -\cos(u_2) \\ e^{u_1+u_2} & e^{u_1+u_2} \end{bmatrix}\begin{bmatrix} 1/2 \\ -1/2 \end{bmatrix}$$

Simple Neural Networks

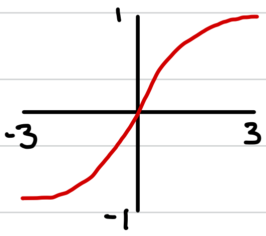
$y = f(x)$



$a^{(1)} = \sigma(wa^{(0)} + b)$

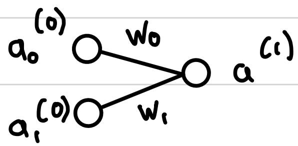
a : activity, w : weight, b : bias

σ : activation function



$\sigma(x) = \tanh(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$

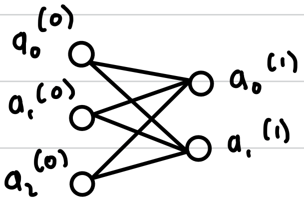
belongs to family of sigmoids



$$a^{(1)} = \sigma(w_0 a_0^{(0)} + w_1 a_1^{(0)} + b)$$

$$= \sigma\left(\left(\sum_{j=0}^{n} w_j a_j^{(0)}\right) + b\right)$$

$$= \sigma(w \cdot a^{(0)} + b)$$

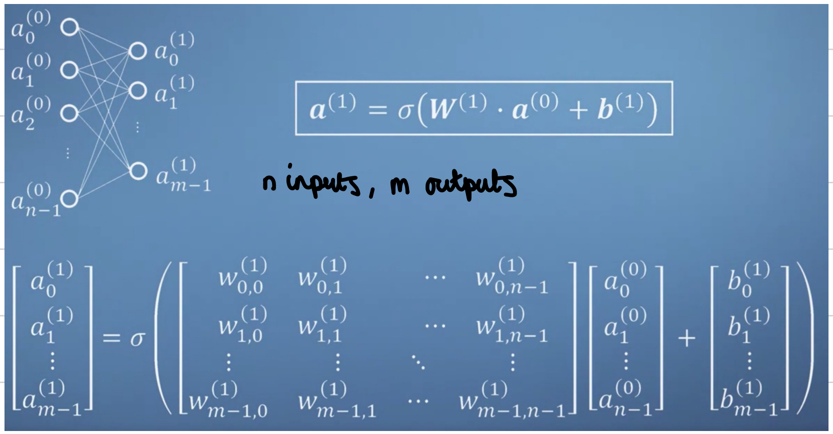


$a_0^{(1)} = \sigma(w_0 \cdot a^{(0)} + b_0)$

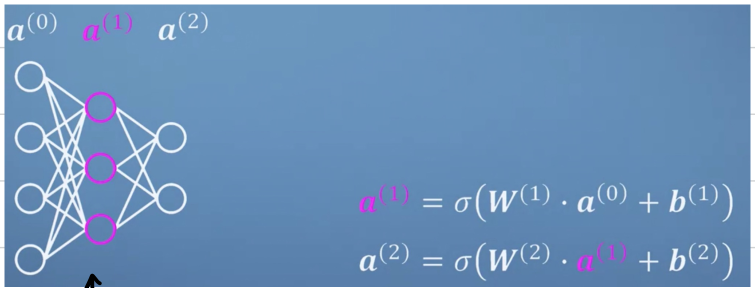
$a_1^{(1)} = \sigma(w_1 \cdot a^{(0)} + b_1)$

$\Rightarrow a^{(1)} = \sigma(W^{(1)} \cdot a^{(0)} + b^{(1)})$

$a^{(L)} = \sigma(W^{(L)} \cdot a^{(L-1)} + b^{(L)})$ (feed-forward nn)



(single layer nn)

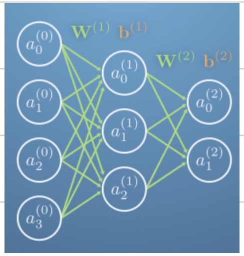


fully-connected feed-forward

Practice Quiz

1. $w^{(1)} = -5$, $b^{(1)} = 5$ (NOT function, using tanh)

2. $W^{(1)} = \begin{bmatrix} -2 & 4 & -1 \\ 6 & 0 & -3 \end{bmatrix}$, $b = \begin{bmatrix} 0.1 \\ -2.5 \end{bmatrix}$, $a^{(0)} = \begin{bmatrix} 0.3 \\ 0.4 \\ 0.1 \end{bmatrix}$, $a^{(1)} = \sigma(W^{(1)}a^{(0)} + b^{(1)}) = \sigma\left(\begin{bmatrix} 0.9 \\ 1.5 \end{bmatrix} + \begin{bmatrix} 0.1 \\ -2.5 \end{bmatrix}\right) = \sigma\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix}\right) = [0.762, -0.762]$

3. 

- has 5 biases
- no. of weights in a layer : product of input and output neurons to that layer

4. $a^{(2)} = \sigma(W^{(2)}\sigma(W^{(1)}a^{(0)} + b^{(1)}) + b^{(2)})$

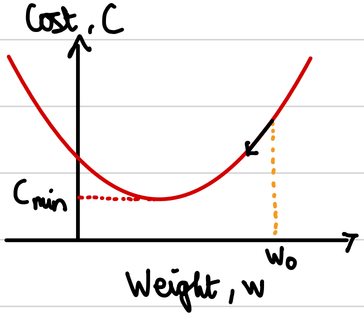
More NN's

Labelled inputs fed to NN, trained using backpropagation

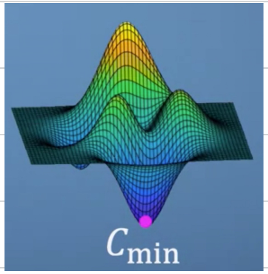
↳ want to find weights and biases that cause network to best match inputs to outputs

Initialise weights randomly

Define cost function: $C = \sum_i (a_i^{(L)} - y_i)^2$



(may have many minima)



multi-dimensional hypersurface

need to find Jacobian: partial derivatives of cost function w.r.t. all relevant variables (to go downhill)

$a^{(0)}$ ○——○ $a^{(1)}$

$$z^{(1)} = w^{(1)} a^{(0)} + b$$
$$a^{(1)} = \sigma(z^{(1)})$$
$$C = (a^{(1)} - y)^2$$

$$\frac{\partial C_k}{\partial w^{(1)}} = \frac{\partial C_k}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial w^{(1)}}$$

$$\frac{\partial C_k}{\partial b^{(1)}} = \frac{\partial C_k}{\partial a^{(1)}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial b^{(1)}}$$

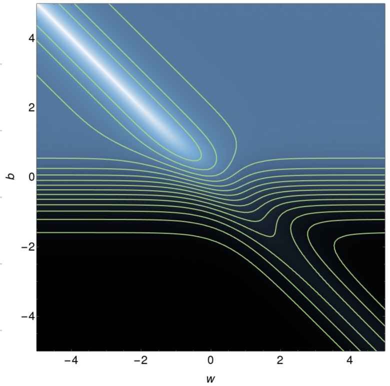
(chain rule expressions required to navigate 2D w, b space to minimise cost of network for set of training examples)

$$z^{(L)} = W^{(L)} \cdot a^{(L-1)} + b^{(L)}, \quad a^{(L)} = \sigma(z^{(L)}), \quad C = \sum_i (a_i^{(L)} - y_i)^2 \qquad (a^{(0)} = x)$$

Neural Networks Quiz

1. $C_n = (a^{(1)} - y)^2$, NOT function : $x = 0$, $w^{(1)} = 1.3$, $\sigma = \tanh$, $b^{(1)} = -0.1$, $C_0 = 1.2$

(cost function as contour map)



white : low cost

black : high cost

2. $C = \frac{1}{N} \sum_n C_n$, N = training set examples

examples: $(x = 0, y = 1)$, $(x = 1, y = 0)$, $C = \frac{1}{2}(C_0 + C_1)$ (2D parameter space $w^{(1)}$, $b^{(1)}$)

Descending perpendicular to contours will improve performance of network

Optimal configuration lies somewhere along line $b = -w$

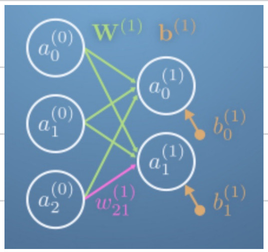
(System asymptotically approaches minimum along that line)

3. Can vary w and b to improve performance : chain rule to calculate derivative of cost

$$\frac{\partial z^{(1)}}{\partial b^{(1)}} = 1, \quad \frac{\partial a^{(1)}}{\partial z^{(1)}} = \sigma'(z^{(1)}), \quad \frac{\partial C_n}{\partial a^{(1)}} = 2(a^{(1)} - y), \quad \frac{\partial z^{(1)}}{\partial w^{(1)}} = a^{(0)}$$

if bias is increased by a small amount, weighted sum increases by same amount

4. $\frac{d}{dz}\tanh(z) = \frac{1}{\cosh^2 z}$, (code for derivative calculation)

5. 

training example

$a^{(1)} = \sigma(z^{(1)})$, $z^{(1)} = W^{(1)} a^{(0)} + b^{(1)}$, $C_n = \sum_i (a_i^{(1)} - y_i)^2$ output neuron

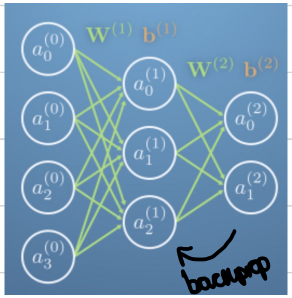
$x \to x$ (Same no. of elements as input neurons)

$y \to y$ (Same no. of elements as output neurons)

$C_n = |a^{(1)} - y|^2$ (code to calculate loss function)

$= 1.8$

6. NN. with hidden layers



$$\frac{\partial C_n}{\partial W^{(2)}} = \frac{\partial C_n}{\partial a^{(2)}} \frac{\partial a^{(2)}}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial W^{(2)}}$$

(derivative of cost w.r.t. weights of final layer)

$$\frac{\partial C_n}{\partial W^{(1)}} = \frac{\partial C_n}{\partial a^{(2)}} \underbrace{\frac{\partial a^{(2)}}{\partial a^{(1)}}}_{\frac{\partial a^{(2)}}{\partial z^{(2)}} \frac{\partial z^{(2)}}{\partial a^{(1)}}} \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial W^{(1)}}$$

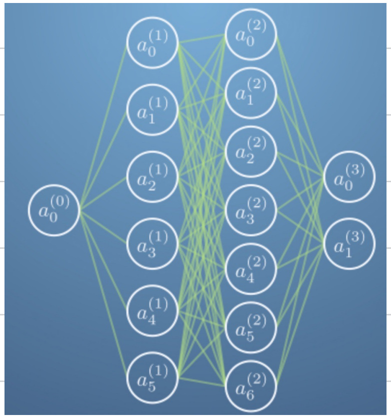
(derivative of cost w.r.t. weights of previous layer)

$$\frac{\partial C_n}{\partial W^{(i)}} = \frac{\partial C_n}{\partial a^{(N)}} \underbrace{\frac{\partial a^{(N)}}{\partial a^{(N-1)}} \frac{\partial a^{(N-1)}}{\partial a^{(N-2)}} \cdots \frac{\partial a^{(i+1)}}{\partial a^{(i)}}}_{\text{from layer } N \text{ to layer } i} \frac{\partial a^{(i)}}{\partial z^{(i)}} \frac{\partial z^{(i)}}{\partial W^{(i)}}$$

$$a^{(n)} = \sigma(z^{(n)})$$
$$z^{(n)} = w^{(n)} a^{(n-1)} + b^{(n)}$$

$$\frac{\partial a^{(j)}}{\partial a^{(j-1)}} = \frac{\partial a^{(j)}}{\partial z^{(j)}} \frac{\partial z^{(j)}}{\partial a^{(j-1)}} = \sigma'(z^{(j)}) W^{(j)}$$

Backpropagation Lab



Element-wise multiplication : *

Matrix multiplication : @ $\quad (n \times \overset{\text{same}}{m \; @ \; n} \times m)$

(train network to draw a curve : 1 input (amount travelled along curve from 0 to 1), 2 outputs (2D coordinates of position of points on the curve))

Feed-forward: $a^{(n)} = \sigma(z^{(n)})$, $z^{(n)} = W^{(n)} a^{(n-1)} + b^{(n)}$, $\sigma(z) = \dfrac{1}{1+\exp(-z)}$ (see notebook for code)

Backprop: $J_{W^{(3)}} = \dfrac{\partial C}{\partial W^{(3)}}$, $J_{b^{(3)}} = \dfrac{\partial C}{\partial b^{(3)}}$, $C = \frac{1}{N} \sum_k C_k$

$$\dfrac{\partial C}{\partial W^{(3)}} = \dfrac{\partial C}{\partial a^{(3)}} \dfrac{\partial a^{(3)}}{\partial z^{(3)}} \dfrac{\partial z^{(3)}}{\partial W^{(3)}}, \quad \dfrac{\partial C}{\partial b^{(3)}} = \dfrac{\partial C}{\partial a^{(3)}} \dfrac{\partial a^{(3)}}{\partial z^{(3)}} \dfrac{\partial z^{(3)}}{\partial b^{(3)}}$$

$$\dfrac{\partial C}{\partial a^{(3)}} = 2(a^{(3)} - y), \quad \dfrac{\partial a^{(3)}}{\partial z^{(3)}} = \sigma'(z^{(3)}), \quad \dfrac{\partial z^{(3)}}{\partial W^{(3)}} = a^{(2)}, \quad \dfrac{\partial z^{(3)}}{\partial b^{(3)}} = 1$$

