

NetLogo-simuleringen

Simuleringer og fysiske modeller (henfaldsloven)

```
breed [N1s N1]
breed [N2s N2]

globals [
  Sandsynlighed-for-henfald-N1
  Startantal-N1
]

to setup
  clear-all

  set-default-shape N1s "circle"
  set-default-shape N2s "circle"

  set Sandsynlighed-for-henfald-N1
  set Startantal-N1 8000

  create-N1s Startantal-N1
  [set color cyan
   move-to one-of patches with [
]]

  reset-ticks
end

to go
  if not any? N1s [ stop ]
  kor-henfald1
  tick
end

to kor-henfald1
  ask N1s [
    if random-float 100.0 < Sandsynlighed-for-henfald-N1
    set breed N2s
    set color red
  ]
end
```

HVAD ER DETTE?

Dette er et lille program, der simulerer radioaktivt henfald. De ustabile kerner N1 henfalder til de stabile kerner N2.

N1 → N2 (henfald)

Antallet af N1-kerner og N2-kerner kan følges i de to grafer samt i hovedvinduet, hvor de vises som henholdsvis lyseblå og røde prikker.

Det er meningen, at du skal lave små ændringer og forbedringer i programmet, så det passer til de problemer, som du skal løse. Du skal også udbygge programmet, så det kan simulere mere komplekse scenarier. Du kommer altså til at lave ændringer i programkoden.

HVORDAN DET VIRKER

Ved hvert tidsstep ("tick") – som for eksempel kan repræsentere ét sekund eller ét år – har hver N1-kerne, der endnu ikke er henfaldet, en hvis sandsynlighed for at henfalde. Når en N1-kerne henfalder, skifter den farve fra lyseblå til rød. Simuleringen stopper, når alle N1-kerner er henfaldet.

MODELLEN BAG SIMULERINGEN

Modellen i simuleringen bygger på en enkelt simpel antagelse:

For hvert tidsstep ("tick") har hver N1-kerne en given sandsynlighed for at henfalde.

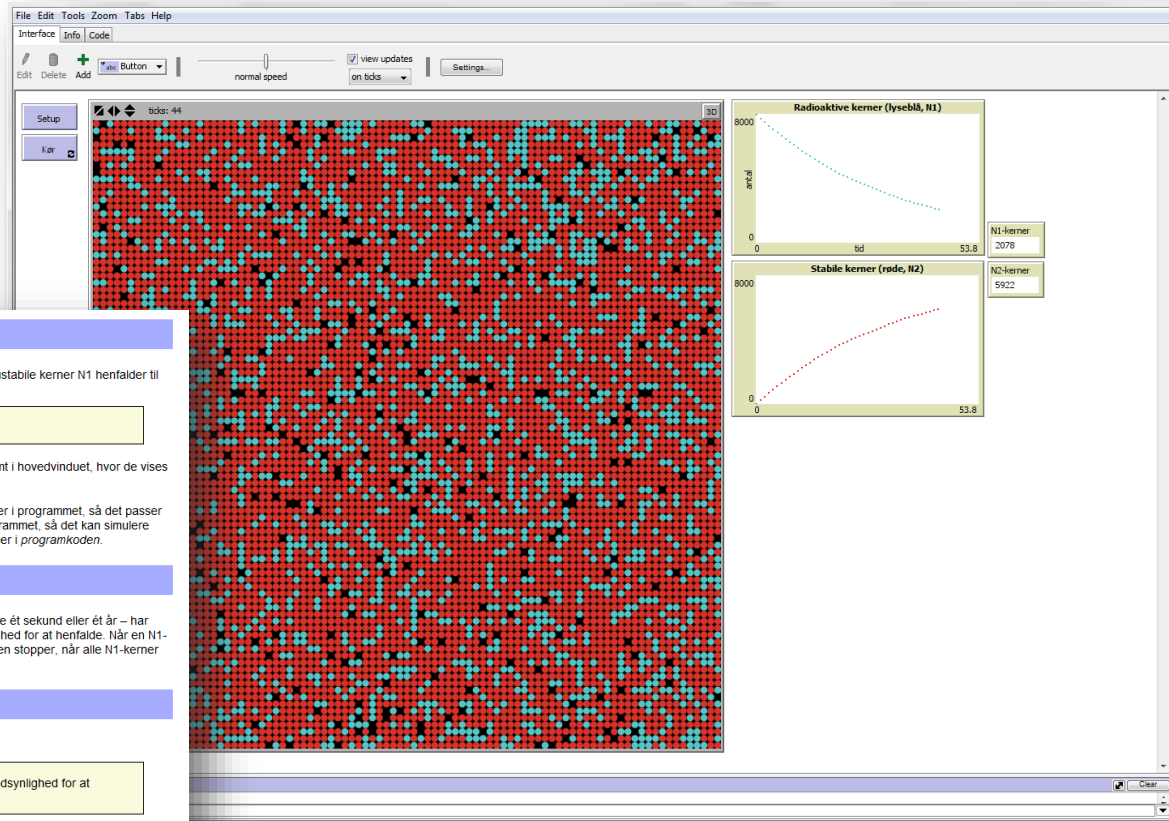
Det er denne antagelse, som vi med simuleringen ønsker at teste. Det gør vi ved at sammenligne resultaterne fra simuleringen med virkeligheden, dvs. med resultaterne fra et virkeligt forsøg med radioaktivt henfald.

