Data Analytics and Machine Learning

Newton methods in *n*D

Henrik Karstoft
Carl Schultz
Alexandros Iosifidis

landscape of approaches for “solving” polynomial equations/inequalities

- we will learn:
  - some of most popular, useful methods
  - situations they are most effective
  - their limitations

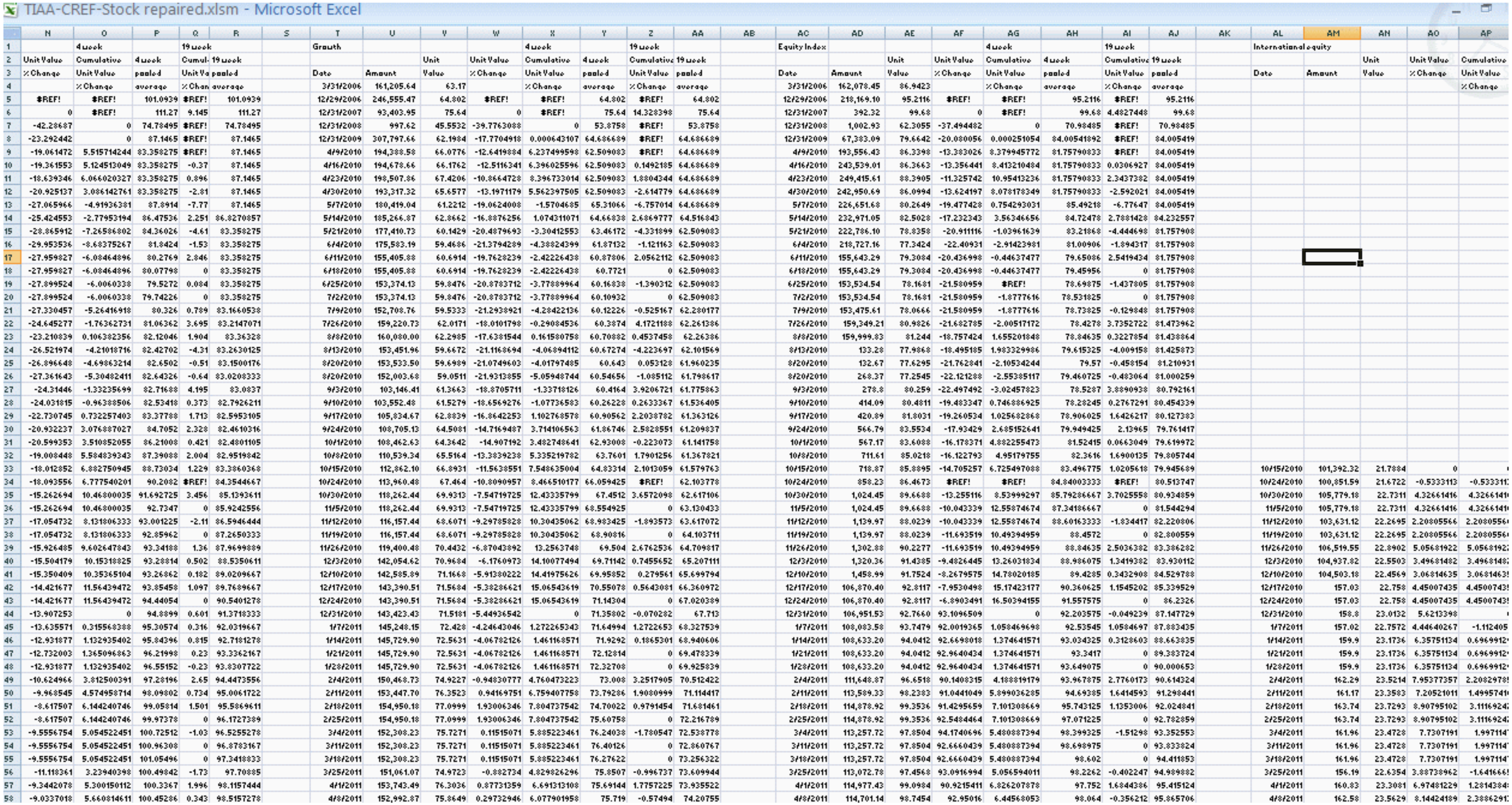
machine learning

optimization

communication,
visualization

today: data overload

data **useless** without interpretation



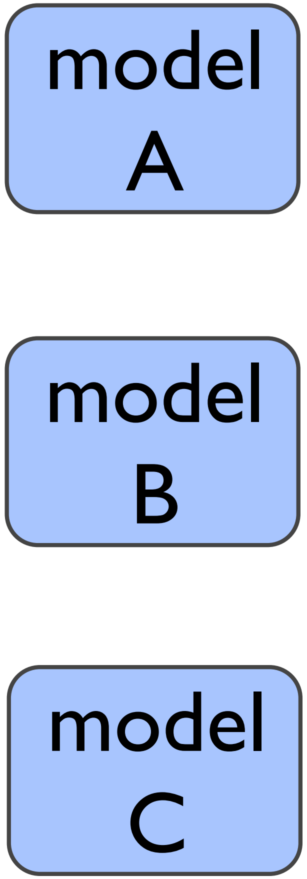
anything interesting happening here?

models can "find patterns"

image

model B → cat!

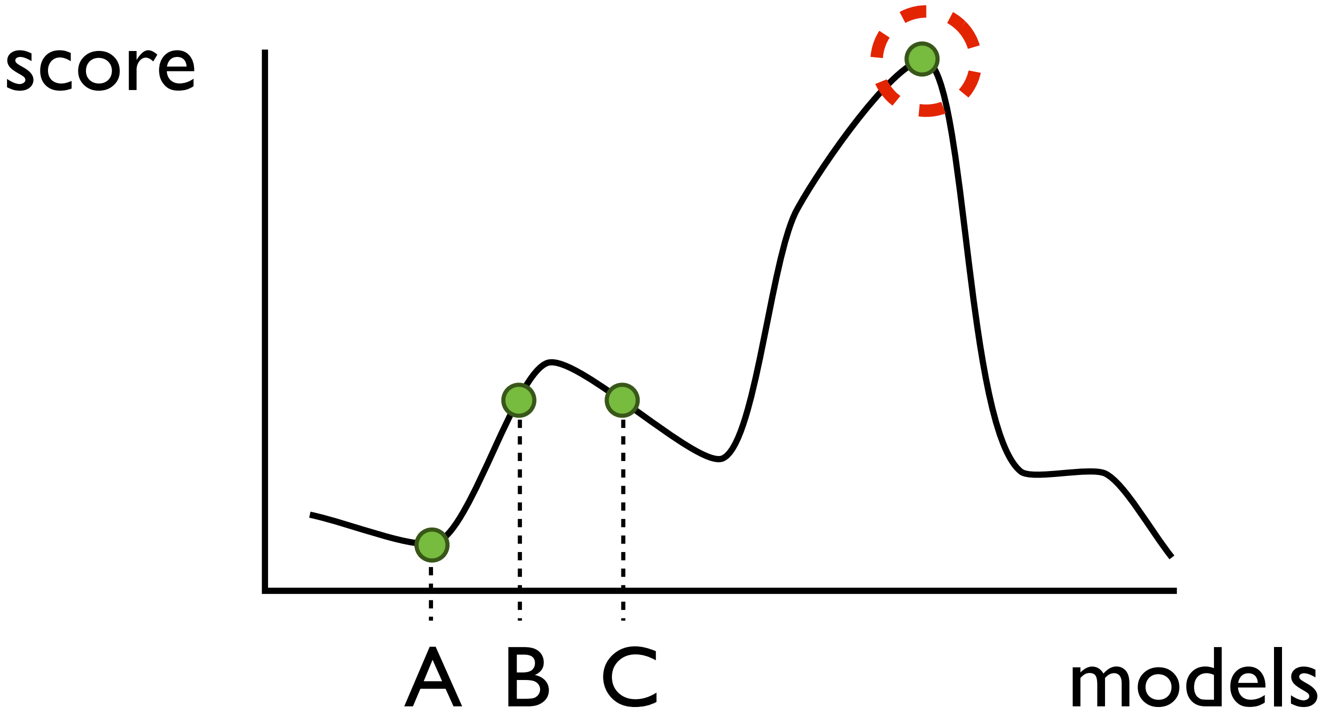
imagine set of **all possible** models



...

let's give each model a **score**

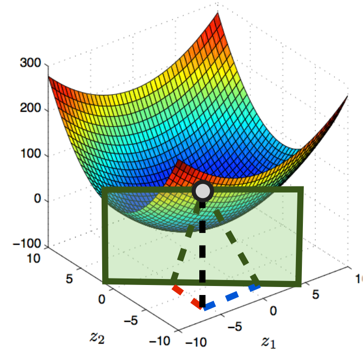
we want **best model** we can find

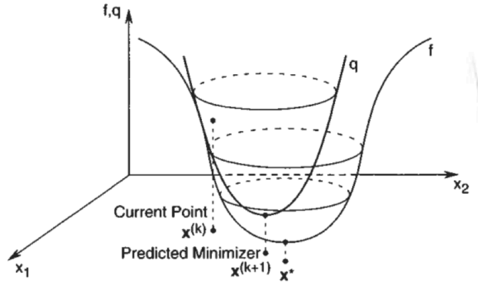


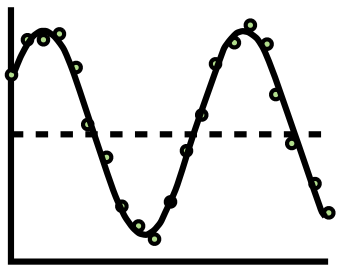
optimization: find model that maximises score

TODAY'S OUTLINE

- preliminaries
- Newton *from* **1D** *to* **nD**
- *from* **root finding** *to* **optimization**
- *from* **Jakobian** *to* **Hessian**



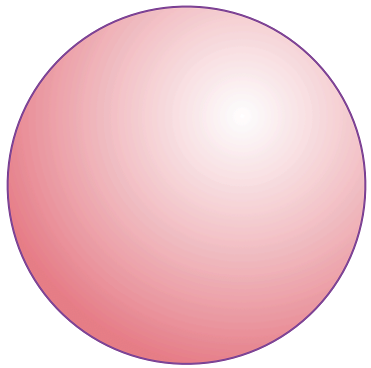




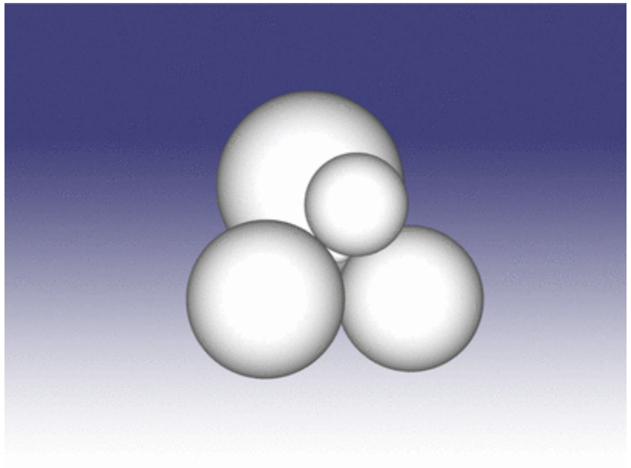
Part 1

some preliminary concepts

spatial puzzle solving

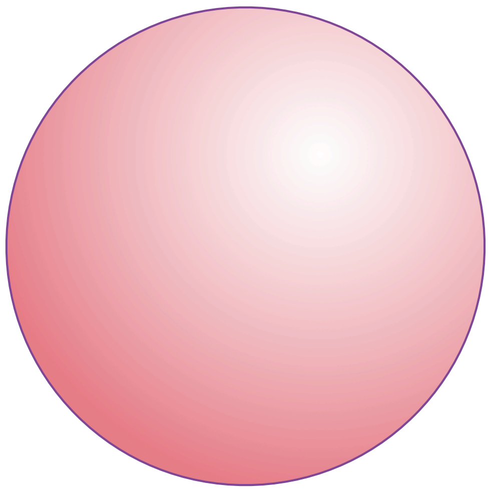


how many spheres (of any size)
can be mutually touching?

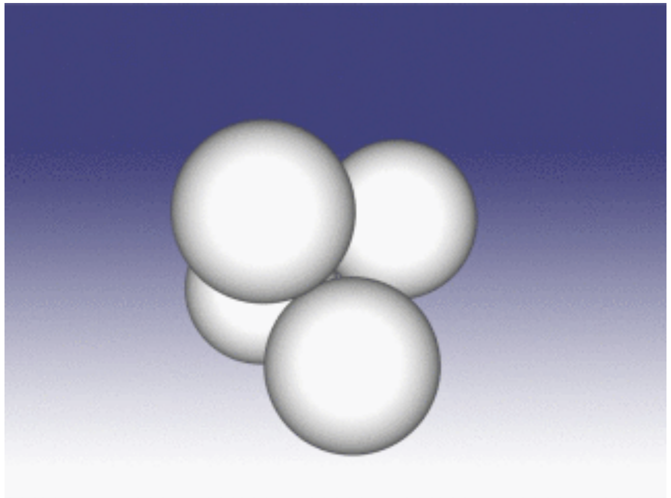


5 spheres

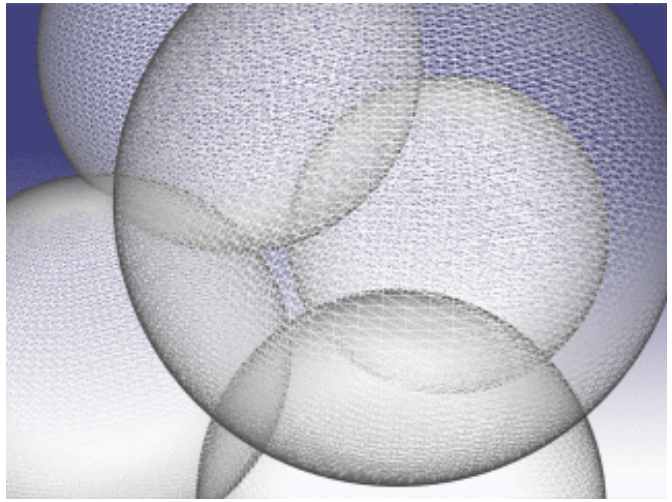
spatial puzzle solving



how many **same-sized** spheres
can be mutually touching?

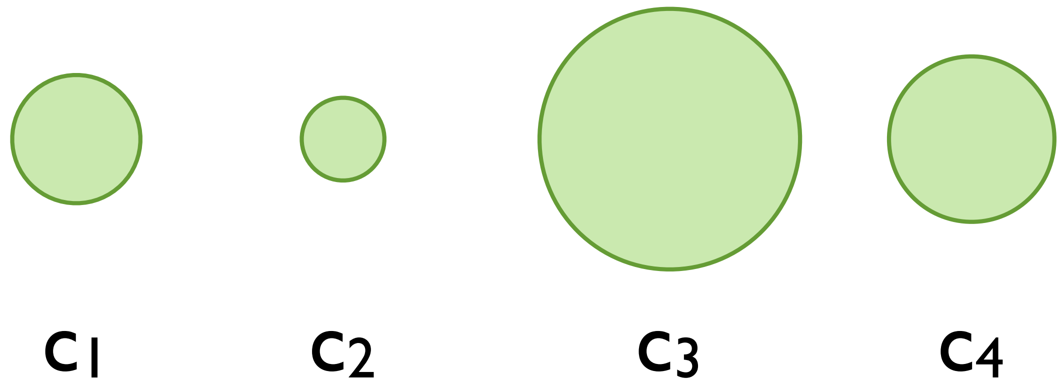


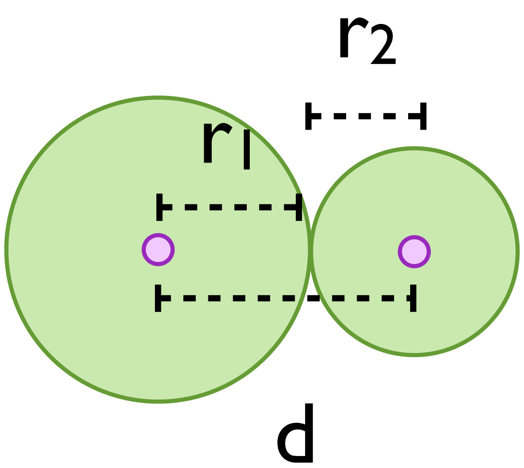
4 spheres



5 spheres

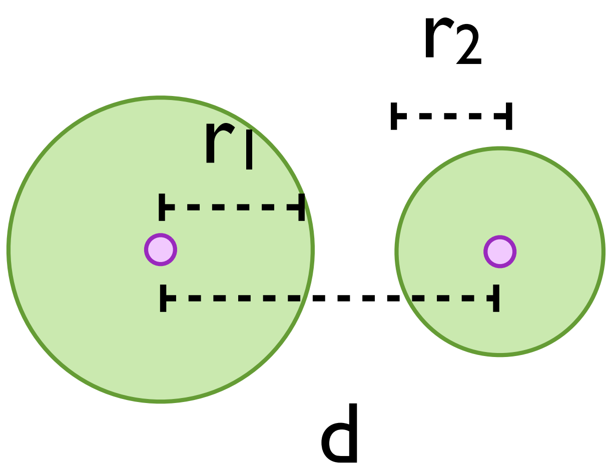
can four circles all touch each other?





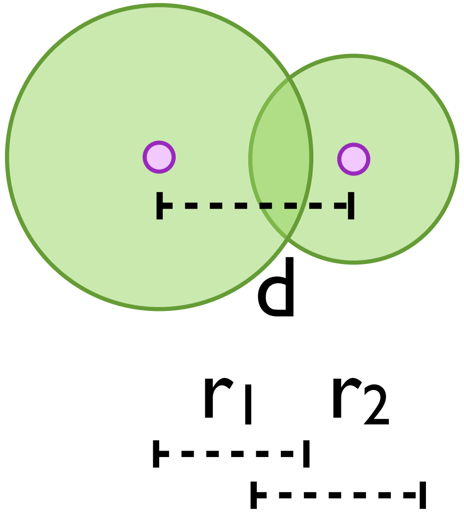
2 circles “touch” IF
(distance between centres) EQUALS (sum of radii)

$$d = (r_1 + r_2)$$



2 circles "touch" IF
(distance between centres) EQUALS (sum of radii)

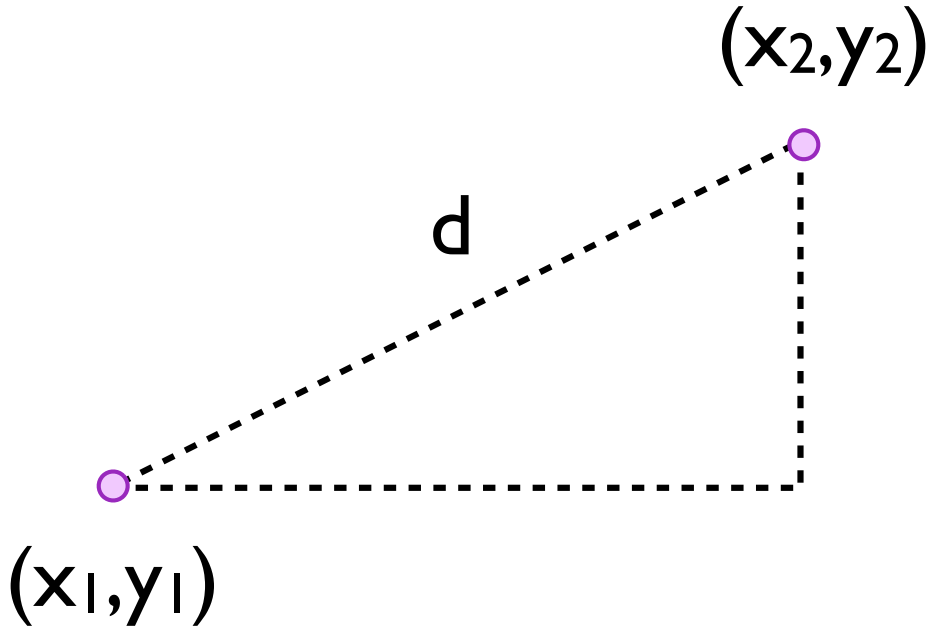
$$d = (r_1 + r_2)$$



2 circles "touch" IF

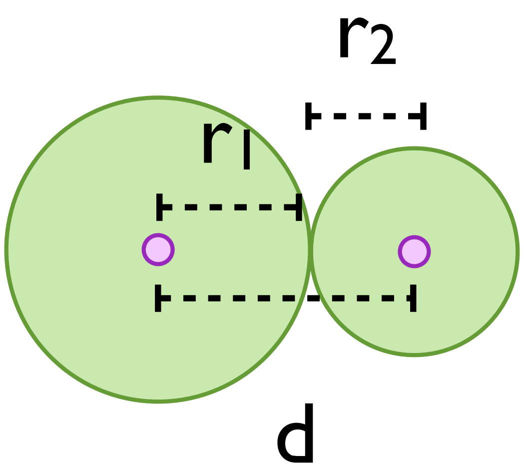
(distance between centres) EQUALS (sum of radii)

$$d = (r_1 + r_2)$$

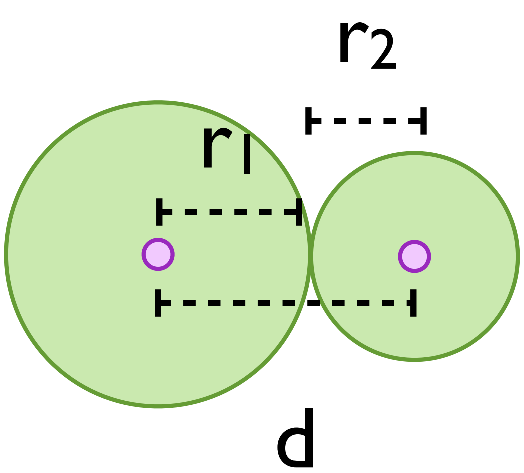


$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

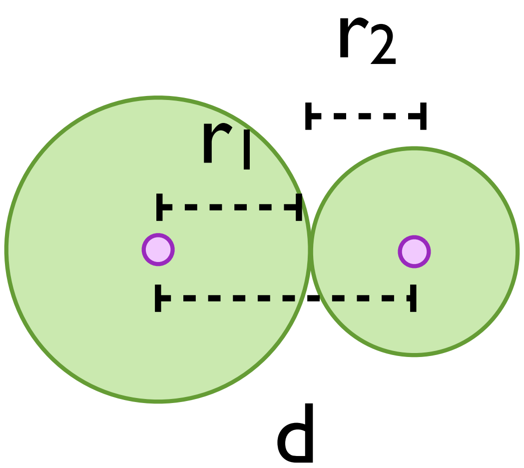
$$d^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2$$



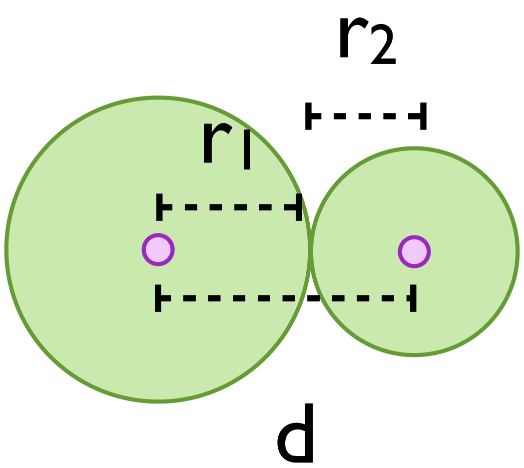
$$d = (r_1 + r_2)$$



$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} = (r_1 + r_2)$$

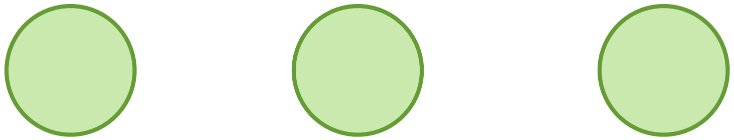


$$(x_1 - x_2)^2 + (y_1 - y_2)^2 = (r_1 + r_2)^2$$



$$(x_1 - x_2)^2 + (y_1 - y_2)^2 - (r_1 + r_2)^2 = 0$$

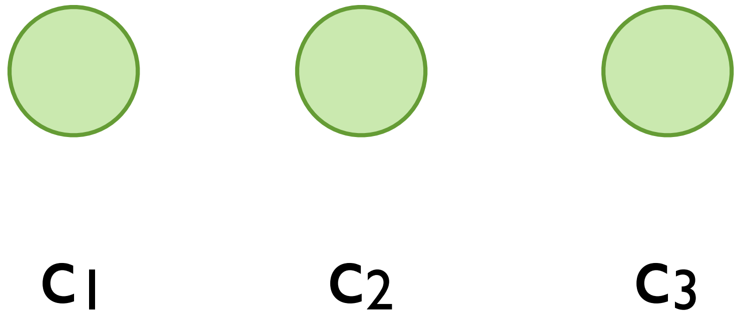
can three **same-sized** circles all touch each other?



c_1 c_2 c_3

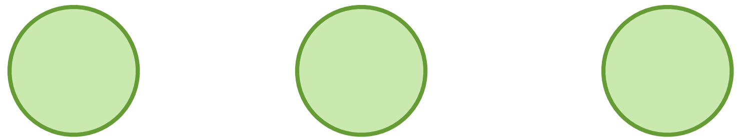
?

can three same-sized circles all touch each other?



c_1 touches c_2
c_1 touches c_3
c_2 touches c_3

can three same-sized circles all touch each other?



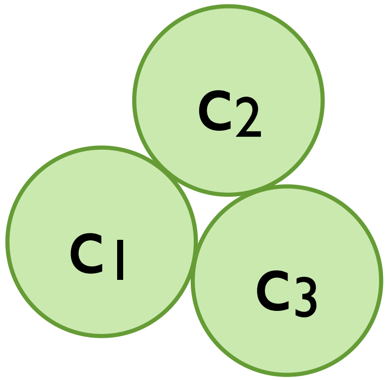
c_1 c_2 c_3

$$(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2 = 0$$

$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$

$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$

can three same-sized circles all touch each other?

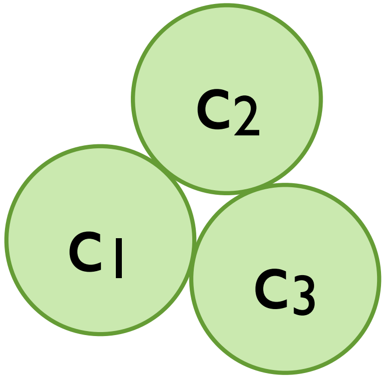


$$(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2 = 0$$

$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$

$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$

can three same-sized circles all touch each other?



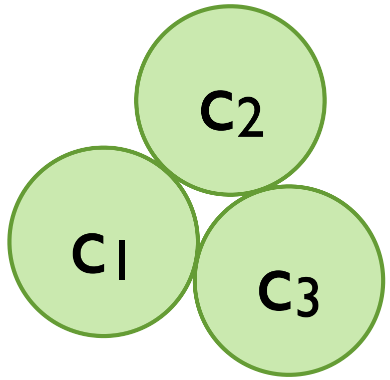
polynomial expression

$$\boxed{(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2} = 0$$

$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$

$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$

can three same-sized circles all touch each other?



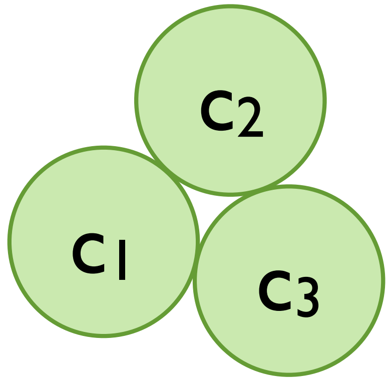
number of variables?

$$\boxed{(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2} = 0$$

$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$

$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$

can three same-sized circles all touch each other?



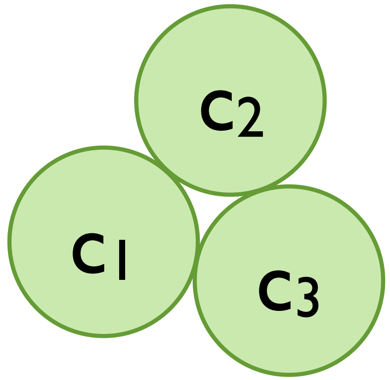
5 variables (“multi-variate” expression)

$$\boxed{(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2} = 0$$

$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$

$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$

can three same-sized circles all touch each other?



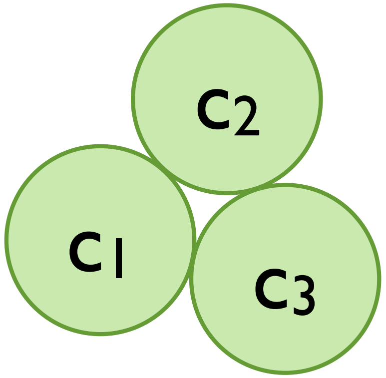
5 “dimensions”

$$\boxed{(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2} = 0$$

$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$

$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$

can three same-sized circles all touch each other?



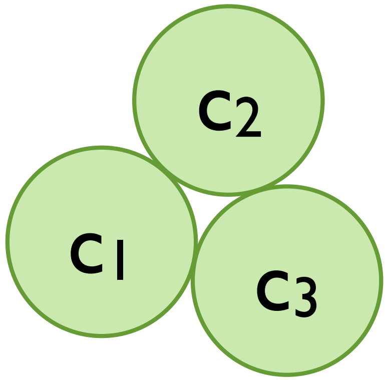
linear?

$$\boxed{(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2} = 0$$

$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$

$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$

can three same-sized circles all touch each other?



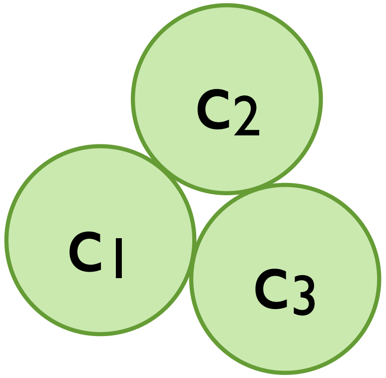
non-linear polynomial expression

$$\boxed{(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2} = 0$$

$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$

$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$

can three same-sized circles all touch each other?



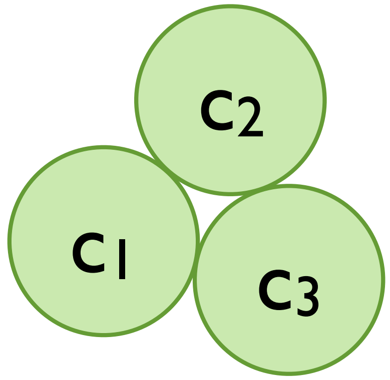
polynomial equation

$$\boxed{(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2 = 0}$$

$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$

$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$

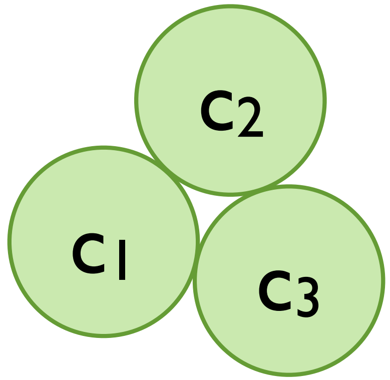
can three same-sized circles all touch each other?



$$(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2 = 0$$
$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$
$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$

system of polynomial equations

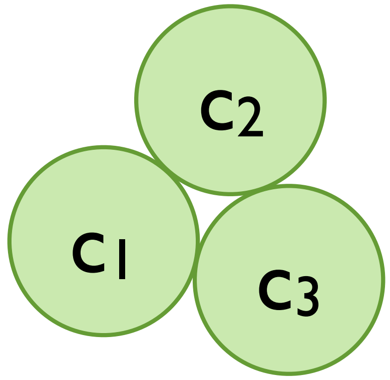
can three same-sized circles all touch each other?



$$(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2 = 0$$
$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$
$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$

number of variables?

can three same-sized circles all touch each other?



$$(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2 = 0$$

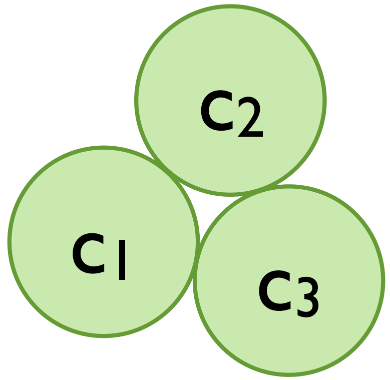
$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$

$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$

7 variables

(7 “dimensions”)

can three same-sized circles all touch each other?

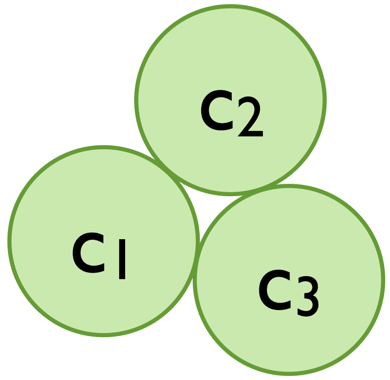


$$(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2 = 0$$
$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$
$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$

our task: **solving** a system of equations

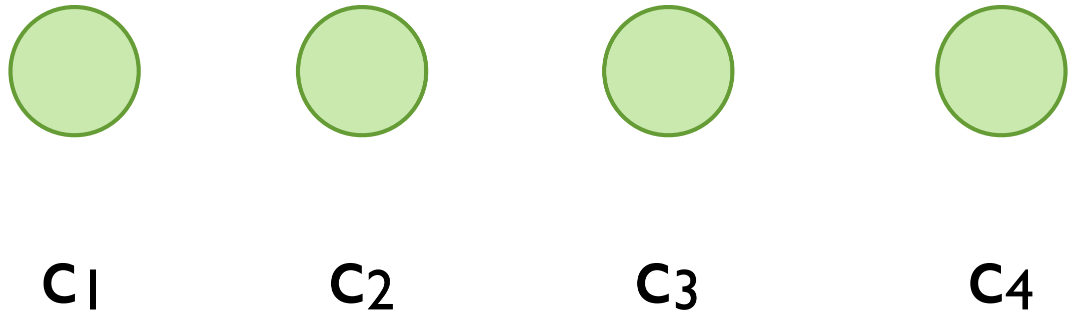
what does "solving" mean?

can three same-sized circles all touch each other?

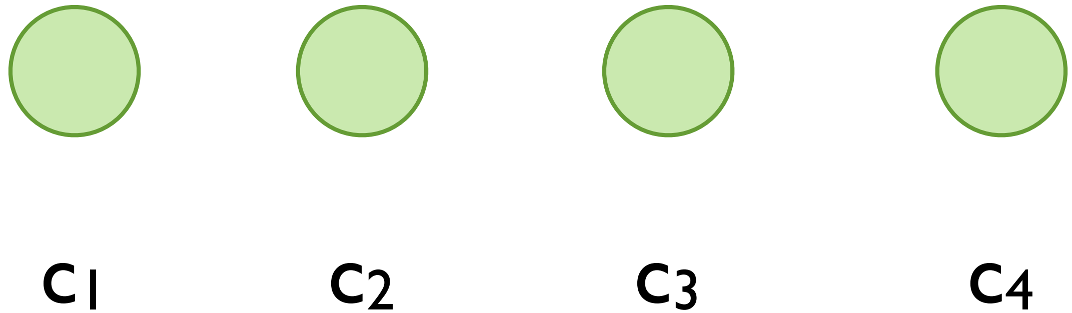


$$(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2 = 0$$
$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$
$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$

our task: finding roots of system of equations can four **same-sized** circles all touch each other?



can four **same-sized** circles all touch each other?



no.

