

MILJØVERNDIREKTORATET I CELANDIA

Modellering av Bjarnøyas Økosystem

Dr. Hans Ekkehard Plesser

MILJØVERNDIREKTØR

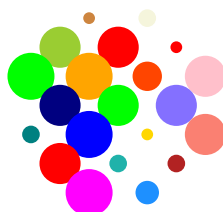
Eilen Nordlie, MSc

SPESIALRÅDGIVER

Håkon Enger, dr. scient.

TEKNISK KONSULENT

25. februar 2008



Prosjektoppgave i Programmering II ved IMT/UMB, Våren 2008.

Innhold

1	Innledning	3
1.1	Oppgave	4
2	Naturen på Bjarnøya	6
2.1	Geografi	6
2.1.1	Hav og fjell	6
2.1.2	Ørken	7
2.1.3	Savanne	7
2.1.4	Jungel	8
2.2	Fauna	9
2.2.1	Byttedyr	11
2.2.2	Rovdyr	11
2.3	Årets løp på Bjarnøya	12
2.4	Førtilbud i det første året	12
3	Programkjøring og materialpakke	13
3.1	Kjøring av programmet	13
3.2	Generering av inputfiler	13
3.3	Code::blocks-pakke	14
3.4	Matlab-verktøy	14
3.5	C++ toolbox	15
3.6	Doxygen dokumentasjon	15
4	Filformater	16
4.1	Spesifikasjonsfil	16
4.2	Geografi	17
4.3	Parameter	18
4.4	Populasjon	18
4.4.1	Populasjonsfil som utdata	19
4.5	Utdatafiler	20
4.5.1	Befolkningsfil	20

<i>INNHold</i>	2
4.5.2 Dyrefordeling	20
4.5.3 Fôrfordeling	21
5 Veiledning	22

Kapittel 1

Innledning

Bjarnøya er en liten øy midt i det store havet som tilhører arkipelstaten Celandia. Økosystemet på Bjarnøya er forholdsvis uforstyrret, men Celandias miljøverndirektorat ønsker å utrede stabiliteten av økosystemet. Det langsiktige målet er å sikre Bjarnøya som naturpark for fremtidige generasjoner.

Økosystemet på Celandia kjennetegnes av flere forskjellige landskaps typer, som jungel, savanne, fjell, og ørken. Faunaen omfatter bare to dyrearter, en planteetende art (bytte), og en rovdyrart (rovdyr). Det skal utredes om begge artene kan overleve på sikt. En detaljert beskrivelse av Celandias geografi og fauna finnes i kapittel 2. De viktigste karakteristikkene er

Bytte er en planteetende dyreart, som er avhengig av god tilgang på mat for å overleve og reproducere.

Rovdyr er avhengig av tilgang på bytte. Rovdyrene er noe mer mobile enn byttedyrene.

Jungel tilbyr store mengder fôr selv ved intensiv beiting.

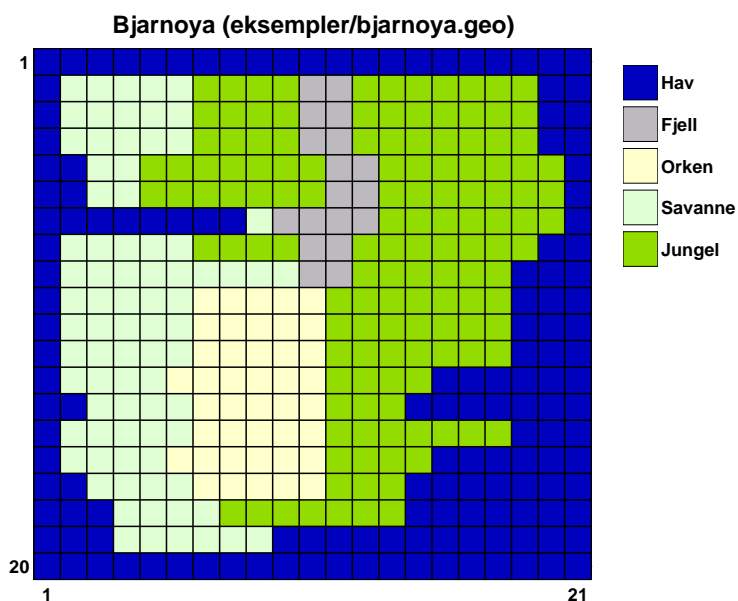
Savanne kan ødelegges ved overbeiting.