Vær opmærksom på at vores antagelse kunne have været anderledes, fx "En N1-kerne er 10 sekunder om at henfalde" eller "Der henfalder 50 N1-kerner per sekund". I så fald ville graferne (sammenhængen mellem tid og antal kerner) se meget anderledes ud.

HVORDAN MAN BRUGER PROGRAMMET

Sandsynligheden sættes til den ønskede værdi i programkoden:

```
set Sandsynlighed-for-henfald-N1 3
```



Hvad er en simulering?

For at kunne arbejde med en simulering er der to vigtige elementer, man må have en grundlæggende forståelse for:

1. Programmering: At læse og skrive computerkode/programkode – det sprog der skrives for at få computeren til at køre simuleringen.
2. De fysiske/naturvidenskabelige modeller, der beskriver fænomenet/processen, der simuleres. Modellerne ligger i maskinrummet af simuleringen og bestemmer, hvad der skal ske i simuleringen.

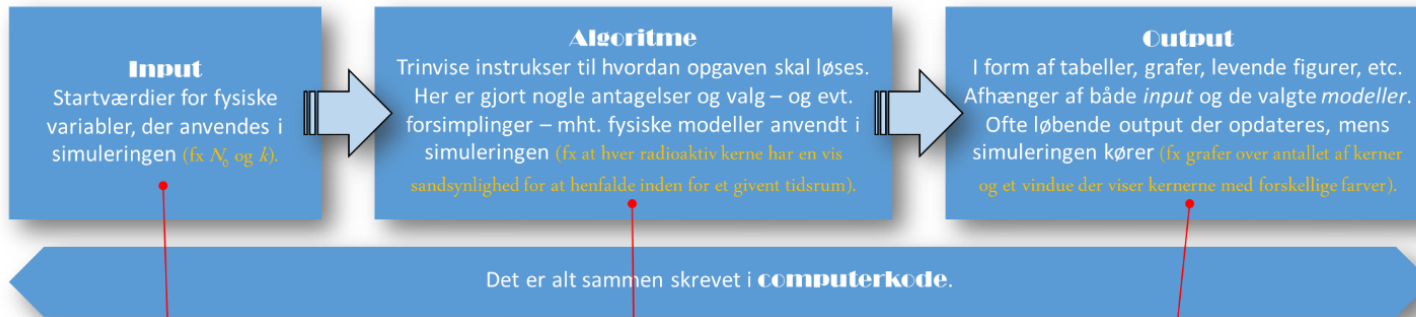
Modellen eller modellerne er skrevet ind i computerkoden. *Vil man ændre sin model for det fysiske system i simuleringen, må man altså ændre i computerkoden.* Det var det i gjorde med de forskellige arbejdsark, og det er derfor, det er vigtigt, at I kan læse computerkode og rette i den, så den passer til jeres situation.

Resultaterne af simuleringen (outputtet) afhænger af ens *startværdier* (input) og de anvendte *modeller*. Det er derfor vigtigt at gøre sig klart hvad de anvendte modellers styrker og svagheder er – og vælge de modeller, der passer til det, som man ønsker at undersøge.