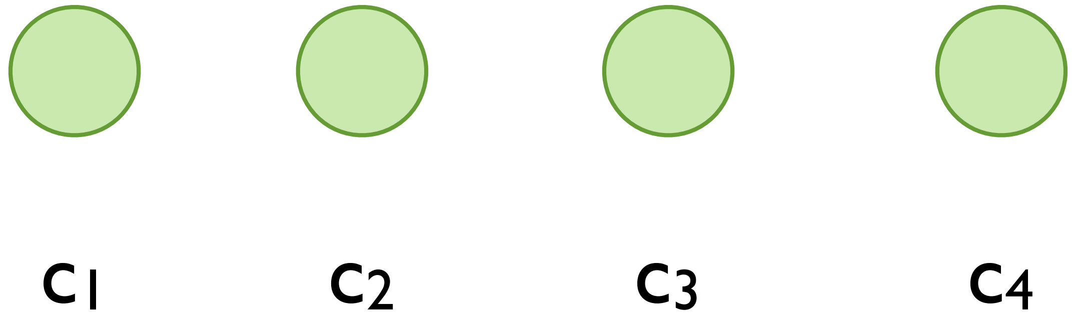
can four same-sized circles all touch each other?

c_1 c_2 c_3 c_4

$$(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2 = 0$$
$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$
$$(x_1-x_4)^2 + (y_1-y_4)^2 - (r+r)^2 = 0$$
$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$
$$(x_2-x_4)^2 + (y_2-y_4)^2 - (r+r)^2 = 0$$
$$(x_3-x_4)^2 + (y_3-y_4)^2 - (r+r)^2 = 0$$

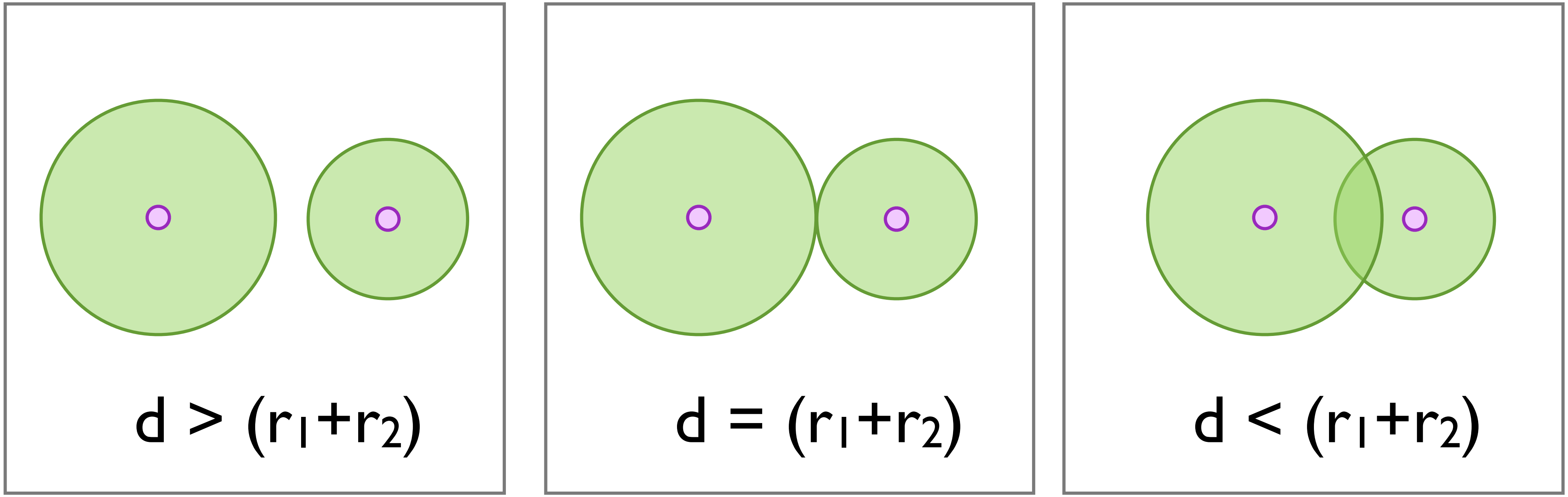
no solution to this system of equations

can four same-sized circles all touch each other?

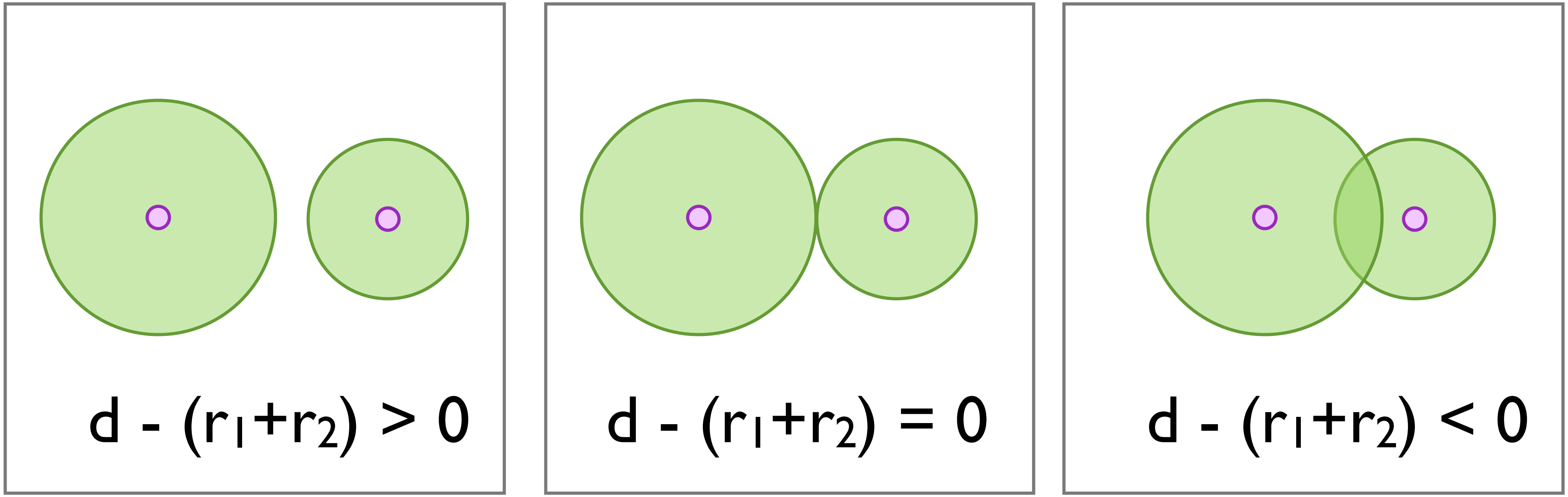


$$(x_1-x_2)^2 + (y_1-y_2)^2 - (r+r)^2 = 0$$
$$(x_1-x_3)^2 + (y_1-y_3)^2 - (r+r)^2 = 0$$
$$(x_1-x_4)^2 + (y_1-y_4)^2 - (r+r)^2 = 0$$
$$(x_2-x_3)^2 + (y_2-y_3)^2 - (r+r)^2 = 0$$
$$(x_2-x_4)^2 + (y_2-y_4)^2 - (r+r)^2 = 0$$
$$(x_3-x_4)^2 + (y_3-y_4)^2 - (r+r)^2 = 0$$

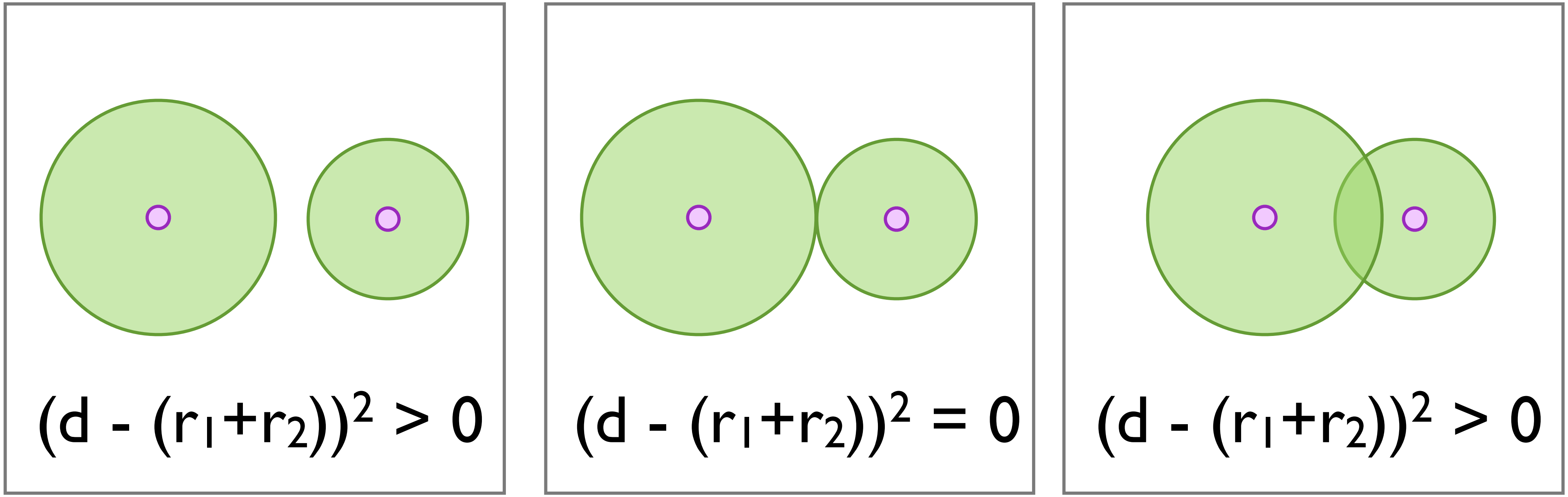
how “close” can we get?



can we measure "cost" of not satisfying constraint?



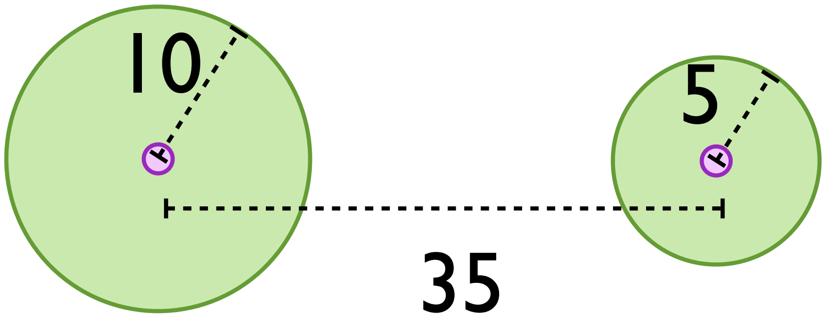
can we measure "cost" of not satisfying constraint?



idea: measure "how far are we from solution" by taking square of expression?

want to find variable values that **minimise** this "cost"

$$(d - (r_1+r_2))^2$$



$$(d - (r_1+r_2))^2$$

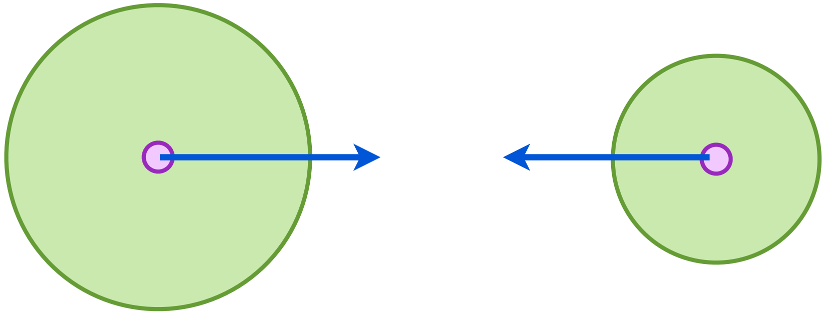
$$= (35 - (10+5))^2$$

$$= 400$$

$$(d - (r_1+r_2))^2$$

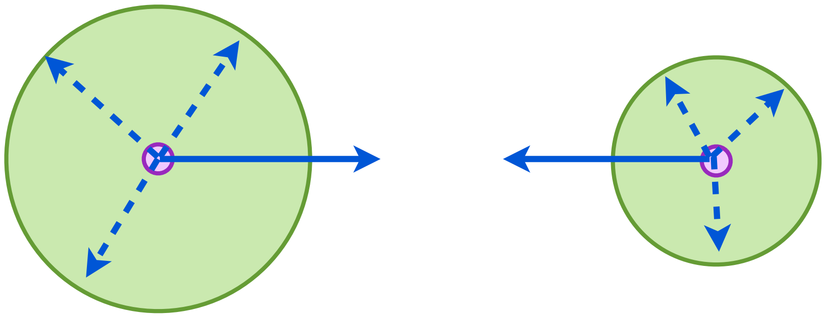
what **direction** should each variable move to get a better score?

centres should move towards each other...



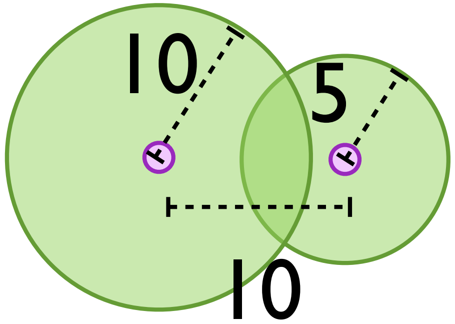
$$(d - (r_1+r_2))^2$$

radii should increase



$$(d - (r_1+r_2))^2$$

the **gradient** of our function can give us that information

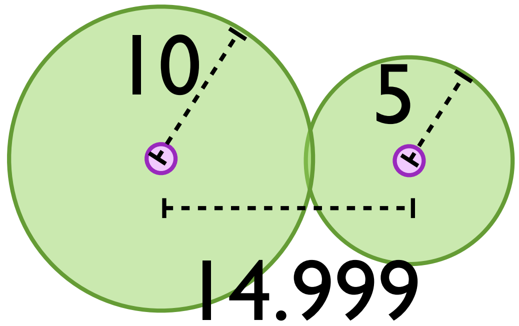


$$(d - (r_1+r_2))^2$$

$$= (10 - (10+5))^2$$

$$= 25$$

closer!



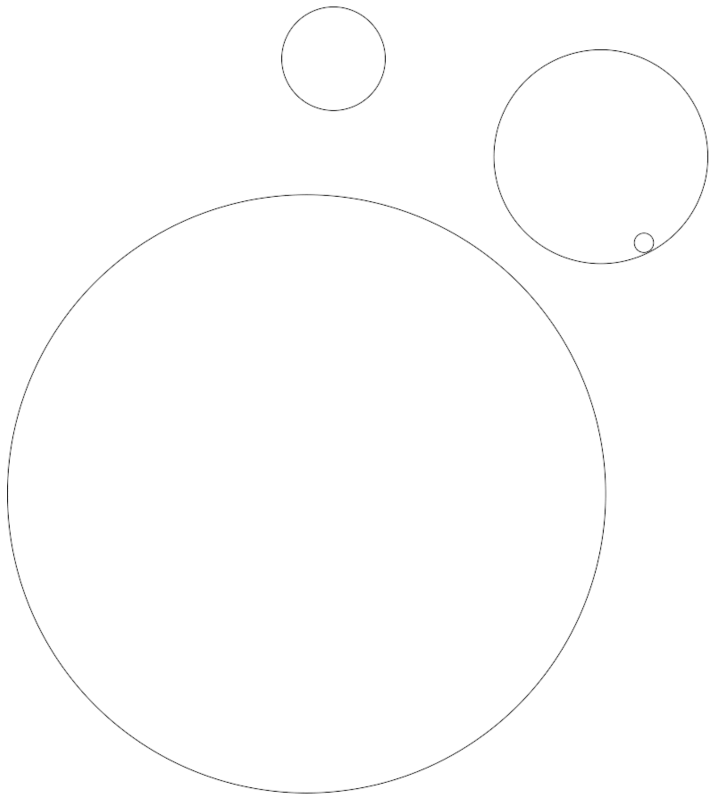
$$(d - (r_1+r_2))^2$$

$$= (14.999 - (10+5))^2$$

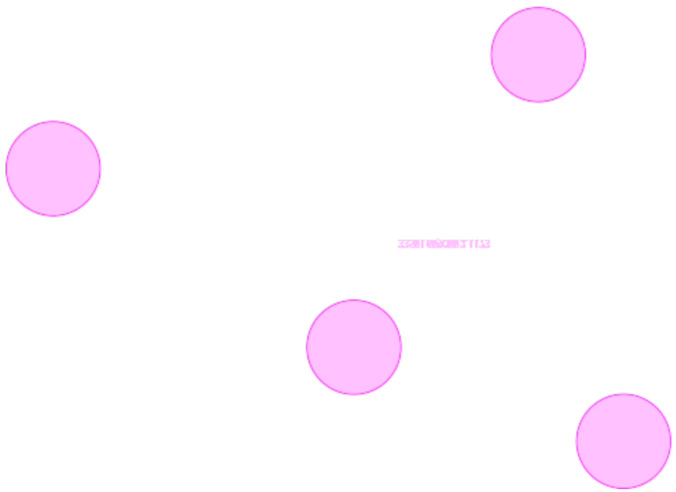
$$= 0.000001$$

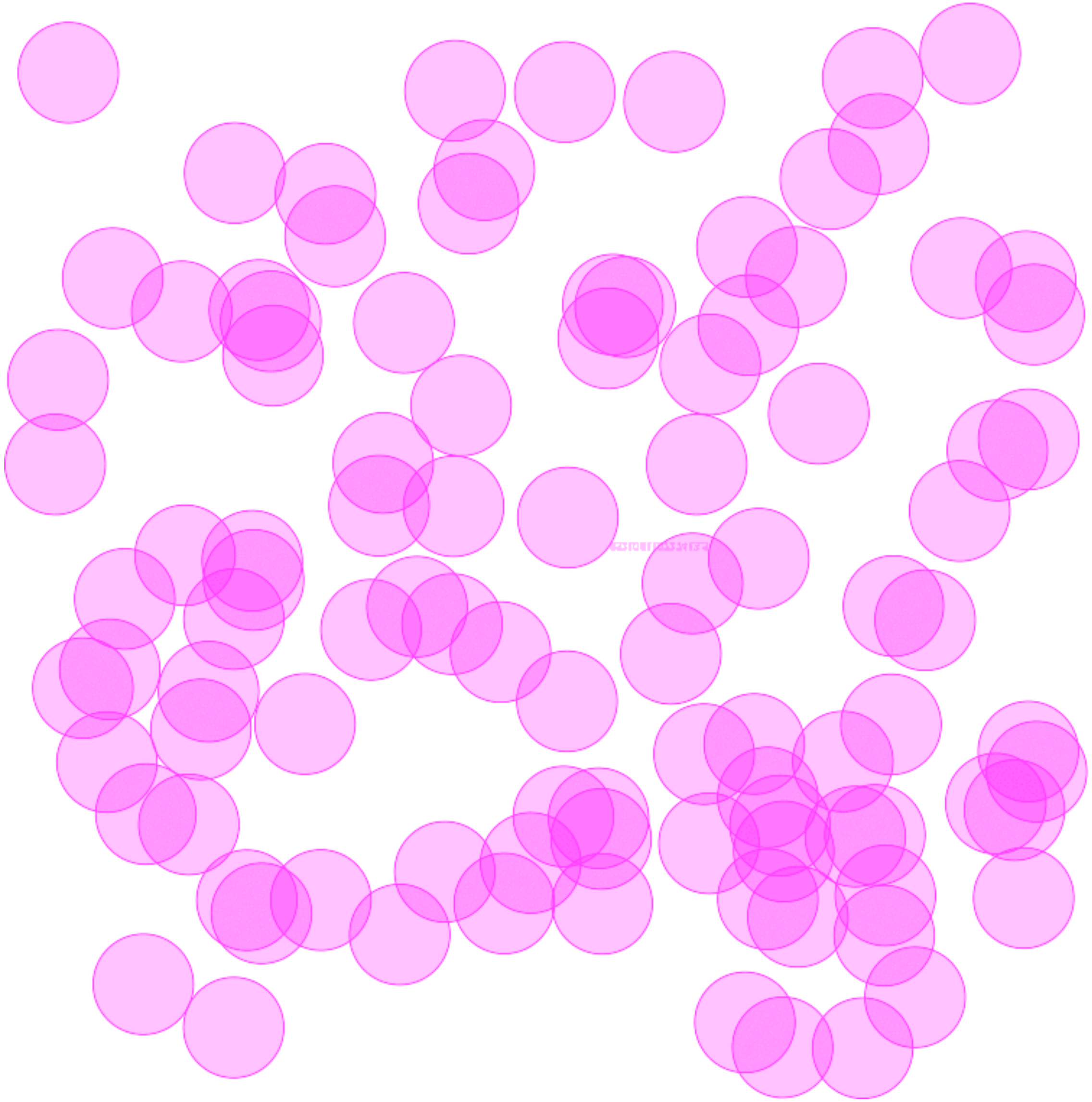
close enough?

4 circles touching



4 circles
touching, same size





100 circles?

SUMMARY Part 1. prelliminaries

- expression, equation, inequality
- system of equations
- linear vs nonlinear
- 1 dimension (univariate) vs
  n dimensions (multivariate)
- “solving”: root-finding vs optimisation

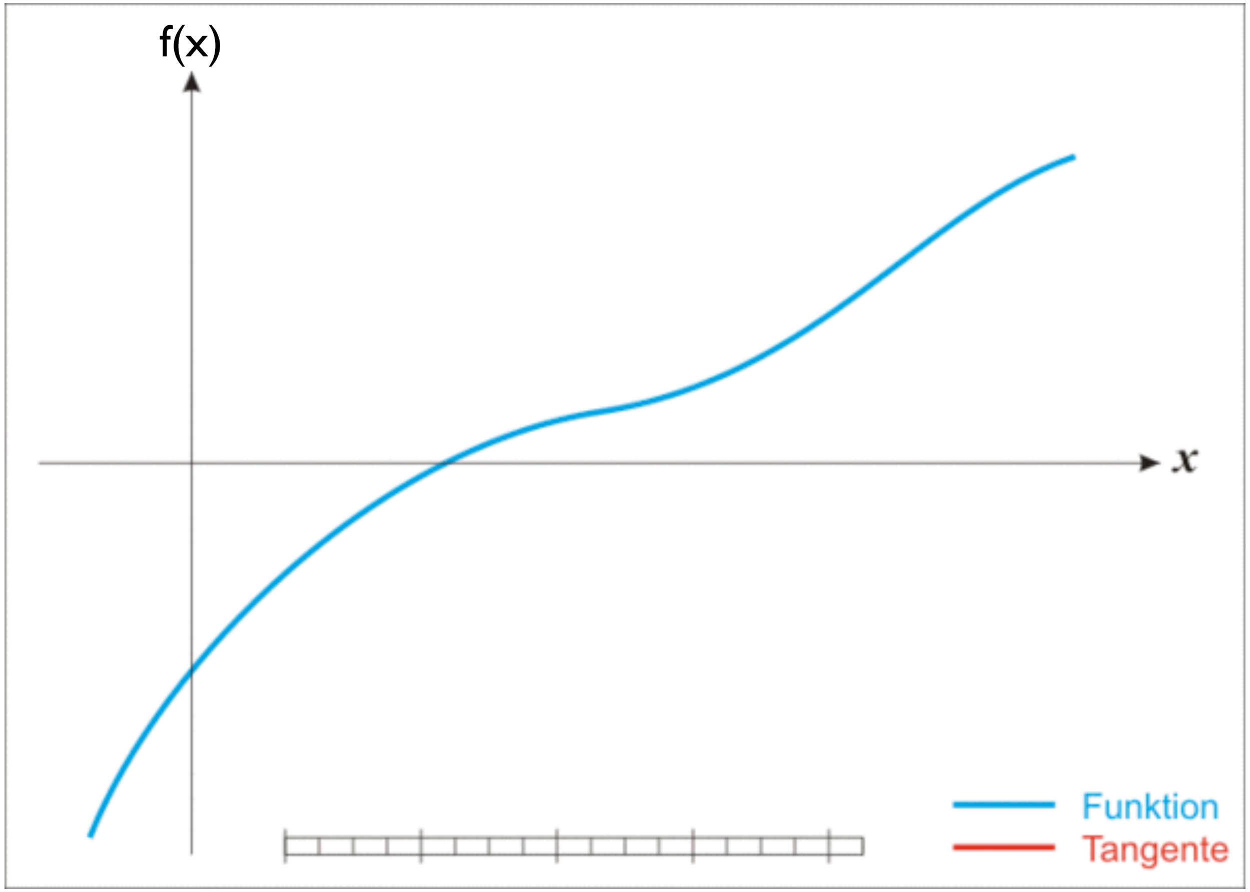
Part 2

iterative methods

guess - check - new guess - check - new guess - check ...

Newton's method for root-finding

- you've seen this for 1 function, for 1 dimension
- first we'll extend this to *n* dimensions
- then we'll see correspondance to optimisation task



http://en.wikipedia.org/wiki/Newton's_method

Newton's method for root-finding

$$x^{(i+1)} = x^{(i)} + \text{update}$$

Newton's method for root-finding

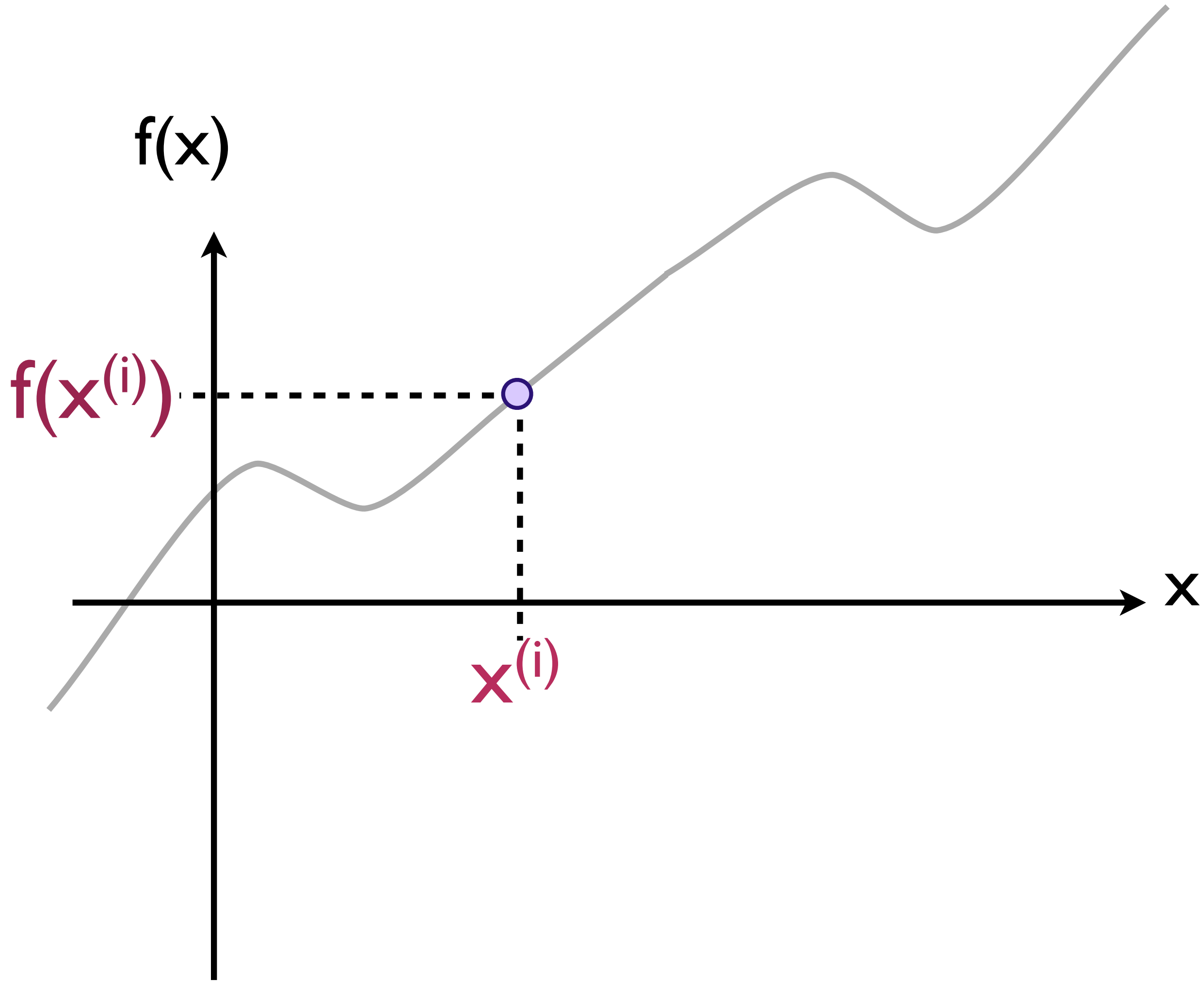
$$x^{(i+1)} = x^{(i)} + \frac{-f(x^{(i)})}{f'(x^{(i)})}$$

deriving Newton's method

deriving Newton's method

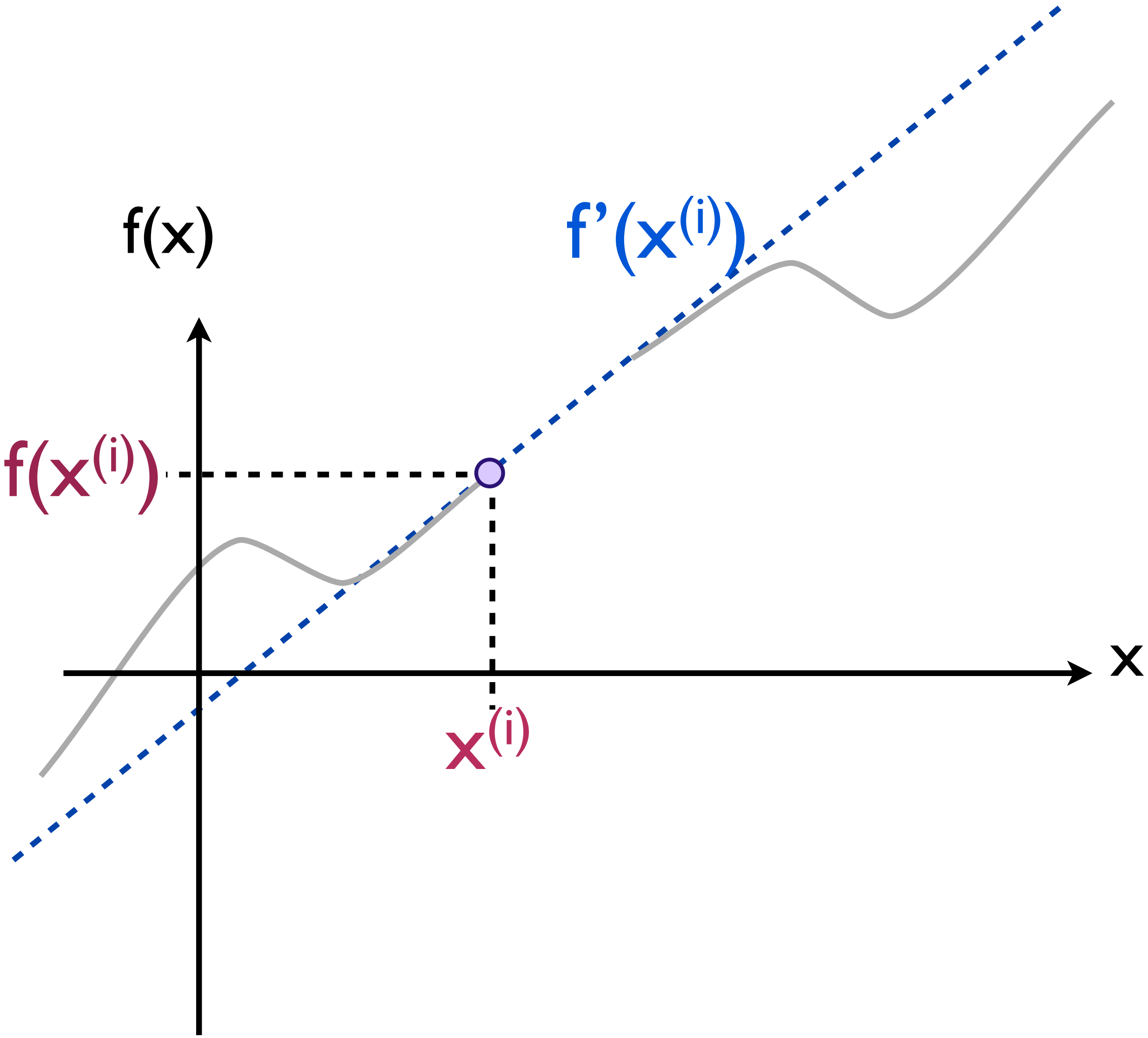
first we'll derive it in the "naive" way

then we'll derive it using Taylor series expansion say we're at $x^{(i)}$ and we want the next step to find the root, $f(x^{(i+1)})=0$



