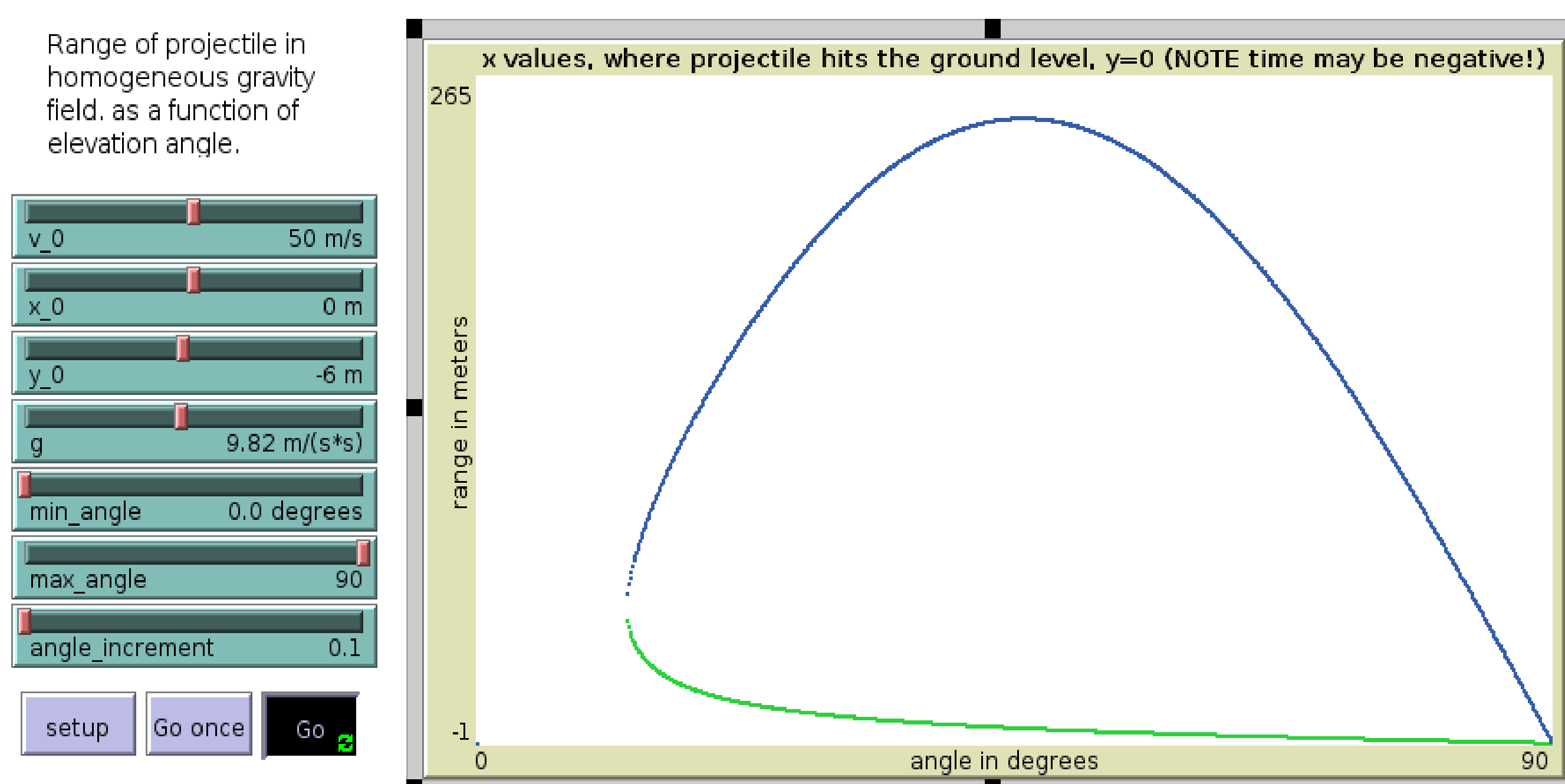


## Introduktion

Rækkevidden af et skråt kast, når affyringen sker over eller under jordniveau har ingen simpel løsning. Dette, i modsætning til affyring fra jordniveau, hvor det simpelt kan vises at den optimale affyringsvinkel er 45 grader. Hensigten med dette forløb er at undersøge den optimale affyringsvinkel og skuddets rækkevidde i tilfælde hvor affyringsniveauet kan være enten højere eller lavere end jordniveau.