Hvad er en simulering?

En simulering består som figuren viser ofte af et input, en model og et output.

En computer-simulering (til undersøgelse af en fysisk problemstilling):



```
set Sandsynlighed-for-henfald-N1 3
set Startantal-N1 8000
```

```
breed [N1s N1]
breed [N2s N2]

globals [
  Sandsynlighed-for-henfald-N1
  Startantal-N1
]

to setup
  clear-all

  set-default-shape N1s "circle"
  set-default-shape N2s "circle"

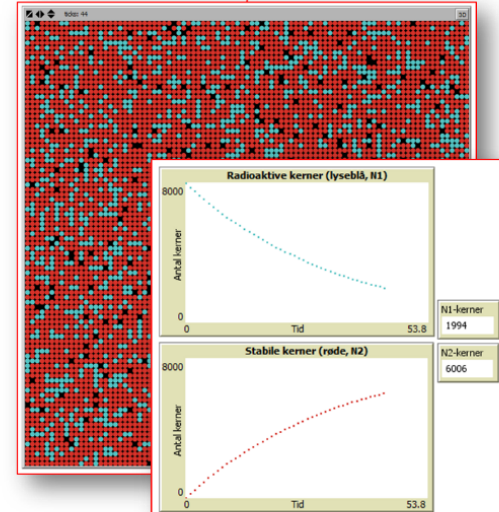
  set Sandsynlighed-for-henfald-N1 3
  set Startantal-N1 8000

  create-N1s Startantal-N1
  [set color cyan
  move-to one-of patches with [ not any? turtles-here ]
  ]

  reset-ticks
end

to go
  if not any? N1s [ stop ]
  køn-henfald1
  tick
end

to køn-henfald1
  ask N1s [
    if random-float 100.0 < Sandsynlighed-for-henfald-N1 [
      set breed N2s
      set color red
    ]
  ]
end
```



Netlogo-simuleringen og koden

Tre faneblade:

- *Interface*: Brugerflade som man bruger til at starte og følge simuleringen.
- *Info*: Information om simuleringen, hvad der simuleres og hvordan den bruges
- *Code*: Simuleringens maskinrum (som man kan ændre i og tilpasse) – det er her ens modeller og antagelser ligger gemt (og dermed også viden om simuleringens begrænsninger)

Instruktioner:

Interface-fanen

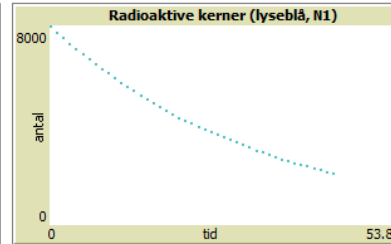
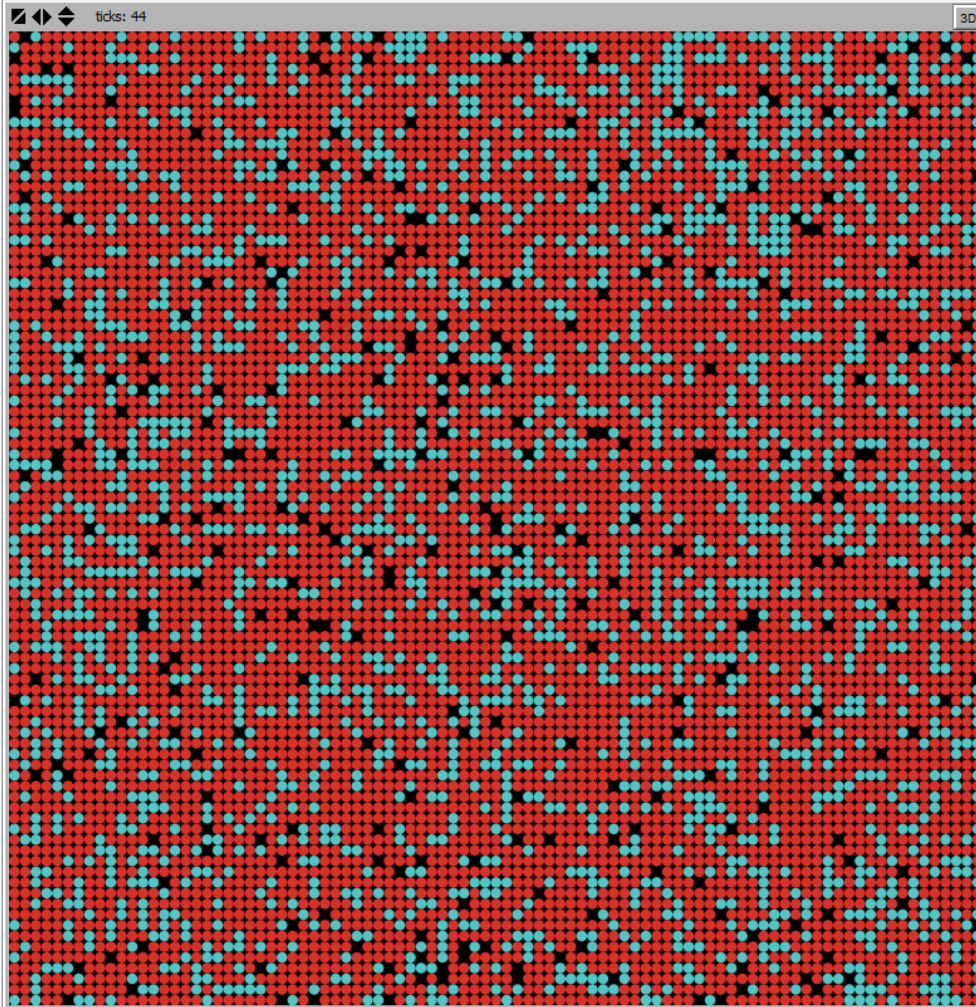
- Tilføj knapper og grafer med mere (ved at højreklikke)
- Knappen "Settings"
- "speed" og "view updates"
- Knappen "Setup" og knappen "Kør"
- Begrebet "ticks" (tids-tælleren)