Build Jacobian for $W^{(3)}$ and $b^{(3)}$ using steps

layer 3

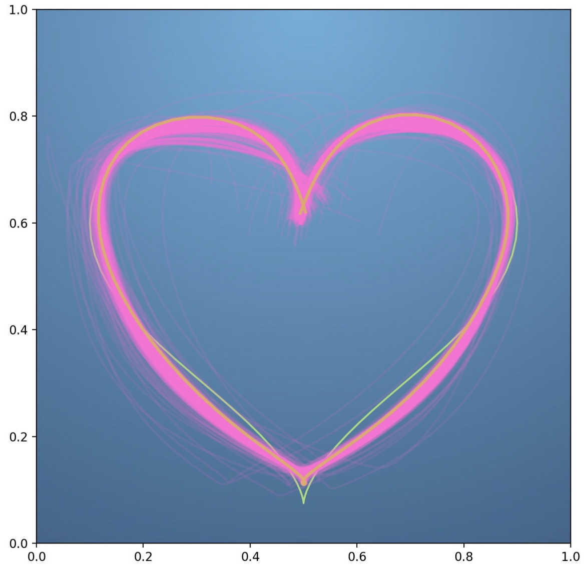
stays same

layer 2: same as before, terms in parentheses are 1 layer lower, $\dfrac{\partial a^{(3)}}{\partial a^{(2)}}$ in both partial derivative equations after $\dfrac{\partial C}{\partial a^{(3)}}$

$$\rightarrow \dfrac{\partial a^{(3)}}{\partial z^{(3)}} \dfrac{\partial z^{(3)}}{\partial a^{(2)}} = \sigma'(z^{(3)}) W^{(3)}$$

$$\text{layer 1: } \frac{\partial C}{\partial W^{(1)}} = \frac{\partial C}{\partial a^{(3)}} \left(\frac{\partial a^{(3)}}{\partial a^{(2)}} \frac{\partial a^{(2)}}{\partial a^{(1)}} \right) \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial W^{(1)}}$$

$$\frac{\partial C}{\partial b^{(1)}} = \frac{\partial C}{\partial a^{(3)}} \left(\frac{\partial a^{(3)}}{\partial a^{(2)}} \frac{\partial a^{(2)}}{\partial a^{(1)}} \right) \frac{\partial a^{(1)}}{\partial z^{(1)}} \frac{\partial z^{(1)}}{\partial b^{(1)}}$$



pink : each iteration

green : training data

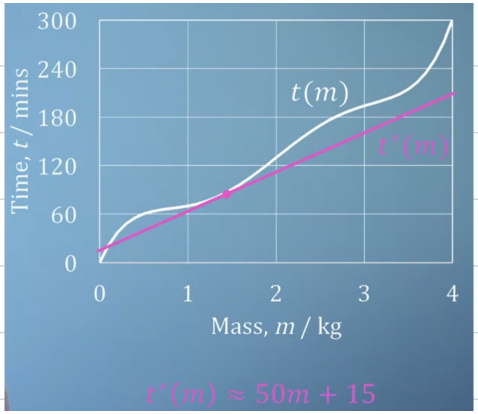
orange : final output

(iterates through steepest descent algorithm using Jacobians)

Building Approximate Functions

$$t(m, T, \text{OvenFactor}, \text{ChickenShapeFactor}) = 7.33m^5 - 72.3m^4 + 253m^3 - 368m^2 + 250m + 0.02 + \text{OvenFactor} + \text{ChickenShapeFactor}$$

$$t(m) = 7.33m^5 - 72.3m^4 + 253m^3 - 368m^2 + 250m + 0.02$$



Power Series

$g(x) = a + bx + cx^2 + dx^3 + \dots$ (generalised power series)

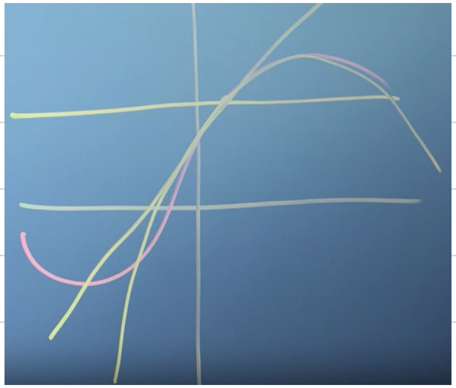
$g_0(x) = a$ — 0'th order approximation

$g_1(x) = a + bx$ — 1^{st} order approximation

$g_2(x) = a + bx + cx^2$ (parabola) …

$g_3(x) = a + bx + cx^2 + dx^3$ …

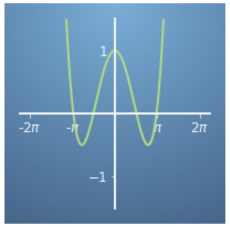
} truncated Taylor Series approximations



Matching Functions and Approximations Quiz

Taylor series approximation: power series, approximations used to build simpler functions easier to evaluate when using numerical methods

1. Fourth order approximation:



$$f_4(x) = 1 - \frac{x^2}{2} + \frac{x^4}{24}$$

Maclaurin Series

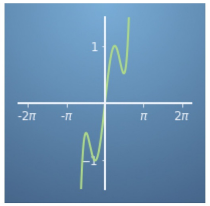
$f(x) = \cos(x)$

(Symmetric about $x = 0$)

$x = 0$, $f(0) = 1$

(which is in 0^{th} order approximation)

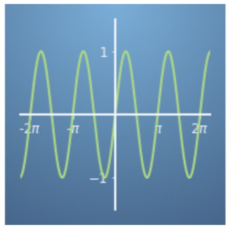
2. Fifth Order:



$$f_5(x) = 2x - \frac{4x^3}{3} + \frac{4x^5}{15}$$

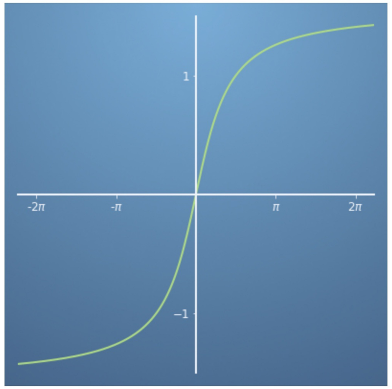
$f(x) = \sin(2x)$

$f(x) = \sin(2x)$

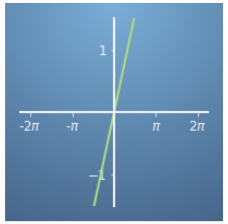


(has rotational symmetry about the origin, and shorter period)

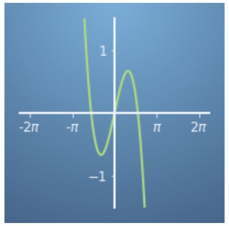
3. The graph below shows the function $f(x) = \tan^{-1}(x)$, select all the power series approximations that can be used to obtain an approximation for this function.



$f(x) = x\ldots$

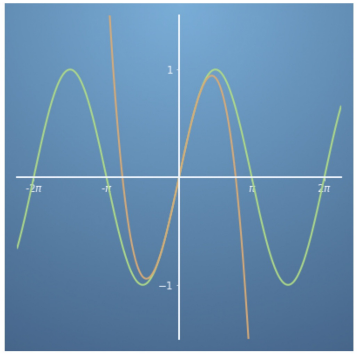


$f(x) = x - \frac{x^3}{3}\ldots$



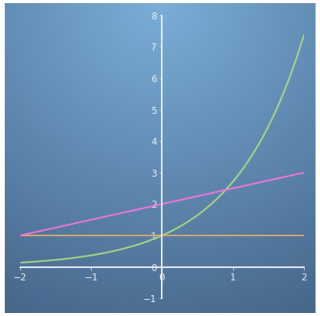
(both sit well between $-0.5 < x < 0.5$ and both go through origin)

4. The sinusoidal function $f(x) = \sin(x)$ (green line) centered at $x = 0$ is shown in the graph below. The approximation for this function is shown through the series $f(x) = x - \frac{x^3}{6}\ldots$ (orange line). Determine what polynomial order is represented by the orange line.



Third order (highest power is 3)

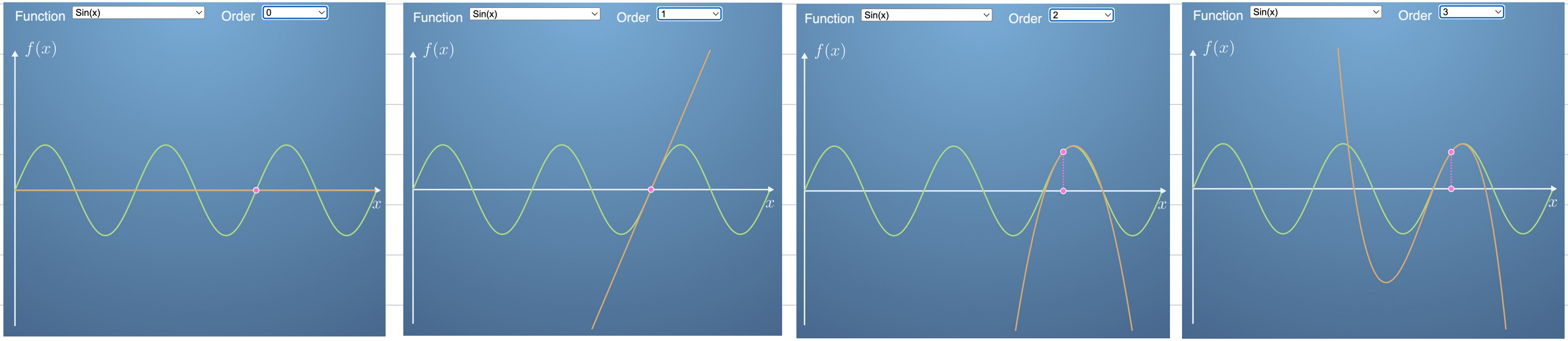
5. The graph below shows the function $f(x) = e^x$ (green line), the exponential function so widely used in science and mathematics today. The orange line represents the zeroth order approximation for the exponential function, centred at $x = 0$. Determine if the pink line shown on the graph is, in fact, an approximation and if so, what order is this approximation.



Not correct approximation: not a tangent to this point,

$\therefore$ poor approximation

Visualising Taylor Series



Power Series Derivation

Power series representation of functions : can use a series of increasing powers of x to re-express functions

$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \dots$ (re-express e^x as power series)

Taylor series method : if you know everything about function at any point (value, all derivatives etc), can use info to reconstruct function everywhere else

↳ only true for well behaved functions (continuous and can differentiate as many times as you want)

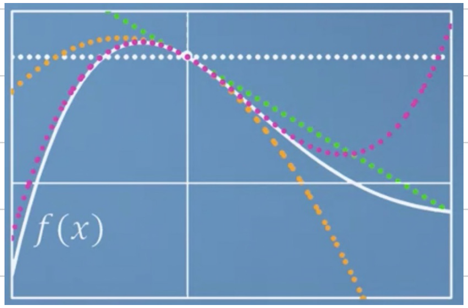
(evaluate at $x=0$)

$g_0(x) = f(0)$ $\quad$ x^2, $\therefore$ parabola $\quad$ (0^{th} order approximation)

$g_1(x) = f(0) + f'(0)x$ $\quad$ (1^{st} order approximation)

$g_2(x) = f(0) + f'(0)x + \frac{1}{2}f''(0)x^2$ $\quad$ (2^{nd} order approximation)

$g_3(x) = f(0) + f'(0)x + \frac{1}{2}f''(0)x^2 + \frac{1}{6}f'''(0)x^3$ $\quad$ (3^{rd} order approximation)



$$\left.\begin{aligned} y &= ax^2+bx+c = f(0) \Rightarrow c = f(0) \\ y' &= 2ax+b = f'(0) \Rightarrow b = f'(0) \\ y'' &= 2a = f''(0) \Rightarrow a = \frac{f''(0)}{2} \end{aligned}\right\} \text{2nd order approximation}$$

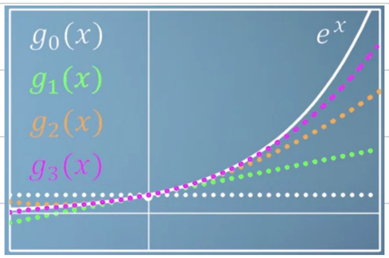
$$\left.\begin{aligned} y &= ax^3+bx^2+cx+d \\ y' &= 3ax^2+2bx+c = f'(0) \Rightarrow c = f'(0) \\ y'' &= 6ax+2b = f''(0) \Rightarrow b = f''(0)/2 \\ y''' &= 6a = f'''(0) \Rightarrow a = f'''(0)/6 \end{aligned}\right\} \text{3rd order approximation}$$