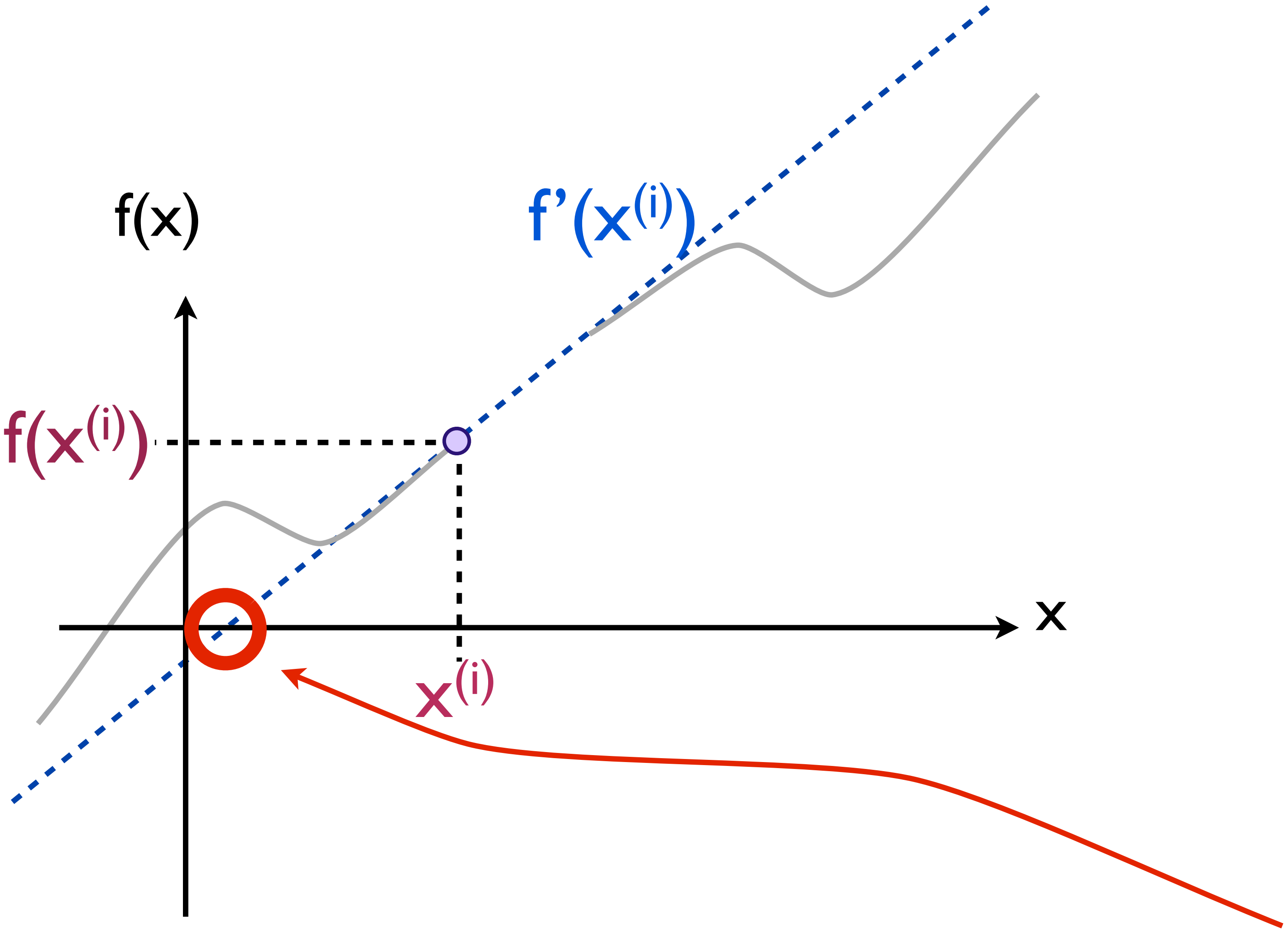
If our function *f* was linear, then this approximation would be **exact,**

that's the approximation we're making



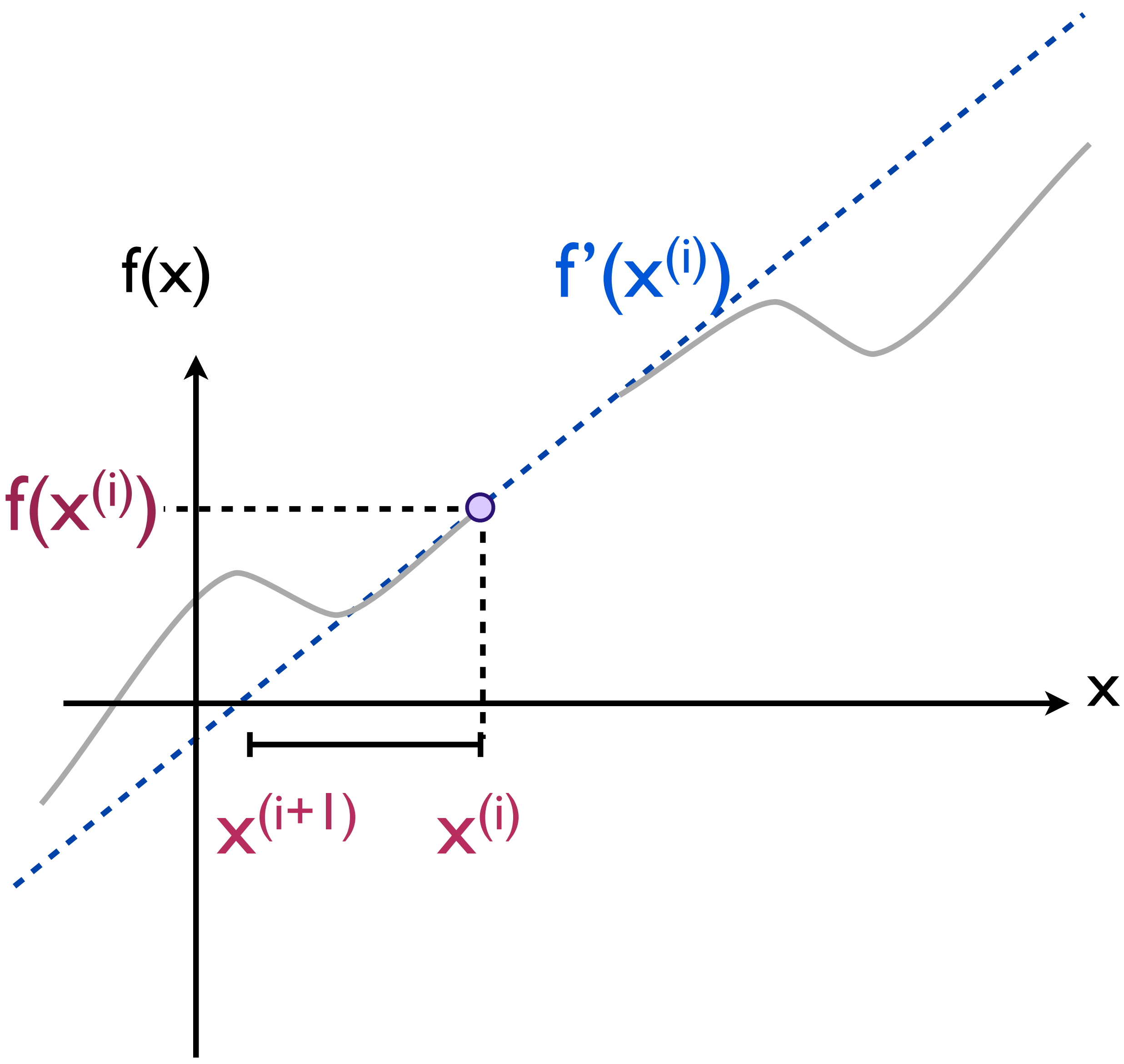
If our function *f* was linear, then this approximation would be **exact,**

that's the approximation we're making



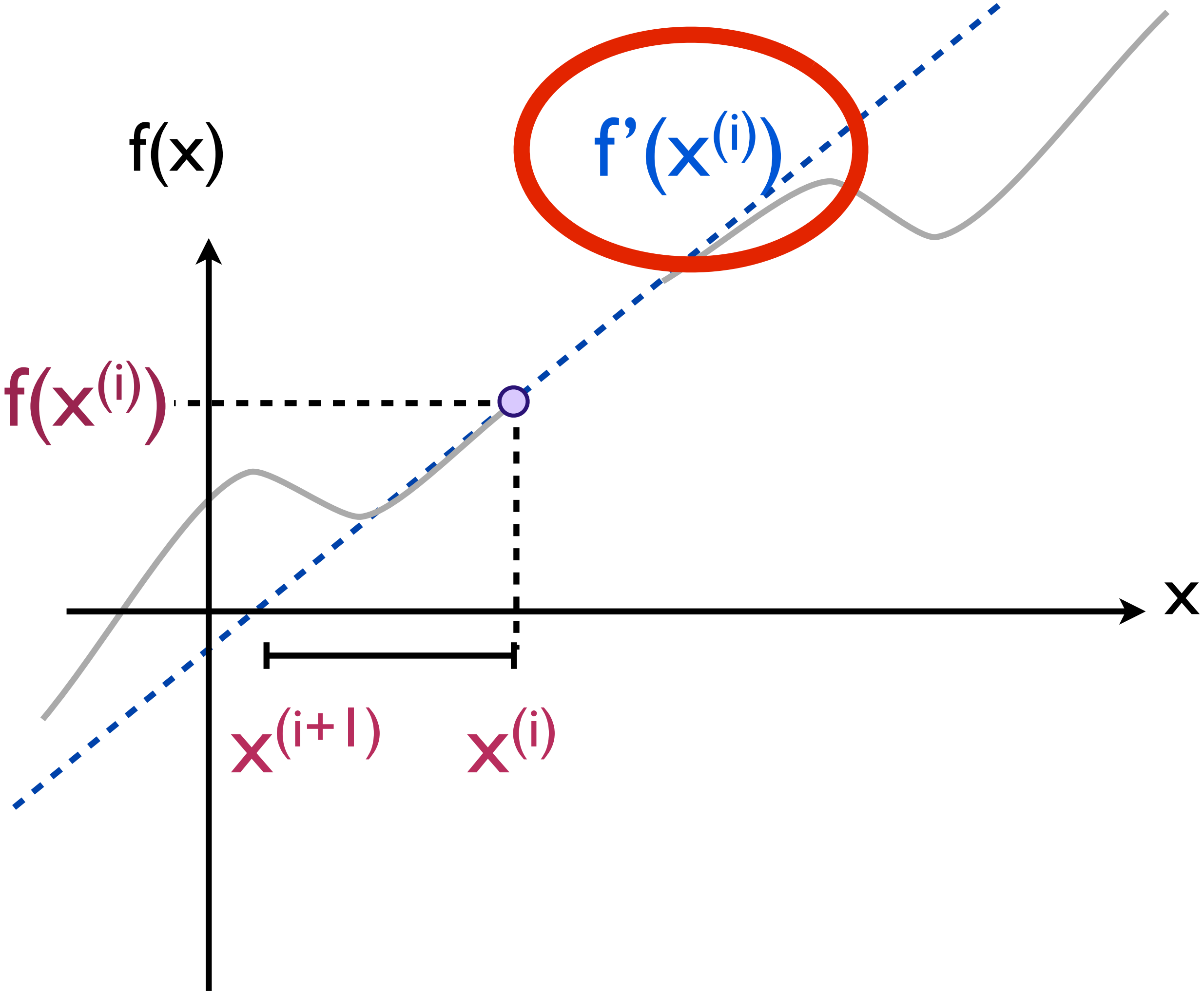
if that was the case, then the root of *f* would be here

so let's find the change in x



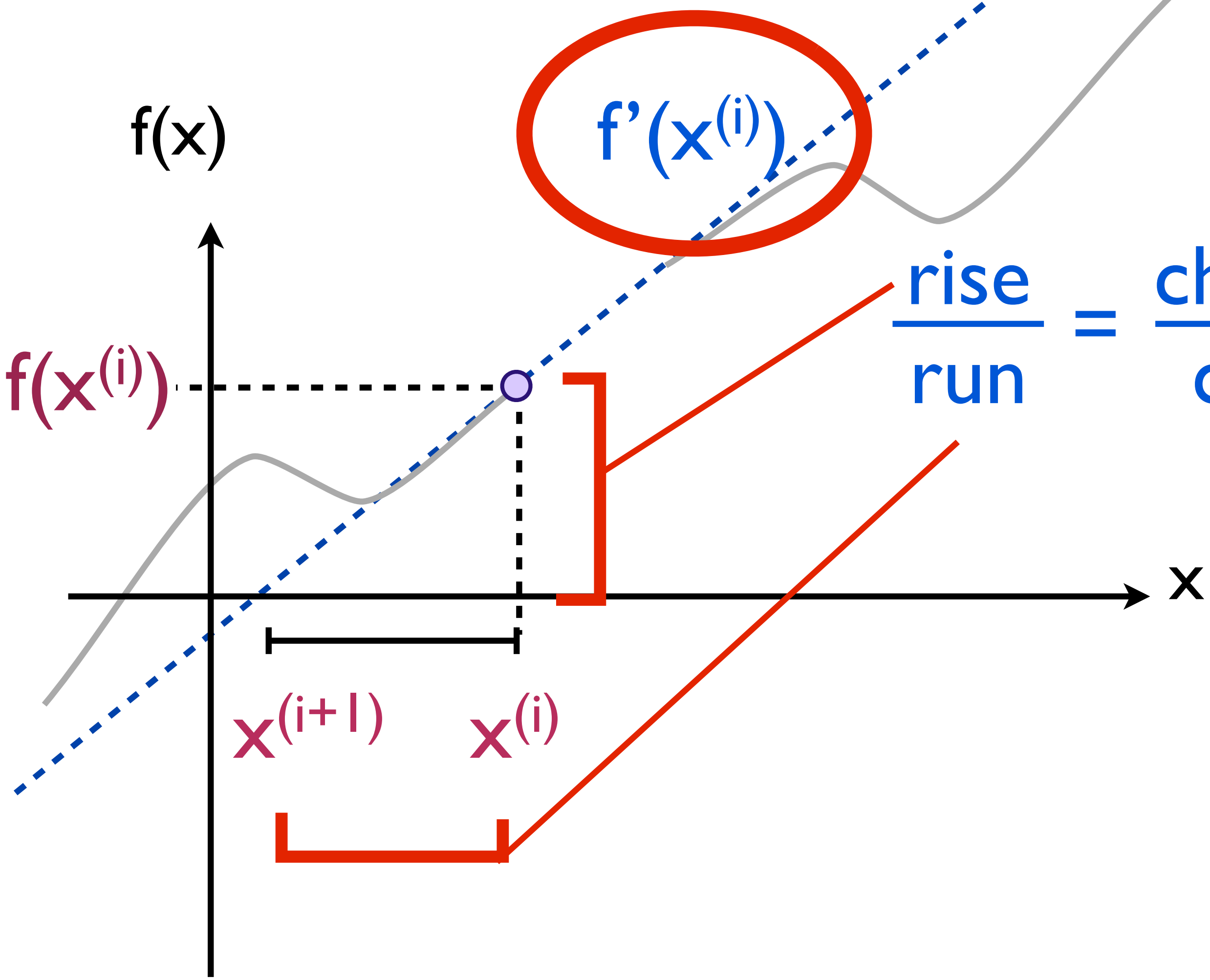
so let's find the change in x

f'($x^{(i)}$) is the "slope" of this line, i.e. it is a ratio - what is it a ratio of?



so let's find the change in x

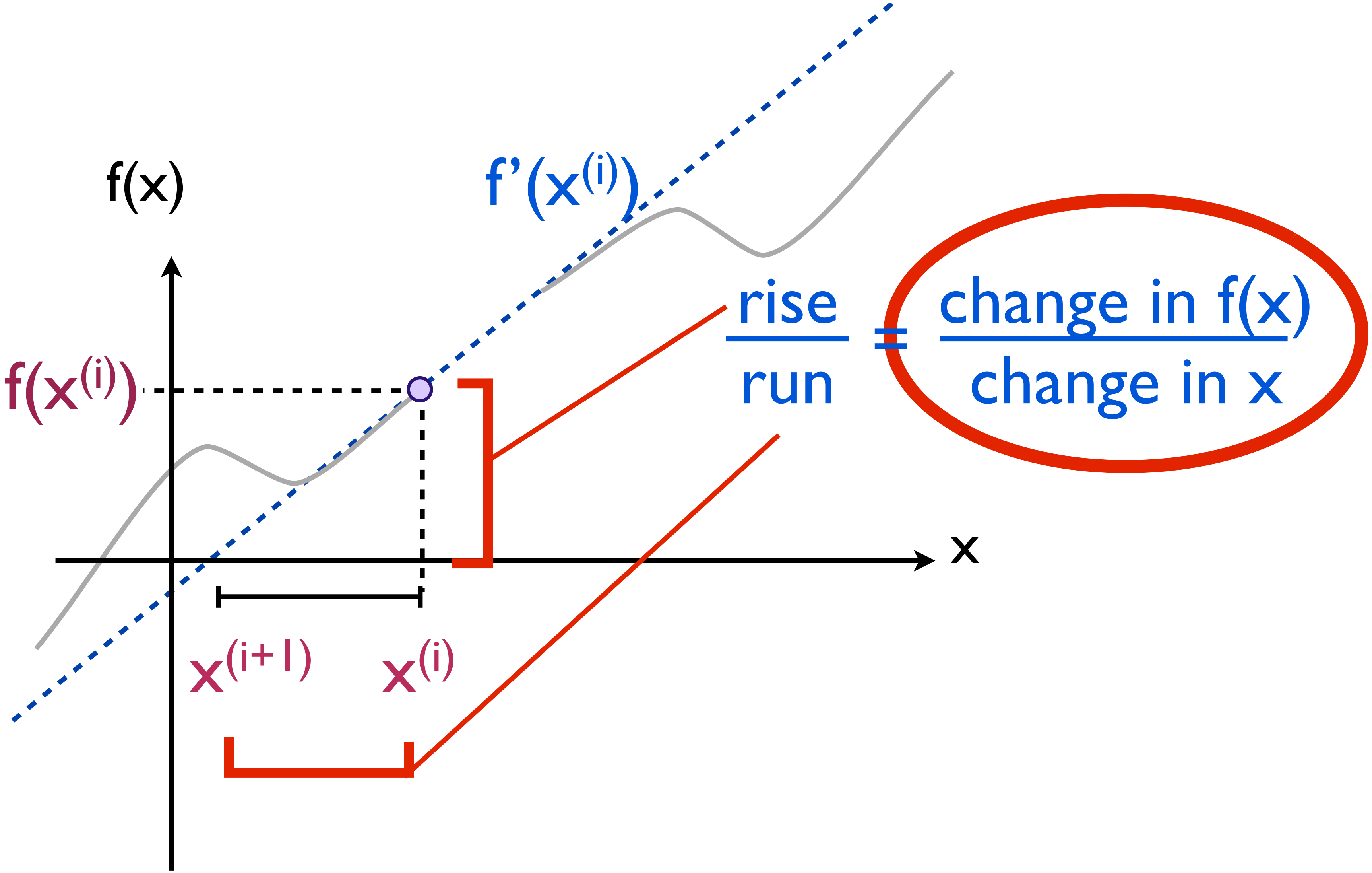
f'($x^{(i)}$) is the "slope" of this line, i.e. it is a ratio - what is it a ratio of?



$$\frac{\text{rise}}{\text{run}} = \frac{\text{change in } f(x)}{\text{change in } x}$$

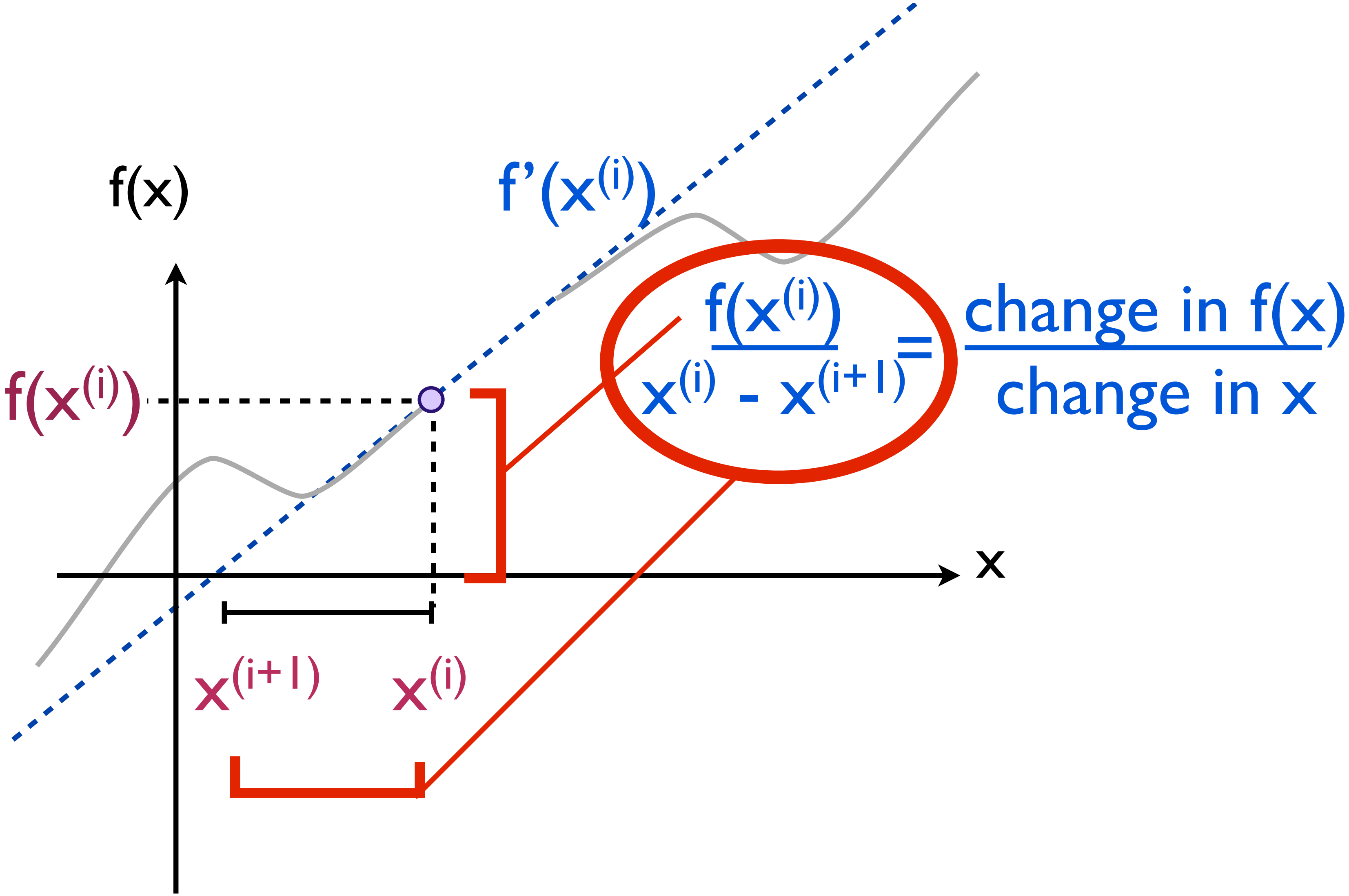
so let's find the change in x

What is "change in f(x)"? What is "change in x"?



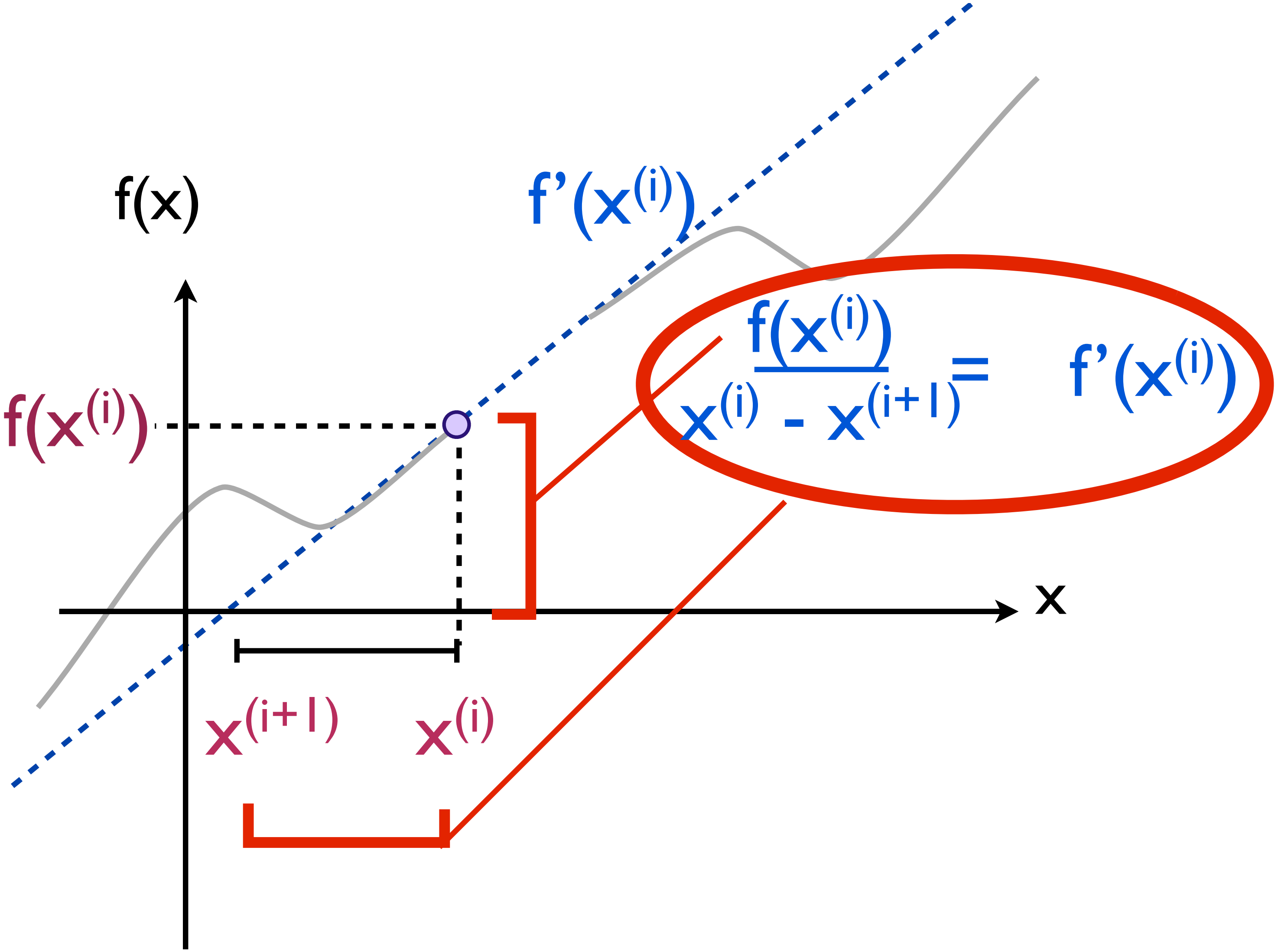
so let's find the change in x

What is "change in f(x)"? What is "change in x"?



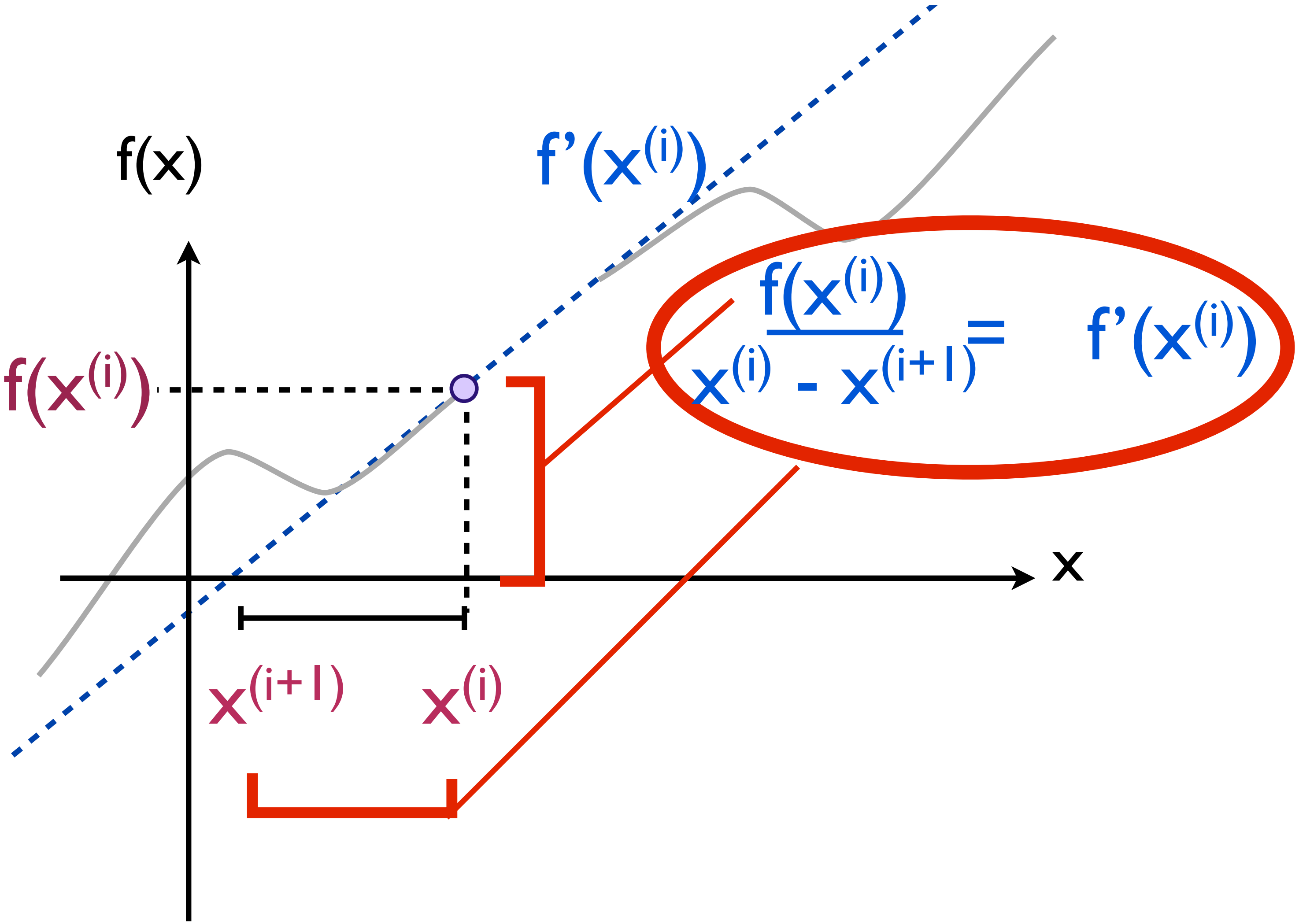
so let's find the change in x

What is "change in f(x)"? What is "change in x"?



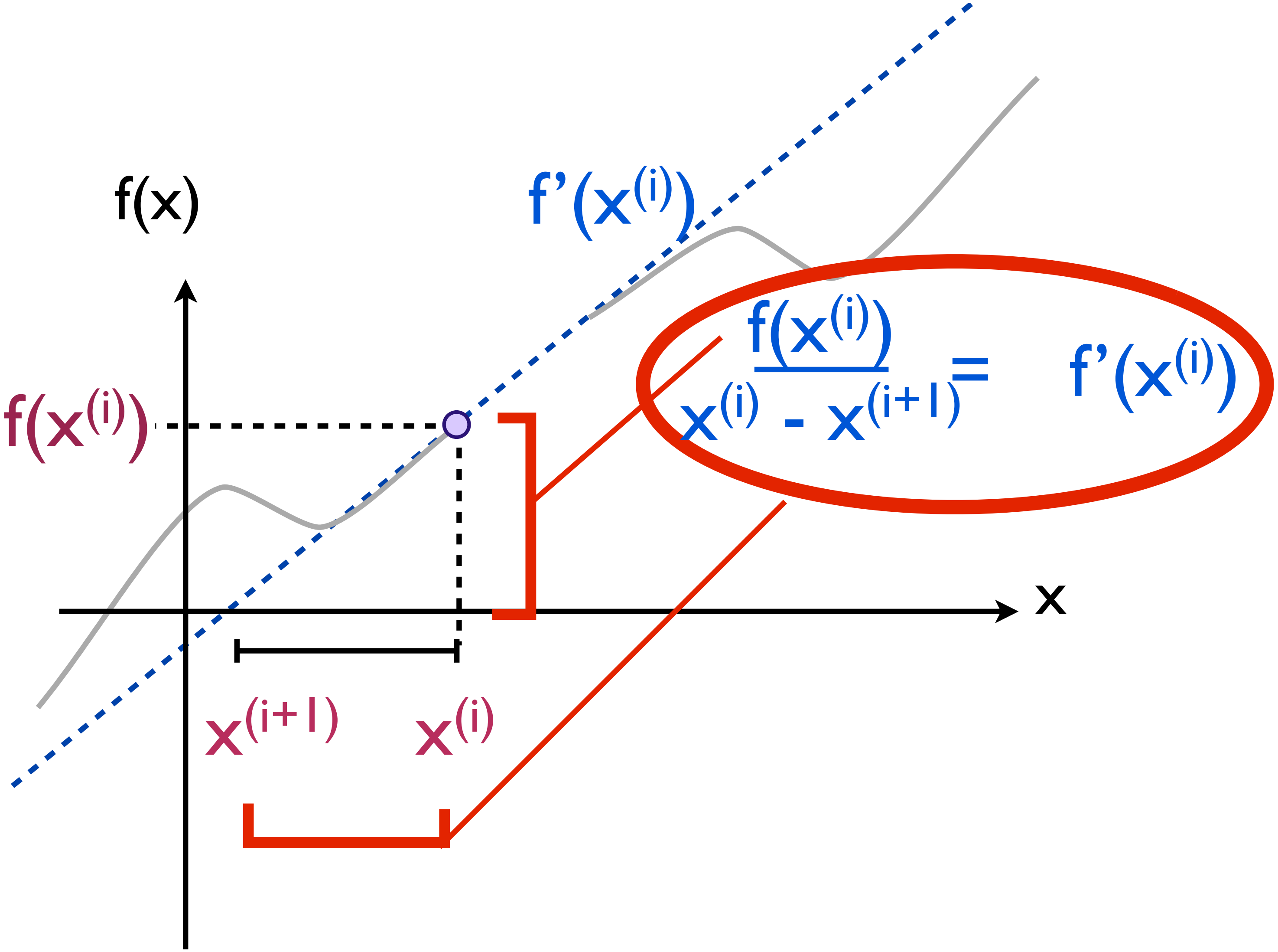
so let's find the change in x

What variable values do we have, and what do we want to solve for again?

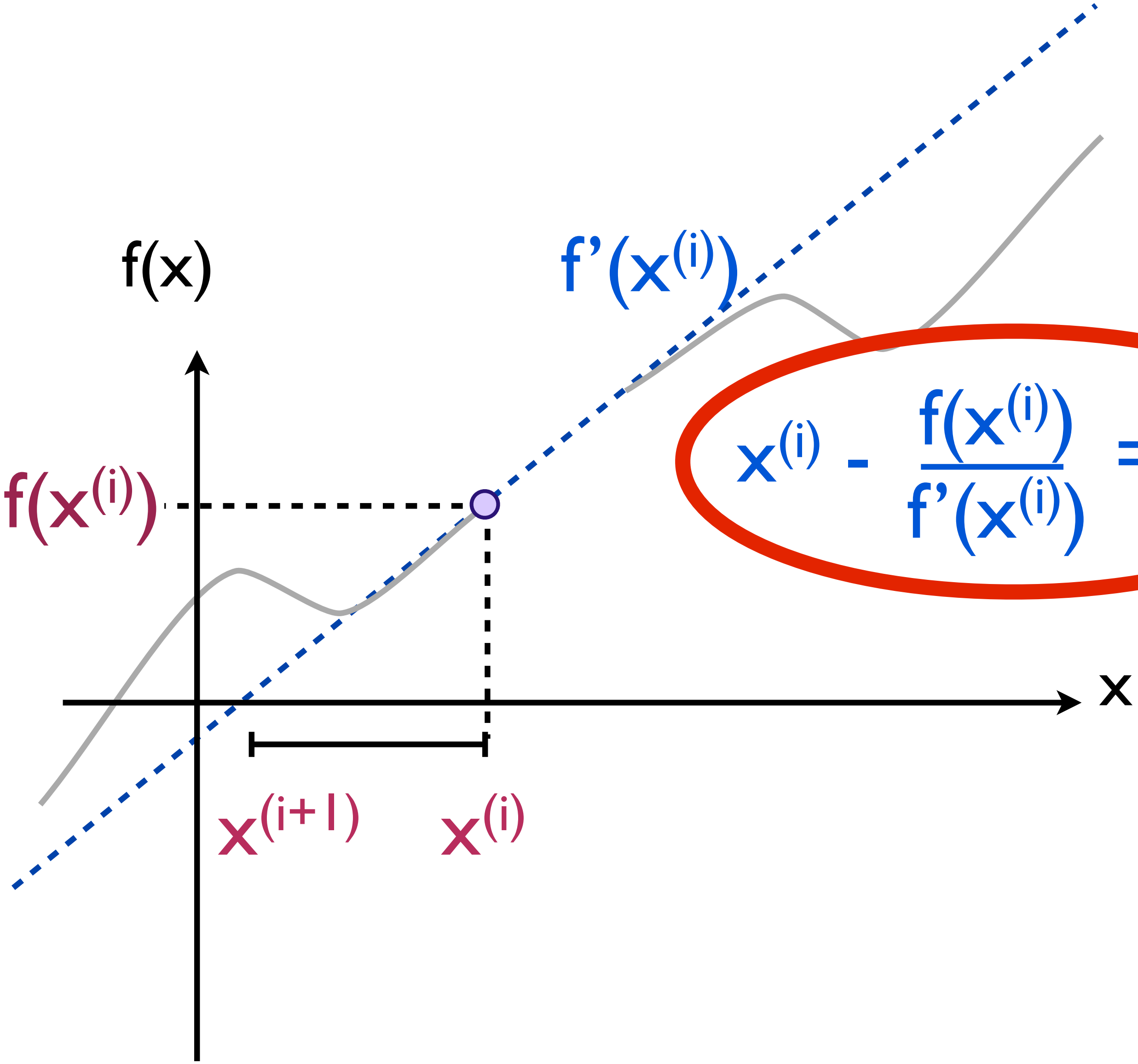


so let’s find the change in x

We want $x^{(i+1)}$ - can we rearrange to solve for $x^{(i+1)}$?