Code-fanen

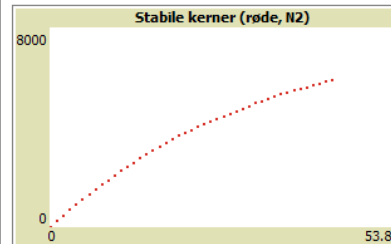
- To semikoloner før tekst
- Variabler
- Globale variabler (globals)
- Løkker
- Forgreninger (hvis-så-ellers)
- Tælleren "ticks"
- Agenter (turtles, patches, mm) og arter (breed)
- Procedurer (fra "to" til "end")
- Indentering
- "to setup" og "to go" (flueben i "Forever")
- "Check"-knappen
- Højreklik og Quick Help
- Brug . (punktum) i stedet for , (komma)

Edit Delete Add abc Button | normal speed | view updates | on ticks | Settings...

Setup
Kør



N1-kerner
2078



N2-kerner
5922



HVAD ER DETTE?

Dette er et lille program, der simulerer radioaktivt henfald. De ustabile kerner N1 henfalder til de stabile kerner N2:

N1 → N2 (henfald)

Antallet af N1-kerner og N2-kerner kan følges i de to grafer samt i hovedvinduet, hvor de vises som henholdsvis lyseblå og røde prikker.

Det er meningen, at du skal lave små ændringer og forbedringer i programmet, så det passer til de problemer, som du skal løse. Du skal også udbygge programmet, så det kan simulere mere komplekse scenarier. Du kommer altså til at lave ændringer i *programkoden*.

HVORDAN DET VIRKER

Ved hvert tidsstep ("tick") – som for eksempel kan repræsentere ét sekund eller ét år – har hver N1-kerne, der endnu ikke er henfaldet, en hvis sandsynlighed for at henfalde. Når en N1-kerne henfalder, skifter den farve fra lyseblå til rød. Simuleringen stopper, når alle N1-kerner er henfaldet.

MODELLEN BAG SIMULERINGEN

Modellen i simuleringen bygger på en enkelt simpel antagelse:

For hvert tidsstep ("tick") har hver N1-kerne en given sandsynlighed for at henfalde.

Det er denne antagelse, som vi med simuleringen ønsker at teste. Det gør vi ved at sammenligne resultaterne fra simuleringen med virkeligheden, dvs. med resultaterne fra et virkeligt forsøg med radioaktivt henfald.