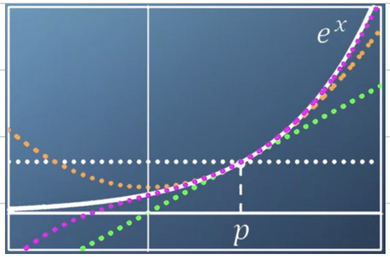
$$g_4(x) = \frac{f^{(0)}(0)}{0!} + \frac{f^{(1)}(0)}{1!}x + \frac{f^{(2)}(0)}{2!}x^2 + \frac{f^{(3)}(0)}{3!}x^3 + \frac{f^{(4)}(0)}{4!}x^4$$

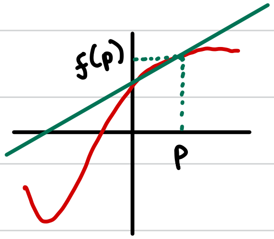
$$g(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)x^n}{n!}$$ (Maclaurin series), because we evaluate at $x=0$ (does count as Taylor series)

Taylor series method: nothing special about $x=0$

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$ (stays same when differentiating)







$y = mx + c$

$y = f'(p)x + c$

$f(p) = f'(p)p + c\,,\quad c = f(p) - f'(p)p$

$y = f'(p)x + f(p) - f'(p)p$

$y = f'(p)(x-p) + f(p)$ (building approximation around point p, apply $f'(p)$ to $x-p$ instead of just x, i.e. how far are you away from p)

$g_0(x) = f(p)$

$g_1(x) = f(p) + f'(p)(x-p)$

$g_2(x) = f(p) + f'(p)(x-p) + \frac{1}{2} f''(p)(x-p)^2$

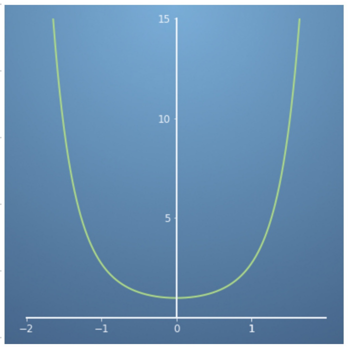
Taylor Series : $g(x) = \displaystyle\sum_{n=0}^{\infty} \frac{f^{(n)}(p)}{n!}(x-p)^n$ (allows us to re-express functions into polynomial series)

$f''''(x) = 8x^2e^{x^2} + 4e^{x^2} + 16x^2e^{x^2} + 8e^{x^2}$
$+ 16x^4e^{x^2} + 24x^2e^{x^2}$

Applying Taylor Series Quiz

$f(x) = e^{x^2},\quad f'(x) = 2xe^{x^2},\quad f''(x) = 2e^{x^2} + 4x^2e^{x^2},\quad f'''(x) = 4xe^{x^2} + 8xe^{x^2} + 8x^3e^{x^2}$

1.



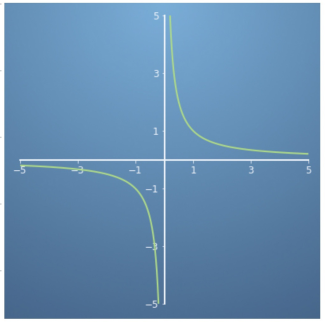
For the function $f(x) = e^{x^2}$ about $x = 0$, using the the Maclaurin series formula, obtain an approximation up to the first three non zero terms.

$$g(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)x^n}{n!} = \frac{f^{(0)}(0)}{0!} + \frac{f^{(1)}(0)}{1!}x + \frac{f^{(2)}(0)}{2!}x^2 + \frac{f^{(3)}(0)}{3!}x^3 + \frac{f^{(4)}(0)}{4!}x^4$$

$$= 1 + 0 + \frac{2x^2}{2} + 0 + \frac{12}{24}x^4$$

$$= 1 + x^2 + \frac{1}{2}x^4$$

2.

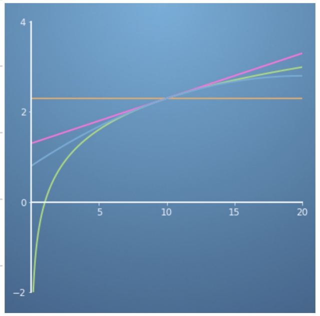


Use the Taylor series formula to approximate the first three terms of the function $f(x) = 1/x$, expanded around the point $p = 4$.

$$g(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(p)}{n!}(x-p)^n = \frac{1}{4} + \left(-\frac{1}{16}(x-4)\right) + \frac{\frac{2}{64}}{2}(x-4)^2 = \frac{1}{4} - \frac{(x-4)}{16} + \frac{(x-4)^2}{64}$$

$$f(x) = \frac{1}{x} = x^{-1}, \quad f'(x) = -\frac{1}{x^2} = -x^{-2}, \quad f''(x) = 2x^{-3}$$

3.



By finding the first three terms of the Taylor series shown above for the function $f(x) = \ln(x)$ (green line) about $x = 10$, determine the magnitude of the difference of using the second order taylor expansion against the first order Taylor expansion when approximating to find the value of $f(2)$.

$$g(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(p)}{n!}(x-p)^n = \overbrace{\underbrace{\ln(10) + \frac{(x-10)}{10}}_{}\!{}^{1^{st}\text{ order}} - \frac{(x-10)^2}{200}}^{2^{nd}\text{ order}} \quad \therefore$$

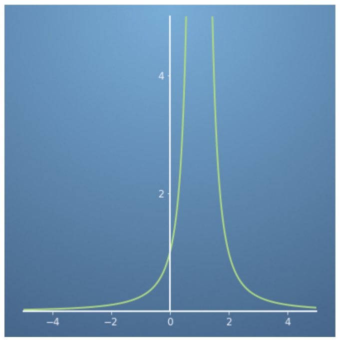
2^{nd} 1^{st}

$$\Delta f(2) = |2.7825\ldots - 3.1025\ldots| = 0.32$$

$$f(x) = \ln(x), \quad f'(x) = \frac{1}{x} = x^{-1}, \quad f''(x) = -x^{-2}$$

4.

In some cases, a Taylor series can be expressed in a general equation that allows us to find a particular n^{th} term of our series. For example the function $f(x) = e^x$ has the general equation $f(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!}$. Therefore if we want to find the 3^{rd} term in our Taylor series, subsitituting $n = 2$ into the general equation gives us the term $\frac{x^2}{2}$. We know the Taylor series of the function e^x is $f(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \ldots$. Now let us try a further working example of using general equations with Taylor series.



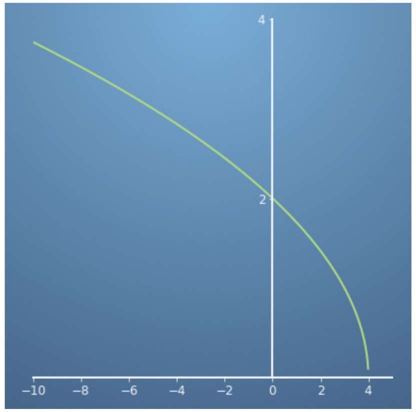
By evaluating the function $f(x) = \frac{1}{(1-x)^2}$ about the origin $x = 0$, determine which general equation for the n^{th} order term correctly represents $f(x)$.

$$f(x) = \frac{1}{(1-x)^2} = (1-x)^{-2}, \quad f'(x) = 2(1-x)^{-3}, \quad f''(x) = 6(1-x)^{-4}$$

$$g(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)x^n}{n!} = 1 + 2x + 3x^2 + 4x^3 + \ldots$$

$$f(x) = \sum_{n=0}^{\infty} (1+n)x^n$$

5.



By evaluating the function $f(x) = \sqrt{4-x}$ at $x = 0$, find the quadratic equation that approximates this function.

$$g(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)x^n}{n!} = 2 - \frac{x}{4} - \frac{x^2}{64} \ldots$$

$$f(x) = (4-x)^{1/2}, \quad f'(x) = \frac{1}{2}(4-x)^{-1/2}, \quad f''(x) = -\frac{\frac{1}{4}}{2}(4-x)^{-3/2}$$

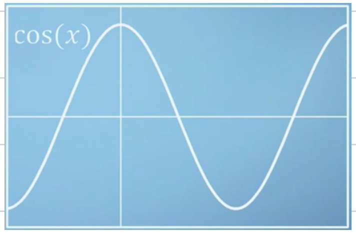
Examples

Maclaurin Series expansion of $f(x) = \cos(x)$

↳ well behaved

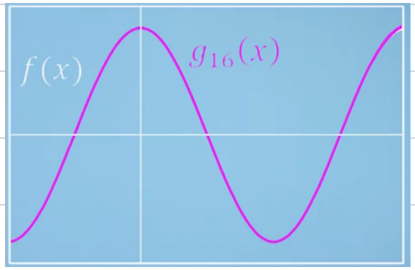
(continuous, infinitely differentiable)

$$g(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)x^n}{n!}$$



$f(x) = \cos(x)$, $f'(x) = -\sin(x)$, $f''(x) = -\cos(x)$, $f^3(x) = \sin(x)$, $f^4(x) = \cos(x)$

@ $x=0$: $=1$ $=0$ $=-1$ $=0$ $=1$



(every other term has a 0 coefficient, every odd powers of x will be absent)

↳ even powers of x: even functions (symmetrical around vertical axis)

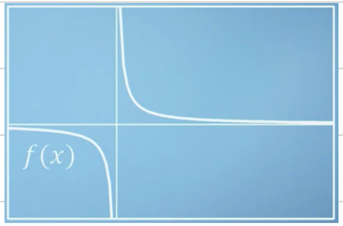
$$\cos(x) = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \ldots$$

↓

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

(careful with series approximations: need to know domain which is acceptable)

$f(x) = \frac{1}{x}$



discontinuity at $x=0$ (not well behaved : $\frac{1}{0} = NaN$)

Use $x=1$ instead, $\therefore$ need Taylor Series $g(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(p)}{n!}(x-p)^n$

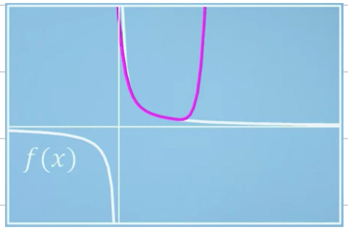
$f(1) = 1,\ f'(1) = {-1}/{1^2} = -1,\ f''(1) = {2}/{1^3} = 2,\ f^3(1) = {-6}/{1^4} = -6,\ f^4(1) = {24}/{1^5} = 24$

$g(x) = \frac{1}{x} = 1 - (x-1) + (x-1)^2 - (x-1)^3 + \ldots$

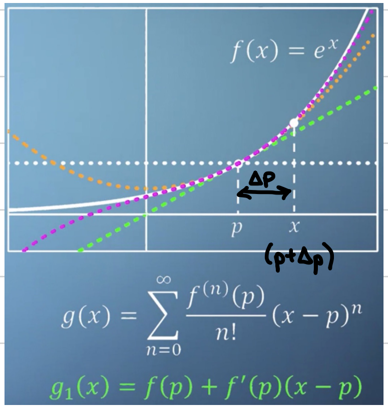
$\hookrightarrow = \sum_{n=0}^{\infty} (-1)^n (x-1)^n$ $\Rightarrow$ plot function over points, start @ $x=1$, progressively get more complex.

$\hookrightarrow$ below 0 not described by approximations

$\hookrightarrow$ addition of new term causes approximation to flip



Linearisation



Starting from height $f(p)$, as you move away from p corresponding change in height = distance away from p x grad. of function at p $\left(\underbrace{(x-p)}_{\text{Run}} \times \underbrace{f'(p)}_{\text{Grad}} = \text{Rise}\right)$

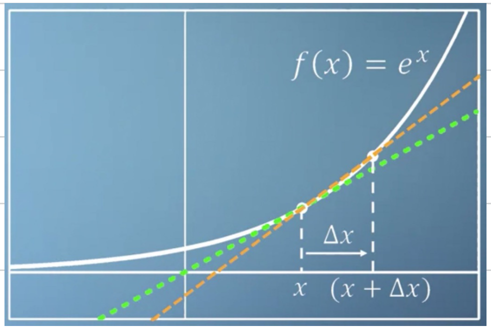
Going to use approximation to evaluate function near p as you already know about it at p

$g_1(p+\Delta p) = f(p) + f'(p)(\Delta p)$ (then swap p for x)

ignoring terms above Δx = linearisation

$f(x+\Delta x) = f(x) + f'(x)(\Delta x) + O(\Delta x^2)$ (error term : on the order of Δx^2, second order accurate)

$f(x+\Delta x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x)}{n!}\Delta x^n$ (increasing Δx power terms result in increasingly smaller values if Δx is small)



$$f(x+\Delta x) = f(x) + f'(x)\Delta x + \frac{f''(x)}{2}\Delta x^2 + \frac{f^3(x)}{6}\Delta x^3 + \ldots$$

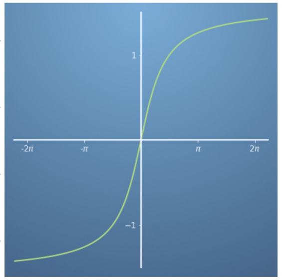
$$f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} - f''(x)\Delta x - f^3(x)\Delta x^2 - \ldots$$ (infinite, but still exact expression for grad. at x)

$$= \frac{f(x+\Delta x) - f(x)}{\Delta x} + O(\Delta x)$$ ← error in approximation

↳ using rise/run between 2 points with finite separation, gives us approximation to gradient that contains an error proportional to Δx (forward difference method is first order accurate)

Taylor Series Special Cases Quiz

1. The graph below shows the function $f(x) = \tan^{-1}(x)$

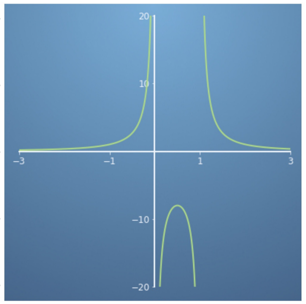


By using the Maclaurin series or otherwise, determine whether the function shown above is even, odd or neither.