Ørken tilbyr ikke fôr for byttedyr.

Fjell er ikke mulig å passere for hverken bytte eller rovdyr.

Hav er heller ikke mulig å passere.

Et kart over Bjarnøya vises i figur 1.1.



Figur 1.1: Landskapstyper på Bjarnøya ved siste oppmåling.

1.1 Oppgave

Miljøverndirektoratet i Celandia oppfordrer forskergrupper til å utvikle dataprogrammer for simulering av populasjonsdynamikken på Bjarnøya. Miljøverndirektoratet forventer at

- **grupper på to eksperter**
- utvikler en **populasjonsdynamikksimulering**
- innen **onsdag, 7. mai 2008, kl 1500**.

Miljøverndirektoratet i Celandia vil holde fortløpende informasjonsseminarer om oppgaven under prosjektperioden. Det henvises spesielt til ukentlige konferanser med teknisk konsulent H. Enger

- mandager, kl 0815–1000
- onsdager, kl 1015–1200
- fredager, kl 1415–1600

som avholdes i rom TF02.

Miljøverndirektoratet forventer delrapporter i henhold til følgende milepæler:

Dato	Milepæl
14. mars 2008	Løsningsskisse (datastrukturer, algoritmer, evt pseudokode).
7. april 2008	Simulering uten romlig utstrekning.
25. april 2008	Simulering med romlig utstrekning, dog uten differansiering av landskapstyper.
7. mai 2008	Fullstendig simulering med dokumentasjon.

Ekspertgruppene skal innen fristen 7. mai 2008 levere kildekoden samt dokumentasjon til direktoratet. Gruppene skal presentere sine resultater ved et offentlig symposium som avholdes 14. mai. I sammenheng med symposiet avholdes det også individuelle vurderingssamtaler med ekspertene. Detaljene legges frem ved et senere tidspunkt.

Direktoratet oppretter en egen nettside, der det informeres om prosjektet. På denne nettsiden vil det om kort tid legges ut hjelpemidler for arbeidet med oppgaven.

Kapittel 2

Naturen på Bjarnøya

2.1 Geografi

Bjarnøya er inndelt i plankvadrater, og hvert plankvadrat er en av følgende fem typer:

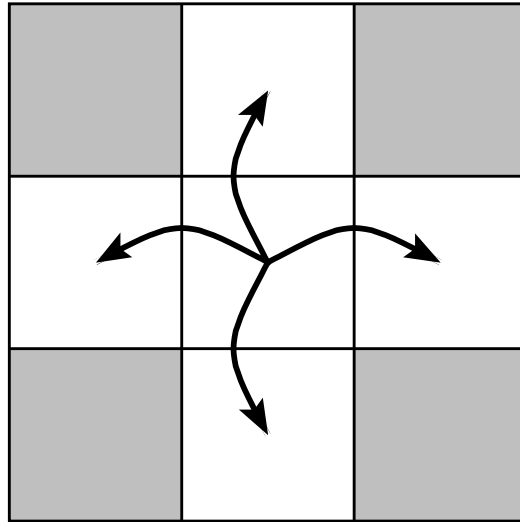
- hav,
- fjell,
- ørken,
- savanne,
- jungel.

Plankvadratene betegnes også som *celler*.

Som øy er Bjarnøya naturligvis omringet av hav: ytterkantene av kart over Bjarnøya (jf fig. 1.1) viser derfor alltid plankvadrat av type "Hav". Dyr kan bare bevege seg fra det plankvadratet de befinner seg i til et av de fire nærmeste nabokvadratene, se fig. 2.1. Intet dyr kan oppholde seg i havet eller til fjells.

2.1.1 Hav og fjell

Hverken hav eller fjell kan passeres av dyr. Celler av disse to typene vil derfor være helt passive i simuleringen. Hvis et dyr prøver å ta seg inn i enten hav eller fjell, vil det treffe på en vegg, og forbli i den cellen det befinner seg i fra før.



Figur 2.1: Et dyr som befinner seg i den midterste cellen kan vandre til en av de fire nabocellene, men ikke til celler som ligger diagonalt (grått).