Vær opmærksom på at vores antagelse kunne have været anderledes, fx "En N1-kerne er 10 sekunder om at henfalde" eller "Der henfalder 50 N1-kerner per sekund". I så fald ville graferne (sammenhængen mellem tid og antal kerner) se meget anderledes ud.

HVORDAN MAN BRUGER PROGRAMMET

Sandsynligheden sættes til den ønskede værdi i programkoden:

```
set Sandsynlighed-for-henfald-N1 3
```

```

breed [N1s N1] ;; Definerer en ny art af kerner med kaldenavnet "N1s" (breed = art, inspireret af dyreverdenen, hvor man fx kunne have ulve og får). I d
breed [N2s N2] ;; Definerer en ny art af kerner med kaldenavnet "N2s". I dette tilfælde er det de stabile datterkerner som N1s henfalder til.

globals [ ;; "globals" er variabler, hvis værdi kan tilgås og ændres overalt i programmet, dvs. fra alle procedurer og fra fanen Interface - variab
Sandsynlighed-for-henfald-N1 ;; Sandsynligheden for at en N1-kerne henfalder per tidsenhed ("tick")
Startantal-N1 ;; Antal N1-kerner i starten af simuleringen
]

to setup ;; Setup procedure (opsætning eller startprotokol) - køres når der trykkes på knappen "Setup" under fanen "Interface"
clear-all ;; Nulstiller alt fra tidligere kørsler af programmet (værdier af variabler og ticks, turtles, patches, grafer, vinduer mm.)

set-default-shape N1s "circle" ;; Udseendet af N1-kerner sættes til at være cirkler.
set-default-shape N2s "circle" ;; Udseendet af N2-kerner sættes til at være cirkler.

set Sandsynlighed-for-henfald-N1 3 ;; Sætter sandsynligheden for henhald per tidsstep til 3 %
set Startantal-N1 8000 ;; Sætter startantallet af N1-kerner til 8000 (kan være max 8280)

create-N1s Startantal-N1 ;; Opretter et antal N1-kerner. Antallet er den den værdi som variabelen "Startantal-N1" har.
[set color cyan ;; Farven af N1-kernerne sættes til cyan.
move-to one-of patches with [ not any? Turtles-here ] ;; Hver er de nye N1-kerner placeres på et felt (patch), hvor der ikke i forvejen er nogle turtles ("Turtles" er en fællesbetegnelse for
]

reset-ticks ;; Nulstiller værdien af tælleren "ticks", så den starter med at tælle fra nul igen (og opretter plots og opdaterer diverse værdier)
end

to go ;; Hovedløkken der kører/styrer programmet. Denne procedure startes når der trykkes på "Kør" under fanen "Interface" og gentages igen og
if not any? N1s [ stop ] ;; Stopper programmet, hvis der ikke er flere N1-kerner
kør-henfald1 ;; Kører proceduren "kør-henfald1" nedenfor
tick ;; Værdien af tælleren "ticks" øges med 1
end

to kør-henfald1 ;; Procedure der gør, at nogle af N1-kernerne henfalder til N2-kerner. Det er i denne procedure at vores model/antagelse får betydning.
ask N1s [ ;; Spørger alle N1-kerner om de skal henfalde, og hvis det er tilfældet (afhænger af sandsynligheden), så udføres de handlinger der skal
if random-float 100.0 < Sandsynlighed-for-henfald-N1 [ ;; Der vælges et tilfældigt tal for hver af N1-kernerne. Er tallet mindre end den angivende sandsynlighed for at en N1-kerne henfalder, så
set breed N2s ;; Hvis kernen henfalder, så laves den om til en N2-kerne ...
set color red ;; ... og får farven rød.
]
end

```

Programkoden: Definition af arter og globale variabler

```
breed [N1s N1]
breed [N2s N2]

globals [
  Sandsynlighed-for-henfald-N1
  Startantal-N1
]
```

Her defineres de *arter* og de *globale variabler* der skal bruges i løbet af programmet.

En *art* (breed = art) betegner de elementer i vores program, der kan *gøre noget*. Navnet er inspireret af dyreverdenen, hvor man fx kunne have ulve og får. I vores program har vi to arter, henholdsvis de radioaktive moderkerner (N1) og de stabile datterkerner (N2).