so let's find the change in x

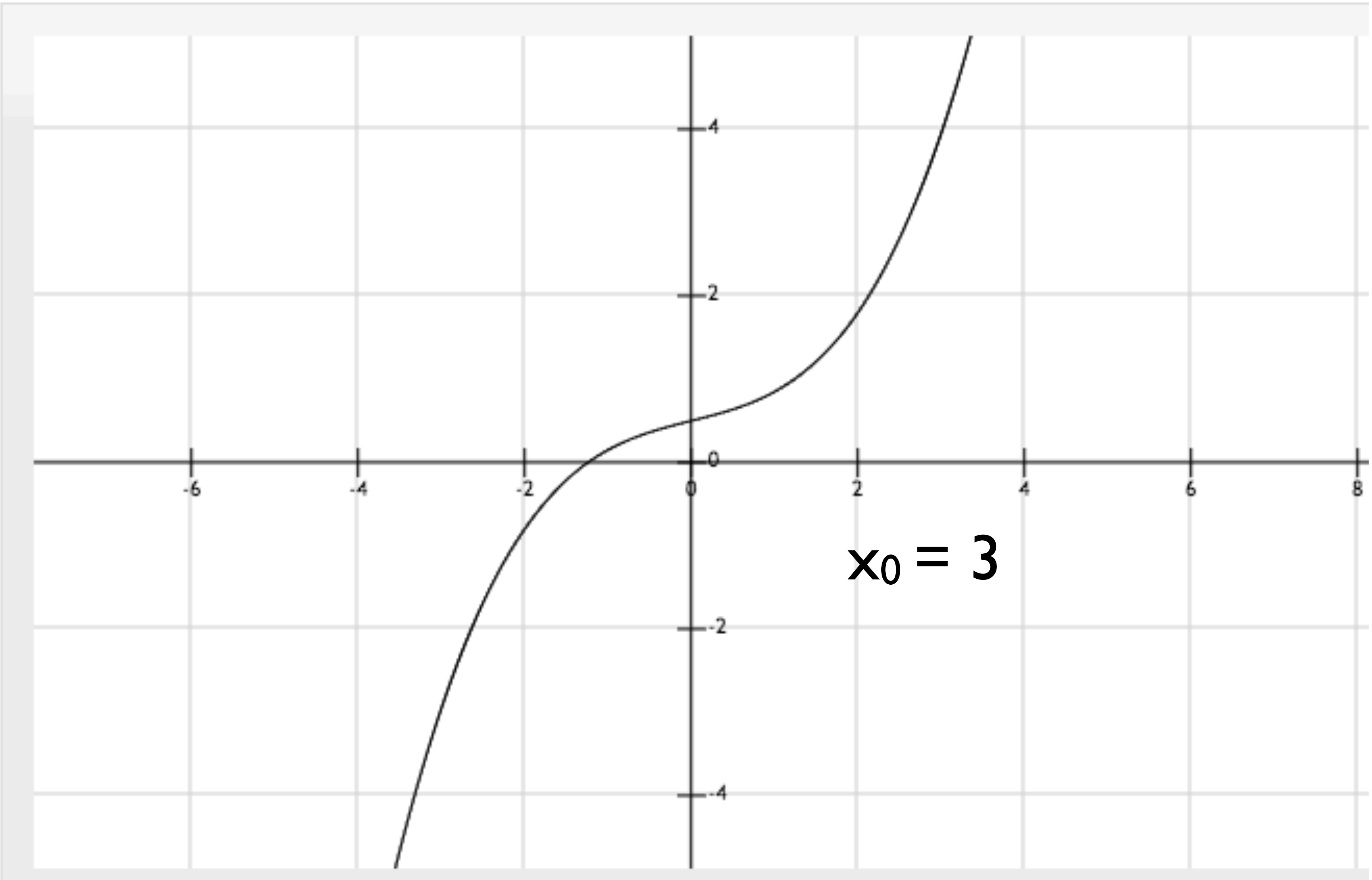


$$x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})} = x^{(i+1)}$$

$$f(x) = 0.1x^3 + 0.25x + 0.5$$

$$f'(x) = ?$$

$$x^{(i+1)} = x^{(i)} + \frac{-f(x^{(i)})}{f'(x^{(i)})}$$



$$f(x) = 0.1x^3 + 0.25x + 0.5$$

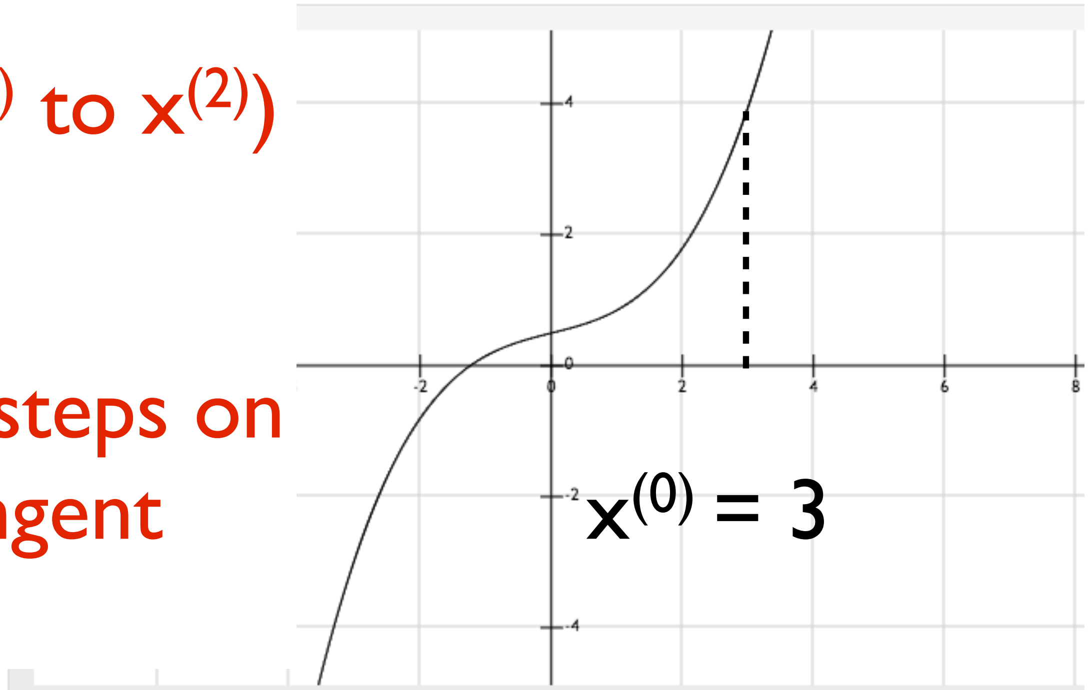
$$f'(x) = 0.3x^2 + 0.25$$

$$x^{(i+1)} = x^{(i)} + \frac{-f(x^{(i)})}{f'(x^{(i)})}$$

Step 1: write pseudo-code that implements this iterative "loop" for 10 iterations

Step 2: do 2 steps ($x^{(0)}$ to $x^{(2)}$) by hand

Step 3: draw the two steps on this graph (including tangent lines)

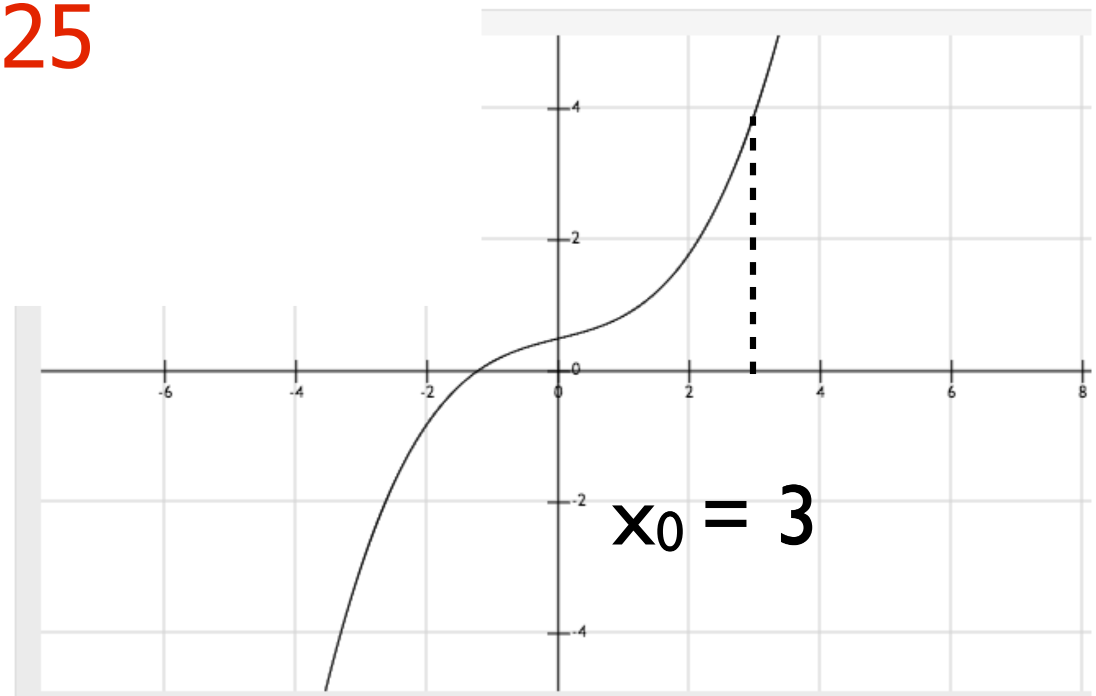


$$f(x) = 0.1x^3 + 0.25x + 0.5$$

$$f'(x) = 0.3x^2 + 0.25$$

$$x^{(i+1)} = x^{(i)} + \frac{-f(x^{(i)})}{f'(x^{(i)})}$$

```
double xi = 3
for i = 1 to 10 {
  fx = 0.1*xi^3 + 0.25*xi + 0.5
  f'x = 0.3*xi^2 + 0.25
  xi = xi - (fx / f'x)
}
```

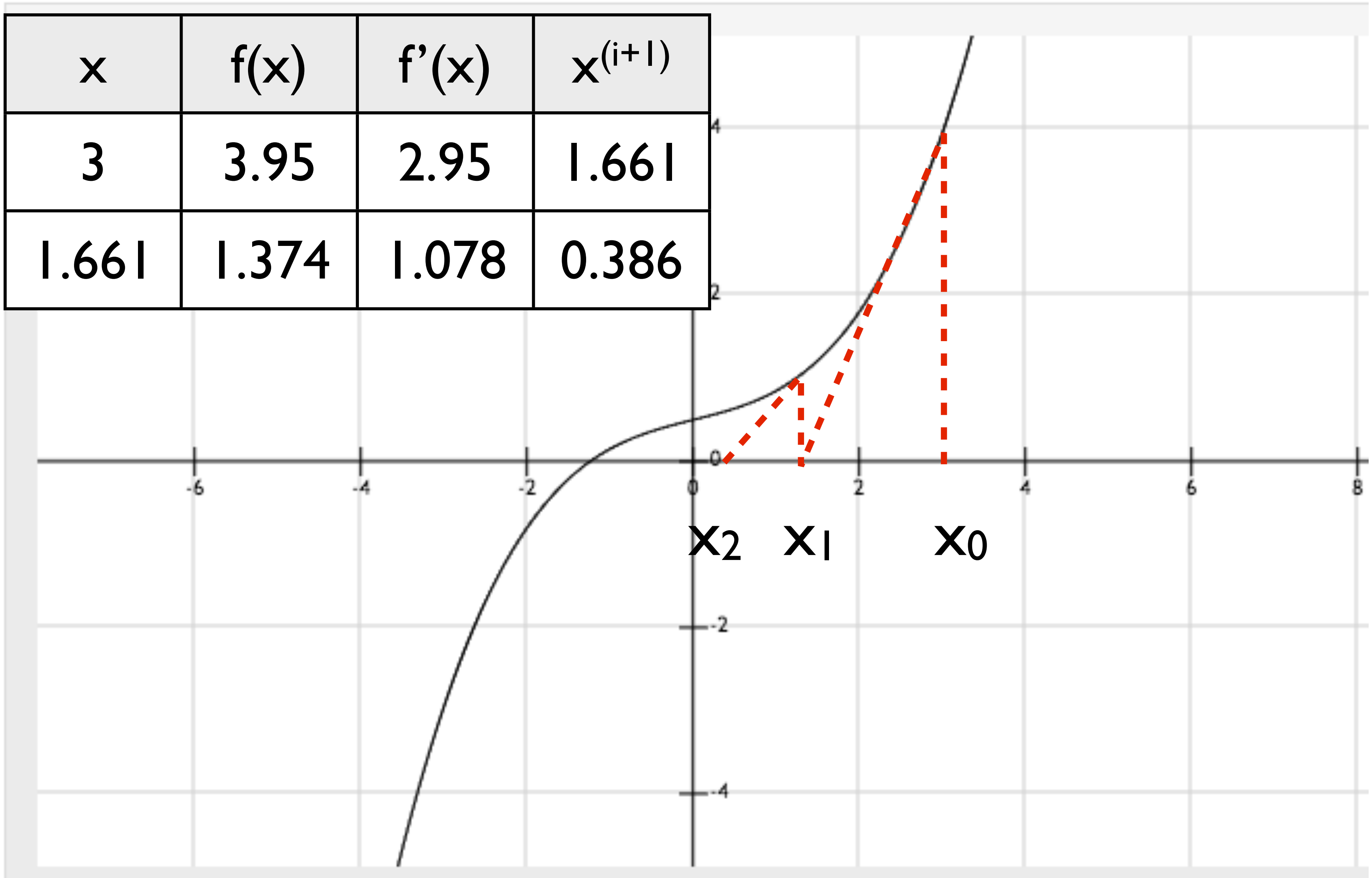


$$f(x) = 0.1x^3 + 0.25x + 0.5$$

$$f'(x) = 0.3x^2 + 0.25$$

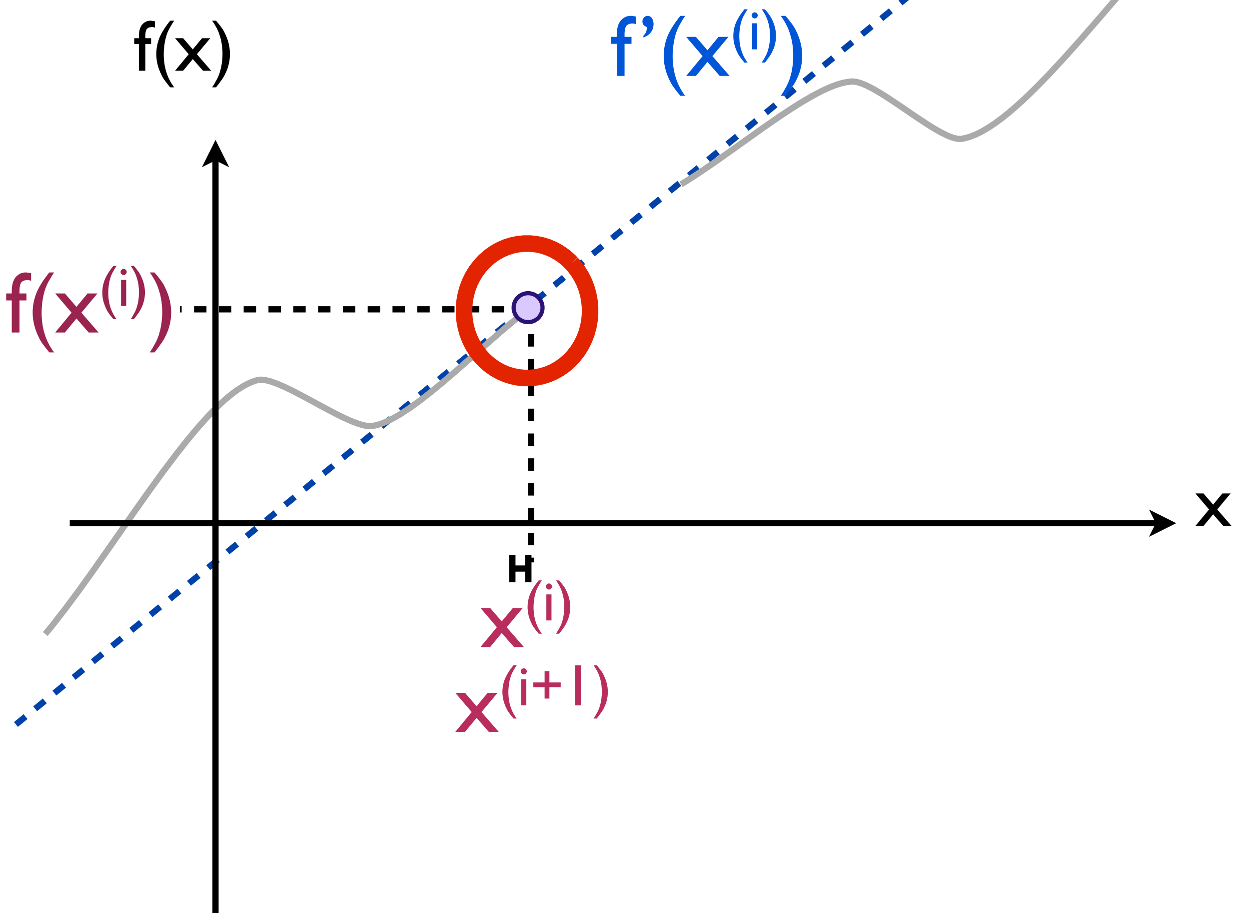
$$x^{(i+1)} = x^{(i)} + \frac{-f(x^{(i)})}{f'(x^{(i)})}$$

x	$f(x)$	$f'(x)$	$x^{(i+1)}$
3	3.95	2.95	1.661
1.661	1.374	1.078	0.386

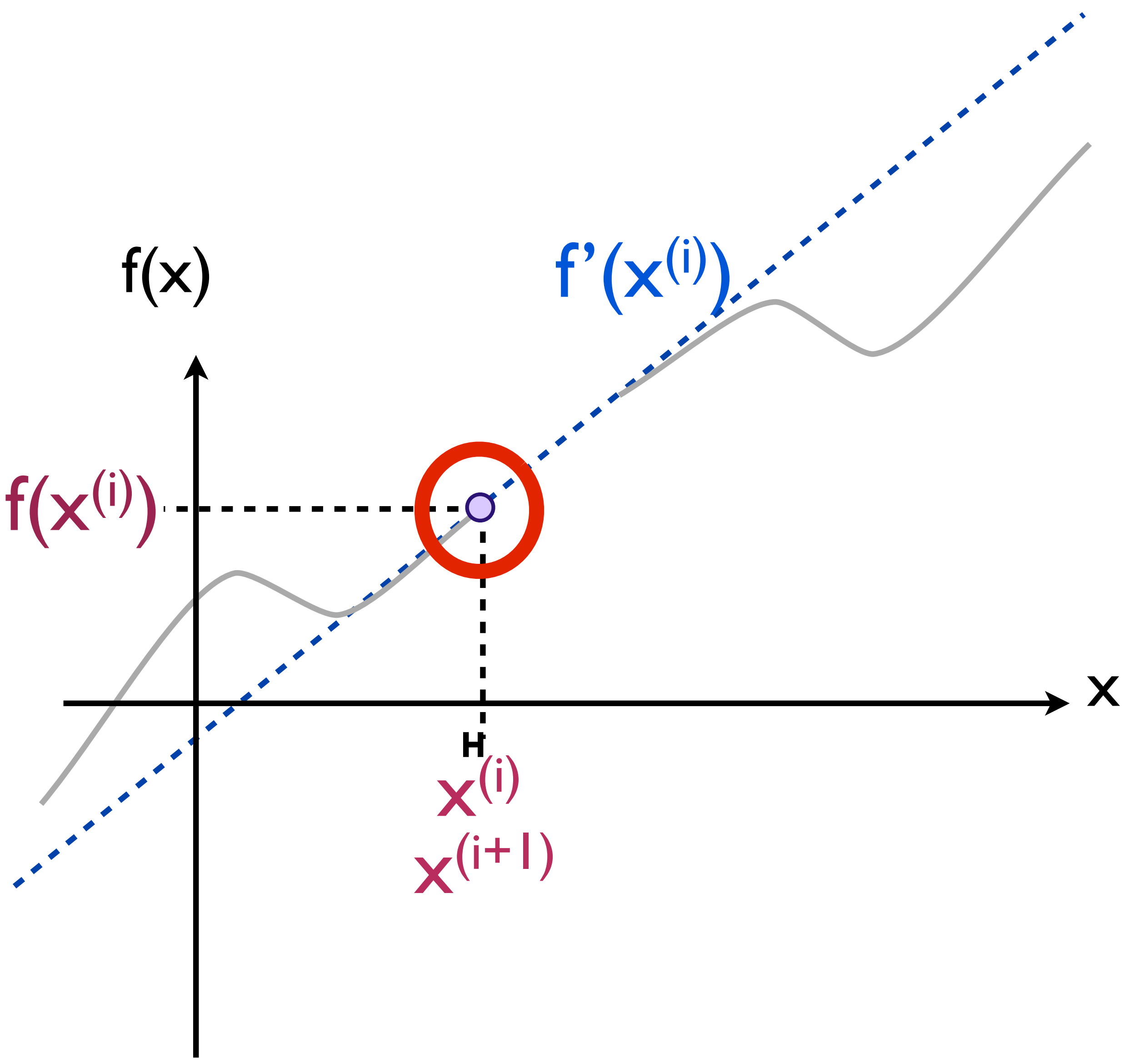


another way to think about it...

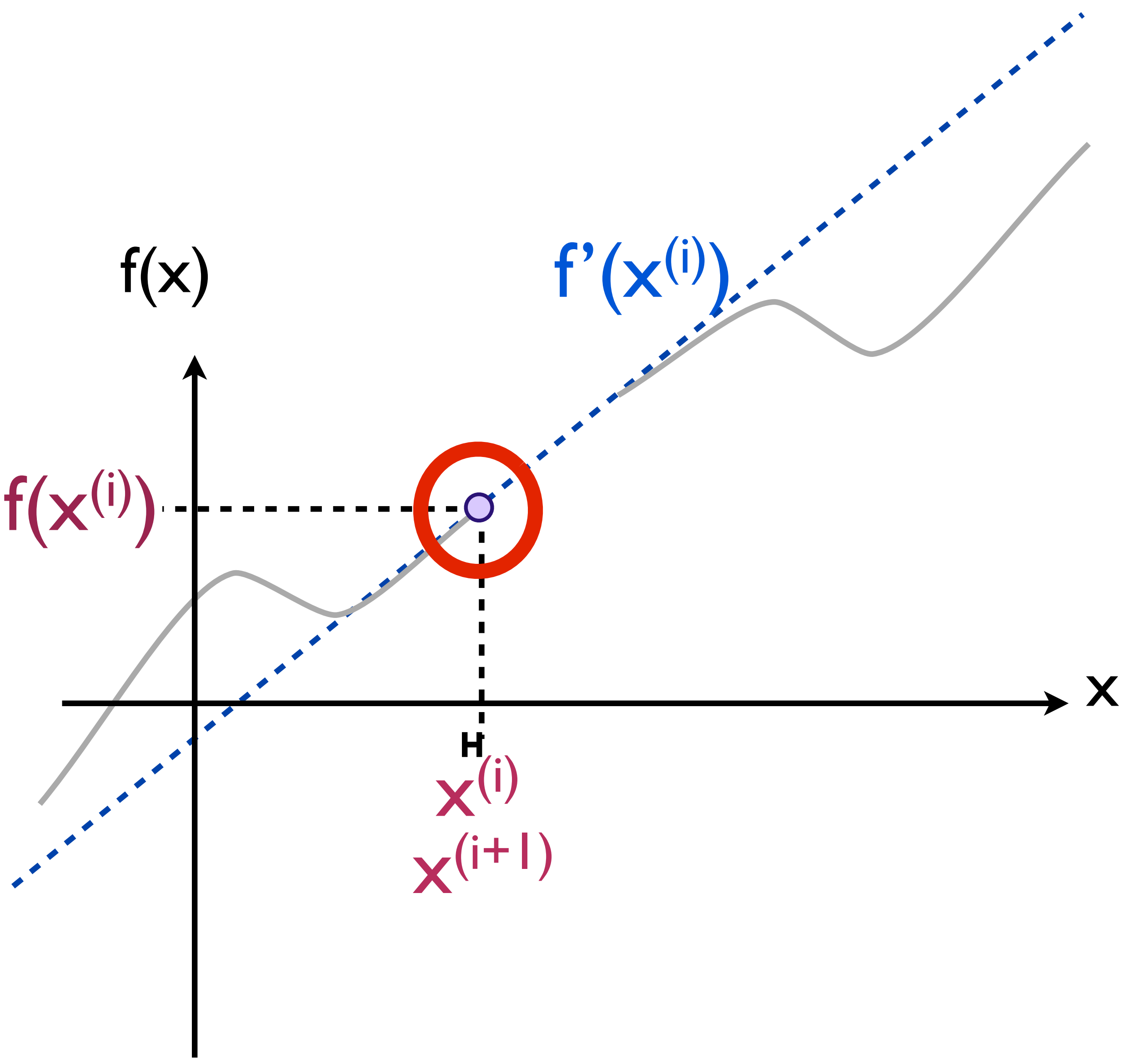
suppose we only move from x(i) by a **really** tiny amount...



$(x^{(i+1)} - x^{(i)})$ is **almost** zero

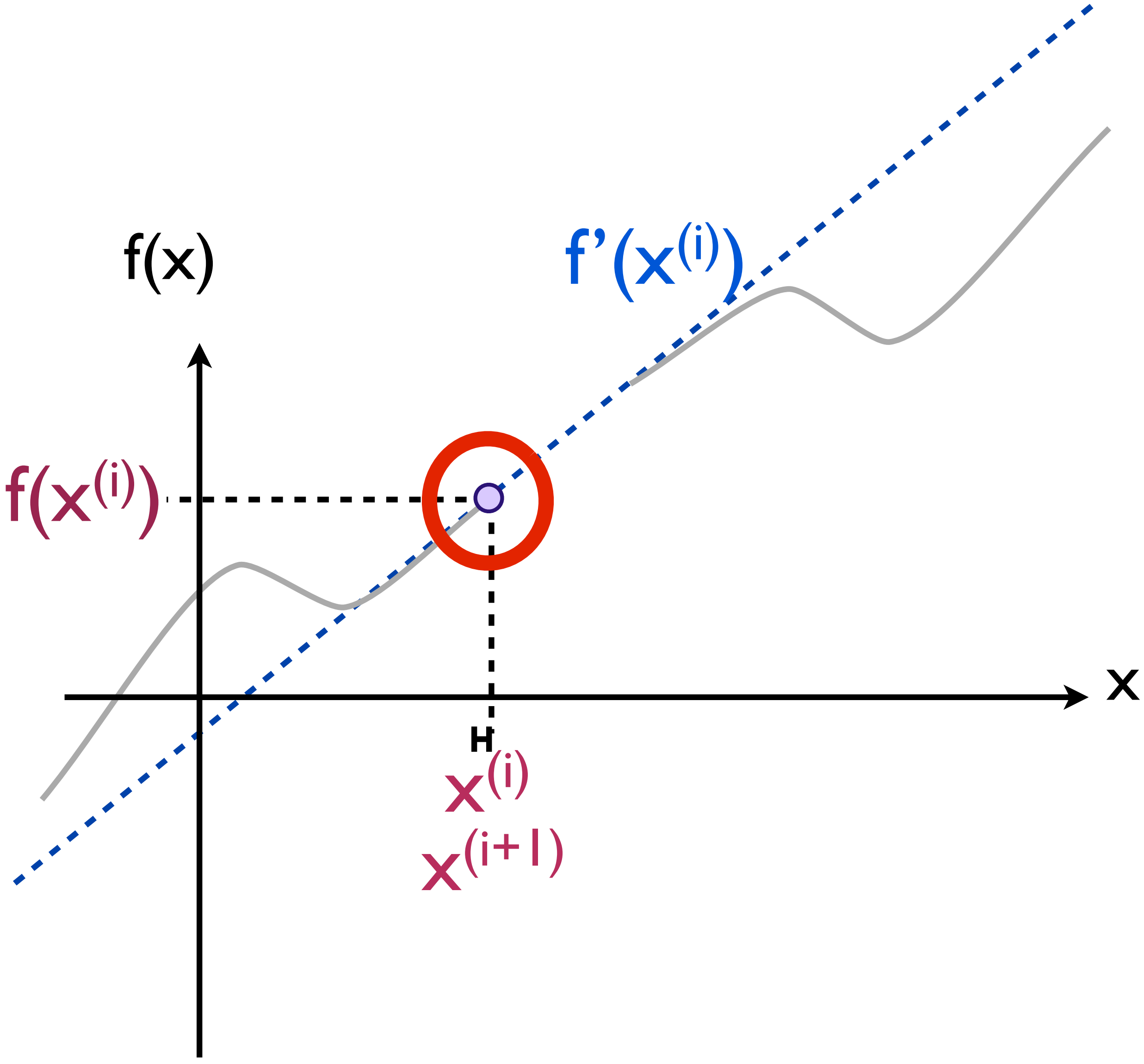


$f'(x^{(i)})(x^{(i+1)} - x^{(i)})$ is also **almost** zero



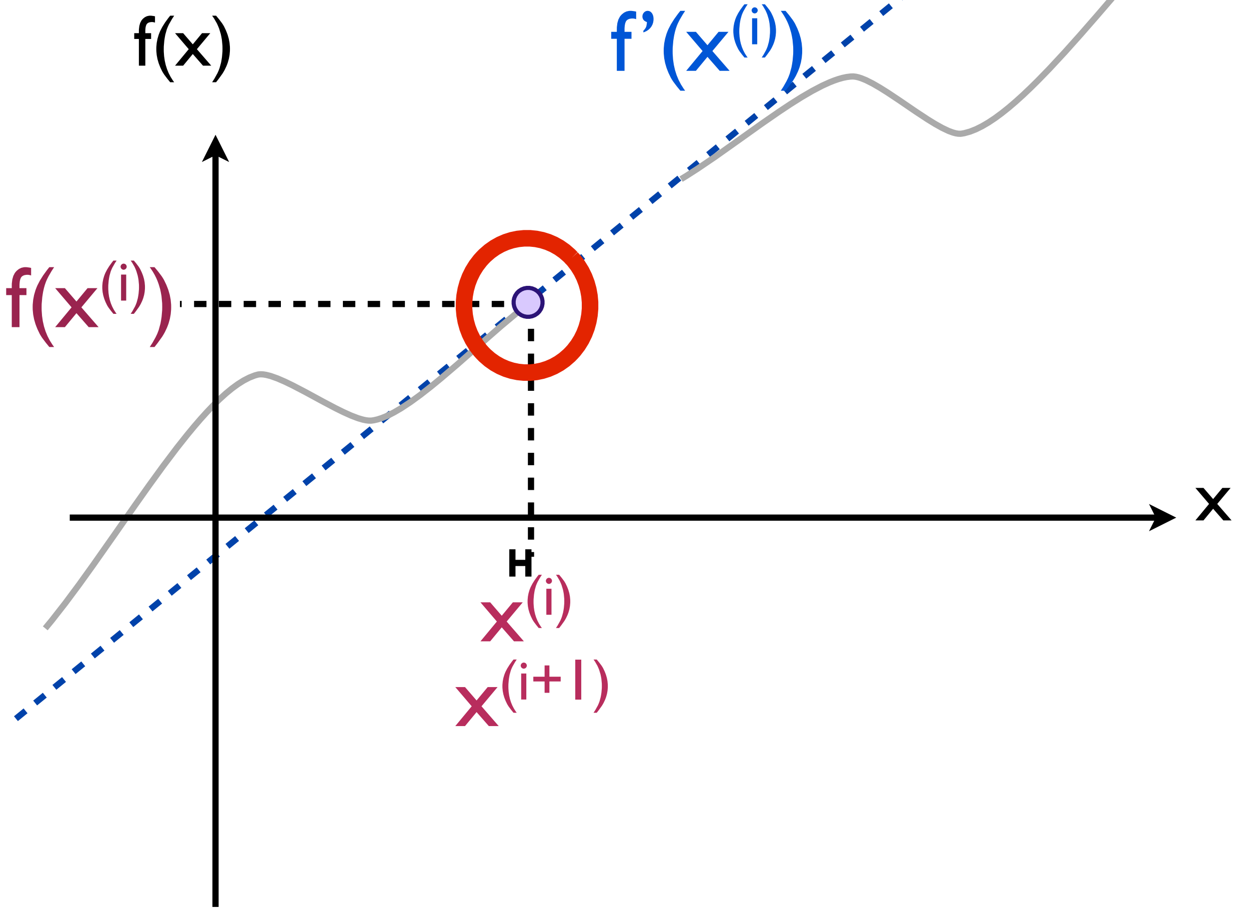
$$f'(x^{(i)})(x^{(i+1)} - x^{(i)}) + f(x^{(i)})$$

is also **almost** equal to what?



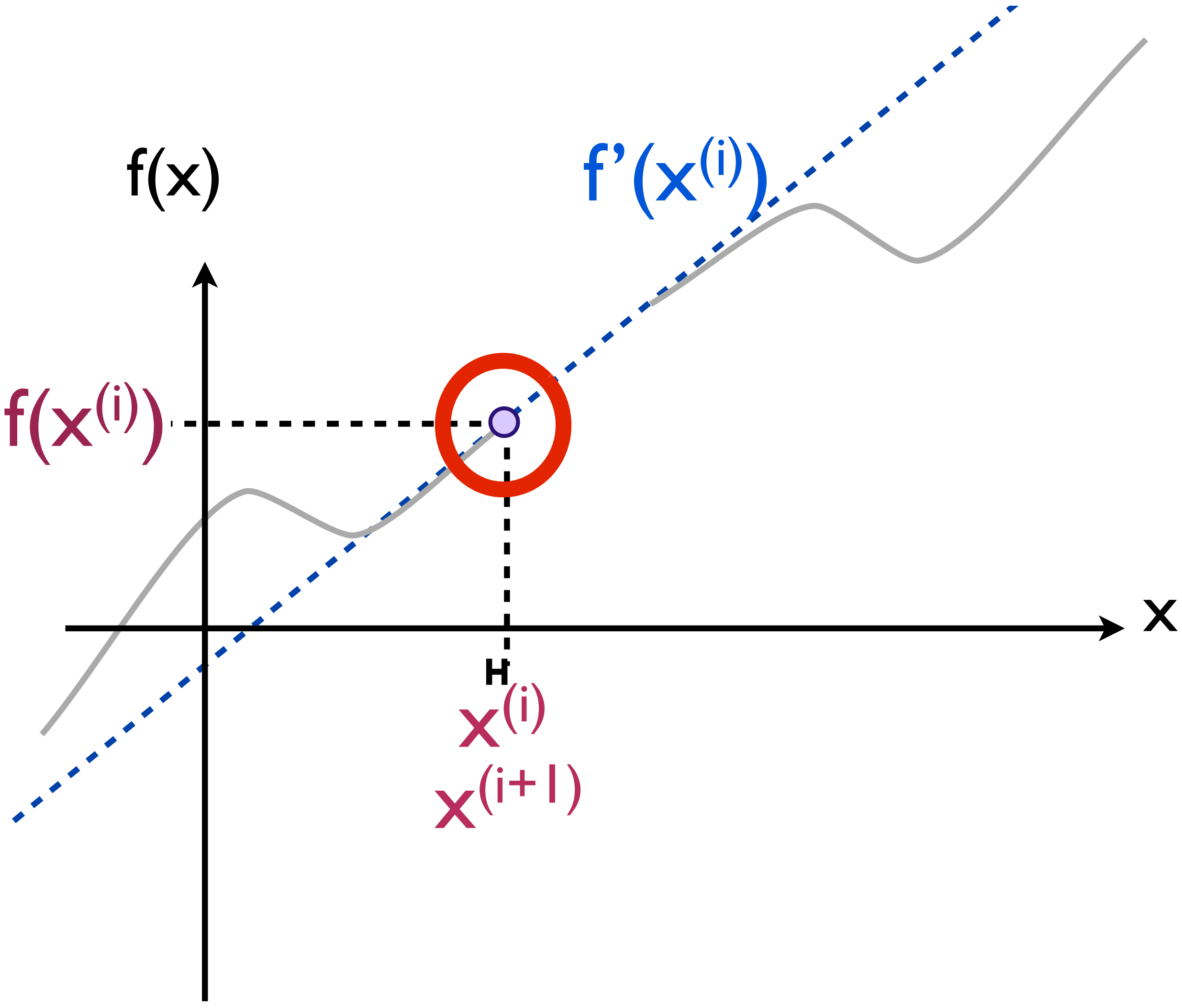
$f'(x^{(i)})(x^{(i+1)} - x^{(i)}) + f(x^{(i)})$

so, for **tiny** change in x, this is a decent approximation of our function at $f(x^{(i+1)})$



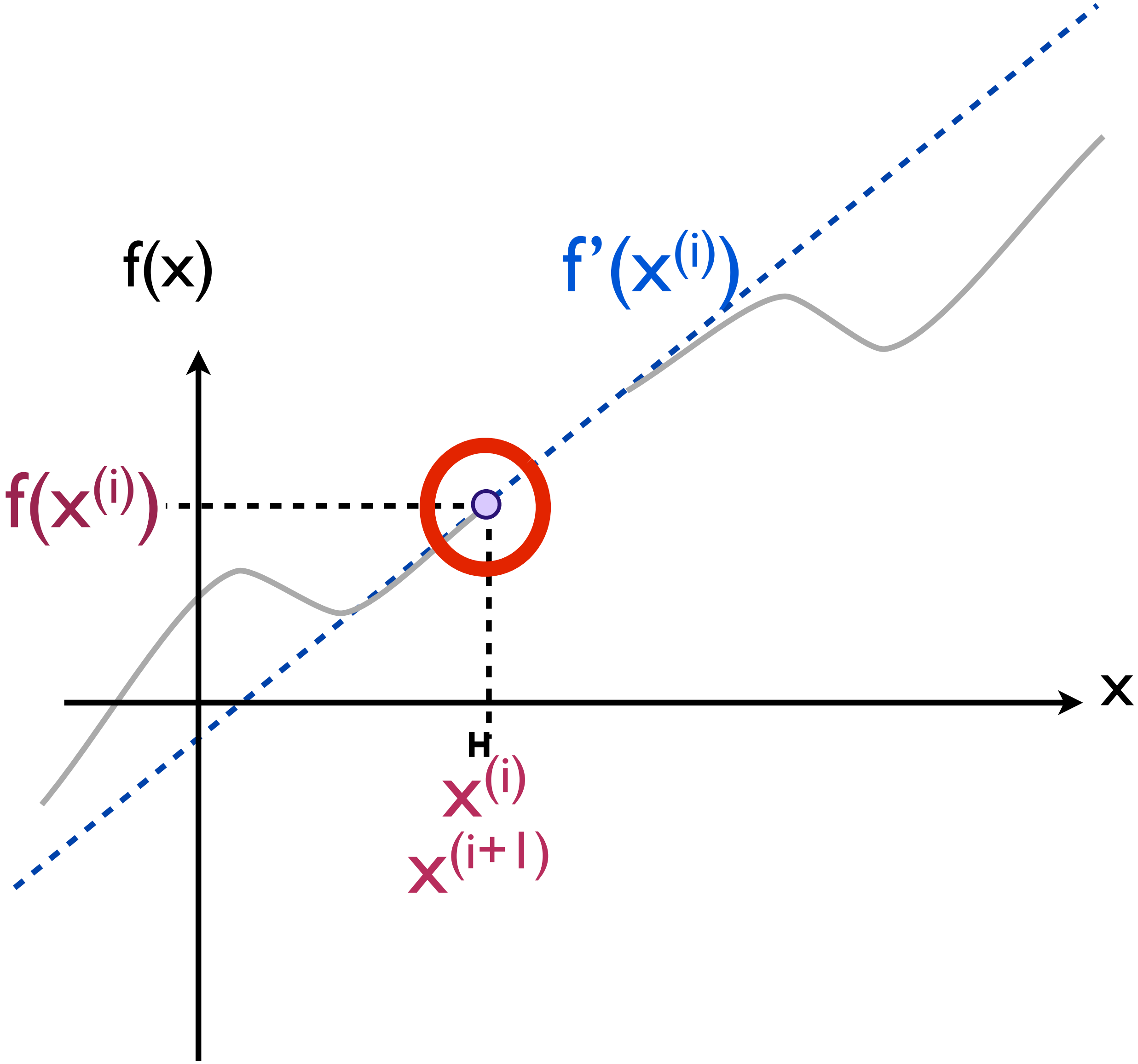
$$\underline{f'(x^{(i)})(x^{(i+1)} - x^{(i)})} + f(x^{(i)})$$