2.1.2 Ørken

Dyr kan oppholde seg i ørkenen, men det finnes ikke noe fôr for byttedyr der. Rovdyr kan felle byttedyr i ørkenen.

2.1.3 Savanne

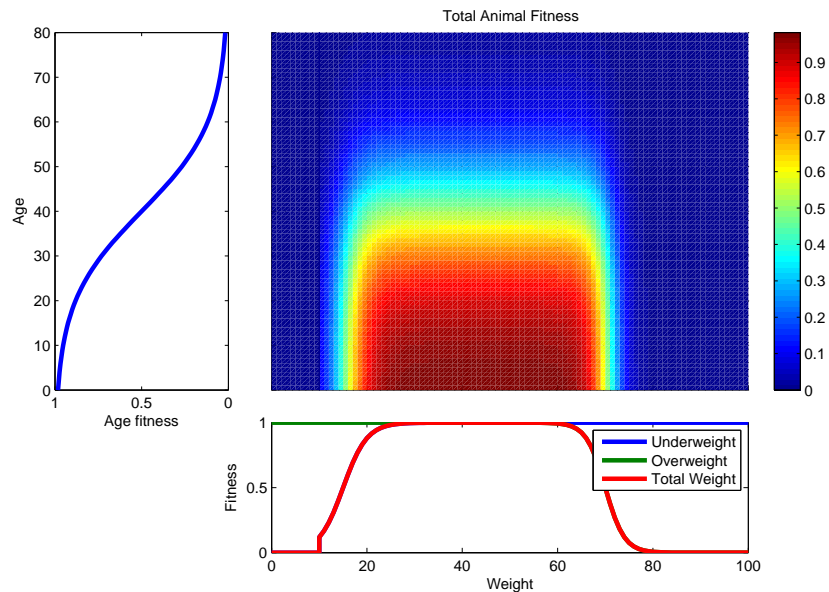
Savannen tilbyr fôr for byttedyr i begrenset mengde, og er ømfintlig for overbeiting. I hver celle finnes en bestemt mengde fôr f_{ij} , der indekserne (i, j) bestemmer cellen. Hver gang et byttedyr spiser, prøver dyret å fortære en viss fôrmengde F . Følgende “spiseregler” gjelder:

- hvis** $F \leq f_{ij}$
så finnes det nok fôr, dyret får i seg fôrmengden F og
fôrmengden f_{ij} i cellen reduseres med F
- hvis** $0 < f_{ij} < F$
får dyret i seg det som er igjen av fôr, dvs f_{ij} , og
 f_{ij} settes lik 0
- hvis** $f_{ij} = 0$
får ikke dyret noe mat.

Hvert år vokser nytt fôr til etter formelen

$$f_{ij} \leftarrow f_{ij} + \alpha \times (f_{\max}^{\text{Sav}} - f_{ij}) . \quad (2.1)$$

Rovdyr kan felle byttedyr på savannen.



Figur 2.2: Fitnessfunksjon: Grafikken til venstre viser aldersrelatert fitness, figuren nederst over- og undervektskomponent i vektrelatert fitness, samt samlet effekt av disse to. Den store figuren viser total fitness basert på alder og vekt.

2.1.4 Jungel

I jungelen gjelder i utgangspunktet de samme “spisereglene” som i savannen, jf avsnitt 2.1.3. Tilvekst av fôr i jungelen er dog så rask, at jungelen ikke er utsatt for langvarig skade pga overbeiting. I begynnelsen av hvert år er fôrmengden

$$f_{ij} \leftarrow f_{\max}^{\text{Jungle}} \quad (2.2)$$

tilgjengelig. Rovdyr kan felle byttedyr i jungelen.