Odd function: $-f(x) = f(-x)$

(if it has rotational symmetry w.r.t. origin)

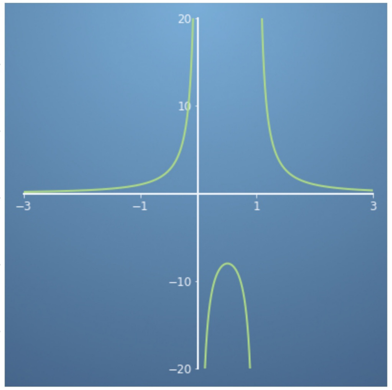
2. The graph below shows the discontinuous function $f(x) = \frac{2}{(x^2-x)}$. For this function, select the starting points that will allow a Taylor approximation to be made.



$x = 2$, $x = 0.5$, $x = -3$

3.

For the same function as previously discussed, $f(x) = \frac{2}{(x^2-x)}$, select all of the statements that are true about the resulting Taylor approximation.



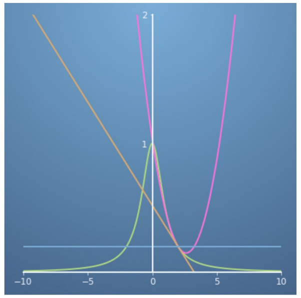
Approximation ignores segments of the function (discontinuities, starting point dictates domain)

Approximation converges quickly (no! sign flip in approximation causes slow convergence)

Approximation ignores asymptotes (Taylor series struggle)

4.

The graph below highlights the function $f(x) = \frac{1}{(1+x^2)}$ (green line), with the Taylor expansions for the first 3 terms also shown about the point $x = 2$. The Taylor expansion is $f(x) = \frac{1}{5} - \frac{4(x-2)}{25} + \frac{11(x-2)^2}{125} + \dots$. Although the function looks rather normal, we find that the Taylor series does a bad approximation further from its starting point, not capturing the turning point. What could be the reason why this approximation is poor for the function described.

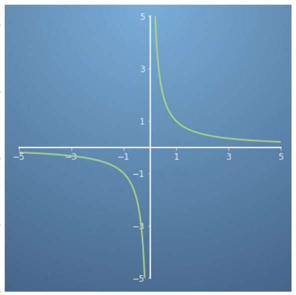


Discontinuous function in the complex plane (well behaved in real plane, but in imaginary plane asymptotes limit its convergence and behaviour of Taylor expansion)

Asymptotes in the complex plane

5.

For the function $f(x) = \frac{1}{x}$, provide the linear approximation about the point $x = 4$, ensuring it is second order accurate.



$$g(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(p)}{n!}(x-p)^n$$

$$f(x) = \frac{1}{x} = x^{-1}, \quad f'(x) = -x^{-2}, \quad f''(x) = 2x^{-3}$$

Second order accurate

$$g(4) = \frac{1}{4} - \frac{(x-4)}{16} + O(\Delta x^2)$$

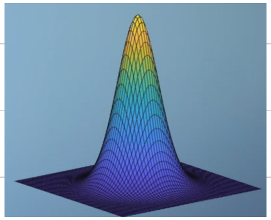
Multivariate Taylor

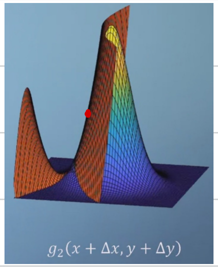
$$f(x+\Delta x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x)}{n!}\Delta x^n \quad \text{(1D case)}$$

$$f(x+\Delta x, y+\Delta y) = ??$$

(truncated Taylor series expressions enable us to approximate function at nearby point $x+\Delta x,\ y+\Delta y$)

2D Gaussian function : $f(x,y) = e^{-(x^2+y^2)}$

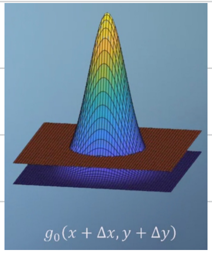




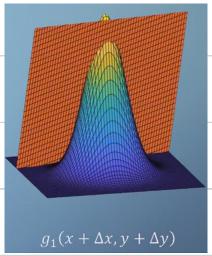
at peak : would have to look inside

parabolic surface

2^{nd} order approximation : Saddle function (but not useful outside of a small region around the point)



0^{th} order approximation : flat surface



1^{st} order approximation : surface with same height and gradient

$$\text{2D Taylor Series expansion of } f(\overset{\nearrow \text{ at point } x,y}{x+\Delta x}, y+\Delta y) = f(x,y)$$

$$+ \left(\partial_x f(x,y)\Delta x + \partial_y f(x,y)\Delta y\right) \Rightarrow \begin{bmatrix} \partial_x f(x,y), & \partial_y f(x,y) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \Rightarrow \overset{\text{Jacobian}}{\overset{\downarrow}{J_f}} \Delta x$$

$$+ \frac{1}{2}\left(\partial_{xx} f(x,y)\Delta x^2 + 2\partial_{xy} f(x,y)\Delta x \Delta y + \partial_{yy} f(x,y)\Delta y^2\right)$$

$$\Rightarrow \frac{1}{2}\begin{bmatrix} \Delta x, & \Delta y \end{bmatrix} \begin{bmatrix} \partial_{xx} f(x,y) & \partial_{xy} f(x,y) \\ \partial_{yx} f(x,y) & \partial_{yy} f(x,y) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \Rightarrow \frac{1}{2}\Delta x^t H_f \Delta x$$

$$+ \dots$$

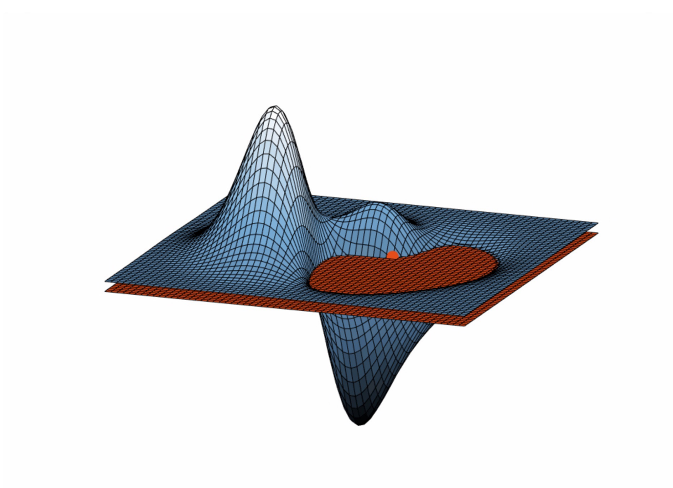
$$f(x+\Delta x) = f(x) + J_f \Delta x + \frac{1}{2}\Delta x^t H_f \Delta x + \dots$$

(2^{nd} order multivariate Taylor Series expansion)

↳ generalises from 2D to multi-dimensional hypersurfaces

2D Taylor Series Quiz

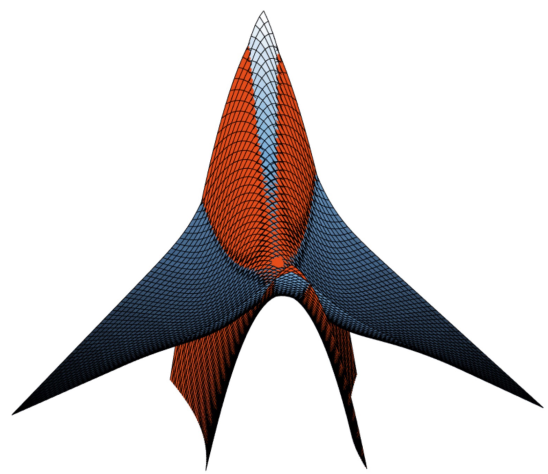
1. The following plot features a surface and its Taylor series approximation in red, around a point given by a red circle. What order is the Taylor series approximation?



0^{th} order

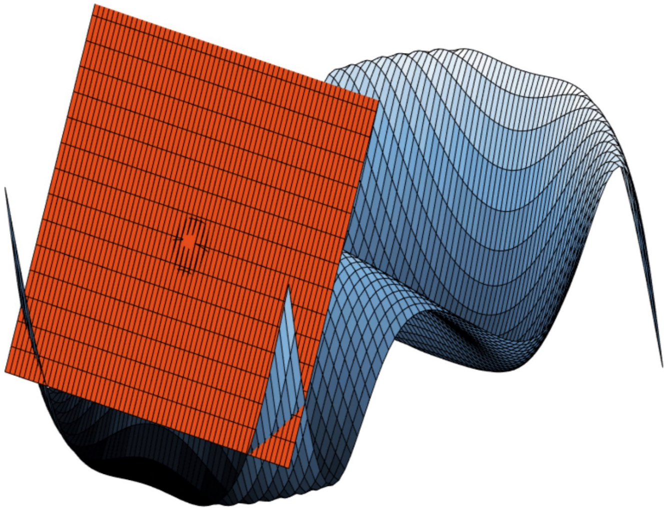
2. What order Taylor series approximation, expanded around the red circle, is the red surface in the following plot?

2^{nd} order



3. Which red surface in the following images is a first order Taylor series approximation of the blue surface? The original functions are given, but you don't need to do any calculations.

◉ $f(x,y) = x\sin(x^2/2 + y^2/4)$



4. Recall that up to second order the multivariate Taylor series is given by $f(\mathbf{x} + \boldsymbol{\Delta}\mathbf{x}) = f(\mathbf{x}) + J_f \boldsymbol{\Delta}\mathbf{x} + \frac{1}{2}\boldsymbol{\Delta}\mathbf{x}^T H_f \boldsymbol{\Delta}\mathbf{x} + \dots$

Consider the function of 2 variables, $f(x,y) = xy^2 e^{-x^4 - y^2/2}$. Which of the following is the first order Taylor series expansion of f around the point $(-1, 2)$?

$$f_1(-1+\Delta x, 2+\Delta y) = -4e^{-3} - 12e^{-3}\Delta x + 4e^{-3}\Delta y$$

$$\frac{d}{dx}\left[y^2 x e^{-x^4 - \frac{y^2}{2}}\right] = y^2 \cdot \frac{d}{dx}\left[x \overbrace{e^{-x^4 - \frac{y^2}{2}}}^{e^{u(x)} \cdot u'(x)}\right]$$

$$= y^2 \cdot \left(\frac{d}{dx}[x] e^{-x^4 - \frac{y^2}{2}} + x \frac{d}{dx}\left[e^{-x^4 - \frac{y^2}{2}}\right]\right)$$

$$= y^2 \cdot \left(e^{-x^4 - \frac{y^2}{2}} + x e^{-x^4 - \frac{y^2}{2}} \cdot \frac{d}{dx}\left[-x^4 - \frac{y^2}{2}\right]\right)$$

$$= y^2 \cdot \left(x e^{-x^4 - \frac{y^2}{2}}\left(\frac{d}{dx}\left[-\frac{y^2}{2}\right] - \frac{d}{dx}[x^4]\right) + e^{-x^4 - \frac{y^2}{2}}\right)$$

$$= y^2\left(x e^{-x^4 - \frac{y^2}{2}}(0 - 4x^3) + e^{-x^4 - \frac{y^2}{2}}\right)$$

$$= y^2\left(e^{-x^4 - \frac{y^2}{2}} - 4x^4 e^{-x^4 - \frac{y^2}{2}}\right)$$

$$= -y^2(4x^4 - 1)e^{-x^4 - \frac{y^2}{2}}$$

5. Now consider the function $f(x,y) = \sin(\pi x - x^2 y)$. What is the Hessian matrix H_f that is associated with the second order term in the Taylor expansion of f around $(1, \pi)$?

$$\frac{\partial}{\partial x}[\sin(\pi x - x^2 y)] = (\pi - 2yx)\cos(yx^2 - \pi x),$$

$$\frac{\partial}{\partial xx}[(\pi - 2yx)\cos(yx^2 - \pi x)] = -(\pi - 2yx)(2yx - \pi)\sin(x(yx - \pi)) - 2y\cos(x(yx - \pi))$$

$$\frac{\partial}{\partial xy}[(\pi - 2yx)\cos(yx^2 - \pi x)] = -x^2(\pi - 2xy)\sin(x^2 y - \pi x) - 2x\cos(x^2 y - \pi x)$$

$$\frac{\partial}{\partial y}[\sin(\pi x - x^2 y)] = -x^2\cos(x^2 y - \pi x),$$

$$\frac{\partial}{\partial yy}[-x^2\cos(x^2 y - \pi x)] = x^4\sin(x^2 y - \pi x)$$

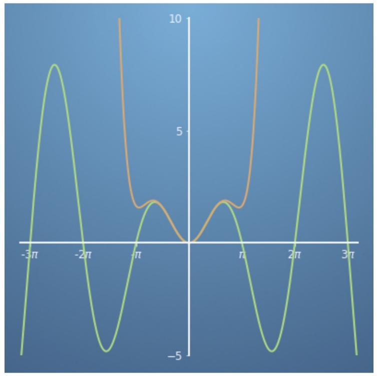
$$\frac{\partial}{\partial yx}[-x^2\cos(x^2 y - \pi x)] = x^2(2yx - \pi)\sin(yx^2 - \pi x) - 2x\cos(yx^2 - \pi x)$$

@ $(1, \pi)$

$$H_{f(x,y)} = \begin{bmatrix} f_{xx}(x,y) & f_{yx}(x,y) \\ f_{xy}(x,y) & f_{yy}(x,y) \end{bmatrix} = \begin{bmatrix} -2\pi & -2 \\ -2 & 0 \end{bmatrix}$$

Taylor Series Assessment

1. For the function $f(x) = x\sin(x)$ shown below, determine what order approximation is shown by the orange curve, where the Taylor series approximation was centered about $x = 0$.