...and ***if*** our function was linear, it would perfectly describe our function for all change in x



suppose our function was linear: $y = mx + c$

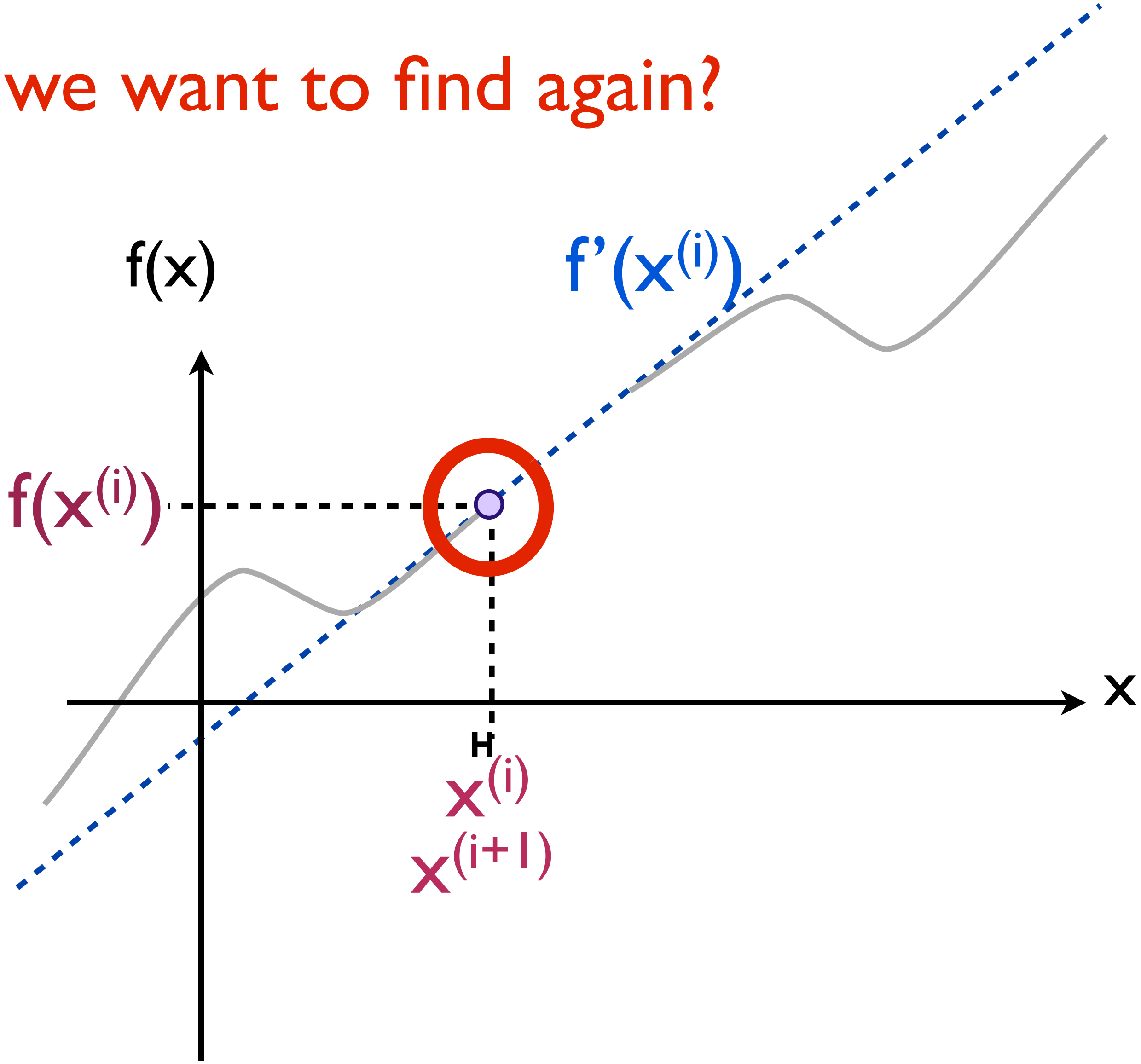
$$f(x^{(i+1)}) = f'(x^{(i)})(x^{(i+1)} - x^{(i)}) + f(x^{(i)})$$



suppose our function was linear: $y = mx + c$

$$f(x^{(i+1)}) = f'(x^{(i)})(x^{(i+1)} - x^{(i)}) + f(x^{(i)})$$

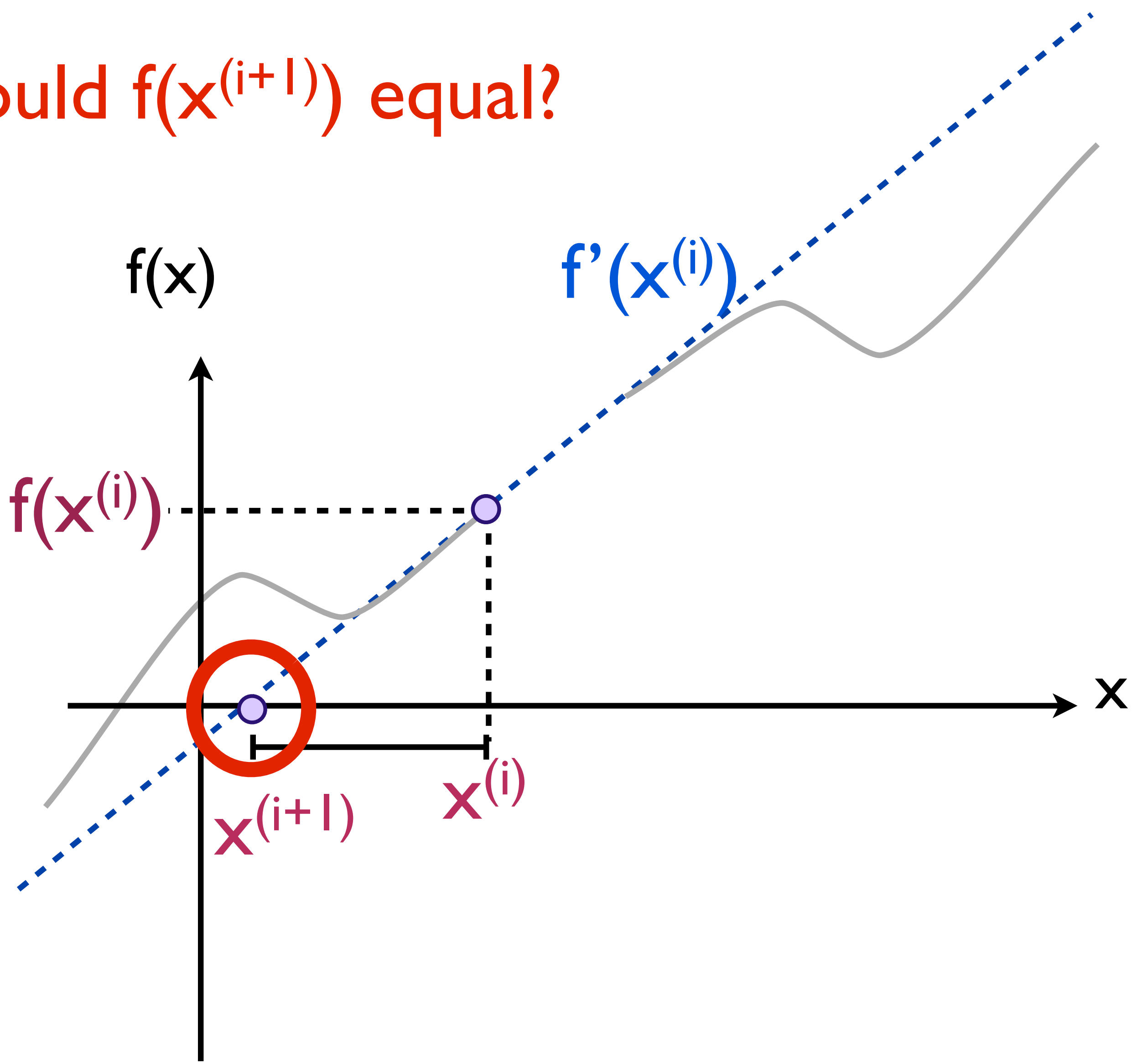
what do we want to find again?



suppose our function was linear: $y = mx + c$

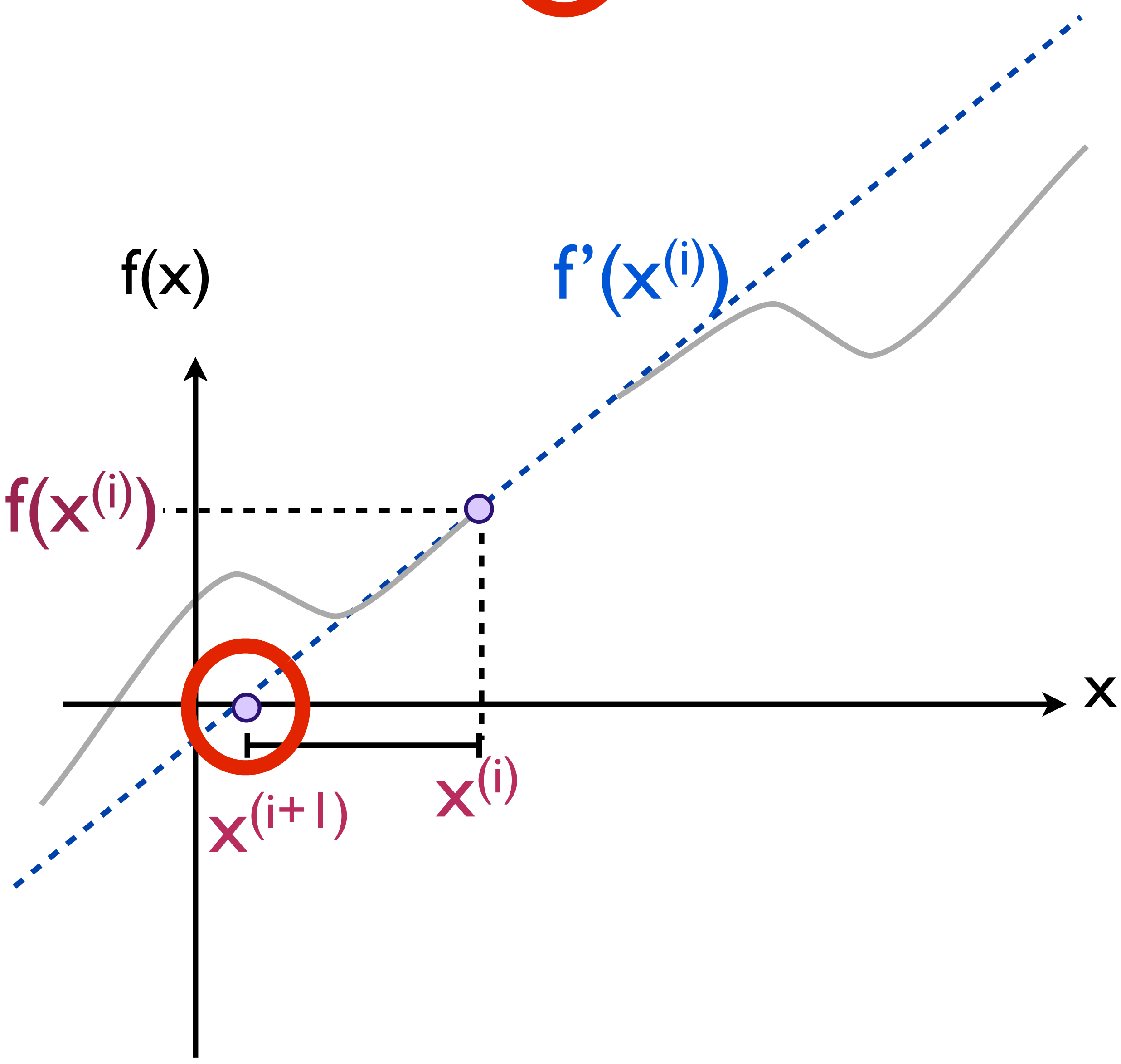
$$f(x^{(i+1)}) = f'(x^{(i)})(x^{(i+1)} - x^{(i)}) + f(x^{(i)})$$

what should $f(x^{(i+1)})$ equal?



suppose our function was linear: $y = mx + c$

$$0 = f'(x^{(i)})(x^{(i+1)} - x^{(i)}) + f(x^{(i)})$$



First-order Taylor series expansion around $x^{(i)}$:

$$f_{linear}(x) \equiv f'(x^{(i)})\,(x - x^{(i)}) + f(x^{(i)})$$

Find root of this linearised system:

$$f_{linear}(x^{(i+1)}) \equiv f'(x^{(i)})\,(x^{(i+1)} - x^{(i)}) + f(x^{(i)}) = 0$$

Rearrange to solve for $x^{(i+1)}$:

$$x^{(i+1)} = x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})}$$

when do we stop?

in your pseudo-code we iterated for 10 loops

but what other stopping conditions could we check for?

```
double xi = 3
for i = 1 to 10 {
  fx = 0.1*xi^3 + 0.25*xi + 0.5
  f'x = 0.3*xi^2 + 0.25
  xi = xi - (fx / f'x)
}
```

when do we stop?

change your pseudo-code to implement **all three** of these:

- **residual size:**
  when function gets "close enough" to root
  $|f(x^{(i)})| < e$
- **increment size:**
  when step size is "small"
  $|x^{(i+1)} - x^{(i)}| < e$
- **number of iterations:**
  when done "too many" steps
  $i > i_{max}$

Part 3

Newton's method for higher dimensions

we'll use a nice example from:

Math, Numerics, & Programming (for Mechanical Engineers)
by Masayuki Yano, James Douglass Penn, George Konidaris, Anthony T Patera

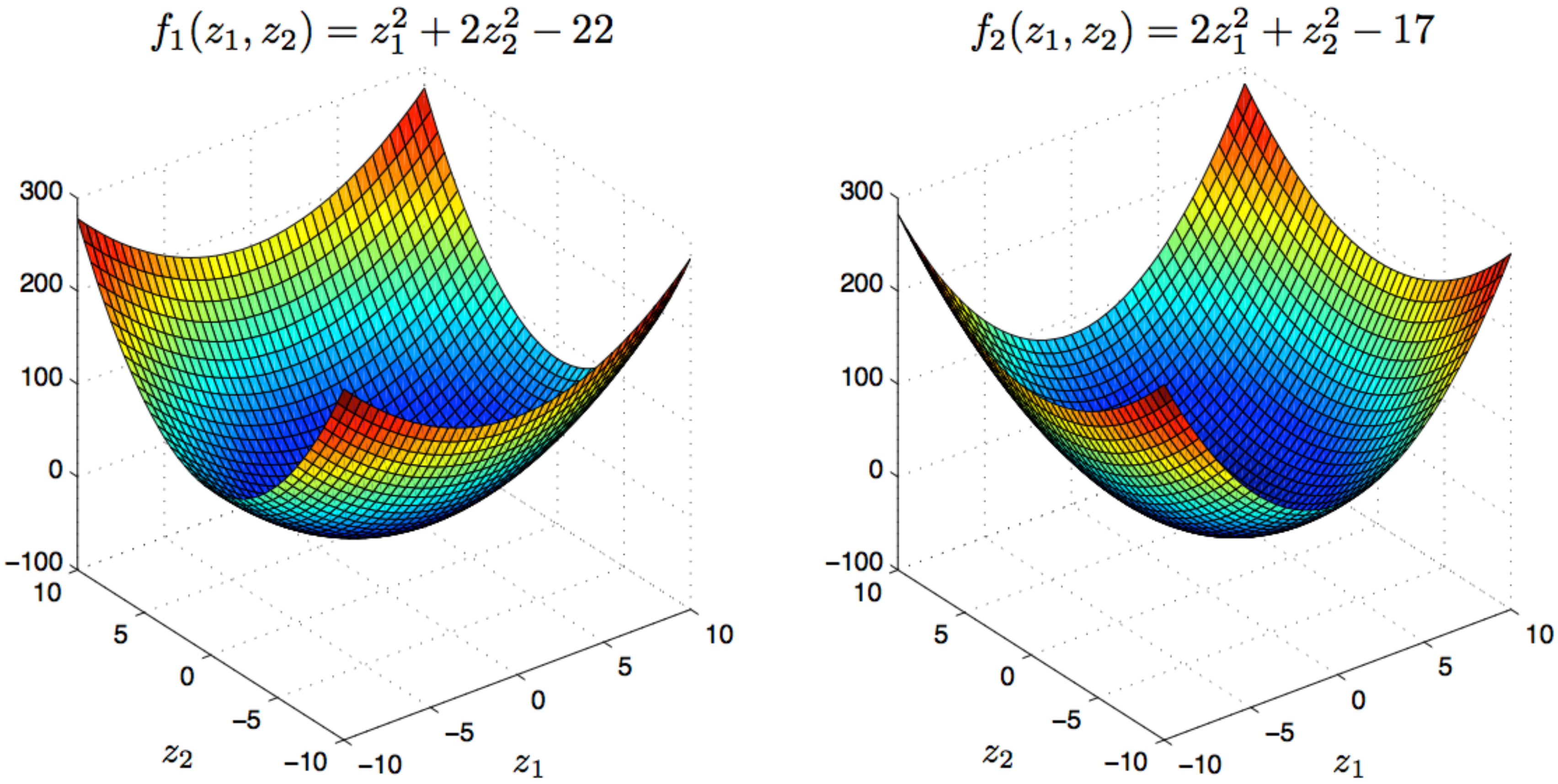
https://ocw.mit.edu/courses/mechanical-engineering/2-086-numerical-computation-for-mechanical-engineers-fall-2012/readings/MIT2_086F12_notes_unit4.pdf say we want to solve this system:

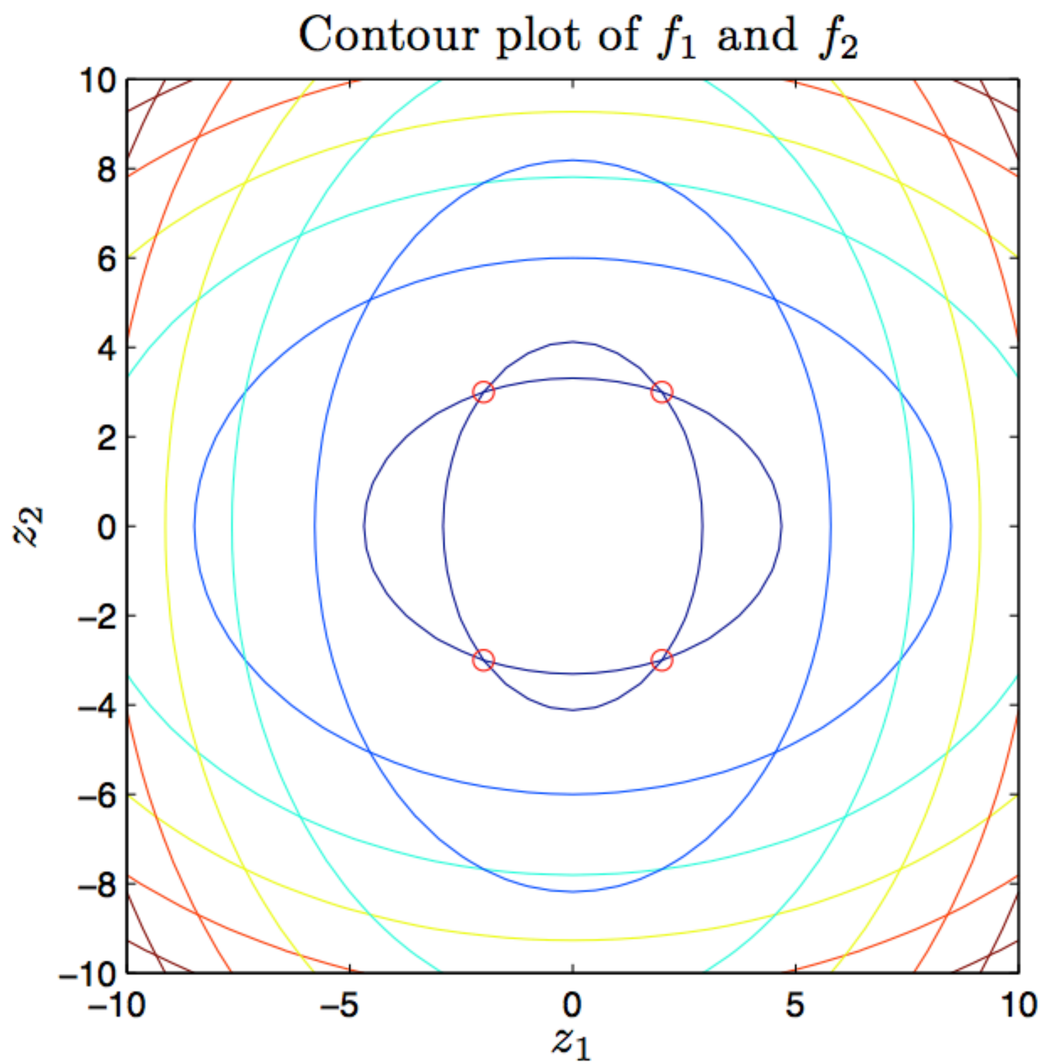
$$\begin{aligned} f_1(z_1, z_2) &= z_1^2 + 2z_2^2 - 22 = 0 \,, \\ f_2(z_1, z_2) &= 2z_1^2 + z_2^2 - 17 = 0 \,. \end{aligned}$$

Newton procedure is same:

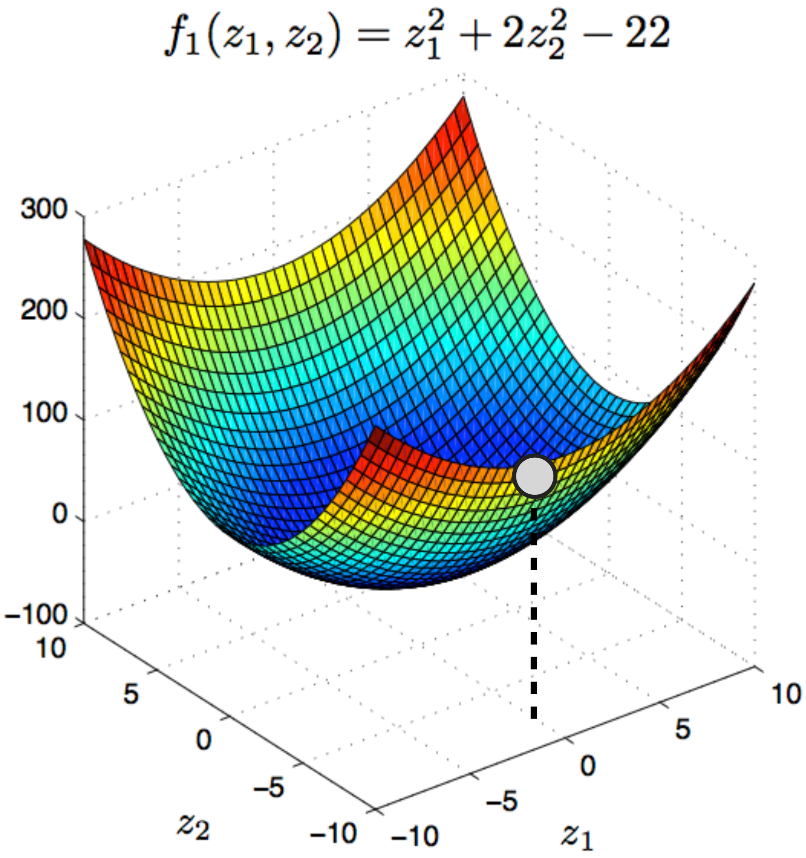
1. guess initial value $X^{(0)}$
2. make linear approximation at $X^{(0)}$
3. solve linearised system to get better guess $X^{(1)}$
4. repeat from step 2 with $X^{(1)}$

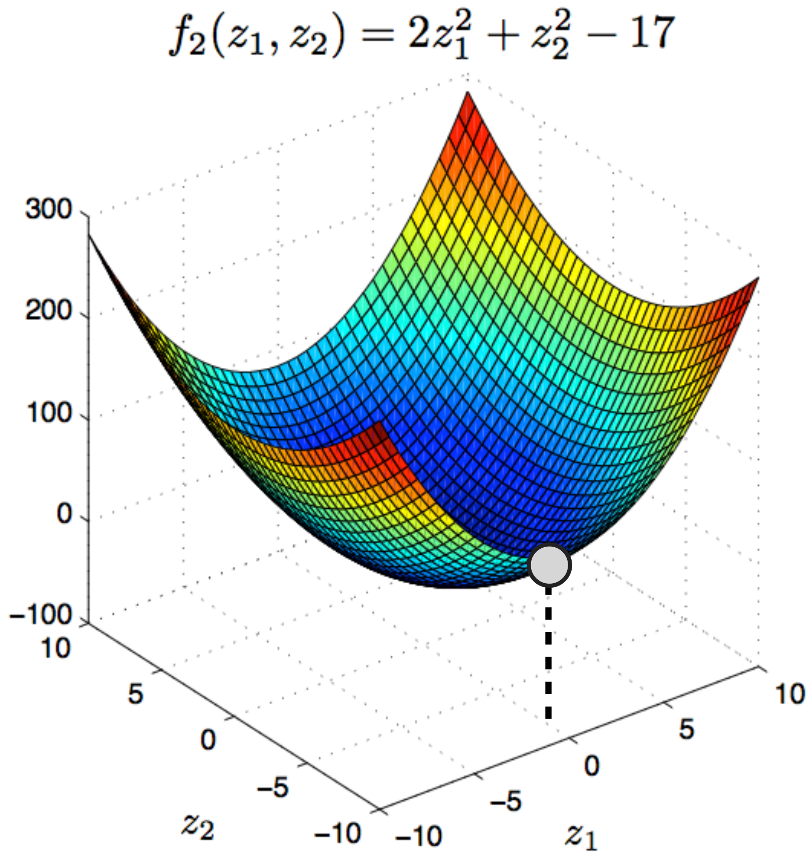
$$
\begin{aligned}
f_1(z_1, z_2) &= z_1^2 + 2z_2^2 - 22 = 0\,, \\
f_2(z_1, z_2) &= 2z_1^2 + z_2^2 - 17 = 0\,.
\end{aligned}
$$





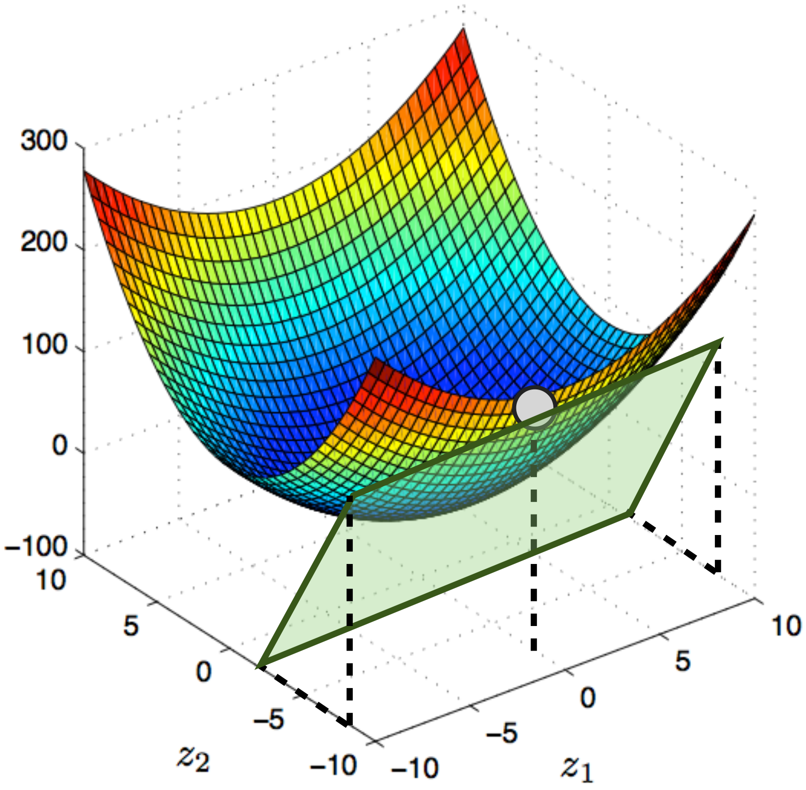
Contour plot of f_1 and f_2
z_2
z_1
10
8
6
4
2
0
−2
−4
−6
−8
−10
−5
0
5
10



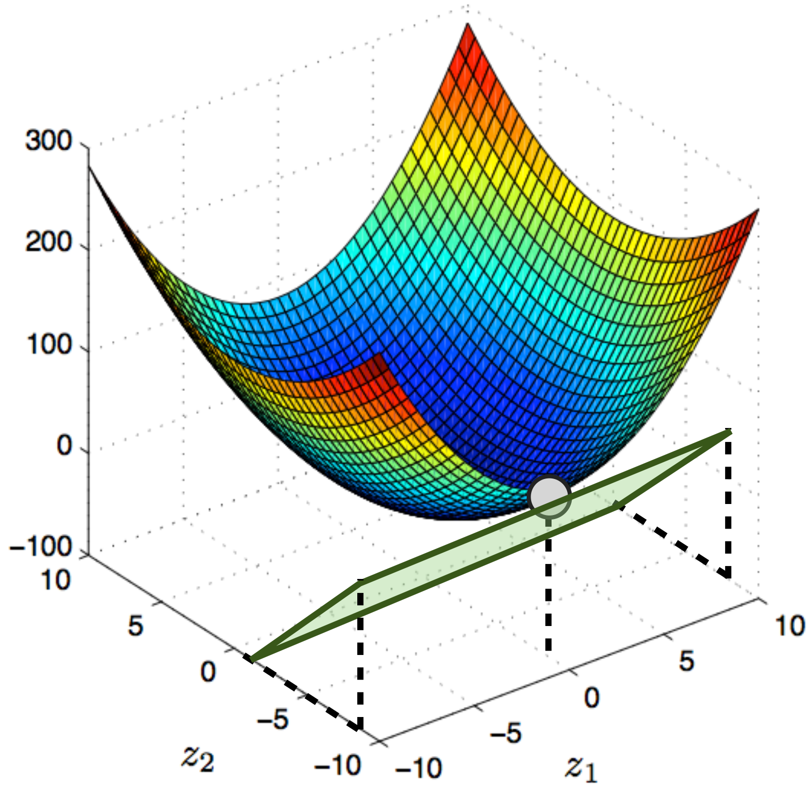


STEP 1. guess a point, e.g. $X^{(0)}=(0,-8)$

$$f_1(z_1, z_2) = z_1^2 + 2z_2^2 - 22$$

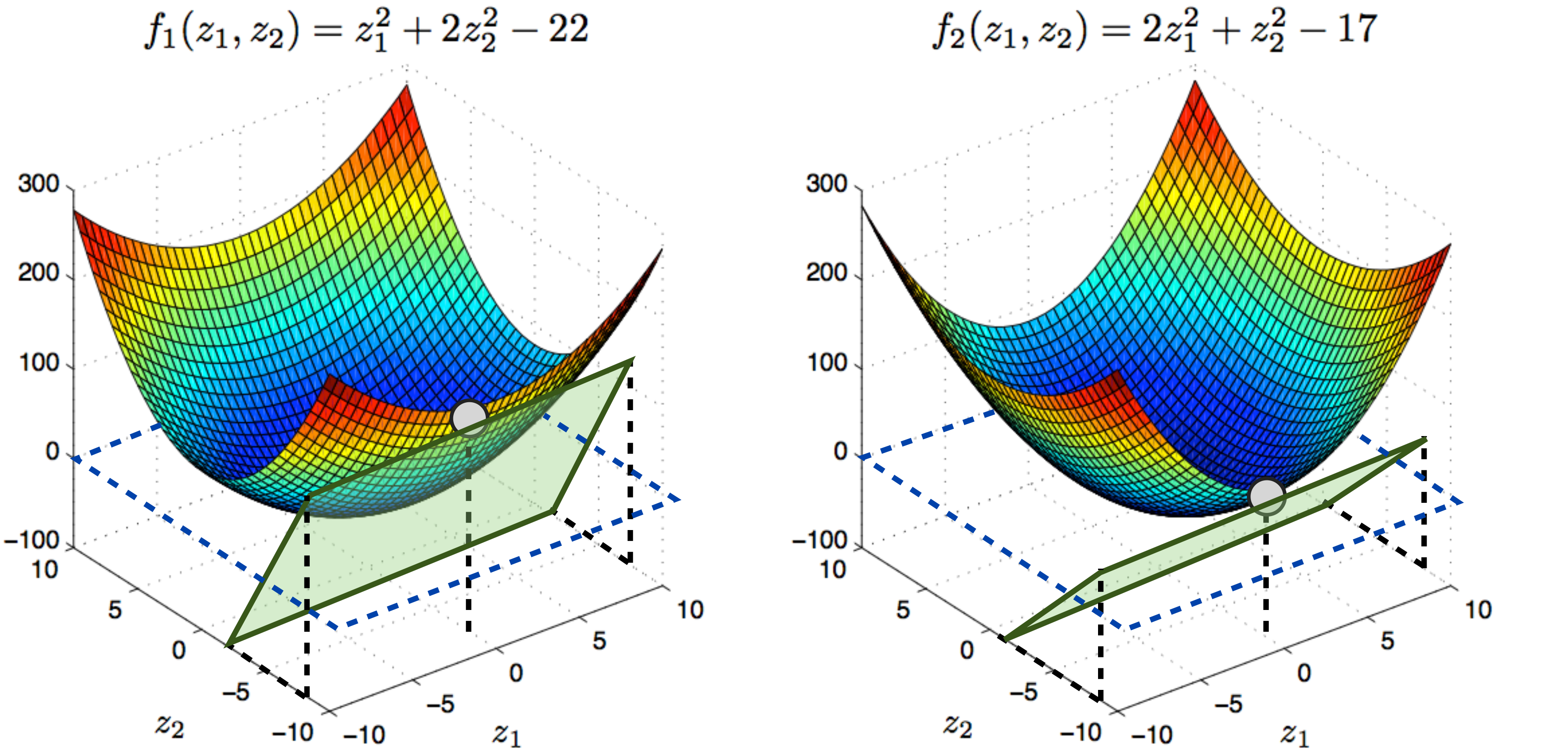


$$f_2(z_1, z_2) = 2z_1^2 + z_2^2 - 17$$

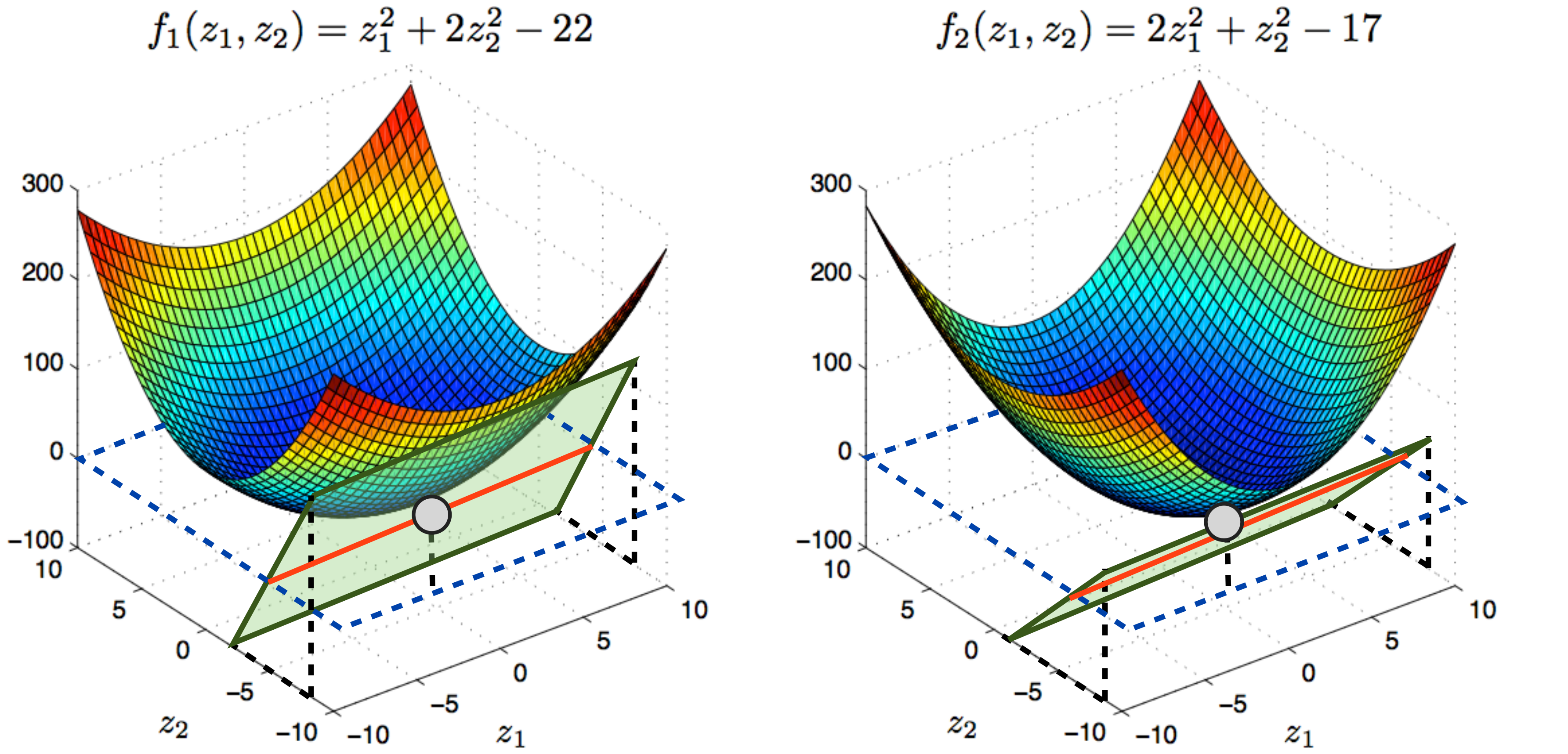


STEP 2. "linearise" function at point $X^{(0)}$=(0,-8)