2.2 Fauna

Rovdyr og byttedyr har bestemte egenskaper til felles:

1. Alder	Ved fødsel har hvert dyr alder $a = 0$. Alderen øker med ett år for hvert år som går.
2. Vekt	Dyr fødes med vekt $v = v_{\text{fod}}$. Når et dyr spiser en førmengde F , øker vekten med βF . Hvert år avtar vekten med σv .
3. Fitness	Dyrets allmenntilstand beskrives av sin fitness, som beregnes utifra alder og vekt etter følgende formel

$$\Phi = \begin{cases} 0 & \text{hvis } v < v_{\min} \\ q^+(a; a_{\frac{1}{2}}, \phi^{\text{alder}}) \\ \quad \times q^-(v; v_{\frac{1}{2}}^{\text{under}}, \phi^{\text{under}}) & \text{ellers} \\ \quad \times q^+(v; v_{\frac{1}{2}}^{\text{over}}, \phi^{\text{over}}) \end{cases} \quad (2.3)$$

med funksjonen

$$q^{\pm}(x; x_{\frac{1}{2}}, \phi) = \frac{1}{1 + e^{\pm\phi(x-x_{\frac{1}{2}})}} \cdot \quad (2.4)$$

Det vises til at $0 \leq \Phi \leq 1$. En typisk fitnessfunksjon vises i fig. 2.2.

4. Vandring	Med sannsynlighet $\mu\Phi$ vandrer dyret til en nabocelle. Det velges med lik sannsynlighet mellom alle nabocellene. Hvis den cellen som er valgt som mål er hav eller fjell, forblir dyret i sin celle.
-------------	---

5. Fødsel Dyr kan pare seg hvis det finnes minst to dyr av samme art i en celle.

- For hvert dyr i cellen er sannsynligheten for å føde et avkom i løpet av et år

$$\gamma\Phi \times (N - 1) , \quad (2.5)$$

der N er antall dyr av samme art i starten av formeringssesongen.

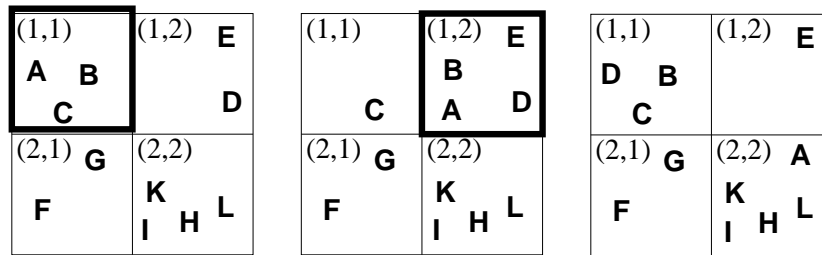
- Sannsynlighet er altså 0 hvis bare ett individ av arten er i cellen
- Sannsynlighet for fødsel er også 0 hvis vekt $v < v_{\min} + \zeta v_{\text{fod}}$
- Hvis (2.5) gir en sannsynlighet større en 1, settes sannsynlighet til 1, dvs moderdyret føder med sikkerhet.
- Det antas at kjønnet ikke spiller noen rolle for paringen.
- Hvert dyr kan føde maks ett barn per år.
- Ved fødsel trekkes ζv_{fod} fra moderdyrets vekt ($\zeta > 1$).

6. Død Dyr kan dø. Et dyr dør med sannsynlighet

$$\omega(1 - \Phi) \quad (2.6)$$

der $0 \leq \omega \leq 1$

Parameterne v_{fod} , β , σ , $a_{\frac{1}{2}}$, $v_{\frac{1}{2}}^{\text{under}}$, $v_{\frac{1}{2}}^{\text{over}}$, ϕ^{alder} , ϕ^{under} , ϕ^{over} , μ , γ , og ζ er identiske for alle dyr av samme art, men kan være forskjellige mellom bytte og rovdyr. Typiske verdier oppgis ved et senere tidspunkt. Den vesentlige forskjellen mellom rovdyr- og byttedyr er deres spisevaner, som beskrevet i avsnitt 2.2.1 og 2.2.2.



Figur 2.3: Venstre: dyrefordeling før vandringen fra (1, 1) (tykk ramme) gjennomføres. Midten: dyrefordelingen etter vandring fra (1, 1) er avsluttet, og vandringen fra (1, 2) (tykk ramme) begynnes. Høyre: fordelingen etter avsluttet vandring fra (1, 2). Legg merke til at A og B først vandrer fra (1, 1) til (1, 2); så vandrer B tilbake til (1, 1), mens A fortsetter til (2, 2). I tillegg vandret D til (1, 1).