Symmetry tells us there should be no odd power terms in approximation

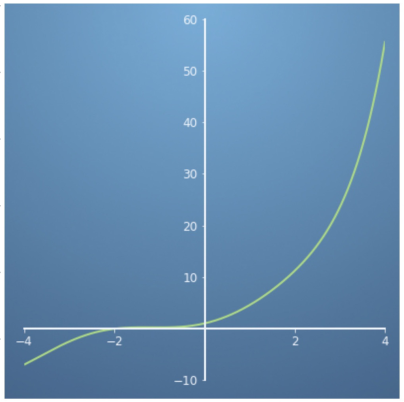
Sign for x^4 is -ve

$\sin(x)$: we expect sign in approximation to constantly change from +ve to -ve

Approximation always +ve: indicating higher order term dominating approximation

$\therefore$ 6th order: 6th term is +ve, dominates over 4th order term, approximation of $f(x)$ always +ve

2. Find the first four non zero terms of the Taylor expansion for the function $f(x) = e^x + x + \sin(x)$ about $x = 0$. The function is shown below:

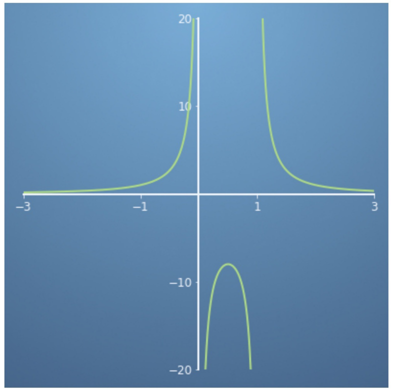


$$g(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)x^n}{n!}, \quad f'(x) = e^x + 1 + \cos(x), \; f''(x) = e^x - \sin(x),$$

$$f'''(x) = e^x - \cos(x), \; f''''(x) = e^x + \sin(x)$$

$$g(0) = 1 + 3x + \frac{x^2}{2} + \frac{x^4}{24} + \ldots$$

3. The graph below shows the discontinuous function $f(x) = \frac{2}{(x^2-x)}$. Approximate the section of this function that covers the domain $0 < x < 1$. Use the Taylor series forumla and $x = 0.5$ as your starting point, find the first two non zero terms.



$$g(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(p)}{n!}(x-p)^n$$

$$f(x) = 2(x^2-x)^{-1}$$

$$f'(x) = -2(x^2-x)^{-2} \cdot 2x-1 = \frac{-2(2x-1)}{(x^2-x)^2} = \frac{(-4x+2)}{(x^2-x)^2} = \overset{u}{(-4x+2)}\overset{v}{(x^2-x)^{-2}}$$

$$f''(x) = (-4x+2) \cdot -2(x^2-x)^{-3} \cdot (2x-1) + (x^2-x)^{-2} \cdot -4$$

$$= \frac{(-4x+2)^2}{(x^2-x)^3} - \frac{4}{(x^2-x)^2}$$

$$g(x) = -8 - \frac{64}{2}(x-0.5)^2 + \ldots$$

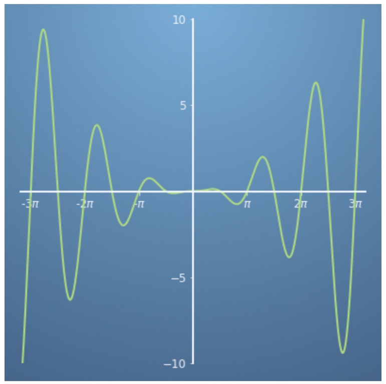
$$= -8 - 32(x-0.5)^2$$

(behaves well within defined domain, but poor approximation for entire function)

4. Determine if the function:

$$f(x) = \left(\frac{x}{2}\right)^2 \frac{\sin(2x)}{2}$$

shown below is odd, even or neither.



Odd function: $-f(x) = f(-x)$

(if it has 180° rotational symmetry w.r.t. origin)

Even function: $f(x) = f(-x)$

(Symmetric about y-axis)

$\therefore$ odd function because of rotational symmetry

5. Take the Taylor expansion of the function

$$f(x) = e^{-2x}$$

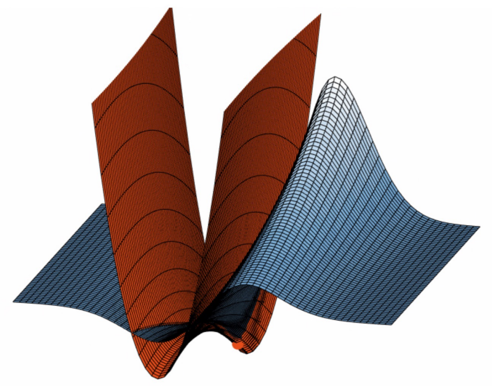
about the point $x = 2$ and subsequently linearise the function.

$$f(x+\Delta x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x)}{n!} \Delta x^n = \frac{1}{e^4} + \left(-\frac{2}{e^4}(x-2)\right) + \ldots = \frac{1}{e^4}\left[1-2(x-2)\right] + O(\Delta x^2)$$

$$f'(x) = -2e^{-2x} \quad , \quad f''(x) = 4e^{-2x}$$

6. The figures below feature functions of two variables with proposed Taylor series approximations in red, expanded around the red circle. Which of the following features a valid second order approximation?

$f(x,y) = xe^{-x^2+y/4}$



has gradient which changes with x and y

function and approximation have same behaviour near point of expansion

Week 5: Intro to optimisation

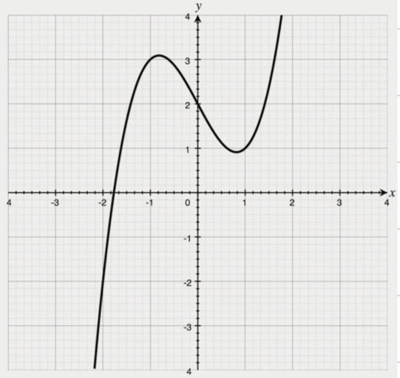
Newton-Raphson Method

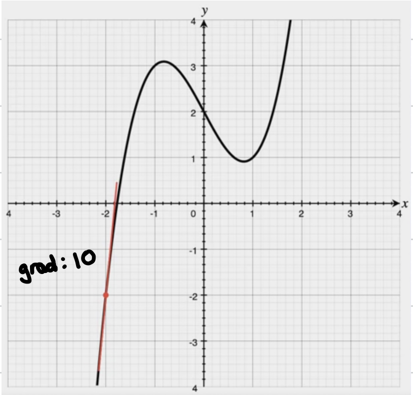
Want to find parameters of model (e.g. μ and σ) to make predictions

$y = x^3 - 2x + 2$

$\frac{\partial y}{\partial x} = 3x^2 - 2$

2 solutions, 2 turning points





extrapolate gradient to intercept with y-axis = first guess to solution of equation, so can use that value of x as new estimate for what solution to equation is (i.e. guessing fn is straight line, 1st order approximation)

Newton-Raphson: $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$

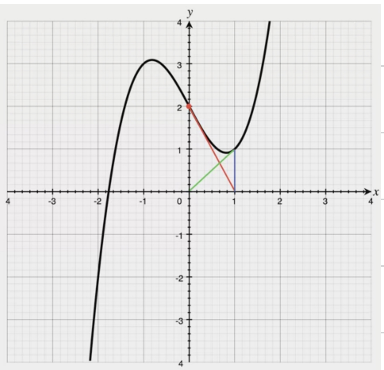
(iteration, root finding algorithm)

i	x_i	$y(x_i)$	$\frac{\partial y(x_i)}{\partial x}$
0	-2	-2	10
1	-1.8	-0.23	7.7
2	-1.77	-0.005	7.4
3	-1.769	-2.3E-6	

↑ gets closer to 0

to solve equation, need to evaluate and differentiate, don't need to graph and visualise everywhere

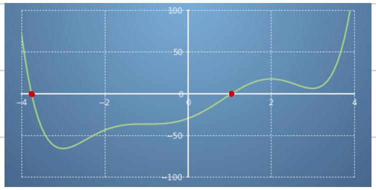
↳ iterate solution to equation by making new estimate for solution and using gradient to extrapolate towards solution



i	x_i	$y(x_i)$	$\frac{\partial y(x_i)}{\partial x}$
0	0	2	-2
1	1	1	1
2	0	2	

(near turning point gradient is small, so dividing by gradient makes it difficult to converge)

Newton-Raphson in 1D Quiz

1. 

2 roots of equation

Short distance away

Linearise about x_0: $f(x_0 + \delta x) = f(x_0) + f'(x_0)\,\delta x$

Assume fn goes through 0 nearby: re-arrange to find how far away (assume $f(x_0+\delta x) = 0$ and solve for δx)

↳ $\delta x = -\dfrac{f(x_0)}{f'(x_0)}$

$f(x)$ not a line, so formula tries to get closer to root but won't exactly hit it (but can repeat to get closer)

↳ $x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$ (Newton-Raphson method to find roots of functions)

Equation: $f(x) = \dfrac{x^6}{6} - 3x^4 - \dfrac{2x^3}{3} + \dfrac{27x^2}{2} + 18x - 30$

$f'(x) = x^5 - 12x^3 - 2x^2 + 27x + 18$

2. Try to find location of root near $x = 1$

Use $x_0 = 1$, $x_1 = 1 - \dfrac{f(1)}{f'(1)} = 1 - \dfrac{-2}{32} = 1.063$ (3.d.p)

3. Let's use code to find the other root, near $x = -4$.

Complete the d_f function in the code block with your answer to Q1, i.e. with $f'(x)$. The code block will then perform iterations of the Newton-Raphson method.

```
def f (x) :
    return x**6/6 - 3*x**4 - 2*x**3/3 + 27*x**2/2 + 18*x - 30

def d_f (x) :
    return x**5 - 12*x**3 - 2*x**2 + 27*x + 18

x = -4.0

d = {"x" : [x], "f(x)": [f(x)]}
for i in range(0, 20):
    x = x - f(x) / d_f(x)
    d["x"].append(x)
    d["f(x)"].append(f(x))

pd.DataFrame(d, columns=['x', 'f(x)'])
```

Run

Reset

```
            x          f(x)
0   -4.000000  7.133333e+01
1   -3.811287  1.223161e+01
2   -3.763093  6.515564e-01
3   -3.760224  2.198858e-03
4   -3.760214  2.531156e-08
5   -3.760214  1.421085e-13
6   -3.760214  4.263256e-14
7   -3.760214  4.263256e-14
8   -3.760214  4.263256e-14
9   -3.760214  4.263256e-14
10  -3.760214  4.263256e-14
11  -3.760214  4.263256e-14
12  -3.760214  4.263256e-14
13  -3.760214  4.263256e-14
14  -3.760214  4.263256e-14
15  -3.760214  4.263256e-14
16  -3.760214  4.263256e-14
17  -3.760214  4.263256e-14
18  -3.760214  4.263256e-14
19  -3.760214  4.263256e-14
20  -3.760214  4.263256e-14
```

What is the x value of the root near $x = -4$? (to 3 decimal places.)

-3.760

4. Step Size: $\delta x = -\dfrac{f(x)}{f'(x)}$, can get big when $f'(x)$ is very small

↳ $f'(x)$ is 0 @ turning points of $f(x)$ (step size is infinite)

↳ $x_0 = 1.99$, method converges to root nearest $x = -4$, takes over 15 iterations to converge

↖ not near starting point!

5. Some starting points on curve don't converge or diverge, but oscillate without settling (e.g. $x_0 = 3.1$)

↳ happens in areas where curve isn't well described by straight line, so linearisation assumption not good for that starting point

In practice, often you will not need to hand craft optimisation methods, as they can be called from libraries, such as scipy. Use the code block below to test $x_0 = 3.1$.

```
from scipy import optimize

def f (x) :
    return x**6/6 - 3*x**4 - 2*x**3/3 + 27*x**2/2 + 18*x - 30

x0 = 3.1
optimize.newton(f, x0)
```

Run

Reset

```
1.063070629709697
```

Did it settle to a root?