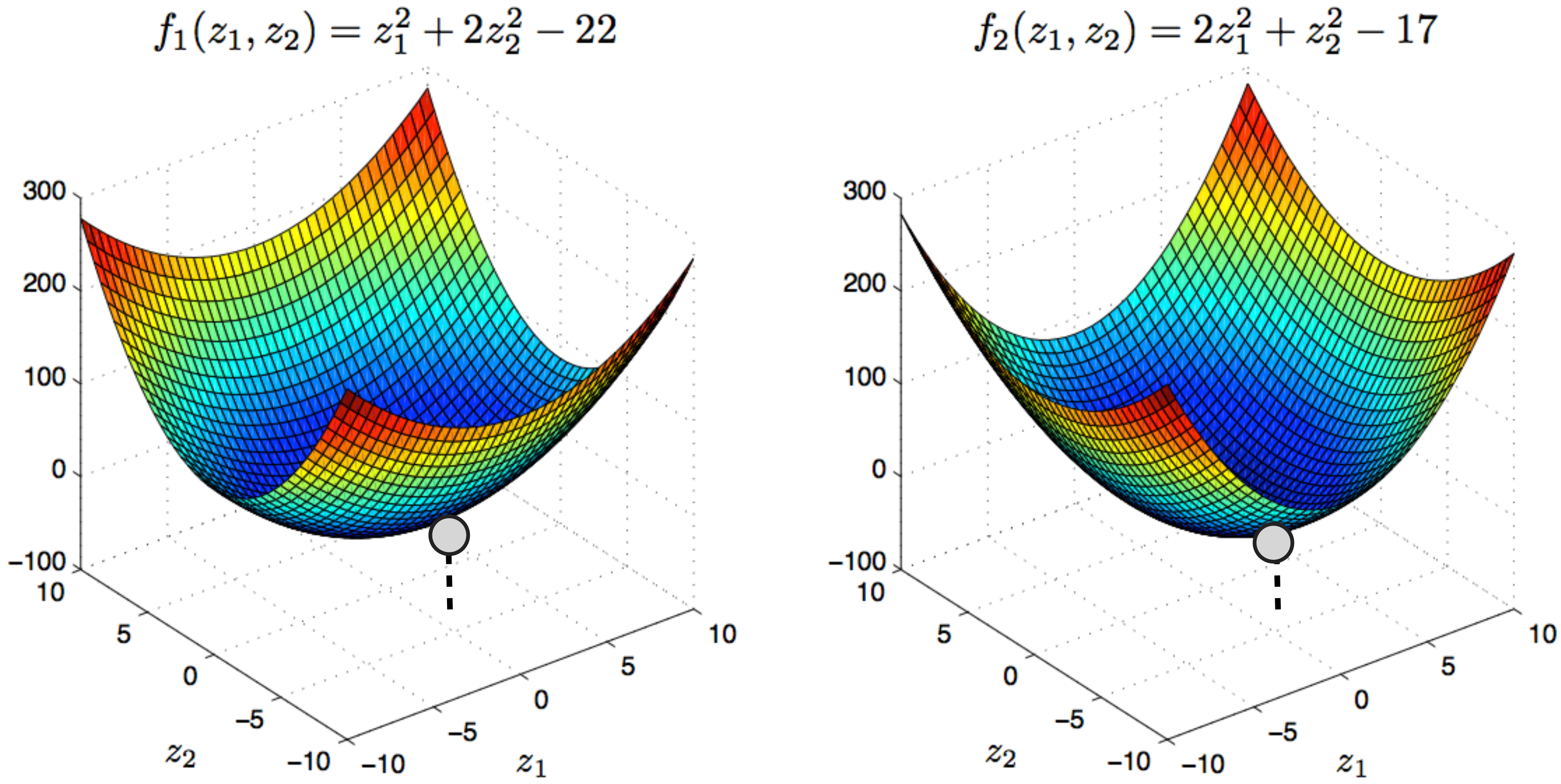
(linear approximation)



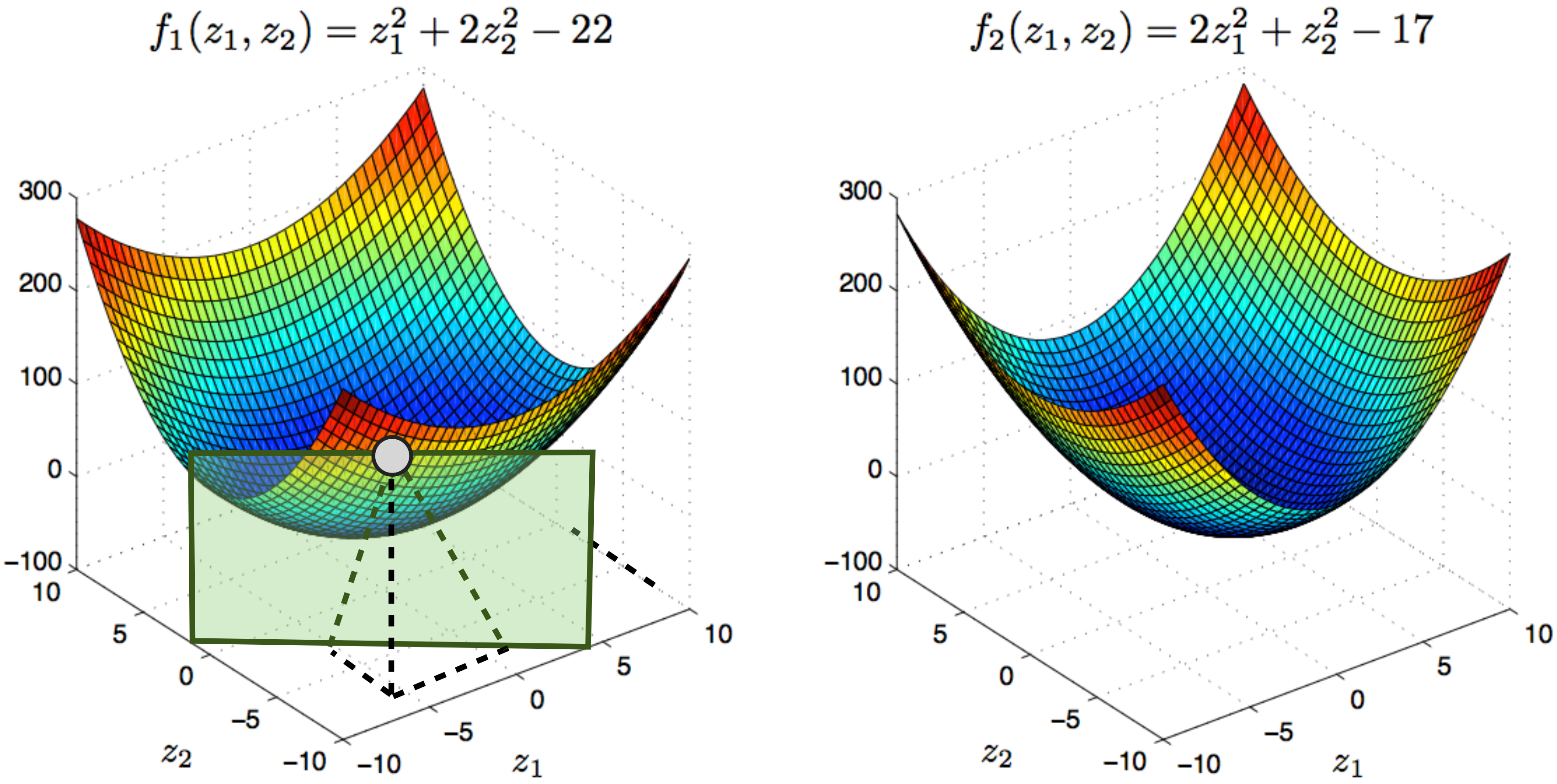
STEP 3. find new $X^{(i+1)}$ where both planes cross 0
(**solve** linearised system)



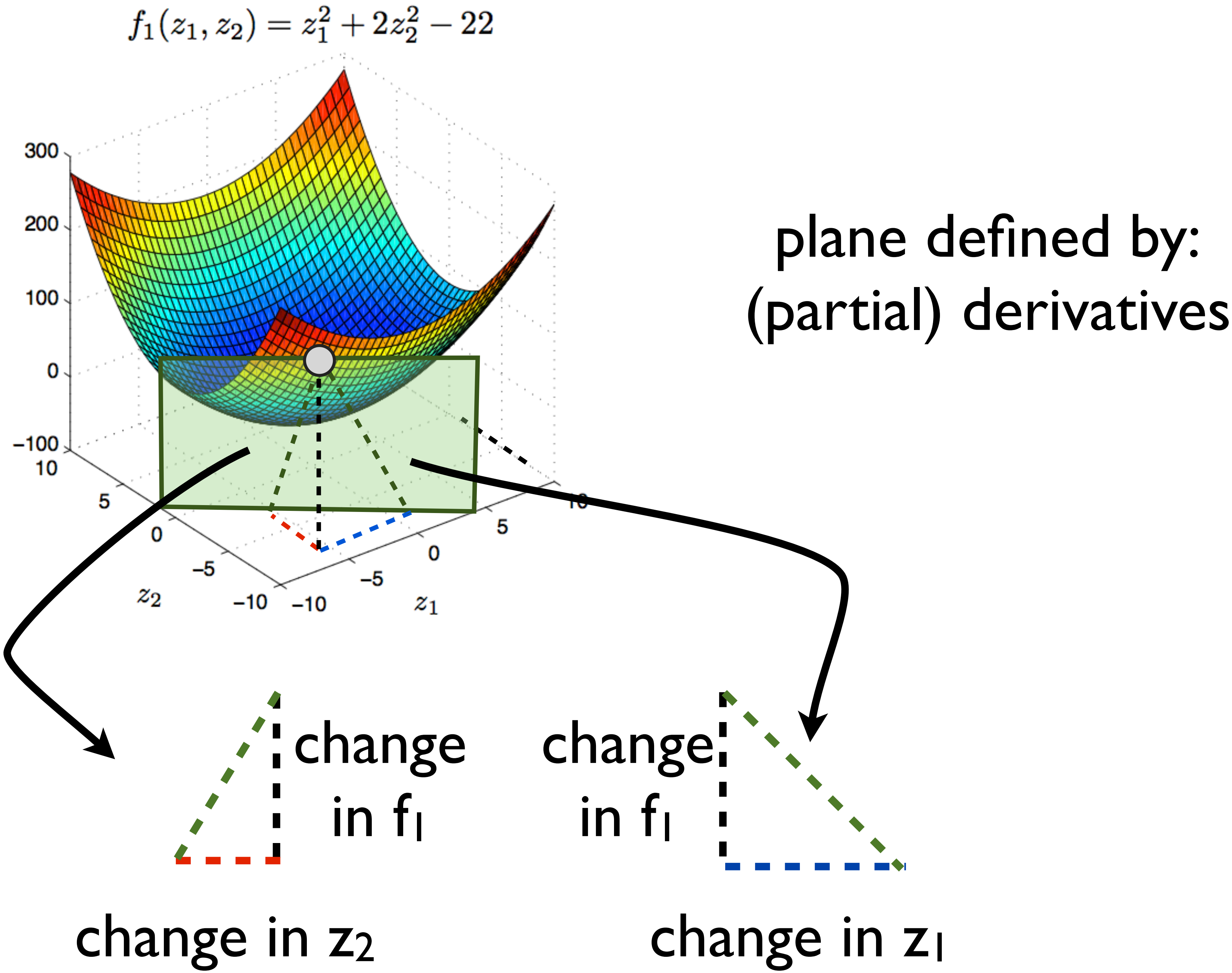
STEP 3. find new $X^{(i+1)}$ where both planes cross 0
(**solve** linearised system)



we've found our next guess $X^{(i+1)}$
(repeat from step 2...)



in step 2, instead of derivative (1D case) we need **gradient** of f_1,f_2 to linearise



$f_1(z_1, z_2) = z_1^2 + 2z_2^2 - 22$
300
200
100
0
−100
10
5
0
−5
−10
z_2
−10
−5
0
5
10
z_1
plane defined by:
(partial) derivatives
change in f_1
change in z_2
change in f_1
change in z_1

$$
\begin{aligned}
f_1(z_1, z_2) &= z_1^2 + 2z_2^2 - 22 = 0 \,, \\
f_2(z_1, z_2) &= 2z_1^2 + z_2^2 - 17 = 0 \,.
\end{aligned}
$$

$$f_1(\underline{z_1}, z_2) \;=\; z_1^2 + 2z_2^2 - 22 \;=\; 0\,,$$

$$f_2(z_1, z_2) \;=\; 2z_1^2 + z_2^2 - 17 \;=\; 0\,.$$

first let's just consider f_1

...and first let's just consider z_1

$$
\begin{aligned}
f_1(\underline{z_1}, z_2) &= z_1^2 + 2z_2^2 - 22 = 0\,, \\
f_2(z_1, z_2) &= 2z_1^2 + z_2^2 - 17 = 0\,.
\end{aligned}
$$

...treat z_2 like a constant

...then how does f_1 change as z_1 changes?

$$\frac{\partial f_1}{\partial z_1} = ?$$

$$f_1(z_1, z_2) = z_1^2 + 2z_2^2 - 22 = 0\,,$$

$$f_2(z_1, z_2) = 2z_1^2 + z_2^2 - 17 = 0\,.$$

...treat z_2 like a constant

...then how does f_1 change as z_1 changes?

$$\frac{\partial f_1}{\partial z_1} = 2z_1$$

$$f_1(z_1, \underline{z_2}) \;=\; z_1^2 + 2z_2^2 - 22 \;=\; 0\,,$$

$$f_2(z_1, z_2) \;=\; 2z_1^2 + z_2^2 - 17 \;=\; 0\,.$$

and now only consider z_2

...treat z_1 like a constant

...then how does f_1 change as z_2 changes?

$$\frac{\partial f_1}{\partial z_2} = \,?$$

$$f_1(z_1, z_2) = z_1^2 + 2z_2^2 - 22 = 0\,,$$
$$f_2(\underline{z_1}, z_2) = 2z_1^2 + z_2^2 - 17 = 0\,.$$

then f_2 and z_1...

$$\frac{\partial f_2}{\partial z_1} = ?$$

$$f_1(z_1, z_2) \; = \; z_1^2 + 2z_2^2 - 22 \; = \; 0 \, ,$$

$$f_2(z_1, \underline{z_2}) \; = \; 2z_1^2 + z_2^2 - 17 \; = \; 0 \, .$$

..and finally f_2 and z_2

$$\frac{\partial f_2}{\partial z_2} = ?$$

$$
\begin{aligned}
f_1(z_1, z_2) &= z_1^2 + 2z_2^2 - 22 = 0 \,, \\
f_2(z_1, z_2) &= 2z_1^2 + z_2^2 - 17 = 0 \,.
\end{aligned}
$$

	∂z_1	∂z_2
∂f_1	$\dfrac{\partial f_1}{\partial z_1}$	$\dfrac{\partial f_1}{\partial z_2}$
∂f_2	$\dfrac{\partial f_2}{\partial z_1}$	$\dfrac{\partial f_2}{\partial z_2}$

$$f_1(z_1, z_2) = z_1^2 + 2z_2^2 - 22 = 0 ,$$
$$f_2(z_1, z_2) = 2z_1^2 + z_2^2 - 17 = 0 .$$

	∂z_1	∂z_2
∂f_1	$2z_1$	$4z_2$
∂f_2	$\frac{\partial f_2}{\partial z_1}$	$\frac{\partial f_2}{\partial z_2}$

$$
\begin{aligned}
f_1(z_1, z_2) &= z_1^2 + 2z_2^2 - 22 = 0\,, \\
f_2(z_1, z_2) &= 2z_1^2 + z_2^2 - 17 = 0\,.
\end{aligned}
$$

	∂z_1	∂z_2
∂f_1	$2z_1$	$4z_2$
∂f_2	$4z_1$	$2z_2$

called the "Jakobian" (J)

$$J_{ij} = \frac{\partial f_i}{\partial z_j}$$

$$\begin{array}{c|cc} & \partial z_1 & \partial z_2 \\ \hline \partial f_1 & 2z_1 & 4z_2 \\ \partial f_2 & 4z_1 & 2z_2 \end{array}$$

called the "Jakobian" (J)

$$J_{ij} = \frac{\partial f_i}{\partial z_j}$$

$$J(X^{(i)}) = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}$$

...where $X^{(i)}=(z_1,z_2)$ at step i

called the "Jakobian" (J)

$$J_{ij} = \frac{\partial f_i}{\partial z_j}$$

$$J(X^{(i)}) = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}$$

sometimes might just write:

$J^{(i)}$

...where $X^{(i)}=(z_1,z_2)$ at step i

1D case:

$$f_{linear}(x^{(i+1)}) \equiv f'(x^{(i)})\,(x^{(i+1)} - x^{(i)}) + f(x^{(i)})$$

nD case:

$$f_{linear}(X^{(i)}) \equiv J(X^{(i)}) \begin{bmatrix} x_1^{(i+1)} - x_1^{(i)} \\ \vdots \\ x_n^{(i+1)} - x_n^{(i)} \end{bmatrix} + \begin{bmatrix} f_1(X^{(i)}) \\ \vdots \\ f_n(X^{(i)}) \end{bmatrix}$$

$$f_{linear}(x^{(i+1)}) \equiv f'(x^{(i)})\,(x^{(i+1)} - x^{(i)}) + f(x^{(i)})$$

1 variable replaced by
n variables, $X=(x_1,...,x_n)$

$$f_{linear}(X^{(i)}) \equiv J(X^{(i)}) \begin{bmatrix} x_1^{(i+1)} - x_1^{(i)} \\ \vdots \\ x_n^{(i+1)} - x_n^{(i)} \end{bmatrix} + \begin{bmatrix} f_1(X^{(i)}) \\ \vdots \\ f_n(X^{(i)}) \end{bmatrix}$$

$$f_{linear}(x^{(i+1)}) \equiv f'(x^{(i)})\,(x^{(i+1)} - x^{(i)}) + f(x^{(i)})$$

derivative
replaced by
Jakobian

$$f_{linear}(X^{(i)}) \equiv J(X^{(i)}) \begin{bmatrix} x_1^{(i+1)} - x_1^{(i)} \\ \vdots \\ x_n^{(i+1)} - x_n^{(i)} \end{bmatrix} + \begin{bmatrix} f_1(X^{(i)}) \\ \vdots \\ f_n(X^{(i)}) \end{bmatrix}$$

$$f_{linear}(x^{(i+1)}) \equiv f'(x^{(i)})\,(x^{(i+1)} - x^{(i)}) + f(x^{(i)})$$

now comparing change in
n variables

$$f_{linear}(X^{(i)}) \equiv J(X^{(i)}) \begin{bmatrix} x_1^{(i+1)} - x_1^{(i)} \\ \vdots \\ x_n^{(i+1)} - x_n^{(i)} \end{bmatrix} + \begin{bmatrix} f_1(X^{(i)}) \\ \vdots \\ f_n(X^{(i)}) \end{bmatrix}$$

$$f_{linear}(x^{(i+1)}) \equiv f'(x^{(i)})\ (x^{(i+1)} - x^{(i)}) + f(x^{(i)})$$

now evaluating
n functions

$$f_{linear}(X^{(i)}) \equiv J(X^{(i)}) \begin{bmatrix} x_1^{(i+1)} - x_1^{(i)} \\ \vdots \\ x_n^{(i+1)} - x_n^{(i)} \end{bmatrix} + \begin{bmatrix} f_1(X^{(i)}) \\ \vdots \\ f_n(X^{(i)}) \end{bmatrix}$$

$$f_{linear}(x^{(i+1)}) \equiv f'(x^{(i)})\,(x^{(i+1)} - x^{(i)}) + f(x^{(i)})$$

$$f_{linear}(X^{(i)}) \equiv J(X^{(i)}) \begin{bmatrix} x_1^{(i+1)} - x_1^{(i)} \\ \vdots \\ x_n^{(i+1)} - x_n^{(i)} \end{bmatrix} + \begin{bmatrix} f_1(X^{(i)}) \\ \vdots \\ f_n(X^{(i)}) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

set to zero to solve linearised system, i.e.
want to find *n*D point $(x_1^{(i+1)}, \ldots, x_n^{(i+1)})$
where all our "hyperplanes" cross 0

$$f_{linear}(x^{(i+1)}) \equiv f'(x^{(i)})\ (x^{(i+1)} - x^{(i)}) + f(x^{(i)})$$

$$f_{linear}(X^{(i)}) \equiv J(X^{(i)}) \begin{bmatrix} x_1^{(i+1)} - x_1^{(i)} \\ \vdots \\ x_n^{(i+1)} - x_n^{(i)} \end{bmatrix} + \begin{bmatrix} f_1(X^{(i)}) \\ \vdots \\ f_n(X^{(i)}) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$f_{linear}(X^{(i+1)}) \equiv J(X^{(i)})\ \Delta X^{(i)} + \mathbf{f}(X^{(i)}) = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$f_{linear}(x^{(i+1)}) \equiv f'(x^{(i)})\,(x^{(i+1)} - x^{(i)}) + f(x^{(i)})$$

$$f_{linear}(X^{(i)}) \equiv J(X^{(i)}) \begin{bmatrix} x_1^{(i+1)} - x_1^{(i)} \\ \vdots \\ x_n^{(i+1)} - x_n^{(i)} \end{bmatrix} + \begin{bmatrix} f_1(X^{(i)}) \\ \vdots \\ f_n(X^{(i)}) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$J(X^{(i)})\,\Delta X^{(i)} = -\mathbf{f}(X^{(i)})$$

rearrange to solve for $\Delta X^{(i)}$

initial guess: $z_1=10$, $z_2=10$

$$J(X) = \begin{bmatrix} 2z_1 & 4z_2 \\ 4z_1 & 2z_2 \end{bmatrix}$$

$$\begin{aligned} f_1(z_1, z_2) &= z_1^2 + 2z_2^2 - 22 \\ f_2(z_1, z_2) &= 2z_1^2 + z_2^2 - 17 \end{aligned}$$

Q 1. what is $X^{(i)}$?

Q 2. what is $J(X^{(i)})$ and $-\mathbf{f}(X^{(i)})$?

$$\underset{?}{\underline{J(X^{(i)})}}\ \Delta X^{(i)} = \underset{?}{\underline{-\mathbf{f}(X^{(i)})}}$$

Q 3. suppose: $\Delta X^{(i)} = \begin{bmatrix} -4.8 \\ -4.55 \end{bmatrix}$

then solve: $X^{(i+1)} = X^{(i)} + \Delta X^{(i)}$

initial guess: $z_1=10$, $z_2=10$

$$J(X) = \begin{bmatrix} 2z_1 & 4z_2 \\ 4z_1 & 2z_2 \end{bmatrix}$$

$$\begin{aligned} f_1(z_1, z_2) &= z_1^2 + 2z_2^2 - 22 \\ f_2(z_1, z_2) &= 2z_1^2 + z_2^2 - 17 \end{aligned}$$

$$\underbrace{J(X^{(i)})}_{?} \, \Delta X^{(i)} = \underbrace{-\mathbf{f}(X^{(i)})}_{?}$$

initial guess: $z_1=10$, $z_2=10$

$$J(X) = \begin{bmatrix} 2z_1 & 4z_2 \\ 4z_1 & 2z_2 \end{bmatrix}$$

$$\begin{aligned} f_1(z_1, z_2) &= z_1^2 + 2z_2^2 - 22 \\ f_2(z_1, z_2) &= 2z_1^2 + z_2^2 - 17 \end{aligned}$$

$$\begin{bmatrix} 20 & 40 \\ 40 & 20 \end{bmatrix} \Delta X^{(i)} = \begin{bmatrix} -278 \\ -283 \end{bmatrix}$$

initial guess: $z_1=10$, $z_2=10$

$$J(X) = \begin{bmatrix} 2z_1 & 4z_2 \\ 4z_1 & 2z_2 \end{bmatrix}$$

$$\begin{aligned} f_1(z_1, z_2) &= z_1^2 + 2z_2^2 - 22 \\ f_2(z_1, z_2) &= 2z_1^2 + z_2^2 - 17 \end{aligned}$$

$$\begin{bmatrix} 20 & 40 \\ 40 & 20 \end{bmatrix} \Delta X^{(i)} = \begin{bmatrix} -278 \\ -283 \end{bmatrix}$$

$$\Delta X^{(i)} = \begin{bmatrix} -4.8 \\ -4.55 \end{bmatrix}$$

initial guess: $z_1=10$, $z_2=10$

$$J(X) = \begin{bmatrix} 2z_1 & 4z_2 \\ 4z_1 & 2z_2 \end{bmatrix}$$

$$\begin{aligned} f_1(z_1, z_2) &= z_1^2 + 2z_2^2 - 22 \\ f_2(z_1, z_2) &= 2z_1^2 + z_2^2 - 17 \end{aligned}$$

$$\begin{bmatrix} 20 & 40 \\ 40 & 20 \end{bmatrix} \Delta X^{(i)} = \begin{bmatrix} -278 \\ -283 \end{bmatrix}$$

$$\Delta X^{(i)} = \begin{bmatrix} -4.8 \\ -4.55 \end{bmatrix}$$

$$X^{(i+1)} = X^{(i)} + \Delta X^{(i)}$$

initial guess: $z_1=10$, $z_2=10$

$$J(X) = \begin{bmatrix} 2z_1 & 4z_2 \\ 4z_1 & 2z_2 \end{bmatrix}$$

$$\begin{aligned} f_1(z_1, z_2) &= z_1^2 + 2z_2^2 - 22 \\ f_2(z_1, z_2) &= 2z_1^2 + z_2^2 - 17 \end{aligned}$$

$$\begin{bmatrix} 20 & 40 \\ 40 & 20 \end{bmatrix} \Delta X^{(i)} = \begin{bmatrix} -278 \\ -283 \end{bmatrix}$$

$$\Delta X^{(i)} = \begin{bmatrix} -4.8 \\ -4.55 \end{bmatrix}$$

$$X^{(i+1)} = \begin{bmatrix} 5.2 \\ 5.45 \end{bmatrix}$$

if you want to solve the following system in MATLAB:

$$J\,\Delta X = -\mathbf{f}$$

...then use backslash operator:

$$\Delta X = -J \backslash \mathbf{f}$$

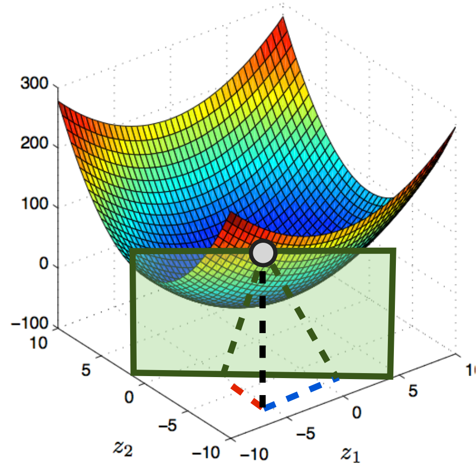
the is equivalent to:

$$\Delta X = -\,J^{-1}\,\mathbf{f}$$

...but faster

notation...

$f'(x)$	1st-order derivative (1D)
$\nabla f(X)$	gradient (nD)
$g(X)$	gradient (nD)



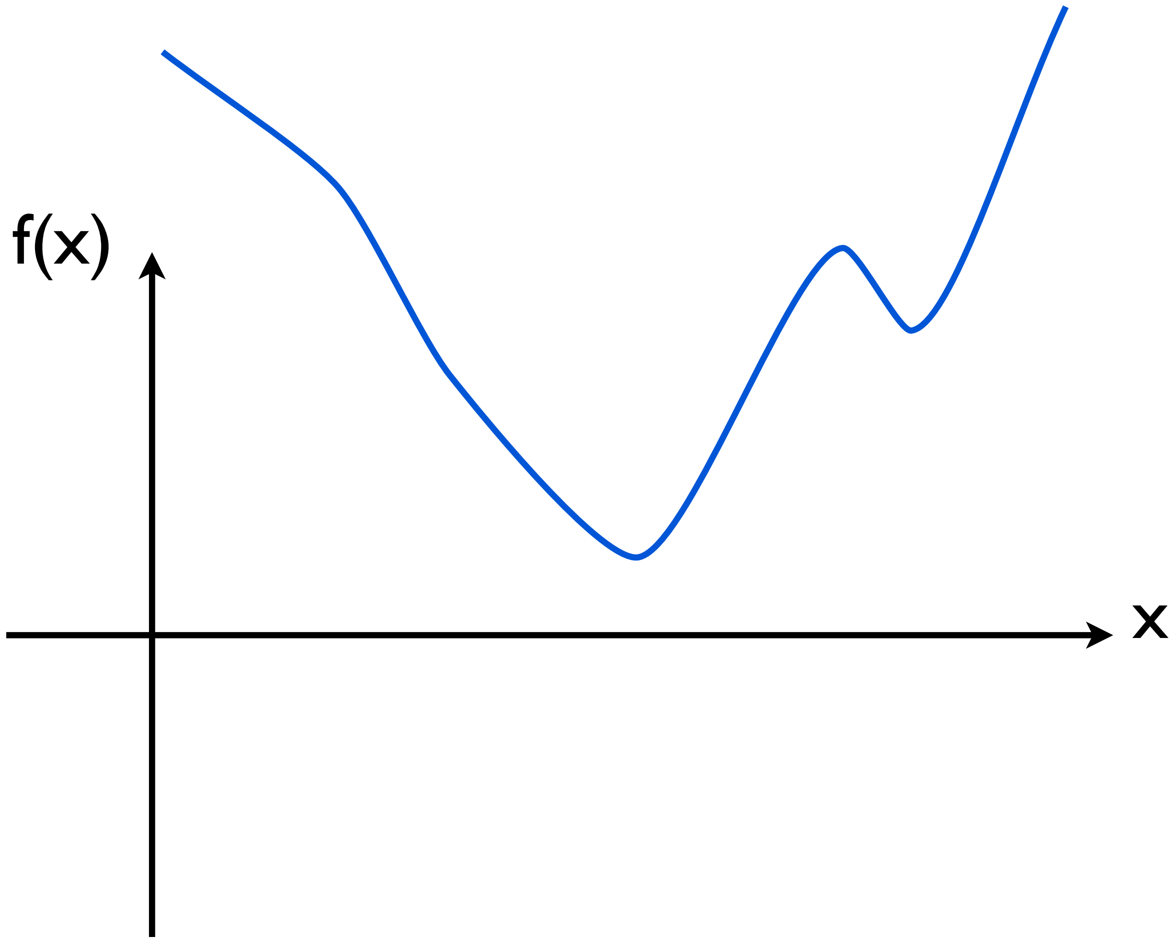
SUMMARY Part 3. nD Newton method

- still linear approximation, but now in *n*D
- Jakobian

Part 4

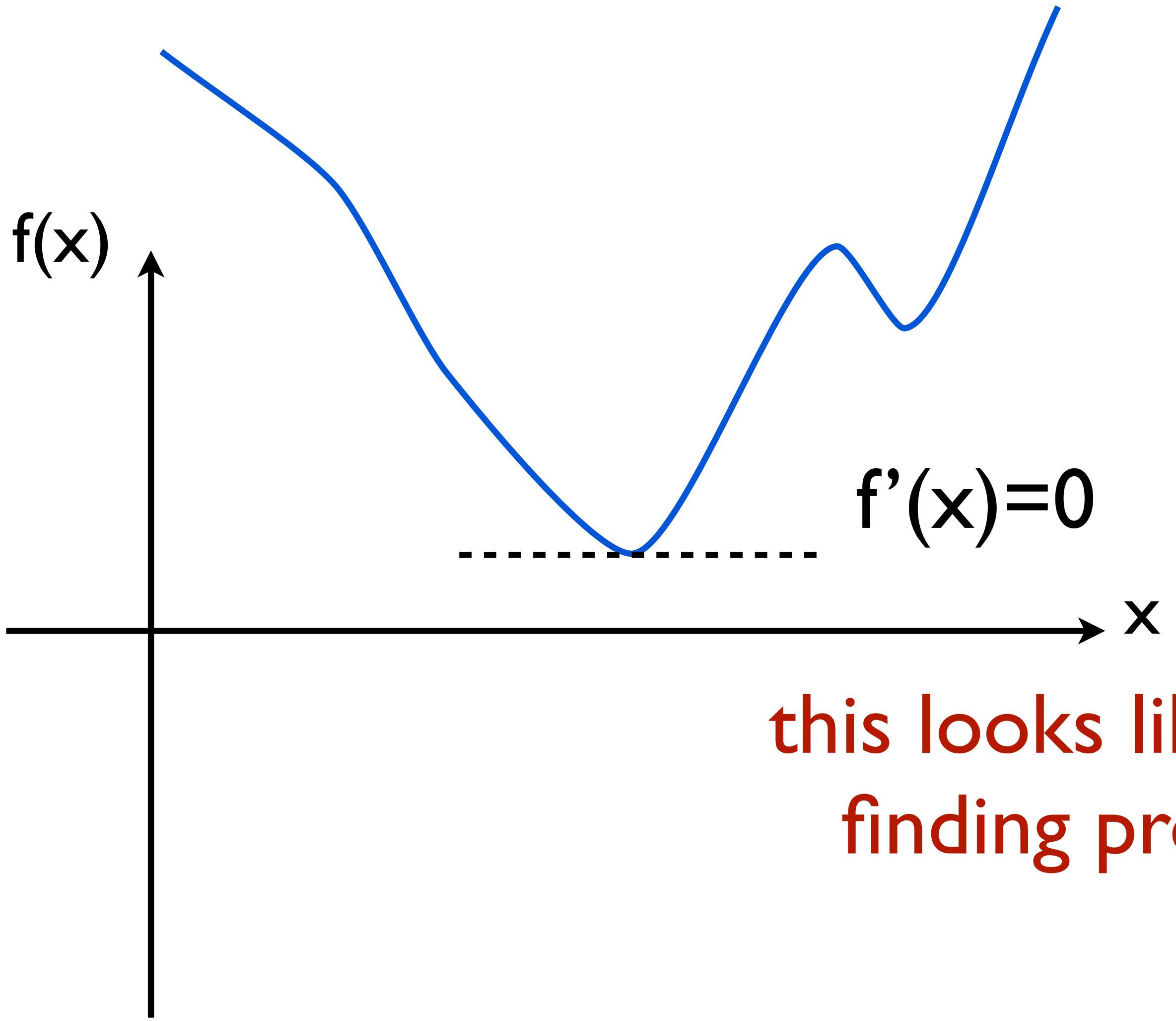
Newton's method for optimisation

can we apply same idea to find minimum?