2.2.1 Byttedyr

Byttedyr finner fôr utelukkende i savannen og jungelen. Dyr som oppholder seg i en celle, spiser i den rekkefølgen deres fitness Φ tilsier, dvs dyret med høyest fitness spiser først. Dyret forsøker hvert år å spise en formengde F , men hvor mye fôr dyret får tak i, avhenger av fôrtilbudet på stedet, jf avsnitt 2.1. Gitt at dyret spiser fôrmengden \tilde{F} , så øker vekten med $\beta\tilde{F}$.

2.2.2 Rovdyr

Rovdyr kan felle bytte alle steder. Rovdyr prøver å felle byttedyr i den rekkefølgen rovdirenes fitness tilsier. Rovdyret vil da prøve seg på et byttedyr om gangen, og vil felle byttedyr med sannsynlighet

$$k = \begin{cases} 0 & \text{hvis } \Phi_{\text{rovdyr}} \leq \Phi_{\text{bytte}} \\ \frac{\Phi_{\text{rovdyr}} - \Phi_{\text{bytte}}}{\Delta\Phi_{\max}} & \text{hvis } 0 < \Phi_{\text{rovdyr}} - \Phi_{\text{bytte}} < \Delta\Phi_{\max} \\ 1 & \text{ellers.} \end{cases} \quad (2.7)$$

Rovdyrets vekt øker med βv_{bytte} , der v_{bytte} er det felte dyrets vekt. Rovdyrets fitness evalueres på nytt etter hver felling.

2.3 Årets løp på Bjarnøya

Naturen på Bjarnøya følger en fast årssyklus. Bestanddelene i årssyklusen er:

1. **Aldring** Hvert dyr blir ett år eldre.
2. **Vekttap** Alle dyr mister vekt, jf avsnitt 2.2, nr. 2.
3. **Vandring** Dyr vandrer til naboceller. Cellene gjennomgås i tilfeldig rekkefølge. Fra hver celle vandrer dyr til naboceller, jf avsnitt 2.2, nr. 4. Det tillatt at et dyr vandrer fram og umiddelbart tilbake mellom to celler i denne årstiden, se også fig. 2.3.
4. **Død** For hvert dyr avgjøres det om dyret dør eller ikke, jf avsnitt 2.2, nr. 6.
5. **Formering** Dyr føder barn, jf avsnitt 2.2, nr. 5. Ved beregning av sannsynlighet for fødsel etter likning (2.5) benyttes antall dyr N i starten av avlssesongen, dvs nyfødte dyr telles ikke med.
6. **Næringsopptak** Dyr spiser: først byttedyr, så rovdyrene, jf avsnitt 2.2.1 og 2.2.2.

Det vises videre til at det er en del usikkerhet i det biologiske forskningsmiljøet omkring den presise sekvensen av hendelser i Bjarnøyas årssyklus. Utviklerne bes derfor om å legge sitt program til rette slik at rekkefølgen av hendelsene lett kan endres.

2.4 Fôrtilbud i det første året

Ved oppstart av simuleringen skal alle jungelceller inneholde fôrmengden f_{\max}^{Jungel} , mens alle savanneceller skal inneholder fôrmengden f_{\max}^{Sav} .

Kapittel 3

Programkjøring og materialpakke

3.1 Kjøring av programmet

Programmet skal kjøres fra kommandolinjen. På kommandolinjen kan spesifikasjonsfil for simuleringen oppgis, jf avsnitt 4.1. Dersom ikke filnavn blir oppgitt på kommandolinjen, skal programmet spørre brukeren om filnavn umiddelbart etter oppstart av programmet.

Under kjøringen kan programmet vise aktuelt årstall samt totalt antall byttedyr og rovdyr for hvert år. Under utviklingen kan det være fornuftig å skrive ut en linje per år. Det ferdige programmet skal derimot viske ut linjen etter hvert år og erstatte den med ny data.

Når programmet er ferdig, skal det gis ut en kort rapport om dyretallet på slutten, samt en oversikt over de filene der resultatene ble lagret. Lagres det en fil per år, skal bare stammen til filnavnene oppgis, ikke alle filer.

Programmet vil komme til å skrive ut svært mange filer, hvis komplett populasjonsinformasjon ol ønskes for hvert år. Utdatafilene bør derfor samles i undermapper. Skal man arbeide med flere forskjellige simuleringsprosjekter, kan det være fornuftig å samle alle filer som tilhører et simuleringsprosjekt i en egen mappe.