Yes, to root nearest $x = 1$

each component evaluated at location (e.g. a,b)

$a.b = ab\cos\theta$

Gradient Descent

∇f θ $\hat{r}$ $\parallel \nabla f$ (parallel)

$\hat{r}$ (unit vector)

$$f(x,y) = x^2 y \,, \quad \frac{\partial f}{\partial x} = 2xy \,, \quad \frac{\partial f}{\partial y} = x^2 \,, \quad \underbrace{\nabla f}_{\text{grad}} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \cdot \begin{bmatrix} c \\ d \end{bmatrix} \,, \quad df = \frac{\partial f}{\partial x} c + \frac{\partial f}{\partial y} d$$

directional gradient

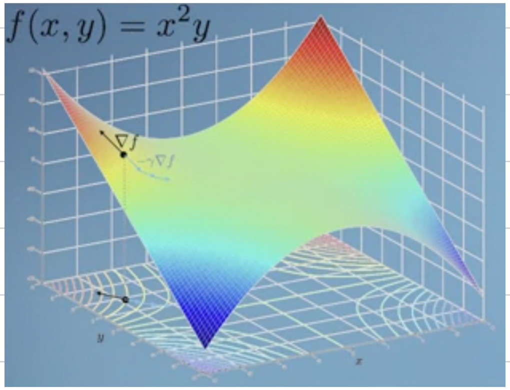
(to find maximum value of directional gradient, want $\hat{r}$ which is normalised version of grad)

$$\nabla f \cdot \frac{\nabla f}{\|\nabla f\|} = \frac{\|\nabla f\|^2}{\|\nabla f\|} = \|\nabla f\|$$

(steepest gradient we can possibly have: sum of squares of components of grad)

Which way does grad point?

↳ points upwards in direction of steepest descent, ⊥er to contour lines



Gradient descent method: $S_{n+1} = S_n - \gamma \nabla f(S_n)$ (iteratively descend downhill towards minimum)

↳ depends on starting point

Gradient Descent in Sandpit Lab

To find minimum: move in direction proportional to Jacobian down the slope $(\delta x = -\gamma J)$ — γ: aggression parameter: how big steps should be

How big to set γ?

↳ use Hessian (second derivative matrix)

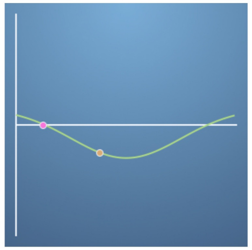
↳ $\delta x = -H^{-1} J$ (but can change direction! can find maxima as well as minima) — but can find stationary point in few steps if sufficiently close

Hybrid method: use Hessian unless step is too big or it would point backwards, then go back to steepest descent

↳ can change direction uphill!

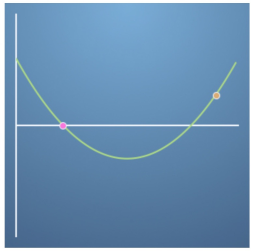
Newton-Raphson Quiz

1. Which functions will converge from orange (start) to pink (specified root)?

- [x] 

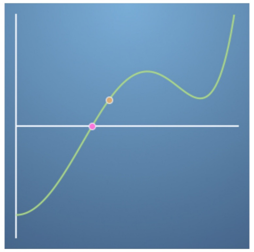
$$$$

Correct
This optimisation will converge to the specified root.

- [x] 

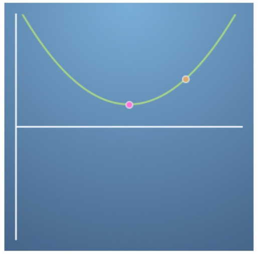
$$$$

This should not be selected
This system will converge, but not to the correct root.

- [x] 

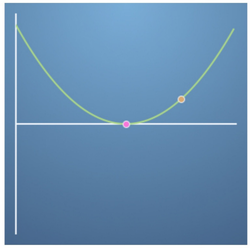
$$$$

Correct
This optimisation will converge to the specified root.

- [x] 

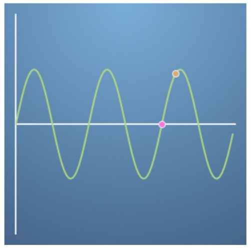
$$$$

This should not be selected
There is no root in this system. A solution does not exist and so cannot be found.

- [x] 

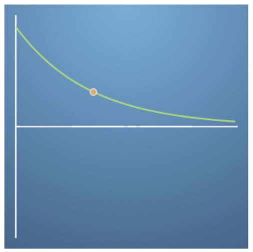
$$$$

Correct
This optimisation will converge to the root, even though it is a minimum.

- [x] 

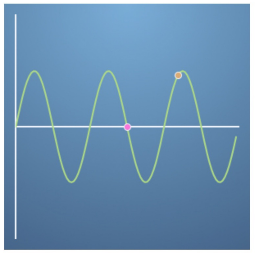
$$$$

This should not be selected
This system will converge, but not to the nearby root. The gradient is shallow at the starting point, so the iteration will shoot into a different basin.

- [x] 

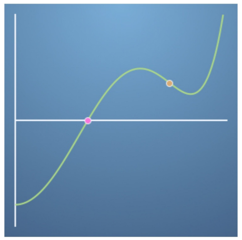
$$$$

This should not be selected
There is no root to this system

- [x] 

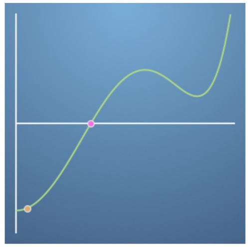
$$$$

Correct
This optimisation will converge to the specified root.

- [x] 

$$$$

This should not be selected
This system may not converge as it is likely to get trapped and oscillate about the minimum that it is sitting in.



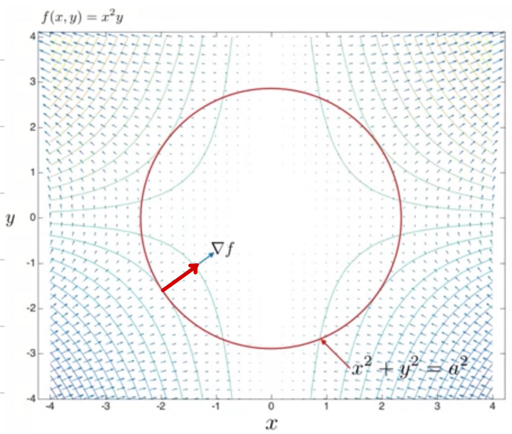
$$$$

This should not be selected

This starting point will throw the starting guess into the basin of the minimum, where it will oscillate in without converging.

Constrained Optimisation

↙ contour map



e.g. want max value within circle

(along the circle, not everywhere in the circle)

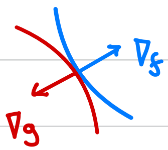
← when line touches contour, we've found minima and maxima of function as we go around the circle

When the contour touches the path, the vector $\perp$er to the contour is in same direction as vector $\perp$er to the path

↳ find grad, can find minimum and maximum points

maximise $f(x,y) = x^2y$

constraint $g(x,y) = x^2+y^2 = a^2$



↙ Lagrange multiplier

Solve $\nabla f = \lambda \nabla g$

$$\nabla f = \nabla(x^2y) = \begin{bmatrix} 2xy \\ x^2 \end{bmatrix} = \lambda \nabla g = \lambda \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

① $2xy = \lambda 2x \Rightarrow y = \lambda$

② $x^2 = \lambda 2y = 2y^2 \Rightarrow x = \pm\sqrt{2}\, y$

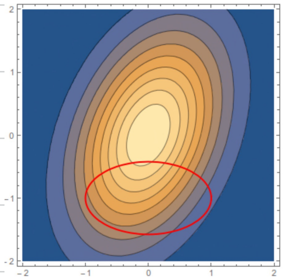
③ $x^2+y^2 = a^2 = 3y^2 \Rightarrow y = \pm a/\sqrt{3}$

Solutions: $\frac{a}{\sqrt{3}}\begin{pmatrix}\sqrt{2}\\1\end{pmatrix}, \frac{a}{\sqrt{3}}\begin{pmatrix}\sqrt{2}\\-1\end{pmatrix}, \frac{a}{\sqrt{3}}\begin{pmatrix}-\sqrt{2}\\1\end{pmatrix}, \frac{a}{\sqrt{3}}\begin{pmatrix}-\sqrt{2}\\-1\end{pmatrix}$

$f(x,y) = \frac{a^3}{3\sqrt{3}}2, \quad \frac{a^3}{3\sqrt{3}}-2, \quad \frac{2a^3}{3\sqrt{3}}, \quad \frac{-2a^3}{3\sqrt{3}}$

max, min, max, min

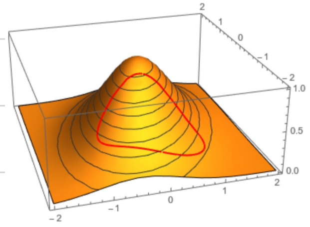
Lagrange Multipliers Quiz



Contour map of functions

1. We can use Lagrange multipliers as a technique to find a minimum of a function subject to a constraint (i.e. solutions lying on a particular curve)

However, their combination can become quite complicated if they were computed directly, as can be inferred from the shape of the constraint on the surface plot,



e.g. $f(x) = \exp\left(-\frac{2x^2+y^2-xy}{2}\right)$ along the curve $g(x) = x^2 + 3(y+1)^2 - 1 = 0$

$f(x)$ has no minima, but there are 2 minima and 2 maxima along the curve

Minima and maxima on the curve will be found where constraint is parallel to the contours of the function

Gradient is $\perp$er to contours, so grad. of func. is parallel to grad. of constraint:

$$\nabla f(x) = \lambda \nabla g(x)$$

In component form: $\begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \lambda \begin{bmatrix} \frac{\partial g}{\partial x} \\ \frac{\partial g}{\partial y} \end{bmatrix}$

Single vector equation: $\nabla L(x,y,\lambda) = \begin{bmatrix} \frac{\partial f}{\partial x} - \lambda \frac{\partial g}{\partial x} \\ \frac{\partial f}{\partial y} - \lambda \frac{\partial g}{\partial y} \\ -g(x) \end{bmatrix} = 0$

(converted q. of finding min. of 2D function constrained to 1D curve to finding zeros of 3D vector equation)

↳ can use root finding methods e.g. Newton-Raphson

$\frac{\partial g}{\partial x} = 2x$, $\frac{\partial g}{\partial y} = 6y+6$

2. Next let's define the vector, $\nabla\mathcal{L}$, that we are to find the zeros of; we'll call this "DL" in the code. Then we can use a pre-written root finding methodin scipy to solve.

```
from scipy import optimize

def DL (xyλ) :
    [x, y, λ] = xyλ
    return np.array([
            dfdx(x, y) - λ * dgdx(x, y),
            dfdy(x, y) - λ * dgdy(x, y),
            - g(x, y)
        ])

(x0, y0, λ0) = (-1, -1, 0)
x, y, λ = optimize.root(DL, [x0, y0, λ0]).x
print("x = %g" % x)
print("y = %g" % y)
print("λ = %g" % λ)
print("f(x, y) = %g" % f(x, y))
```

Run

Reset

```
x = -0.958963
y = -1.1637
λ = -0.246538
f(x, y) = 0.353902
3.62534446907e-11
```

Here, the first two elements of the array are the x and y coordinates that we wanted to find, and the last element is the Lagrange multiplier, which we can throw away now it has been used.

Check that (x, y) does indeed solve the equation $g(x, y) = 0$.

To find other roots of system: use different starting values to find other stationary points on the constraint: y-coordinates = $-1.21, -1.45, -0.43, -1.16$

✓ x which has smallest $f(x,y)$

3. x-coordinate of global minimum of $f(x)$ on $g(x)=0$: 0.93

✓ why last component of ∇L has - sign

4. ∇L: gradient (over x, y and λ) of a scalar function $L(x,y,\lambda)$: $L(x,y,\lambda) = f(x) - \lambda g(x)$

5. Calculate the minimum of

$$f(x, y) = -\exp(x - y^2 + xy)$$

on the constraint,

$$g(x, y) = \cosh(y) + x - 2 = 0$$

```python
# Import libraries
import numpy as np
from scipy import optimize

# First we define the functions, YOU SHOULD IMPLEMENT THESE
def f (x, y) :
    return -np.exp(x**2 - y**2 + x*y)

def g (x, y) :
    return np.cosh(y) + x - 2

# Next their derivatives, YOU SHOULD IMPLEMENT THESE
def dfdx (x, y) :
    return (1+y) * -np.exp(x - y**2 + x*y)

def dfdy (x, y) :
    return (-2*y + x) * -np.exp(x - y**2 + x*y)

def dgdx (x, y) :
    return 1

def dgdy (x, y) :
    return (np.exp(y) - np.exp(-y)) / 2

# Use the definition of DL from previously.
def DL (xyλ) :
    [x, y, λ] = xyλ
    return np.array([
            dfdx(x, y) - λ * dgdx(x, y),
            dfdy(x, y) - λ * dgdy(x, y),
            - g(x, y)
        ])

# To score on this question, the code above should set
# the variables x, y, λ, to the values which solve the
# Langrange multiplier problem.
# I.e. use the optimize.root method, as you did previously.
(x0, y0, λ0) = (55, 55, 0)
x, y, λ = optimize.root(DL, [x0, y0, λ0]).x
```

Run

Reset

```
x = 0.957782
y = 0.289565
λ = -4.07789
f(x, y) = -3.03691
```

Optimisation Scenarios Quiz

QUESTION 1/6

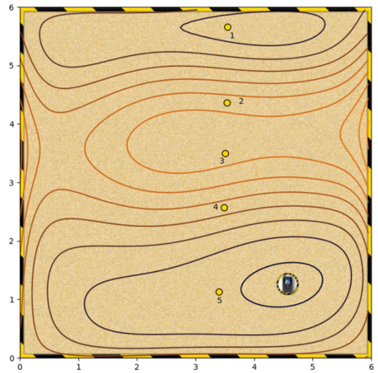
Given the following contour plot,

1, 3, 4

Which starting points (from 1 to 5) are likely to converge to the global minimum (shown by the mobile phone) when using a steepest descent algorithm?