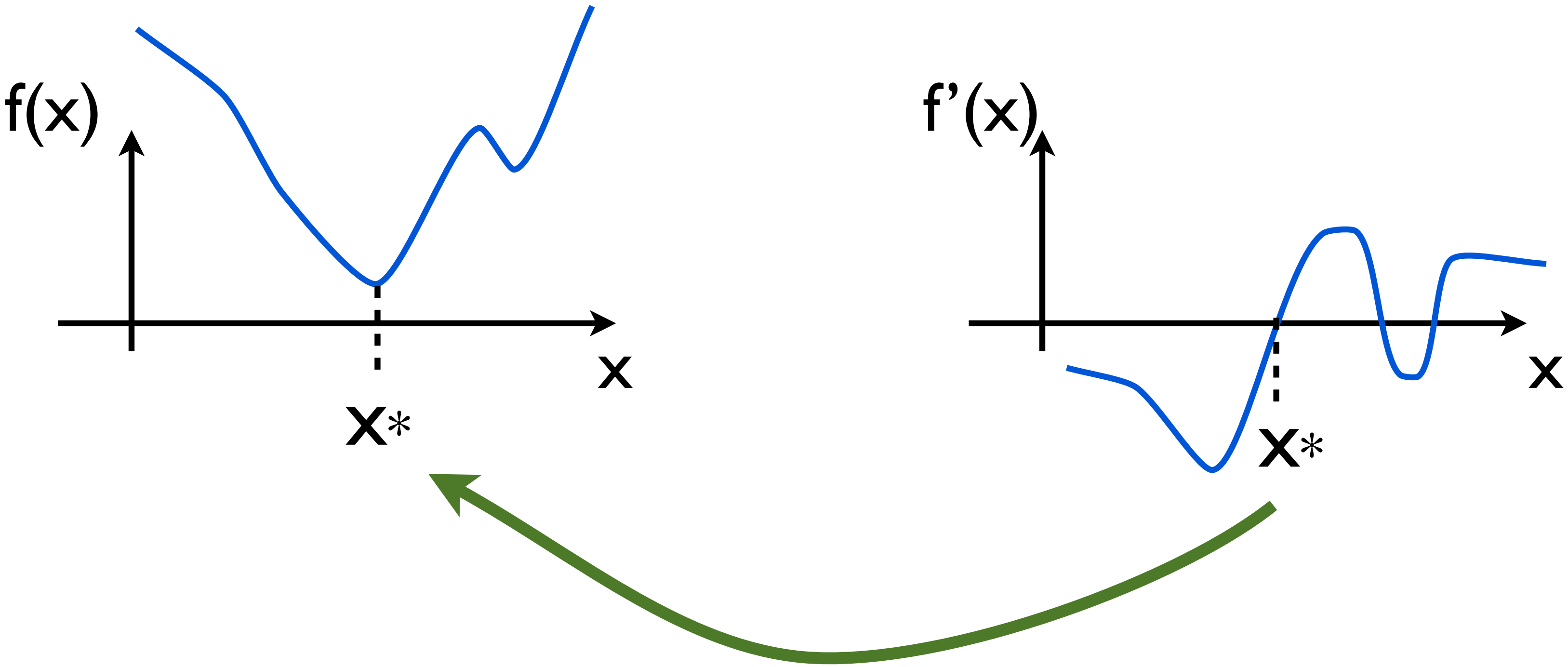
what is true about *f* at the minimum?

can we apply same idea to find minimum?



this looks like a root-finding problem...

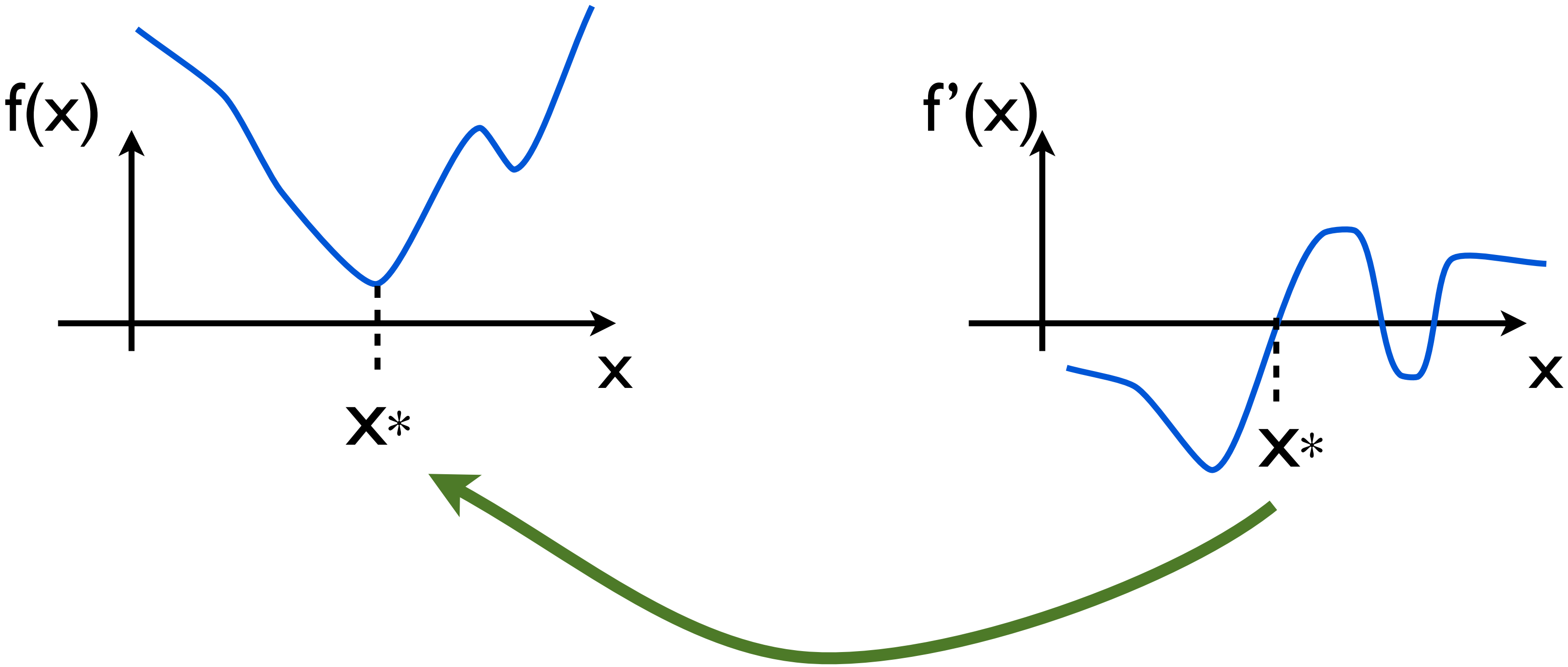
find root of f'(x)



that'll give us the minimum of f(x)

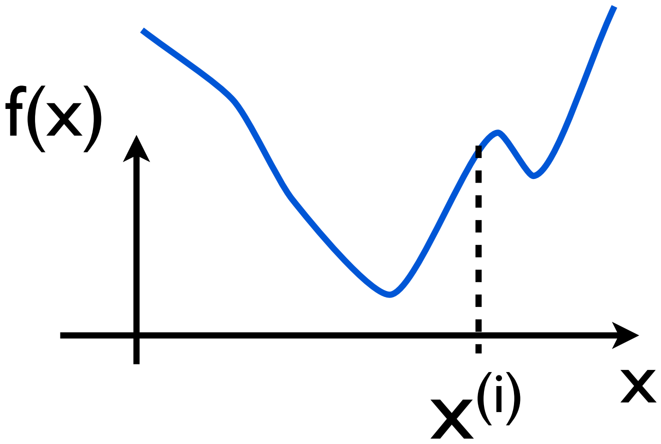
$$x^{(i+1)} = x^{(i)} - \frac{f'(x^{(i)})}{f''(x^{(i)})}$$

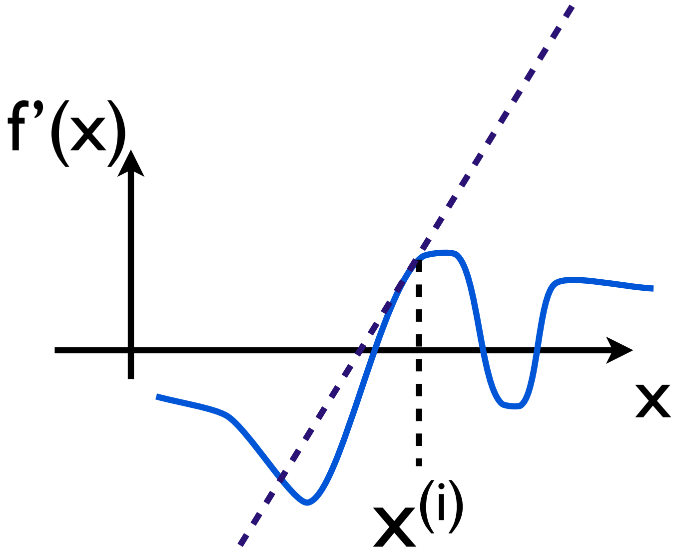
"update" formula
to find minimum



that'll give us the minimum of f(x)

here Newton's method approximates by **lines**





what is approximation in original formula?
(see next slide)

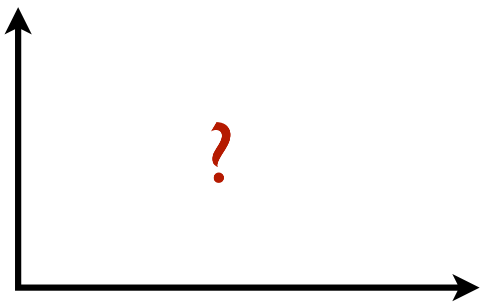
Step 1. discuss with your neighbour (2 mins)

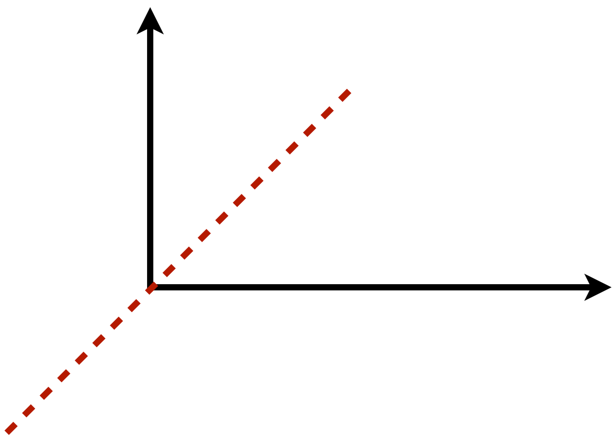
?

$f(x)= ?$

line

$f'(x)=2x$





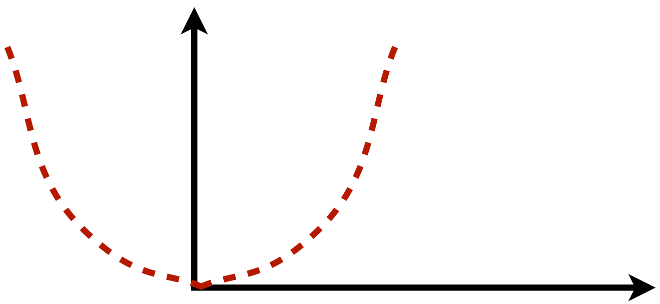
Step 2. ...and in general?

$f(x)=?$

$f'(x)=mx^1 + c$

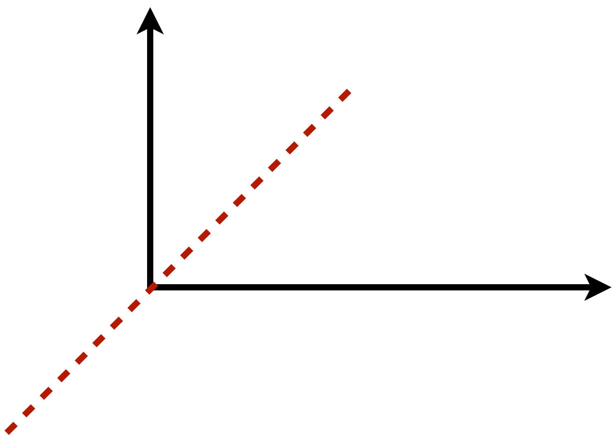
quadratic

$$f(x) = x^2$$



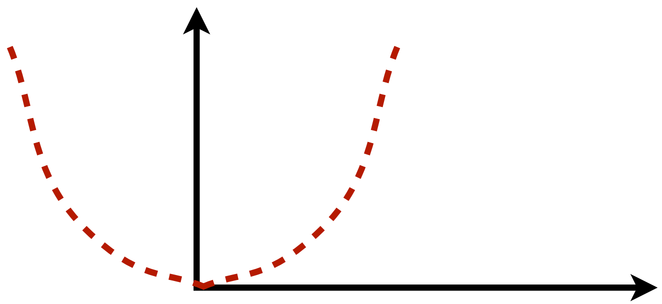
line

$$f'(x) = 2x$$



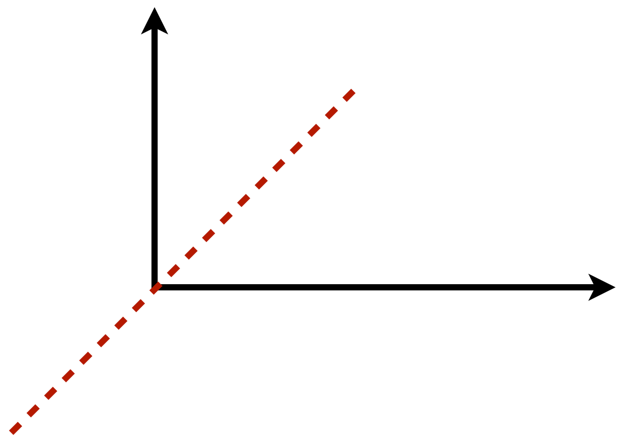
quadratic

$$f(x)=\left(\frac{m}{2}\right)x^2 + cx + k$$

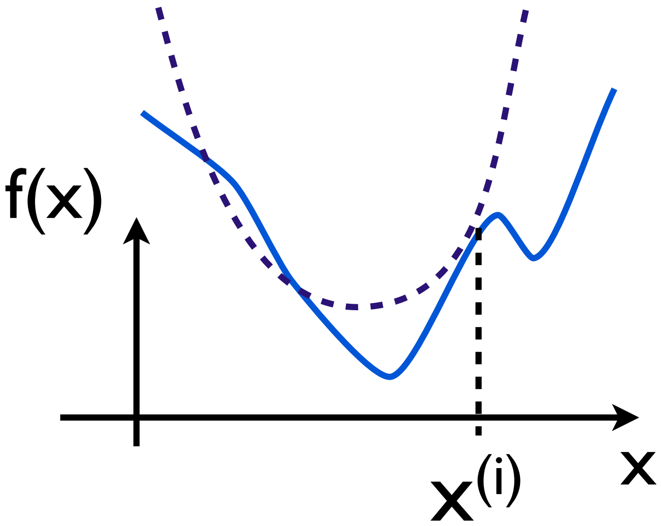


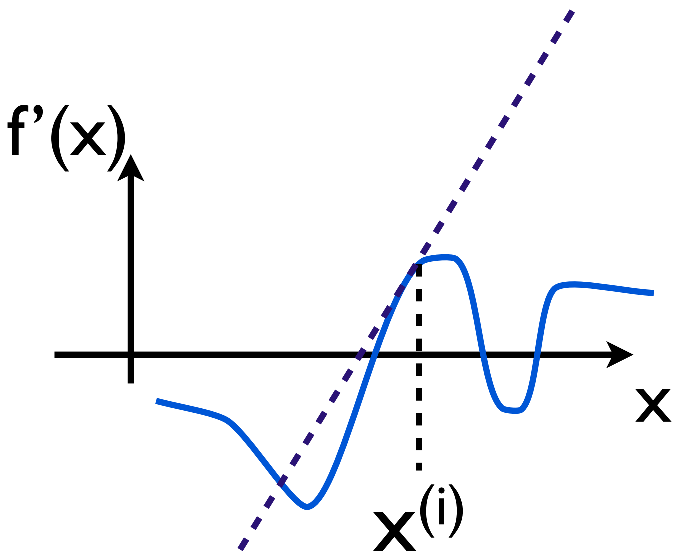
line

$$f'(x)=mx^1 + c$$

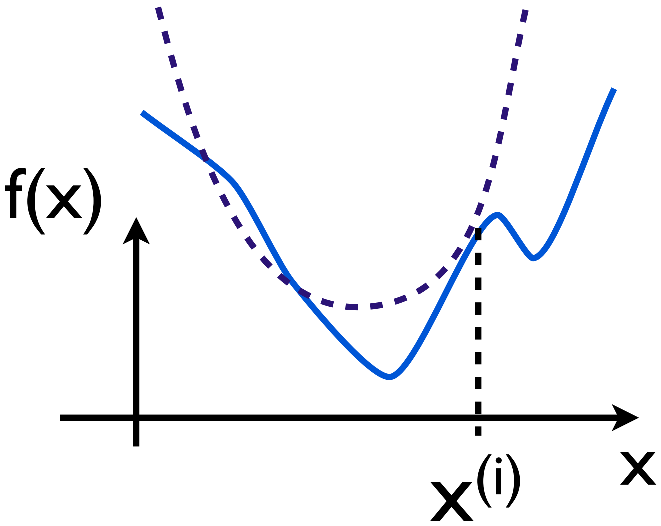


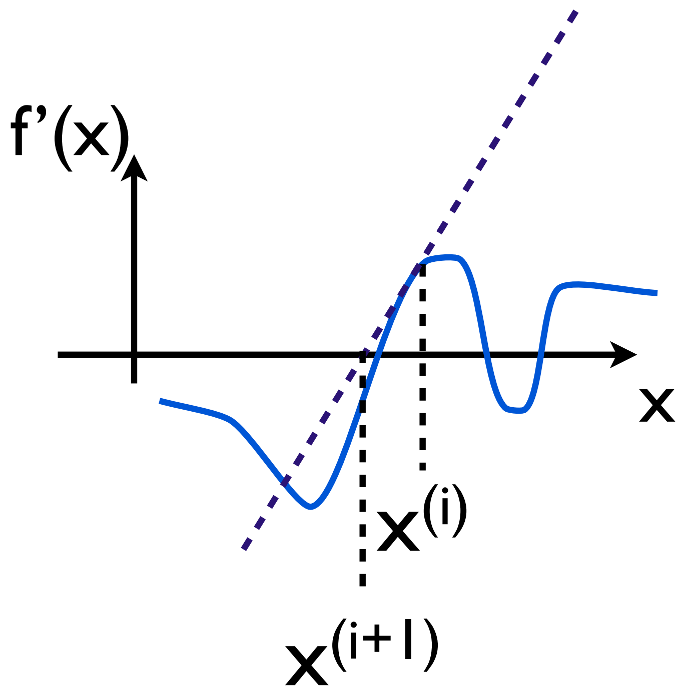
here Newton's method approximates by **lines**





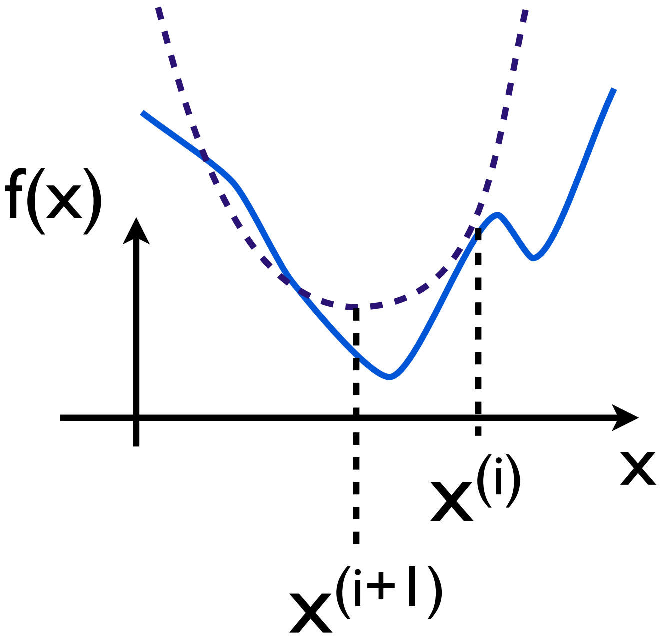
...so here Newton approximates by **quadratics** root of line at each step....

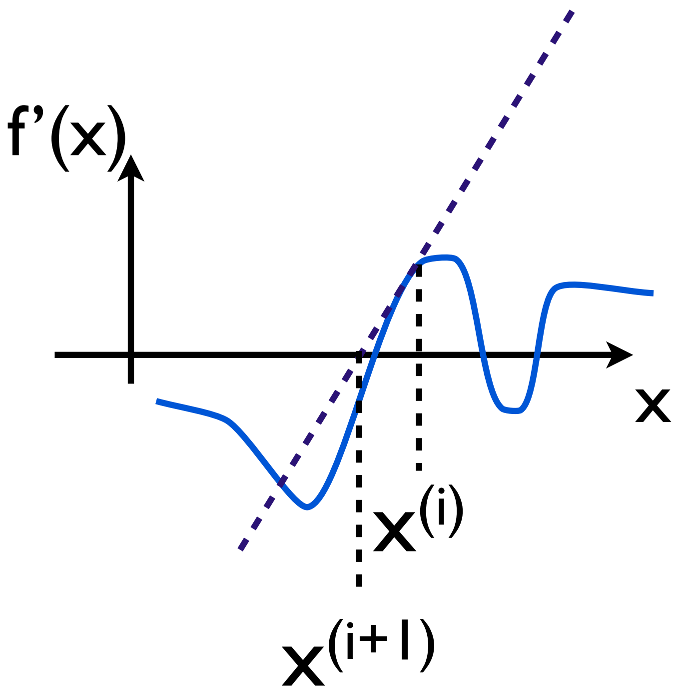




...corresponds to?

root of line at each step....





minimum of quadratic approximation

Second-order Taylor expansion around x_n

$$f_T(x) = f_T(x_n + \Delta x) \approx f(x_n) + \underline{f'(x_n)\Delta x} + \frac{1}{2}\underline{f''(x_n)\Delta x^2}\,.$$

want to find Δx so $(x_n + \Delta x)$ is stationary point:

when is $f_T(x_n + \Delta x)$ a stationary point?

Second-order Taylor expansion around x_n

$$f_T(x) = f_T(x_n + \Delta x) \approx f(x_n) + \underline{f'(x_n)\Delta x} + \frac{1}{2}\underline{f''(x_n)\Delta x^2}\,.$$

want to find Δx so $(x_n + \Delta x)$ is stationary point:

when is $f_T(x_n + \Delta x)$ a stationary point?

...when derivative is **zero**

Second-order Taylor expansion around x_n

$$f_T(x) = f_T(x_n + \Delta x) \approx f(x_n) + \underline{f'(x_n)\Delta x} + \frac{1}{2}\underline{f''(x_n)\Delta x^2} \,.$$

want to find Δx so $(x_n + \Delta x)$ is stationary point:

$$0 = \frac{\mathrm{d}}{\mathrm{d}\Delta \mathrm{x}} \left(f(x_n) + f'(x_n)\Delta x + \frac{1}{2} f''(x_n)\Delta x^2 \right)$$

Second-order Taylor expansion around x_n

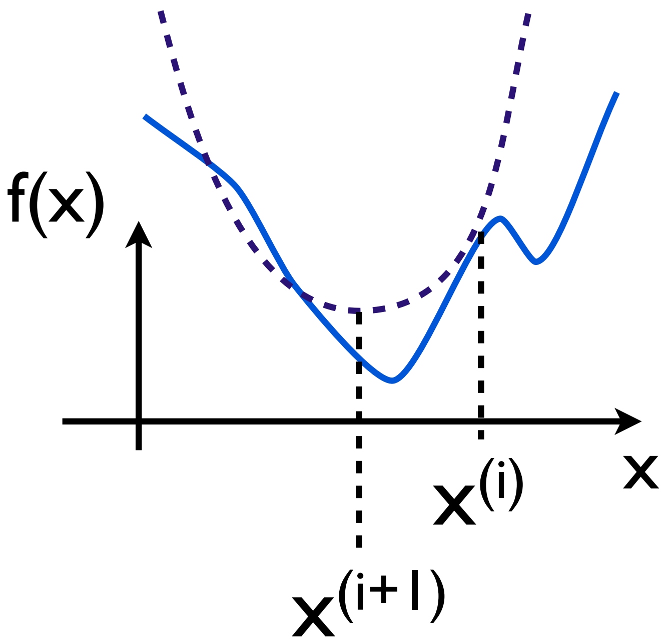
$$f_T(x) = f_T(x_n + \Delta x) \approx f(x_n) + \underline{f'(x_n)\Delta x} + \underline{\frac{1}{2}f''(x_n)\Delta x^2}\,.$$

want to find Δx so $(x_n + \Delta x)$ is stationary point:

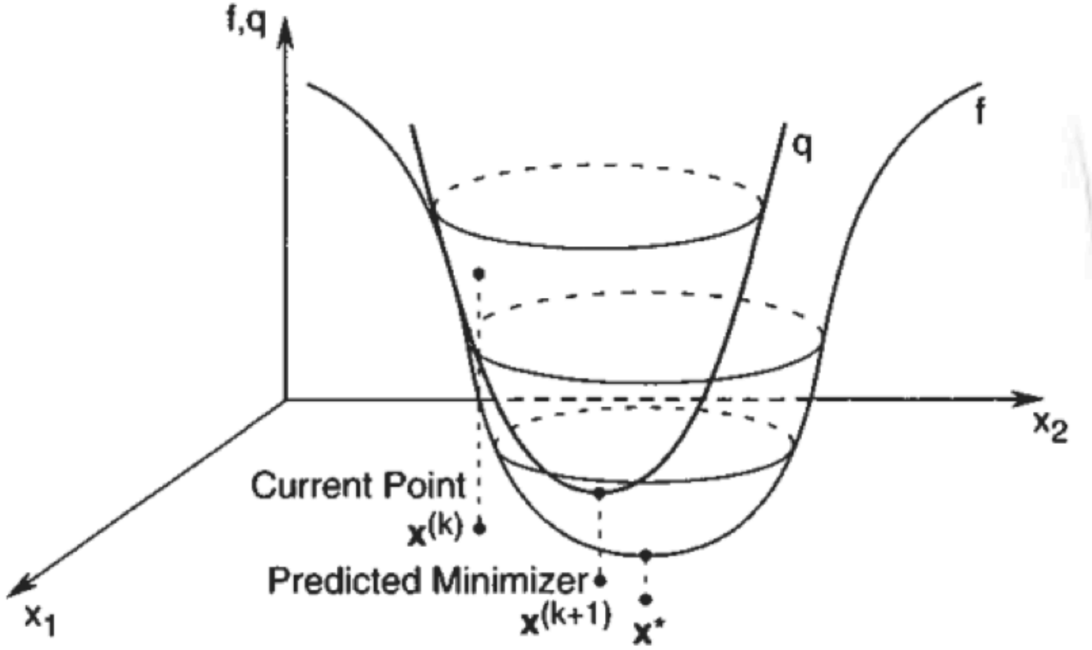
$$0 = \frac{\mathrm{d}}{\mathrm{d}\Delta\mathrm{x}}\left(\cancel{f(x_n)} + f'(x_n)\cancel{\Delta x} + \frac{2}{2}f''(x_n)\Delta x^{\cancel{2}}\right)$$

$$= f'(x_n) + f''(x_n)\Delta x.$$

what about optimisation in higher dimensions?

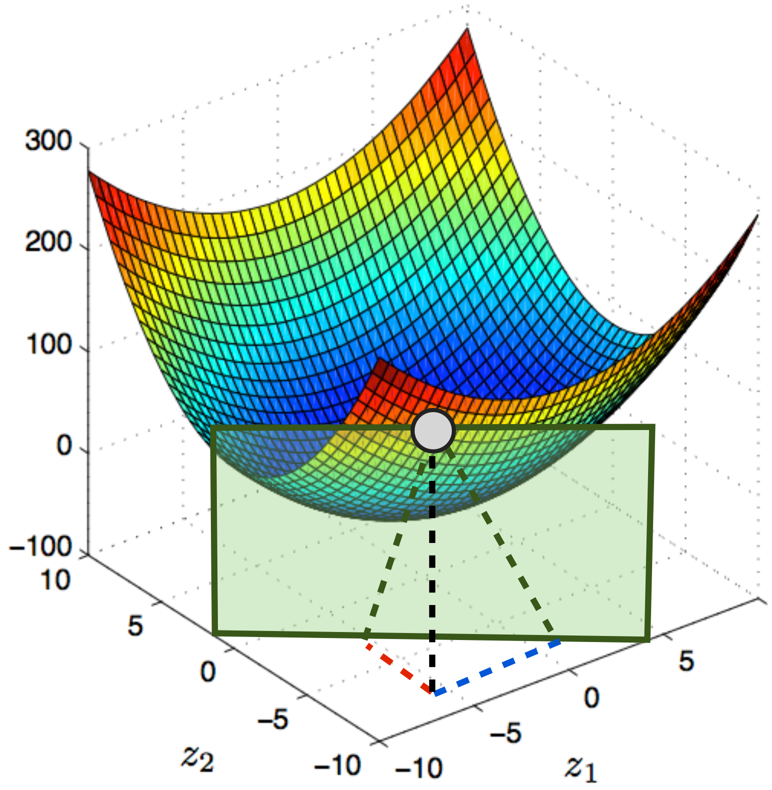


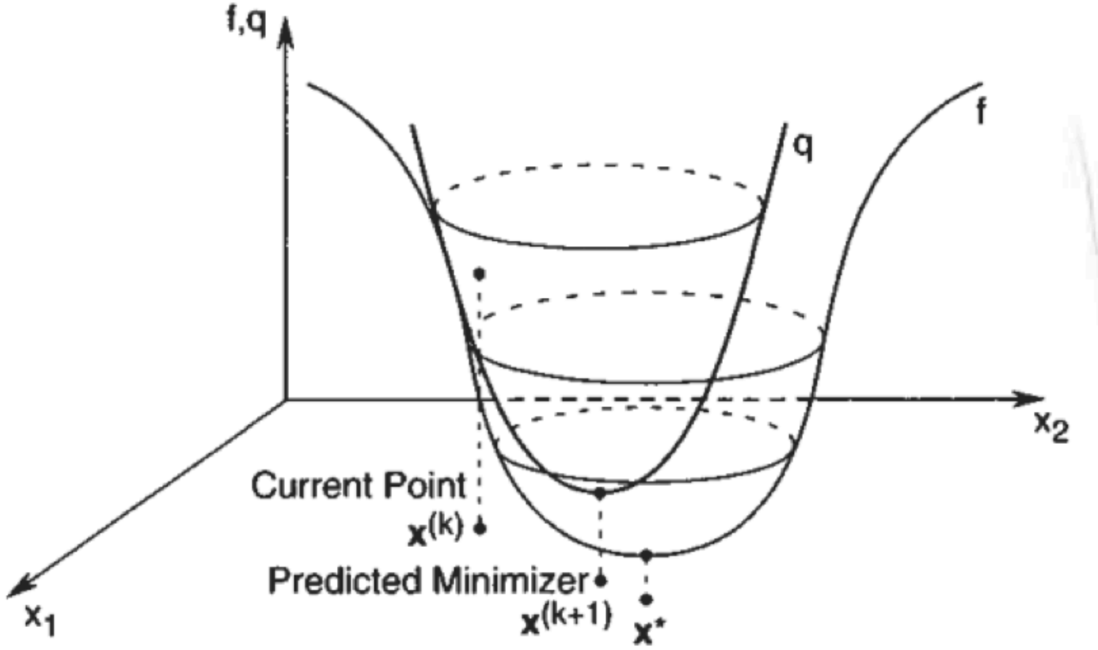
1D case



nD case

what about optimisation in higher dimensions?





nD root-finding:	nD optimisation:
$X^{(i+1)} = X^{(i)} - J(X^{(i)})^{-1}\, \mathbf{f}(X^{(i)})$	$X^{(i+1)} = X^{(i)} - H(X^{(i)})^{-1}\, \nabla\mathbf{f}(X^{(i)})$
"Jakobian"	"Hessian"

Jakobian

$$J_{ij} = \frac{\partial f_i}{\partial x_j}$$

first-order
partial derivatives

Hessian

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \, \partial x_j}$$

second-order
partial derivatives

Second-order Taylor expansion 1D case:

$$f_T(x) = f_T(x_n + \Delta x) \approx f(x_n) + f'(x_n)\Delta x + \frac{1}{2}f''(x_n)\Delta x^2 \, .$$

Second-order Taylor expansion nD case:

$$f_T(X) = f_T(X^{(i)} + \Delta X) \approx f(X^{(i)}) + \nabla f(X^{(i)})\, \Delta X + \frac{1}{2}\, \Delta X^T H(X^{(i)}) \Delta X$$

Second-order Taylor expansion 1D case:

$$f_T(x) = f_T(x_n + \Delta x) \approx f(x_n) + f'(x_n)\Delta x + \frac{1}{2} f''(x_n)\Delta x^2 \,.$$

Second-order Taylor expansion nD case:

$$f_T(X) = f_T(X^{(i)} + \Delta X) \approx f(X^{(i)}) + \nabla f(X^{(i)})\, \Delta X + \frac{1}{2}\, \underline{\Delta X}^T H(X^{(i)}) \underline{\Delta X}$$

notation...

f''(x)	2nd-order derivative (1D)
H(X)	Hessian of f(X)
F(X)	Hessian of f(X)

Summary 3.6. Advantages and disadvantages of Newton's method for unconstrained optimization problems

Advantages

1° Quadratically convergent from a good starting point if $\nabla^2 f(\hat{\boldsymbol{x}})$ is positive definite.

2° Simple and easy to implement.

Disadvantages

1° Not globally convergent for many problems.

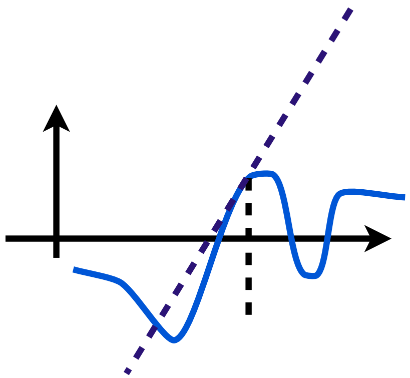
2° May converge towards a maximum or saddle point of f.

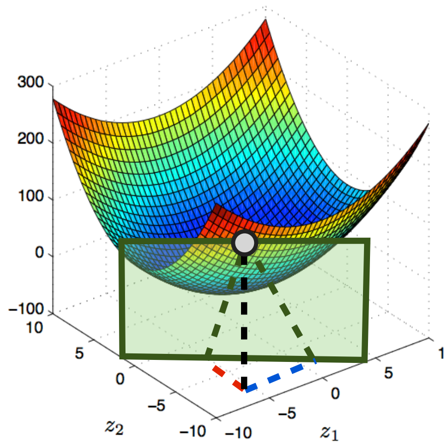
3° The system of linear equations to be solved in each iteration may be ill-conditioned or singular.

4° Requires second order derivatives of f.