3.2 Generering av inputfiler

Spesifikasjonsfiler, geografifiler (jf avsnitt 4.2) og parameterfiler (jf avsnitt 4.3) lages for hånd. Automatisk generering av slike filer i Matlab kan

vurderes, men dette vil gå utover BioSim prosjektet. I materialpakken leveres eksempler på disse filer, som kan benyttes som utgangspunkt for egne filer.

Populasjonsfiler kan lages manuelt. Dette kan være aktuelt hvis man f.eks. ønsker å sette ut en liten rovdyrpopulasjon ett sted på en øy der det bare finnes byttedyr. Materialpakken inneholder et verktøy for automatisert lagring av populasjonsfiler.

3.3 Code::blocks-pakke

Code::blocks kan benyttes til styring av kompilering av BioSim-prosjektet. Materialpakken er laget som et Code::blocks-prosjekt.

3.4 Matlab-verktøy

I undermappen `matlab` i BioSim-pakken, finnes følgende funksjoner til sluttbruker. Se `help function` i Matlab for informasjon om bruk.

<code>vis_kart</code>	Viser geografien på øya.
<code>create_pop</code>	Skaper populasjonsfil for en dyreart med tilfeldig Poissonfordelt antall dyr per celle og tilfeldige alder og vekt (likefordeling). Bruker velger fordelingsparameterne.
<code>tidsrekke</code>	Tegner grafikk over antall dyr både total for øya, og delt etter landskapstyper.
<code>vis_pops</code>	Viser to kart ved siden av hverandre, der den venstre viser fordeling av byttedyr, den høyre fordeling av rovdyr.
<code>vis_for</code>	Viser kart over fôrforedlingen.
<code>anim_pops</code>	Viser animasjon over populasjonskart.
<code>read_pop</code>	Leser populasjonsfil. Stort sett til internt bruk, men kan være nyttig til debugging.
<code>plot_fitness</code>	Viser fitnessfunksjon; kan brukes ved parametervalg.

3.5 C++ toolbox

I BioSim-pakken finnes mappen `toolbox` som inneholder flere nyttige klasser:

Klasse	Hodefil	Beskrivelse
<code>RandomGenerator</code>	<code>random.h</code>	slumptallsgenerator
<code>Filename</code>	<code>filename.h</code>	håndterer oppdeling av filnavn deler, samt produksjon av nummererte filnavn
<code>ReadParameters</code>	<code>read_parameters.h</code>	Klasse til lesing av parameterfiler. Klassen kan lese både <code>.par</code> og <code>.sim</code> filer. Se Doxygen dokumentasjon for et eksempel.
<code>skip_comment()</code>	<code>skip_comment.h</code>	Funksjon som “spiser opp” kommentarer.

Toolbox-klasser benyttes ved å legge `#include "toolbox/deklarasjonsfil"` inn i programfilene. For mer dokumentasjon, kjør Doxygen og se på dokumentasjonen under `doc/html/index.html`!

3.6 Doxygen dokumentasjon

Programmet skal dokumenteres godt. Til dokumentasjon av filer og klasse, skal Doxygen benyttes. Mer informasjon om dette kommer om kort tid.

Kapittel 4

Filformater

I alle datafiler som inngår, betraktes linjer som har % som sitt første tegn, som kommentarlinjer. Dette fordi MATLAB benytter prosenttegnet som kommentartegn.

4.1 Spesifikasjonsfil

En simulering defineres gjennom en spesifikasjonsfil. Spesifikasjonsfilen har følgende format.

Geografi	Bjarnoya.geo
CelleParameter	mincelle.par
BytteParameter	minbytte.par
RovdyrParameter	mittrovdyr.par
Populasjon	bytte.pop
Populasjon	rovdyr.pop
StartAar	0
SluttAar	1000
SlumptallFroe	123456789
UtdataStamme	data/minsim
DumpDyrInterval	1
DumpPoPInterval	5
DumpForInterval	10

Rekkefølgen av linjene skal overholdes! Det kan oppgis så mange Populasjon-linjer som en ønsker.

SlumptallFroe er startverdi til slumptallsgeneratoren.

UtdataStamme er stammen til navn til utdatafilene som produseres. Filnavn bygges i biosim-programmet ved å henge den riktige endelsen på stammen.