En *global variabel* er en variabel, hvis værdi kan tilgås og ændres overalt i programmet, dvs. fra alle *procedurer* og fra fanen Interface. Variabler, der ikke erklæres som globale kan kun tilgås og ændres i den procedure, hvor de er blevet oprettet (en *procedure* er en stump kode der starter med "to" og slutter med "end")

Algoritme

- **Algoritme:** En trinvis beskrivelse af hvordan en opgave skal løses. En algoritme er altså en "opskrift", der fortæller trin for trin hvad computeren skal gøre. Den er skrevet med computerkode, men kan overskueliggøres med såkaldte flowcharts (rutediagrammer). Instruktionerne til computeren skal være utvetydige, ellers forstår computeren dem ikke, da computeren ikke kan tænke selv.

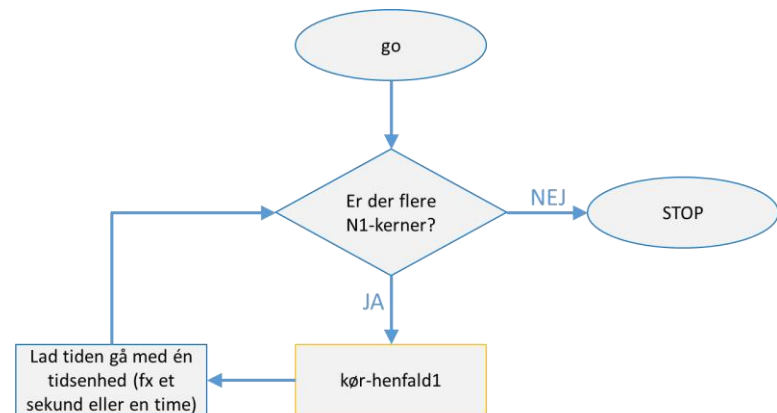
I kom selv med et par gode eksempler på dagligdags algoritmer:

En madopskrift

En IKEA samlevejledning.

```
to go
  if not any? N1s [ stop ]
    kør-henfald1
    tick
  end
```

Et udsnit af computerkoden i simuleringen



Et rutediagram svarende til computerkoden til venstre. Her er det tydeligt at se de enkelte trin, der udføres (algoritmen).