Newton method for root-finding

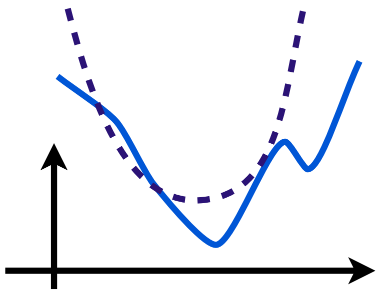
	1D	nD
at root	$f(x)=0$	$f(X)=0$
Newton's method	$x^{(i+1)}=x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})}$	$X^{(i+1)}=X^{(i)} - J^{(i)-1} f(X^{(i)})$

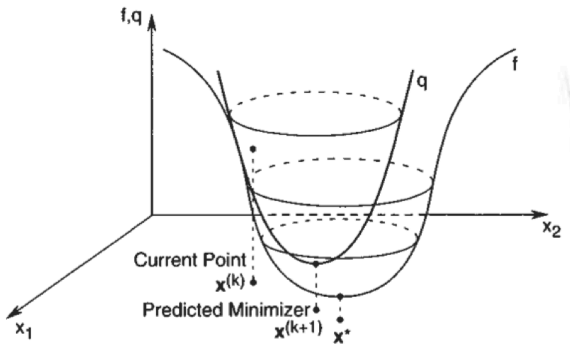




Newton method for optimisation

	1D	nD
at optimum	$f'(x)=0$	$\nabla f(X)=0$
Newton's method	$x^{(i+1)}=x^{(i)} - \dfrac{f'(x^{(i)})}{f''(x^{(i)})}$	$X^{(i+1)}=X^{(i)} - H^{(i)-1}\nabla f(X^{(i)})$

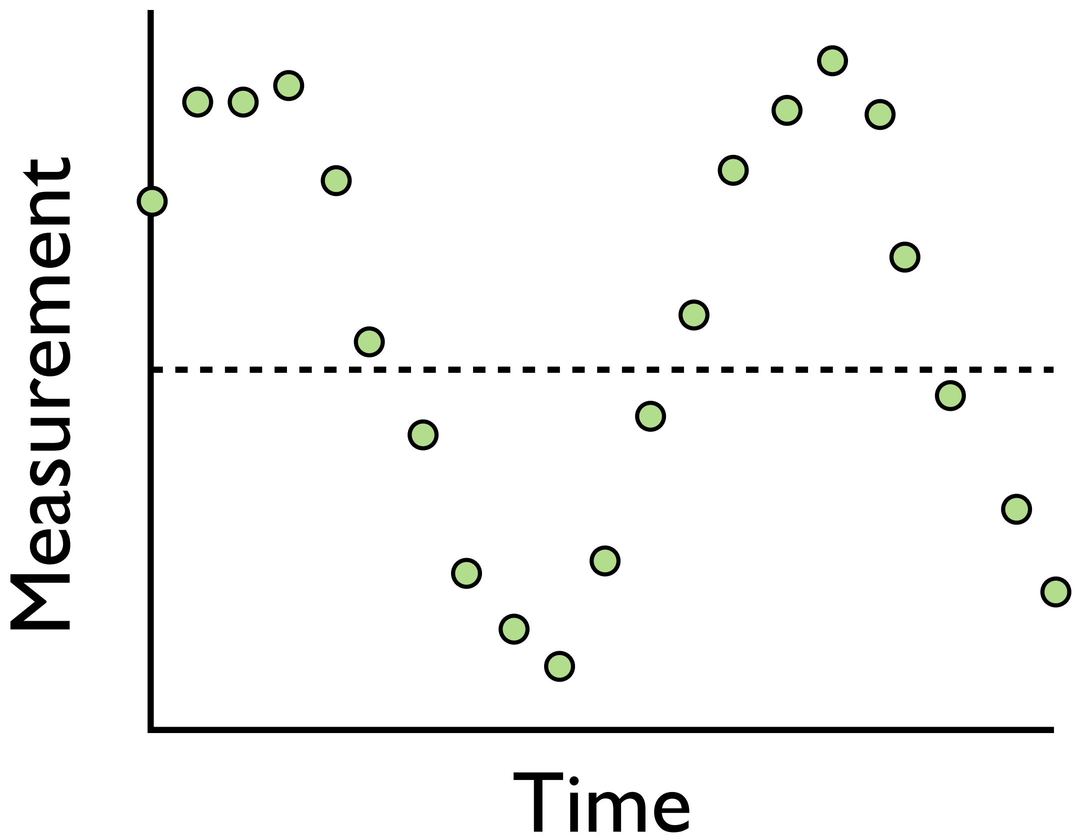




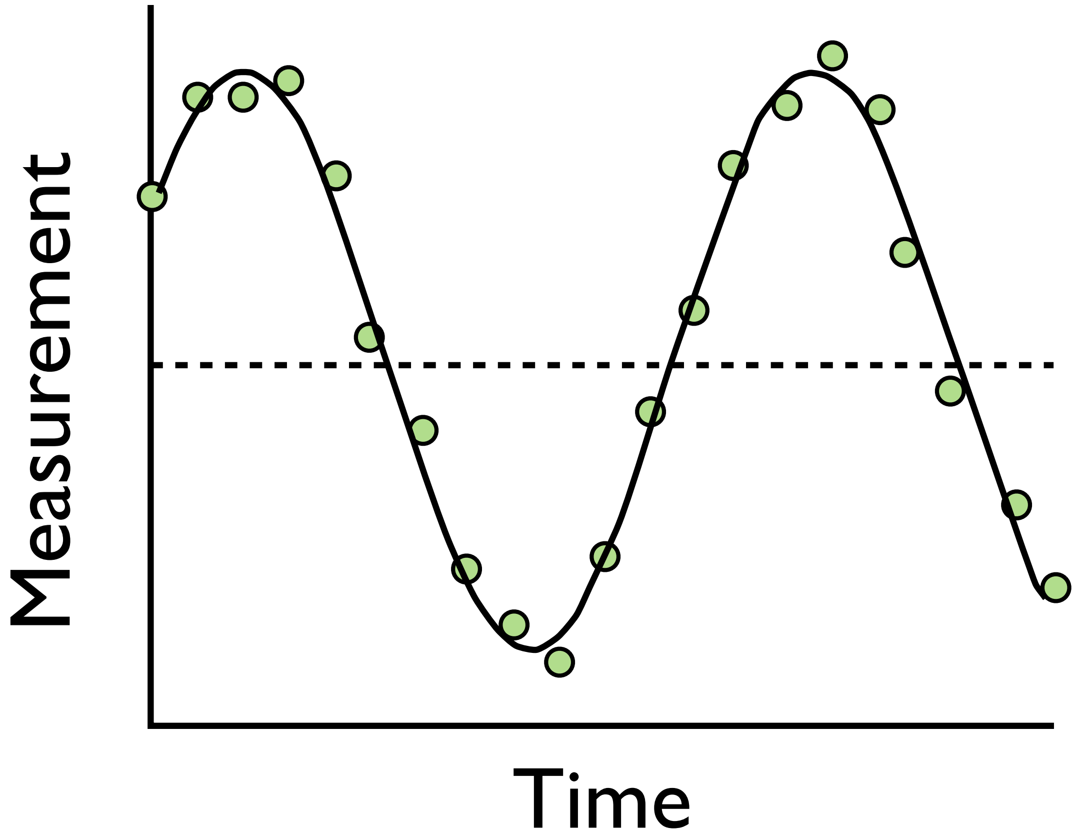
Part 5

curve fitting

curve fitting

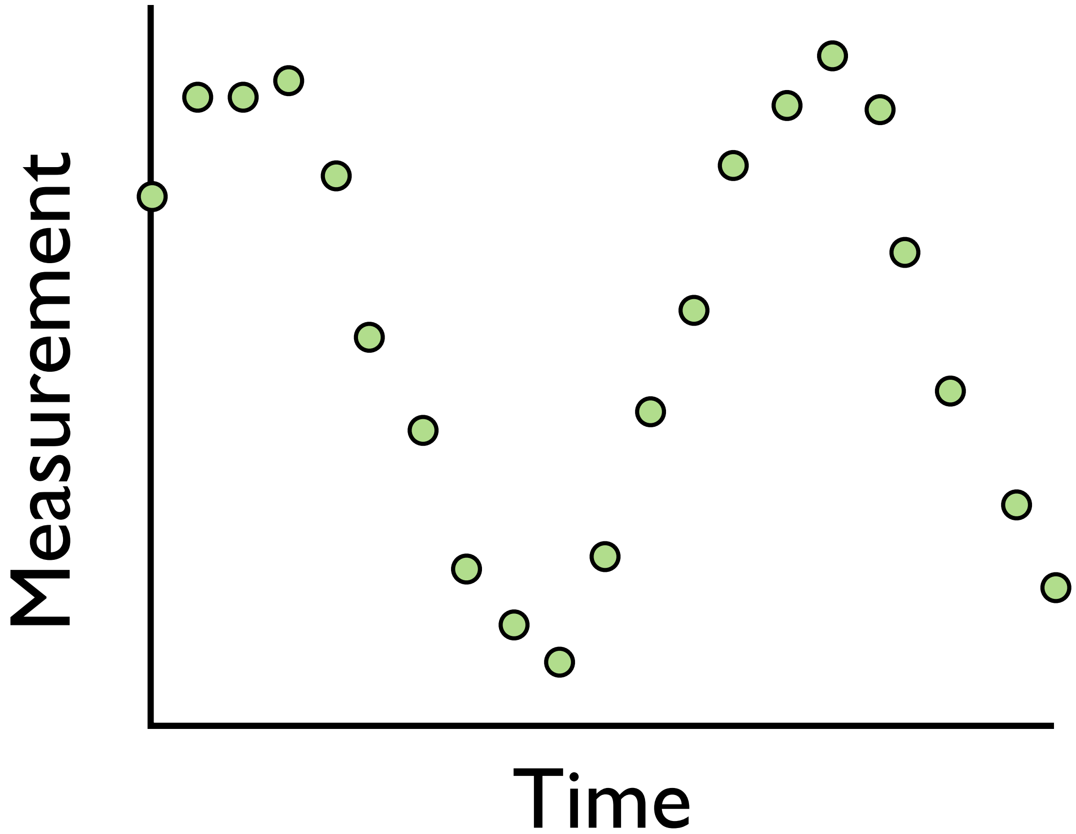


curve fitting



e.g. consider we try fitting the following function with parameters:

$$f(t) = A \sin(\omega t + \Phi)$$

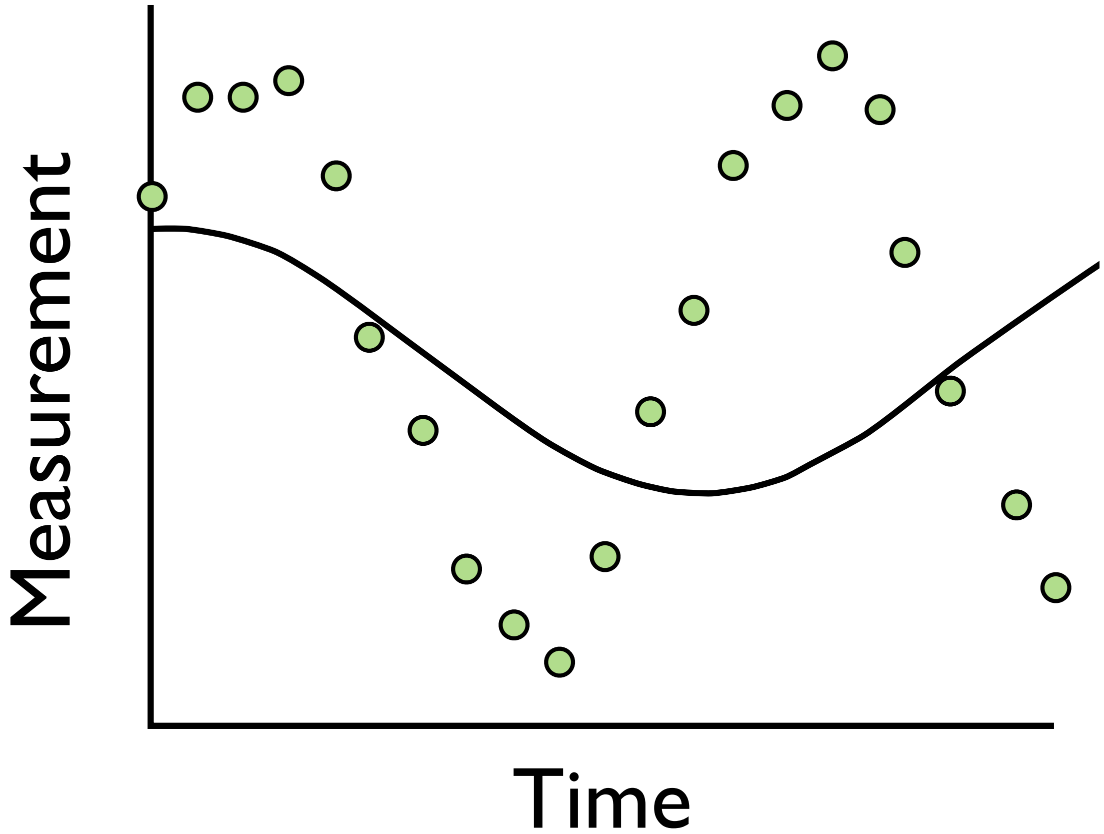


e.g. consider we try fitting the following function with parameters:

$$f(t) = A \sin(\omega t + \Phi)$$

(initial guess)

$A=1.01 \quad \omega=0.592 \quad \Phi=1.541$

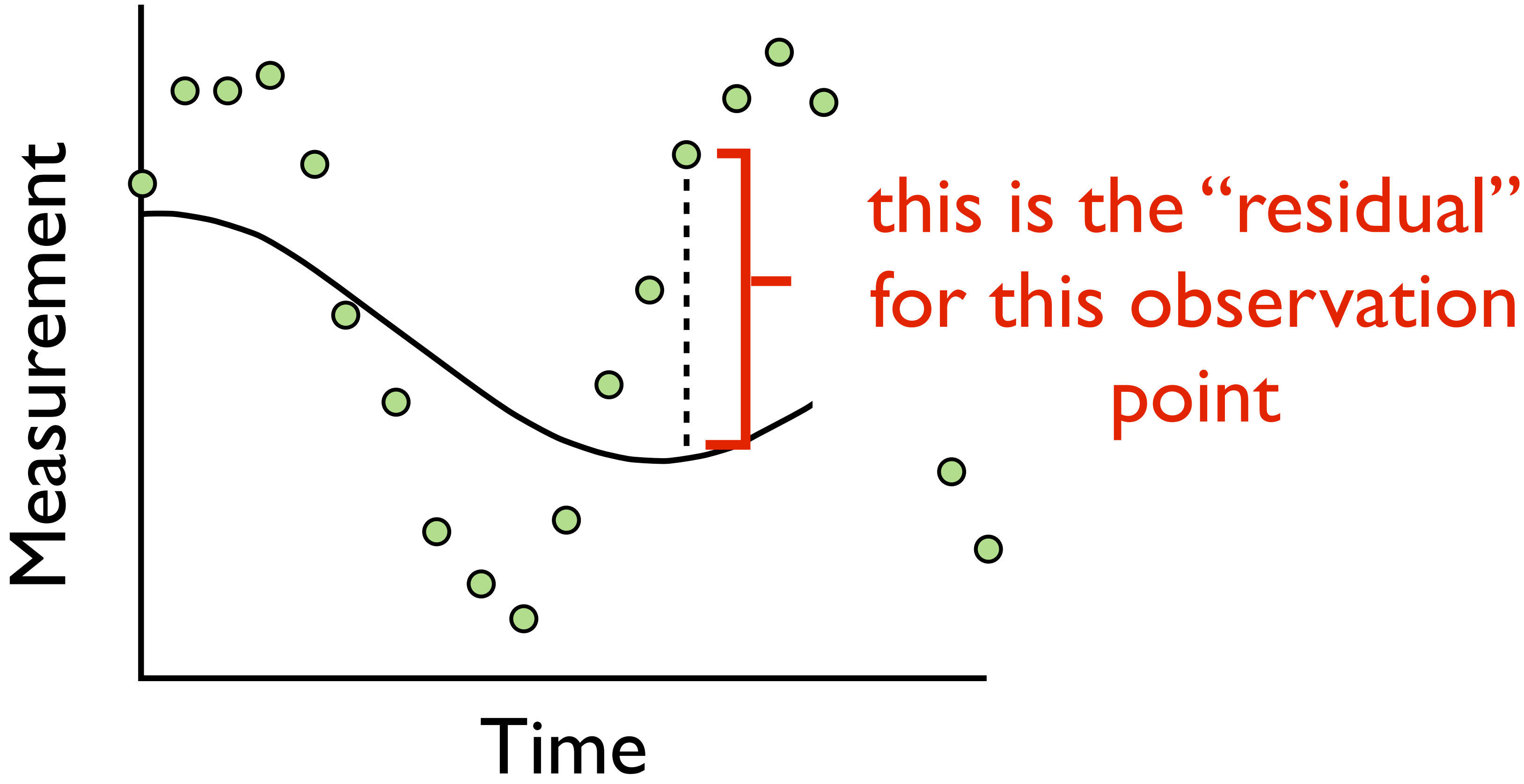


how can we measure how "close" we are to good fit?

e.g. consider we try fitting the following function with parameters:

$$f(t) = A \sin(\omega t + \Phi)$$

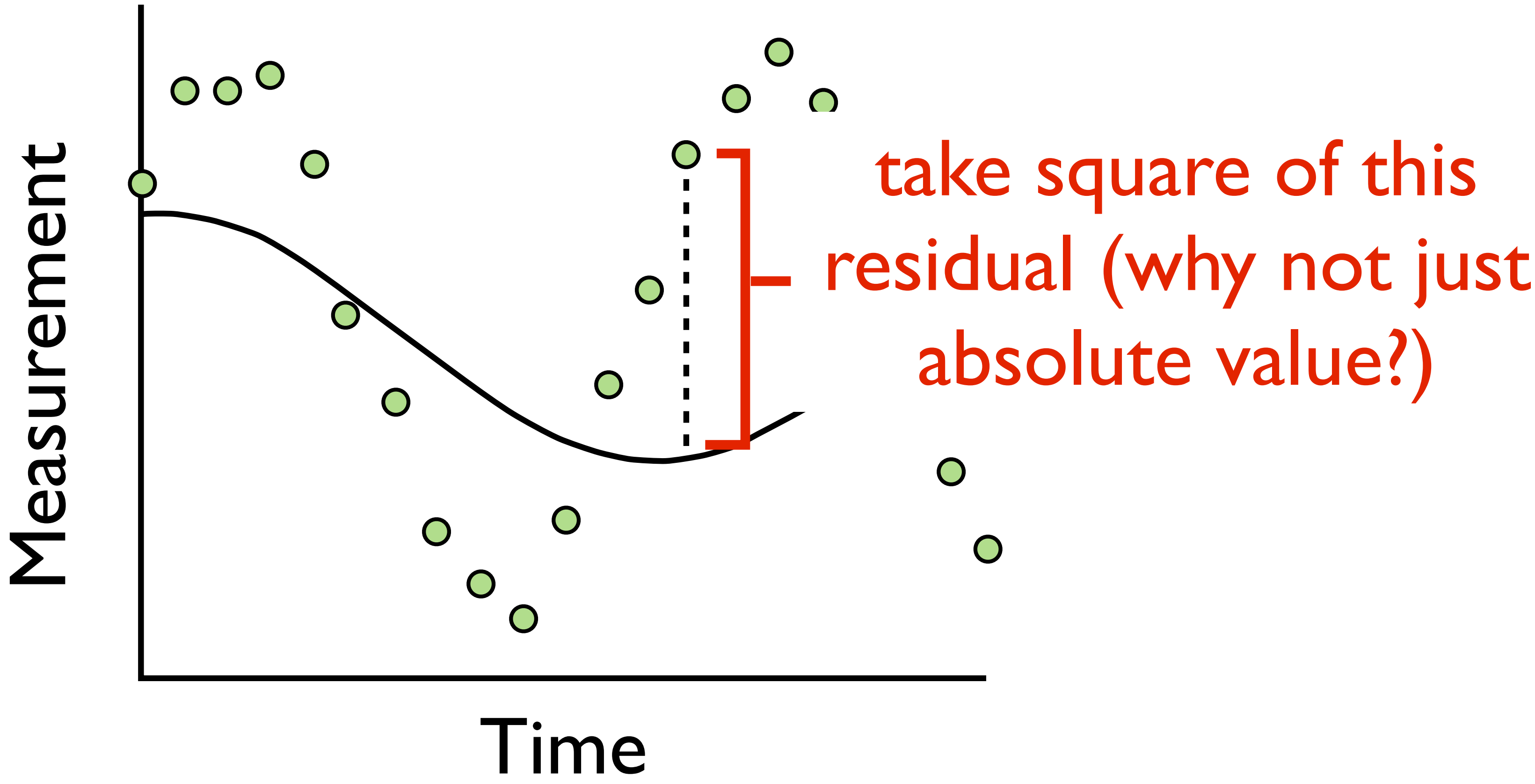
$A=1.01 \quad \omega=0.592 \quad \Phi=1.541$



e.g. consider we try fitting the following function with parameters:

$$f(t) = A \sin(\omega t + \Phi)$$

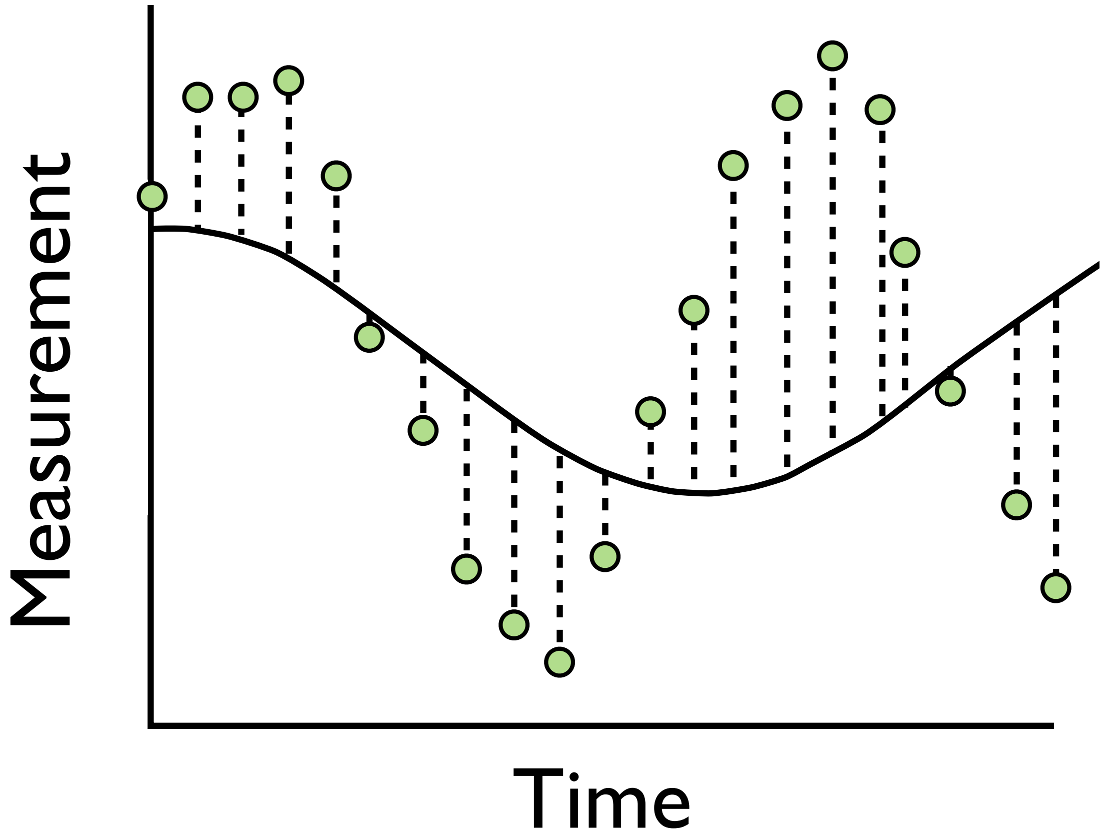
$A=1.01 \quad \omega=0.592 \quad \Phi=1.541$



e.g. consider we try fitting the following function with parameters:

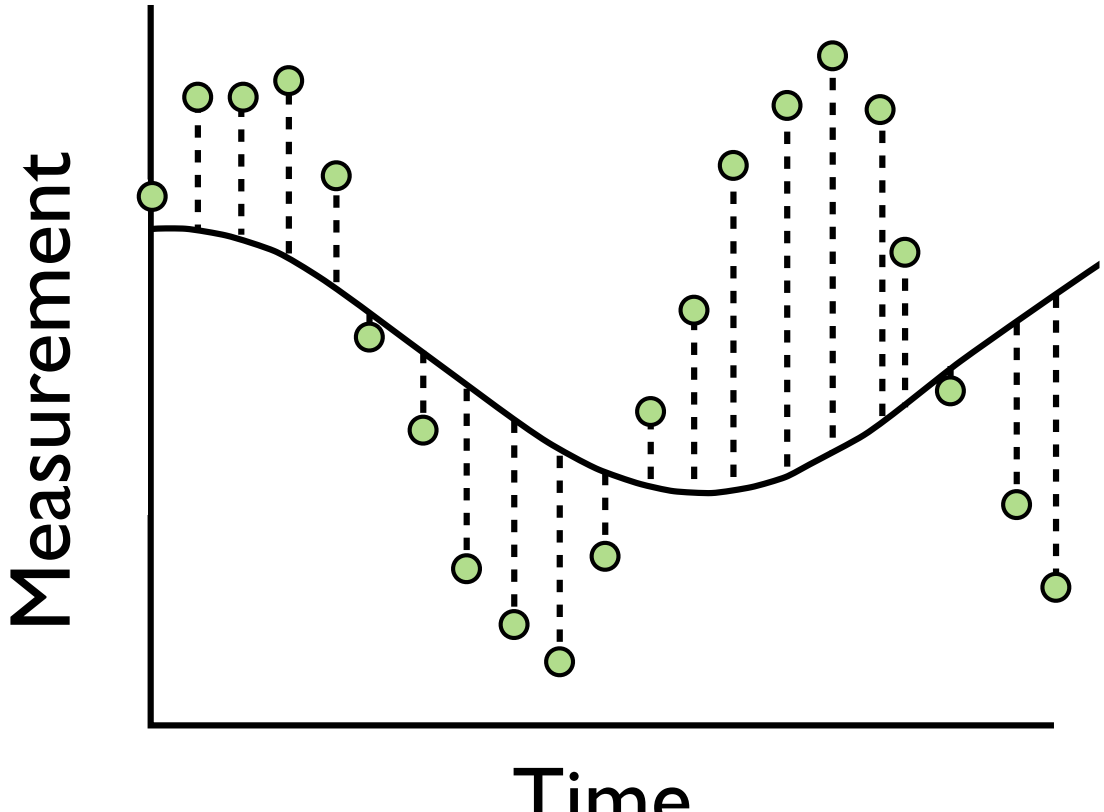
$$f(t) = A \sin(\omega t + \Phi)$$

$A=1.01 \quad \omega=0.592 \quad \Phi=1.541$



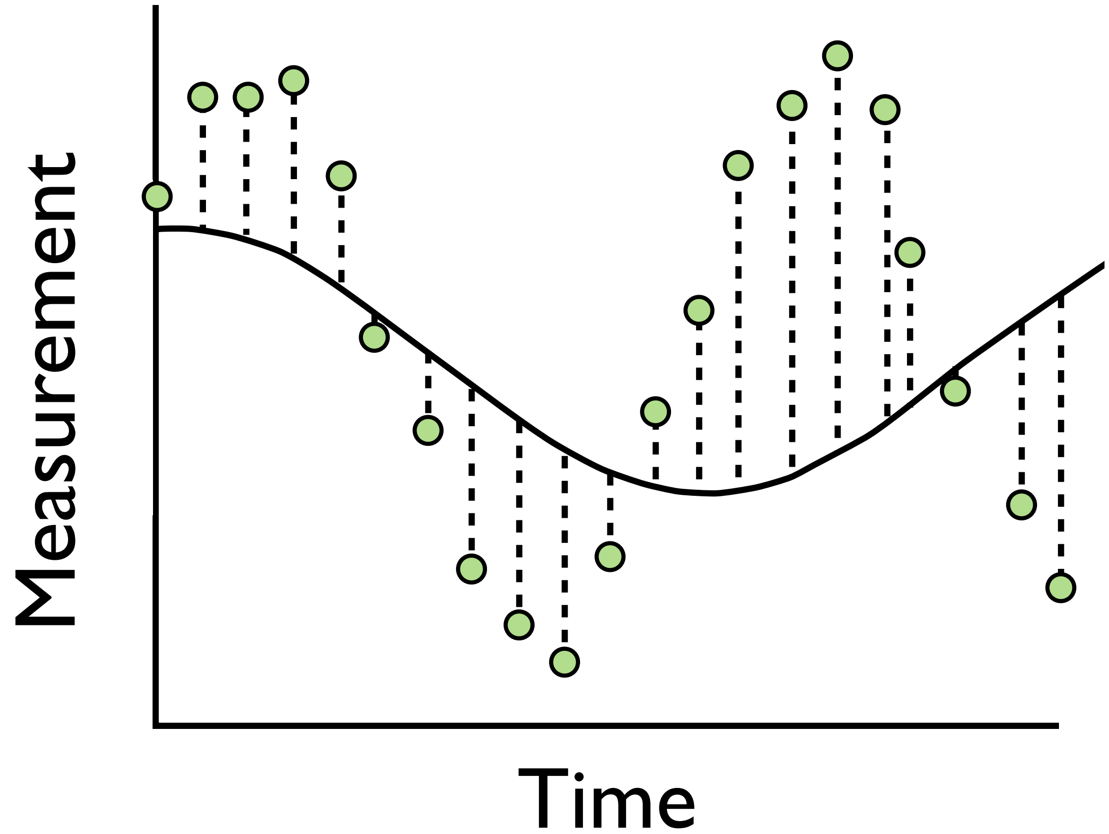
sum all "residual-squared" for all observation points

$$\sum_{j=1}^{m}(y_j - A\sin(\omega t_j + \Phi))^2$$



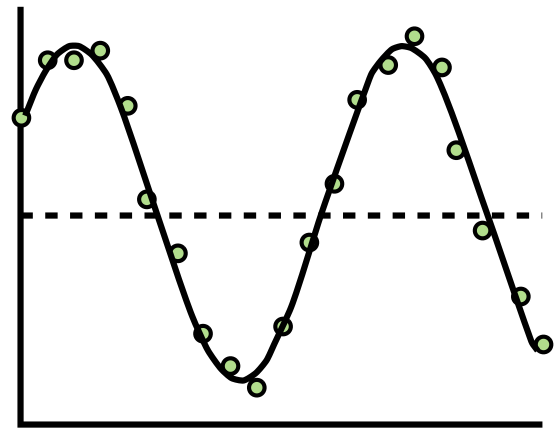
find parameters that **minimise** square residuals minimise $\sum_{j=1}^{m} r_i(X)^2$ ...where $r_j(X) = y_j - A \sin(\omega t_j + \Phi)$

...and $X=(A, \omega, \Phi)$



find parameters that **minimise** square residuals

SUMMARY Part 5. curve fitting



- curve fitting "least squares"
- next time: Gauss-Newton, approximate Hessian

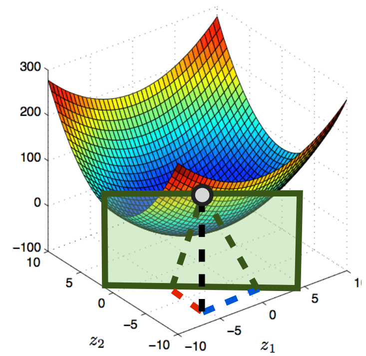
See the last slides for an (optional) homework exercise, get some practice with Hessians.

Together we did pseudo-code for 1D Newton root-finding.

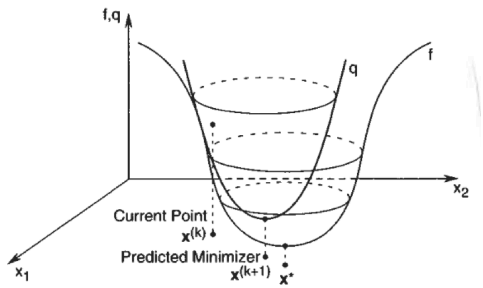
You could try implementing in MATLAB/Python:

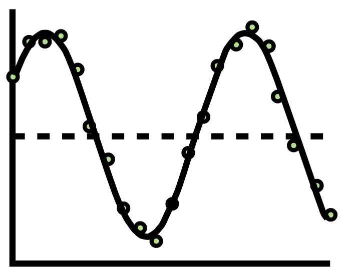
- 1D Newton root-finding
- nD Newton root-finding (Jakobian)
- 1D Newton optimisation
- nD Newton optimisation (Hessian)

SUMMARY



- preliminaries
- Newton *from* **1D** *to* **nD**
- *from* **root finding** *to* **optimization**
- *from* **Jakobian** *to* **Hessian**





exercise - some Newton optimization steps on this one:

$$X = (x_1,x_2,x_3,x_4)$$

$$f(X) = (x_1+10x_2)^2 + 5(x_3-x_4)^2 + (x_2-2x_3)^4 + 10(x_1-x_4)^4$$

exercise - some Newton optimization steps on this one:

$$X = (x_1, x_2, x_3, x_4)$$

$$f(X) = (x_1+10x_2)^2 + 5(x_3-x_4)^2 + (x_2-2x_3)^4 + 10(x_1-x_4)^4$$

STEP 1. partial derivatives

$\nabla f(X) =$

partial derivative of f w.r.t x_1 ($\partial f / \partial x_1$)
partial derivative of f w.r.t x_2 ($\partial f / \partial x_2$)
partial derivative of f w.r.t x_3 ($\partial f / \partial x_3$)
partial derivative of f w.r.t x_4 ($\partial f / \partial x_4$)

exercise - some Newton optimization steps on this one:

$$X = (x_1, x_2, x_3, x_4)$$

$$f(X) = (x_1+10x_2)^2 + 5(x_3-x_4)^2 + (x_2-2x_3)^4 + 10(x_1-x_4)^4$$

STEP 1. partial derivatives

$\nabla f(X) =$

$2(x_1 + 10x_2) + 40(x_1-x_4)^3$
partial derivative of f w.r.t x_2 ($\partial f / \partial x_2$)
partial derivative of f w.r.t x_3 ($\partial f / \partial x_3$)
partial derivative of f w.r.t x_4 ($\partial f / \partial x_4$)

exercise - some Newton optimization steps on this one:

$$X = (x_1, x_2, x_3, x_4)$$

$$f(X) = (x_1+10x_2)^2 + 5(x_3-x_4)^2 + (x_2-2x_3)^4 + 10(x_1-x_4)^4$$

STEP 1. partial derivatives

$$\nabla f(X) = \begin{bmatrix} 2(x_1 + 10x_2) + 40(x_1-x_4)^3 \\ 20(x_1+10x_2) + 4(x_2-2x_3)^3 \\ 10(x_3-x_4) - 8(x_2-2x_3)^3 \\ -10(x_3-x_4) - 40(x_1-x_4)^3 \end{bmatrix}$$

STEP 2. partial derivatives of each "partial derivative" (Hessian)

$\partial f/\partial x_1 =$	$2(x_1 + 10x_2) + 40(x_1 - x_4)^3$
$\partial f/\partial x_2 =$	$20(x_1 + 10x_2) + 4(x_2 - 2x_3)^3$
$\partial f/\partial x_3 =$	$10(x_3 - x_4) - 8(x_2 - 2x_3)^3$
$\partial f/\partial x_4 =$	$-10(x_3 - x_4) - 40(x_1 - x_4)^3$

	∂x_1	∂x_2	∂x_3	∂x_4
$\partial f/\partial x_1$	?	?	?	?
$\partial f/\partial x_2$				
$\partial f/\partial x_3$				
$\partial f/\partial x_4$				

STEP 2. partial derivatives of each "partial derivative" (Hessian)

$\partial f/\partial x_1 =$	$2(x_1 + 10x_2) + 40(x_1 - x_4)^3$
$\partial f/\partial x_2 =$	$20(x_1 + 10x_2) + 4(x_2 - 2x_3)^3$
$\partial f/\partial x_3 =$	$10(x_3 - x_4) - 8(x_2 - 2x_3)^3$
$\partial f/\partial x_4 =$	$-10(x_3 - x_4) - 40(x_1 - x_4)^3$

	∂x_1	∂x_2	∂x_3	∂x_4
$\partial f/\partial x_1$	$2 + 120(x_1 - x_4)^2$	20	0	$-120(x_1 - x_4)^2$
$\partial f/\partial x_2$				
$\partial f/\partial x_3$				
$\partial f/\partial x_4$				

STEP 2. partial derivatives of each "partial derivative" (Hessian)

$\partial f/\partial x_1 =$	$2(x_1 + 10x_2) + 40(x_1-x_4)^3$
$\partial f/\partial x_2 =$	$20(x_1+10x_2) + 4(x_2-2x_3)^3$
$\partial f/\partial x_3 =$	$10(x_3-x_4) - 8(x_2-2x_3)^3$
$\partial f/\partial x_4 =$	$-10(x_3-x_4) - 40(x_1-x_4)^3$

	∂x_1	∂x_2	∂x_3	∂x_4
$\partial f/\partial x_1$	$2 + 120(x_1-x_4)^2$	20	0	$-120(x_1-x_4)^2$
$\partial f/\partial x_2$	20	$200 + 12(x_2-2x_3)^2$	$-24(x_2-2x_3)^2$	0
$\partial f/\partial x_3$	0	$-24(x_2-2x_3)^2$	$10 + 48(x_2-2x_3)^2$	-10
$\partial f/\partial x_4$	$-120(x_1-x_4)^2$	0	-10	$10 + 120(x_1-x_4)^2$

STEP 3. do three iterations (X_1, X_2, X_3)

$$X_0 = [3, -1, 0, 1]$$

$$X^{(i+1)} = X^{(i)} - H(X^{(i)})^{-1} \nabla f(X^{(i)})$$

$\partial f/\partial x_1 =$	$2(x_1 + 10x_2) + 40(x_1 - x_4)^3$
$\partial f/\partial x_2 =$	$20(x_1 + 10x_2) + 4(x_2 - 2x_3)^3$
$\partial f/\partial x_3 =$	$10(x_3 - x_4) - 8(x_2 - 2x_3)^3$
$\partial f/\partial x_4 =$	$-10(x_3 - x_4) - 40(x_1 - x_4)^3$

	∂x_1	∂x_2	∂x_3	∂x_4
$\partial f/\partial x_1$	$2 + 120(x_1 - x_4)^2$	20	0	$-120(x_1 - x_4)^2$
$\partial f/\partial x_2$	20	$200 + 12(x_2 - 2x_3)^2$	$-24(x_2 - 2x_3)^2$	0
$\partial f/\partial x_3$	0	$-24(x_2 - 2x_3)^2$	$10 + 48(x_2 - 2x_3)^2$	-10
$\partial f/\partial x_4$	$-120(x_1 - x_4)^2$	0	-10	$10 + 120(x_1 - x_4)^2$