## Mål med aktiviteten

- at undersøge rækkevidden af et skråt kast, når der affyres fra et niveau forskelligt fra jordniveau. Hvordan ændrer den optimale vinkel og rækkevidden når affyringshøjden bliver større, henholdsvis mindre end nul?
- at demonstrere nytten af programmering til løsning af fysiske problemer, hvor en analytisk løsning er vanskelig eller umulig.
- at træne den fysiske tolkning af multiple løsninger til matematiske problemer og disses grafiske fremstillinger.
- evt. at demonstrere forskelle og ligheder mellem forskellige programmeringssprog.

## Beskrivelse af aktiviteten

Eleverne skal, i grupper, skrive et program i et selvvalgt sprog som løser problemet. Elever med lille programmeringserfaring kan modificere et NetLogo program.

Programmet skal afprøves og anvendes til at få en dybere forståelse for det skrå kast.

I den forbindelse undersøges

- om programmet gør det forventede når der affyres fra jordniveau. Er den optimale vinkel 45 grader og passer rækkevidden?
- hvad sker der med den optimale vinkel, når der affyres over jordniveau? Hvad betyder de to løsninger til 2. gradsligningen og de tilhørende x-værdier?
- hvad sker der med den optimale vinkel når der affyres under jordniveau? Hvad betyder de to løsninger til 2. gradsligningen og de tilhørende x-værdier?
- Hvad betyder det, hvis der ikke er løsninger til 2. gradsligningen?

## Den centrale del af koden

```
globals [angle x_range_1 x_range_2 D]

to setup
  clear-all
  reset-ticks
end

to go-once
  set angle (min_angle + angle_increment * ticks)
  if (angle >= max_angle)[stop]
  let A ( -0.5 * g )
  let B (v_0 * sin( angle ))
  let C (y_0)
  set D (B * B - 4. * A * C)
  if (D >= 0) [
    let t_1 (( -1. * B + sqrt( D )) / ( 2. * A ))
    let t_2 (( -1. * B - sqrt( D )) / ( 2. * A ))
    set x_range_1 (v_0 * cos( angle ) * t_1 + x_0)
    set x_range_2 (v_0 * cos( angle ) * t_2 + x_0)
  ]
  tick
end

to go
  go-once
end
```

## Fremtidige perspektiver

- Der er gode muligheder for at diskutere udformningen af koden. Skal der f.eks. bruges en procedure til at løse 2. gradsligningen.
- Hvordan kan koden optimeres?
- Kan koden modificeres til at udskrive den optimale vinkel, så den ikke skal aflæses på grafen?

## Didaktiske overvejelser

Aktiviteten er designet til at

- demonstrere værdien af CT tilgangen til løsning af et problem som er meget svært at løse analytisk og som ligger indenfor det normale kernestof i fysik.
- være mulig at løse med næsten alle programmeringsværktøjer.
- invitere til skift i repræsentationsformer mellem matematisk beskrivelse, programkode og grafiske afbildninger med tilhørende fortolkninger.
- være interessant og tilgængelig for elever med vidt forskellig programmeringserfaring.

## Kreditering

NetLogo-modellen og undervisningsmaterialet er udviklet af [Peter Snoer Jensen](#), H.C. Ørsted Gymnasiet, Frederiksberg, TEC, i forbindelse med deltagelse i udviklingsprojektet Computational Thinking i Matematik og Naturfag i skoleåret '18/'19. Projektet køres i samarbejde mellem Danske Science Gymnasier og Center for Computational Thinking & Design, Aarhus Universitet.