DumpNumInterval angir hvor ofte populasjonstallene er skrevet til num-filen, mens DumpFullInterval angir hvor ofte en komplett pop-fil skrives. DumpForInterval oppgir hvor ofte fôrfordelingsfilen skal skrives. Hvis null er oppgitt som interval, skal ingen data lagres. Filnavn skal ende på .sim.

4.2 Geografi

Øyas geografi spesifiseres i geografifiler, som har følgende format (tekst i maskinskrift må gjengis som den er gitt her):

```
Rader      3
Kolonner   5
```

```
HHHHH
HFOJH
HHHHH
```

dvs

- Antall rader og kolonner oppgis på de første to linjene; verdiene markeres med tilsvarende nøkkelord. Rekkefølgen på disse to linjene er likegyldige.
- Det følger en linje per rad, med ett tegn per kolonne. Følgende kodebokstaver benyttes: H—hav, F—fjell, J—jungel, O—ørken, og S savanne.
- Først og siste rad samt høyre og venstre kolonne må utelukkende bestå av havceller.
- Alle rader må ha oppgitt antall elementer.
- Geografifiler skal ha navn som ender på .geo.
- Koordinatsystemet er slik at cellen i øvre venstre hjørne (første rad, første kolonne) har koordinatene (0,0). Første koordinat teller rad, andre kolonne.

4.3 Parameter

Parameterverdier for celler, rovdyr og byttedyr og landskap oppgis i separate filer. Alle parameterfil har samme format. Rekkefølgen av navngitte variabler spiller ikke noen rolle. Filnavn skal ende på .par.

Filene inneholder en linje per parameter. På hver linje kommer først navnet til parameteren, så verdien. Følgende navn skal brukes *i nøyaktig denne skrivemåten*:

Celleparameter		Betydning
α	alpha	tilvekstrate, savanne
f_{\max}^{Sav}	fmax_sav	maks formengde, savanne
f_{\max}^{Jungle}	fmax_jngl	maks formengde, jungel
v_{fod}	v_fod	fødselsvekt
β	beta	metabolisk effektivitet
σ	sigma	rel. årlig vekttap
v_{\min}	v_min	nedre vektgrense
$a_{\frac{1}{2}}$	a_half	midpunkt, aldersfitness
ϕ^{alder}	phi_alder	stigning, aldersfitness
$v_{\frac{1}{2}}^{\text{under}}$	v_half_under	midpunkt, nedre vektgrense
ϕ^{under}	phi_under	stigning, nedre vektfitness
$v_{\frac{1}{2}}^{\text{over}}$	v_half_over	midpunkt, øvre vektgrense
ϕ^{over}	phi_over	stigning, øvre vektfitness
μ	mu	sannsynlighet for vandring
γ	gamma	koeffisient for fødselssannsynlighet
ζ	zeta	koeffisient for vekttap ved fødsel
ω	omega	koeffisient for dødssannsynlighet
F	F	förmengden byttedyret ønsker å fortære per år (bare byttedyr)
$\Delta\Phi_{\max}$	DeltaPhiMax	fellingsparameter (bare rovdyr)

4.4 Populasjon

Dyrepopulasjon på en øy spesifiseres gjennom populasjonsfiler. Et program skal være i stand til å lese flere populasjonsfiler ved oppstart. Dyrene skal plasseres på øya etter hvert som de blir lest fra filene. Den totale dyrepopulasjonen i starten vil derfor være summen av alle dyr som

er spesifisert i de leste filene. BioSim-programmet skal også kunne lagre populasjonen i populasjonsfil.

En populasjonsfil har følgende oppbygging:

- Den første linjen spesifiserer geografifilen som skal brukes og har følgende form:

Geografi *filnavn*

Filen skal være en geografifil som spesifisert i avsnitt 4.2.

- Resten av filen består av blokker. Data i hver blokk spesifiserer dyr at en art i en celle. Hver blokk består av en hodelinje, som oppgir dyreart (B eller R), cellens koordinater (rad først) og antall dyr. Etterpå følger en linje per dyr. Hver linje inneholder først alder, så vekt.
- Blokker for forskjellige celler og dyreart kan plasseres fritt i filen. Flere blokke for samme dyreart og celle er tillatt.
- Antall desimaler for vekten skal lett kunne justeres ved utskrift.
- Filnavnet skal slutte på .pop.