QUESTION 2/6

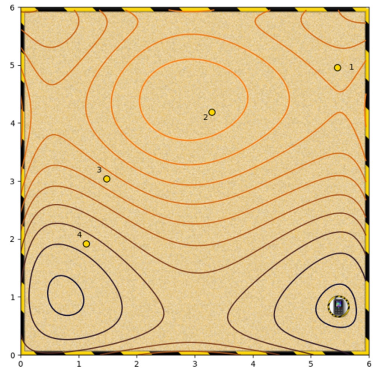
Again, which starting points converge to the global minimum?



3, 4, 5

QUESTION 3/6

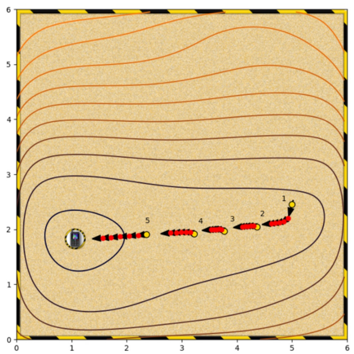
Which starting points converge to the global minimum?



2

QUESTION 4/6

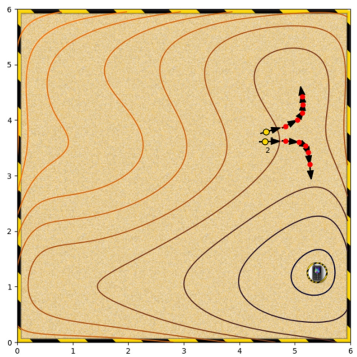
What's happening in this gradient descent?



The global minimum is in a wide and flat basin, so convergence is slow. This could be improved by increasing the aggression.

QUESTION 5/6

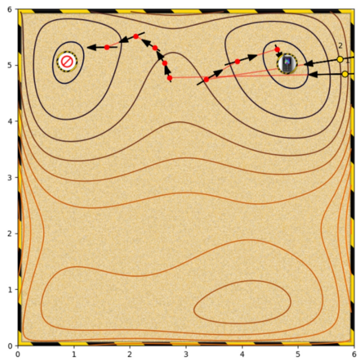
What is happening here?



The algorithm is passing either side of a saddle point.

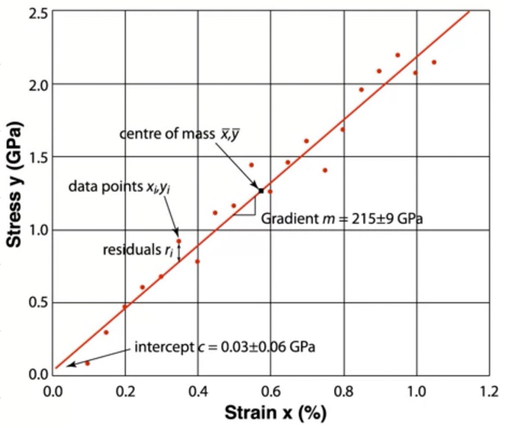
QUESTION 6/6

What is happening here?



The Jacobian at the starting point is very large. This is causing the algorithm to overshoot. In one case into a different basin.

Week 6: Simple Linear Regression



parameters

$$y = y(x; a_i) = mx_i + c$$

$$a = \begin{bmatrix} m \\ c \end{bmatrix}$$

$$r_i = y_i - mx_i - c$$

residual for each point

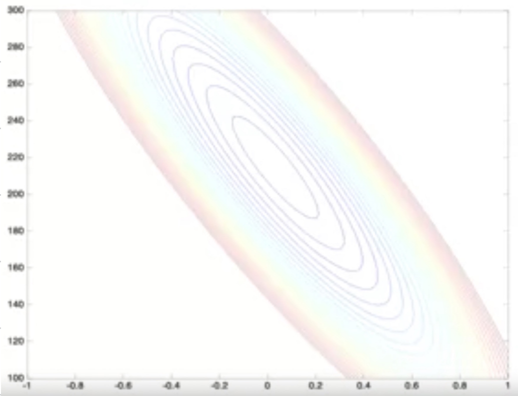
$$\chi^2 = \sum_i r_i^2 = \sum_i (y_i - mx_i - c)^2$$

sum of squares of deviations of fit from the data

chi-squared
(measure of overall quality of the fit: goodness of fit estimator)

↳ sum of squared residuals (squared: penalise data far away from line, and +ve and -ve values don't cancel)

↳ want to minimise χ^2



plot of parameters m and c for χ^2

↳ want to find minimum: $\nabla \chi^2 = \begin{bmatrix} \frac{\partial \chi^2}{\partial m} \\ \frac{\partial \chi^2}{\partial c} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

grad of chi-squared w.r.t. fitting parameters and set to 0

Quality of fit $\chi^2 = \sum_i (y_i - mx_i - c)^2$

$$\nabla \chi^2 = \begin{bmatrix} \frac{\partial \chi^2}{\partial m} \\ \frac{\partial \chi^2}{\partial c} \end{bmatrix} = \begin{bmatrix} -2\sum_i x_i (y_i - mx_i - c) \\ -2\sum_i (y_i - mx_i - c) \end{bmatrix}$$

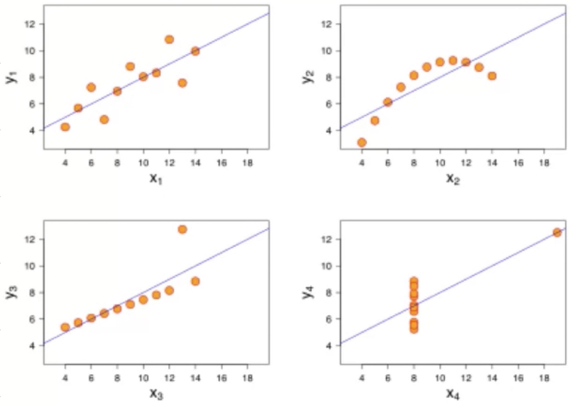
$\frac{d}{dm}(mx_1 + mx_2 + mx_3) = \frac{d}{dm}(m(x_1 + x_2 + x_3)) = x_1 + x_2 + x_3$ (no need to worry about sums) uncertainty in c

$c = \bar{y} - m\bar{x}$

$\sigma_c \simeq \sigma_m \sqrt{\bar{x}^2 + \frac{1}{n}\sum_i (x-x)^2}$

$m = \frac{\sum (x-\bar{x})y}{\sum (x-\bar{x})^2}$

$\sigma_m^2 \simeq \frac{\chi^2}{\sum (x-\bar{x})^2 (n-2)}$



Anscombe's quartet

∴ always plot fitted line as sanity check!

c depends on m

↳ can recast problem: look at deviations from centre of mass at $\bar{x}$

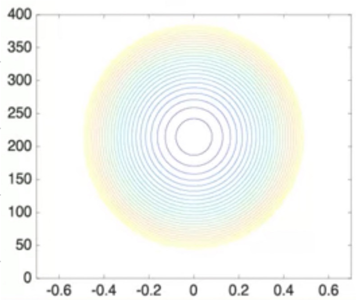
↳ $b = \bar{y}$ (centre of mass in y at $\bar{y}$)

$y = (m \pm \sigma_m)(x - \bar{x}) + (b \pm \sigma_b)$

$m = \frac{\sum (x-\bar{x})y}{\sum (x-\bar{x})^2}$ $\quad \sigma_m^2 \simeq \frac{\chi^2}{\sum (x-\bar{x})^2 (n-2)}$

$b = \bar{y}$ $\quad \sigma_b^2 \simeq \frac{\chi^2}{n(n-2)}$

doesn't depend on gradient anymore

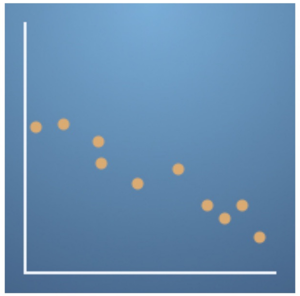


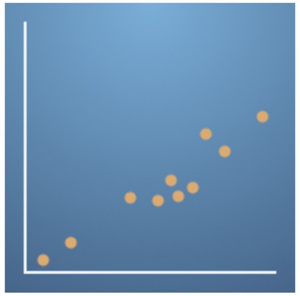
Contour plot for χ^2

↳ isn't slanted: removed interaction between m and c

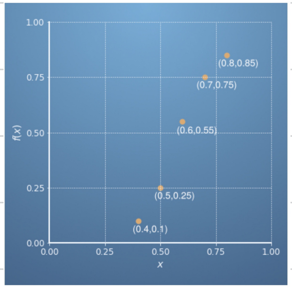
↳ more mathematically reasonable

Linear Regression Quiz

1.  Sensible data for linear fit

2.  Sensible data for linear fit

3. Linear regression by eye: $m = -0.25$, $c = 0.775$ gave $\chi^2 = 0.0388$ (good fit)

4. 

$\chi^2 = \sum (y_i - m x_i - c)^2$ (measure goodness of fit)

$\nearrow c = 0.5 - 2 \times 0.6 = -0.7$

Find min. of χ^2: differentiate and set to 0 $\rightarrow$ $m = \dfrac{\sum (x_i - \bar{x}) y_i}{\sum (x_i - \bar{x})^2}$, $c = \bar{y} - m\bar{x}$ (which minimise χ^2)

$\hookrightarrow \bar{x} = \dfrac{0.4 + 0.5 + 0.6 + 0.7 + 0.8}{5} = 0.6$, $\bar{y} = \dfrac{0.1 + 0.25 + 0.55 + 0.75 + 0.85}{5} = 0.5$

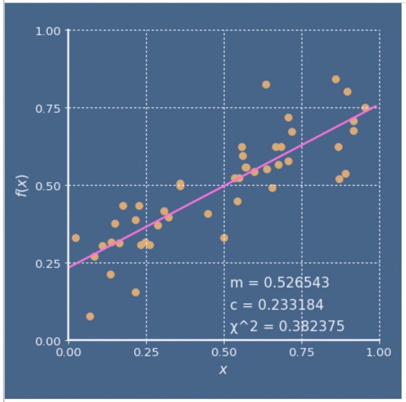
$$\hookrightarrow m = \frac{(0.4-0.6)(0.1) + (0.5-0.6)(0.25) + (0.6-0.6)(0.55) + (0.7-0.6)(0.75) + (0.8-0.6)(0.85)}{(0.4-0.6)^2 + (0.5-0.6)^2 + (0.6-0.6)^2 + (0.7-0.6)^2 + (0.8-0.6)^2} = \frac{0.2}{0.1} = 2$$

5.

```
# Here the function is defined
def linfit(xdat,ydat):
  # Here xbar and ybar are calculated
  xbar = np.sum(xdat)/len(xdat)
  ybar = np.sum(ydat)/len(ydat)

  # Insert calculation of m and c here. If nothing is here the data will be plotted with no l
  m = np.sum((xdat - xbar) * ydat)/np.sum((xdat - xbar) ** 2)
  c = ybar - m*xbar

  # Return your values as [m, c]
  return [m, c]

# Produce the plot - don't put this in the next code block
line()
```

Run

Reset

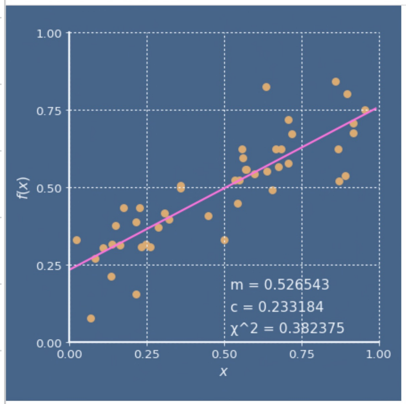


6.

```
from scipy import stats

# Use the stats.linregress() method to evaluate regression
regression = stats.linregress(xdat, ydat)

line(regression)
```

Run

Reset



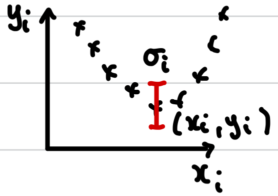
Hopefully it is clear that *linregress()* does everything *linfit()* did and more, without having to write it yourself!