Programkoden: Klargøring af simulering (setup)

```
to setup
  clear-all

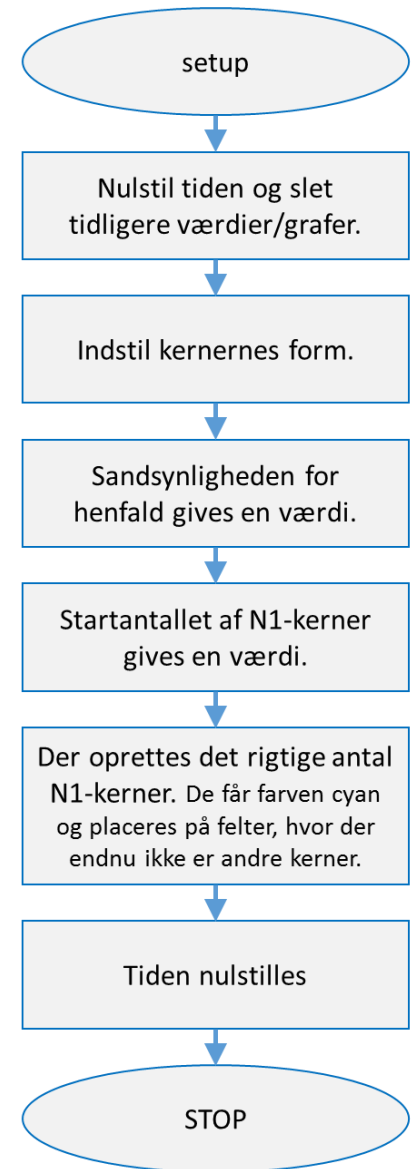
  set-default-shape N1s "circle"
  set-default-shape N2s "circle"

  set Sandsynlighed-for-henfald-N1 3
  set Startantal-N1 8000

  create-N1s Startantal-N1
    [set color cyan
     move-to one-of patches with [ not any? Turtles-here ]
    ]

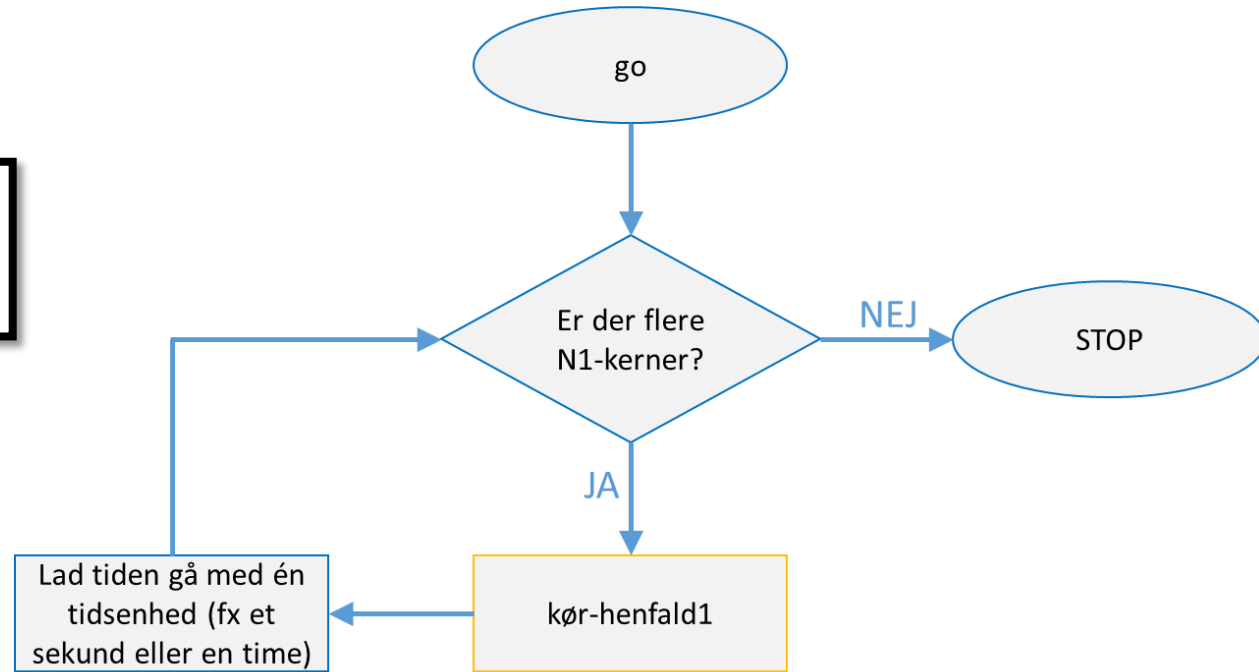
  reset-ticks
end
```

Her gøres simuleringen klar. Vigtige variabler sættes til de ønskede værdier og N1-kernerne oprettes.

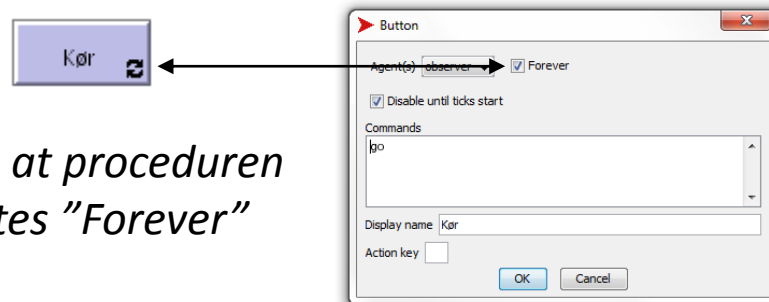


Programkoden: Hovedløkken (go)

```
to go
  if not any? N1s [ stop ]
  kør-henfald1
  tick
end
```



Denne løkke gentages indtil simuleringen får besked på at stoppe. Ved hvert gennemløb går tiden med én tidsenhed)



Her kan man se at proceduren "to go" fortsættes "Forever"

Programkoden: Henfald af N1-kerner

```
to kør-henfald1
  ask N1s [
    if random-float| 100.0 < Sandsynlighed-for-henfald-N1 [
      set breed N2s
      set color red
    ]
  ]
end
```