Eksempel:

```
% en minimal populasjon
Geografi Bjarnoya.geo
```

```
B 2 3 2
12 34.213
10 22.112
```

```
R 3 15 10
20 21.012
8 15.687
3 9.900
```

4.4.1 Populasjonsfil som utdata

Programmet skal også kunne lagre informasjon om den aktuelle populasjonen i en populasjonsfil. I dette tilfelle skal informasjon om hele populasjonen samles i en fil. Filnavn skal i dette tilfelle slutte på .pop.####, der #### står for årstallet.

4.5 Utdatafiler

Programmet skal kunne lagre fullstendig populasjonsdata i pop-filer, se avsnitt 4.4. Videre skal filer over totalbefolkning, dyrefordeling og fôrfordeling kunne lagres.

4.5.1 Befolkningsfil

Denne filen oppgir totalt antall rovdyr og byttedyr for hvert år i hver av de fire landskapstyper. Filen er bygget opp slik:

- Den første linjen spesifiserer geografifilen som skal brukes og har følgende form:
`Geografi filnavn`
- Det følger en linje for hvert år, med syv tall per linje i følgende rekkefølge
 1. årstall
 2. bytte i jungel
 3. rovdyr i jungel
 4. bytte i savanne
 5. rovdyr i savanne
 6. bytte i ørken
 7. rovdyr i ørken
- Første datalinje skal inneholde tall umiddelbart før oppstart av simuleringen.
- Filen skal oppdateres etter hvert år er gått.
- Filnavn skal slutte på `.dat`.

4.5.2 Dyrefordeling

Denne filen viser fordeling av dyr over øya og er bygget opp slik:

- Den første linjen spesifiserer geografifilen som skal brukes og har følgende form:
`Geografi filnavn`

- Det følger en linje per celle, fra øvre venstre til nedre høyre hjørne. Hver linje inneholder antall bytte og antall rovdyr i en celle (i den rekkefølgen). Forskjellige rader i geografien skal skilles med tom linje i fil.
- Filnavn skal slutte på `.dyr.#####`, der `#####` representerer året.

4.5.3 Fôrfordeling

Denne filen viser tilgjengelig fôrmengde i hver celle og er bygget opp slik:

- Den første linjen spesifiserer geografifilen som skal brukes og har følgende form:
`Geografi filnavn`
- Det følger en linje per celle, fra øvre venstre til nedre høyre hjørne. Hver linje inneholder ett tall, som oppgir tilgjengelig fôrmengde f i cellen. Det skal lagres verdien av f etter næringsopptaket har skjedd i det inneværende år.
- For celletyper som ikke tilbyr fôr, oppgis alltid 0.
- Filnavn skal slutte på `.for.#####`, der `#####` representerer året.

Kapittel 5

Veiledning

Denne oppgaven er et forholdsvis stort prosjekt, som krever samarbeid i prosjektgruppen. Samarbeid i gruppen, dvs diskusjon om strategier, framgangsmåte, og programkoden som skal utvikles, er et viktig læringsmål ved denne oppgaven. I arbeidslivet forventes det jo at medarbeidere kan diskutere oppgaver med hverandre og utarbeide løsninger i felleskap.

Siden oppgaven er forholdsvis kompleks, anbefales det sterkt å utrede problemet grundig **før** man begynner å programmere. Sett dere gjerne ned med sjakkbrett og brikker og “lek” Bjarnøya. Følg nøye “spillereglerne” forklart i kapittel 2, for å forstå hva som foregår på Bjarnøya i løpet av året (papir, blyant og lommekalkulator kan være nyttig). Å forstå problemet er første skritt til løsningen.

Når dere går i gang med programmeringen, vil det være veldig lurt å gå frem trinnvis, fra enkle deler frem til simulering av hele naturen. Anbefalt progresjon er:

1. Simuler en dyreart på ett sted. Begynn med byttedyr, og anta at det befinner seg i jungelen.
2. Simuler to dyrearter på ett sted, altså bytte og rovdyr i jungelen.
3. Simuler to dyrearter i et utstrakt område, men anta at alle celler er jungel (med unntak for havet som omgir Bjarnøya).
4. Fullstendig simulering med alle landskapstyper.

Diskuter gjerne oppgaven i *Forum* på kursets nettside!