General non linear least squares

$y(x; a_k) = (x-a_1)^2 + a_2$ $\quad k=1\ldots m$ (example of non linear least squares)

(y_i, x_i, σ_i)

uncertainty



$i = 1 \ldots n$

vector of parameters

$$\chi^2 = \sum_{i=1}^{n} \frac{[y_i - y(x_i; a_k)]^2}{\sigma^2}$$

(want to minimise sum of squares of residuals)

penalising differences by uncertainty so uncertain points have low weight in χ^2

$y(x; a_k) = (x-a_1)^2 + a_2$

$\frac{dy}{da_1} = -2(x-a_1)$, $\frac{dy}{da_2} = 1$

$$\nabla \chi^2 = 0 \rightarrow \frac{d\chi^2}{da_k} = \sum_{i=1}^{n} 2 \frac{[y_i - y(x_i; a_k)]}{\sigma^2} \frac{dy}{da_k} \rightarrow$$

$a_{next} = a_{cur} - cst. \nabla \chi^2$ (iterative, steepest descent formula)

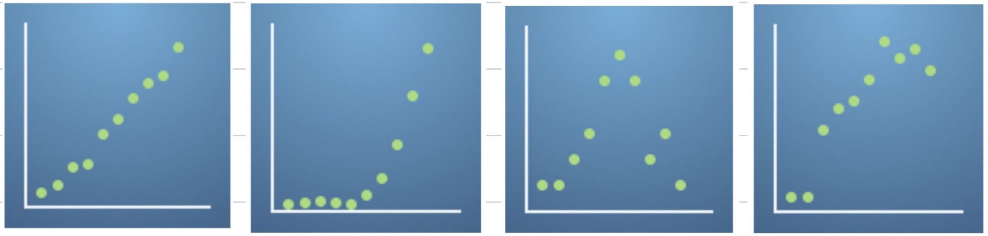
$$= a_{cur} + \sum_{i=1}^{n} \frac{[y_i - y(x_i; a_k)]}{\sigma^2} \frac{dy}{da_k}$$

$$\text{e.g. } f(x; \sigma, x_p, I, b) = b + \frac{I}{\sigma\sqrt{2\pi}} \exp\left\{ \frac{-(x-x_p)^2}{2\sigma^2} \right\}$$

Jacobian required to perform non linear least squares fit using Gaussian function

$$\hookrightarrow J = \left[\frac{\partial(\chi^2)}{\partial \sigma}, \frac{\partial(\chi^2)}{\partial x_p}, \frac{\partial(\chi^2)}{\partial I}, \frac{\partial(\chi^2)}{\partial b} \right]$$

linear curve fitting works here, but unnecessary



Fitting a non-linear function quiz

1. Nonlinear least squares method can be adapted to provide an effective fit to this data

2. $\chi^2 = \sum_{i=1}^{n} \frac{[y_i - y(x_i; a_k)]^2}{\sigma^2}$, taking the grad. of χ^2 and setting to 0 allows us to determine effective fitting parameters
↳ and when calculating χ we $\div$ by uncertainty σ, so χ is squared to minimise effect of dividing by highly uncertain result

3. $\frac{\partial \chi^2}{\partial a_j} = -2 \sum_{i=1}^{n} \frac{y_i - f(x_i; a)}{\sigma_i^2} \frac{\partial f(x_i; a)}{\partial a_j}$ for $j = 1 \ldots n$

Define matrix $[Z_j] = \frac{\partial f(x_i, a)}{\partial a_j}$

e.g. $f(x_i, a) = a_1 x^3 - a_2 x^2 + e^{-a_3 x}$

$\frac{\partial f}{\partial a_1} = x^3$, $\frac{\partial f}{\partial a_2} = -x^2$, $\frac{\partial f}{\partial a_3} = -x e^{-a_3 x}$

4. Jacobian for nonlinear least squares: $J = \left[\frac{\partial \chi^2}{\partial a_k} \right] = \left[\frac{\partial \chi^2}{\partial a_1}, \frac{\partial \chi^2}{\partial a_2} \right]$

e.g. $y(x_i; a) = a_1(1 - e^{-a_2 x_i^2})$, assume $\sigma^2 = 1$

Derivative of χ^2 w.r.t. fitting parameters: $$\frac{\partial \chi^2}{\partial a_j} = -2\sum_{i=1}^{n} \frac{y_i - f(x_i;a)}{\sigma_i^2} \frac{\partial f(x_i,a)}{\partial a_j} \quad \text{for } j=1....n$$

$$\frac{\partial f}{\partial a_1} = 1-e^{-a_2x_i^2}, \quad \frac{\partial f}{\partial a_2} = \frac{\partial}{\partial a_2}\left[a_1 - a_1e^{-a_2x_i^2}\right] = a_1x_i^2e^{-a_2x_i^2}$$

$$\frac{\partial \chi^2}{\partial a_j} = -2\sum_{i=1}^{n} \frac{y_i - f(x_i;a)}{\sigma_i^2} \frac{\partial f(x_i;a)}{\partial a_j} \quad \text{for } j=1....n$$

$$\therefore \frac{\partial \chi^2}{\partial a_1} = -2\sum_{i=1}^{n}\left[y_i - a_1(1-e^{-a_2x_i^2})\right](1-e^{-a_2x_i^2})$$

$$\frac{\partial \chi^2}{\partial a_2} = -2\sum_{i=1}^{n}\left[y_i - a_1(1-e^{-a_2x_i^2})\right](a_1x_i^2e^{-a_2x_i^2})$$

5. e.g. Gaussian distribution with 4 fitting parameters:

$$y(x;\sigma,x_p,I,b) = b + \frac{I}{\sigma\sqrt{2\pi}}\exp\left\{\frac{-(x-x_p)^2}{2\sigma^2}\right\}$$

$$J = \left[\frac{\partial \chi^2}{\partial \sigma}, \frac{\partial \chi^2}{\partial x_p}, \frac{\partial \chi^2}{\partial I}, \frac{\partial \chi^2}{\partial b}\right] \quad \text{where} \quad \frac{\partial \chi^2}{\partial a_j} = -2\sum_{i=1}^{n} \frac{y_i - y(x_i;a)}{\sigma_i^2} \frac{\partial y(x_i;a)}{\partial a_j} \quad \text{for } j=1....n$$

$$\frac{\partial y}{\partial x_p} = \frac{I}{\sqrt{2\pi}} \frac{(x-x_p)}{\sigma^3}\exp\left\{\frac{-(x-x_p)^2}{2\sigma^2}\right\}$$

only 1 partial derivative of Jacobian!

Least squares regression analysis in practice

Taylor series expansion of χ^2 : 2^{nd} term / derivative is Hessian (tells us gradient of gradient)
↳ can go directly to point where Jacobian is 0 like in Newton-Raphson
↳ using Hessian is faster than taking steps along steepest descent algorithm
↳ use Hessian to give guess as to size of step we should take in G.D.
↳ but Hessian often isn't stable (especially further from minimum)
↳ Levenberg-Marquardt method: steepest descent far from minimum, switches to Hessian when getting closer to minimum based on whether χ^2 is getting better or not
↳ also have Gauss-Newton and BFGS methods
↳ different methods may be better than others depending on convergence

Robust fitting
↳ would be unbothered by single outlier
↳ can minimise absolute unsquared deviations instead of least squares, so it doesn't weight points further away as strongly

Important to have good starting guess for curve fitting
Can use scipy to do all this!

Fitting distribution of heights data lab

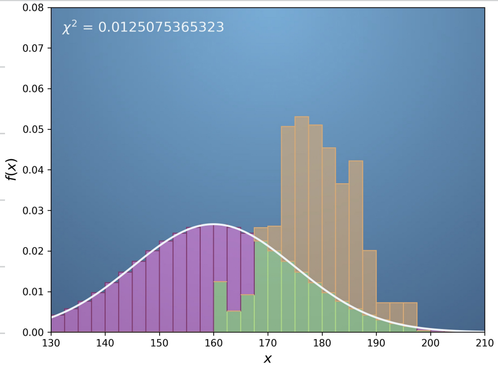
Fit a Gaussian model

Descend the χ^2 function: function of parameters that we are to optimise, and data the model is fit to

e.g. heights of people in population: plot histogram (bar chart with bars having width representing range of heights, and area which is probability of finding person with a height in that range)

↳ can model this data with a function (e.g. Gaussian), specify with 2 parameters, rather than holding all data in the histogram

Gaussian function: $f(x;\mu,\sigma) = \dfrac{1}{\sigma\sqrt{2\pi}} \exp\left(-\dfrac{(x-\mu)^2}{2\sigma^2}\right)$



orange = data

magenta = model

green = overlap

(not a good model fit: not strong overlap!)

χ^2: squared difference of data and the model: $\chi^2 = |\overbrace{y - f(x;\mu,\sigma)}^{\text{vectors}}|^2$

↳ sum of squares of pink and orange bars

To improve the fit: alter parameters μ and σ, see how it changes χ^2

↳ calculate Jacobian:

$$J = \left[\frac{\partial(\chi^2)}{\partial \mu} , \frac{\partial(\chi^2)}{\partial \sigma} \right]$$

```
# GRADED FUNCTION

# This is the Gaussian function.
def f (x,mu,sig) :
    return np.exp(-(x-mu)**2/(2*sig**2)) / np.sqrt(2*np.pi) / sig

# Next up, the derivative with respect to μ.
# If you wish, you may want to express this as f(x, mu, sig) multiplied by chain rule terms.
# === COMPLETE THIS FUNCTION ===
def dfdmu (x,mu,sig) :
    return f(x, mu, sig) * ((x-mu)/sig**2)

# Finally in this cell, the derivative with respect to σ.
# === COMPLETE THIS FUNCTION ===
def dfdsig (x,mu,sig) :
    return f(x, mu, sig) * ((mu**2 - sig**2 + x**2 - 2*mu*x)/sig**3)
```

↳ $$\frac{\partial(\chi^2)}{\partial \mu} = -2(y - f(x;\mu,\sigma)) \cdot \frac{\partial f}{\partial \mu}(x;\mu,\sigma)$$

$$\frac{\partial(\chi^2)}{\partial \sigma} = -2(y - f(x;\mu,\sigma)) \frac{\partial f}{\partial \sigma}(x;\mu,\sigma)$$

and $$\frac{\partial f}{\partial \mu}(x;\mu,\sigma) = \frac{(x-\mu)}{\sigma^3\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$ (can use f in code to simplify)

$$\frac{\partial f}{\partial \sigma}(x;\mu,\sigma) = \frac{(\mu^2-\sigma^2+x^2-2\mu x)}{\sigma^4\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

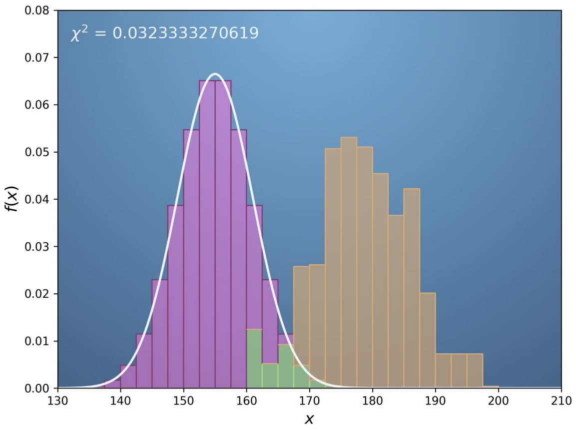
Steepest descent moves around in parameter space proportional to -ve of Jacobian

↳ $\begin{bmatrix} \delta\mu \\ \delta\sigma \end{bmatrix} \propto -J$, constant of proportionality = aggression of algorithm

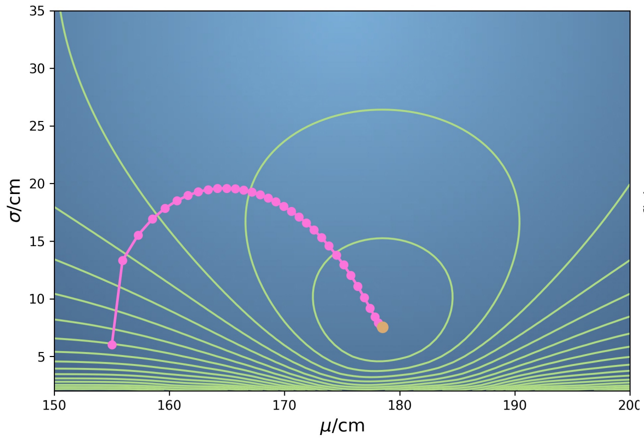
```
# GRADED FUNCTION

# Complete the expression for the Jacobian, the first term is done for you.
# Implement the second.
# === COMPLETE THIS FUNCTION ===
def steepest_step (x, y, mu, sig, aggression) :
    J = np.array([
        -2*(y - f(x,mu,sig)) @ dfdmu(x,mu,sig),
        -2*(y - f(x,mu,sig)) @ dfdsig(x,mu,sig)
    ])
    step = -J * aggression
    return step
```

```
# First get the heights data, ranges and frequencies
x,y = heights_data()

# Next we'll assign trial values for these.
mu = 155 ; sig = 6
# We'll keep a track of these so we can plot their evolution.
p = np.array([[mu, sig]])

# Plot the histogram for our parameter guess
histogram(f, [mu, sig])
# Do a few rounds of steepest descent.
for i in range(50) :
    dmu, dsig = steepest_step(x, y, mu, sig, 2000)
    mu += dmu
    sig += dsig
    p = np.append(p, [[mu,sig]], axis=0)
# Plot the path through parameter space.
contour(f, p)
# Plot the final histogram.
histogram(f, [mu, sig])
```

Initial guess



More $\perp$er to contours



Model fit