**Appendix A – Proof**

We define the length of an individual in the following time-step (*L(t+1)*) to be:

therefore the expected value of *L(t+1)* is:

and the variance of *L(t+1)* is:

Now considering the expected length and variance of an individual after two time-steps:

Therefore, after time-steps the length of an individual will be:

where

**Appendix B – Code for calling *TMB* from within *R***

// Tag recapture model

#include <TMB.hpp>

template<class Type>

Type objective\_function<Type>::operator() ()

{

// Options

DATA\_FACTOR(Options);

// Observations

DATA\_FACTOR(iAge1); // Age of individual at time 1 (with units days, weeks, months, or years, rounded to nearest integer)

DATA\_VECTOR(iLiberty); // Time at liberty of individual (with units days, weeks, months, or years)

DATA\_VECTOR(Length1); // Length of individual at time 1

DATA\_VECTOR(Length2); // Length of individual at time 2

DATA\_FACTOR(Sex); // Sex of individual

DATA\_FACTOR(Time0); // Time at which individual was born (time 0, with units days, weeks, months, or years)

DATA\_FACTOR(Time1); // Time at which individual was caught/tagged (time 1, with units days, weeks, months, or years)

DATA\_FACTOR(Time2); // Time at which individual was recaptured (time 2, with units days, weeks, months, or years)

DATA\_FACTOR(Year0); // Year in which individual was born (time 0)

DATA\_FACTOR(Year1); // Year in which individual was caught/tagged (time 1)

DATA\_FACTOR(Year2); // Year in which individual was recaptured (time 2)

DATA\_FACTOR(Area1); // Area that individual was in at time 1

// Parameters

PARAMETER\_VECTOR(ln\_gamma); // Fixed effect vector (1)

PARAMETER(logit\_psi); // Fixed effect (1)

PARAMETER\_VECTOR(L0); // Fixed effect vector (2, sex-specific)

PARAMETER\_VECTOR(ln\_bmean); // Random effects vector (2, sex-specific)

PARAMETER\_VECTOR(ln\_bdev); // Random effects vector (315, individual)

PARAMETER\_VECTOR(ln\_sd\_bdev); // Random effect standard deviation (2, sex-specific)

PARAMETER(ln\_sd\_obs); // Measurement standard deviation (1)

PARAMETER\_VECTOR(z1); // Random effects vector (315, individual)

PARAMETER\_VECTOR(z2); // Random effects vector (315, individual)

PARAMETER(ln\_sd\_z); // Random effects standard deviation (1)

PARAMETER\_VECTOR(ln\_ydev); // Random effects vector (40, year)

PARAMETER(ln\_sd\_ydev); // Random effects standard deviation (1)

PARAMETER\_VECTOR(ln\_xdev); // Random effects vector (40, area)

PARAMETER(ln\_sd\_xdev); // Random effects standard deviation (1)

// Initialize dimension variables:

int Nindiv = iAge1.size(); // Number of individuals

int Nsex = 2; // Number of sexes

// Initialize population-level parameters:

Type psi = 1 / (1 + exp(-logit\_psi));

Type sd\_obs = exp(ln\_sd\_obs);

Type sd\_z = exp(ln\_sd\_z);

Type sd\_ydev = exp(ln\_sd\_ydev);

// Initialize sex-specific parameters:

vector<Type> gamma(Nsex);

vector<Type> bmean(Nsex);

vector<Type> sd\_bdev(Nsex);

vector<Type> amean(Nsex);

vector<Type> Linf(Nsex);

bmean = exp(ln\_bmean);

sd\_bdev = exp(ln\_sd\_bdev);

gamma = exp(ln\_gamma);

amean = gamma \* pow(bmean, psi);

Linf = (gamma \* pow(bmean, psi)) / bmean;

// Initialize some computational variables:

int sex;

Type ans = 0.;

Type sumj = 0.;

// Distribution of observations x given random effects u (x|u):

vector<Type> a\_indiv(Nindiv);

vector<Type> b\_indiv(Nindiv);

vector<Type> Length1\_hat(Nindiv);

vector<Type> Length2\_hat(Nindiv);

// Time varying individual stuff

vector<Type> sd\_z1(Nindiv);

vector<Type> sd\_z2(Nindiv);

// Year effects

int Nyears = ln\_ydev.size(); // Number of years

int time\_step = 52;

int time, year;

// Random effect probability of each year

if (Options(0) == 1)

{

for (int y = 0; y < Nyears; y++) { ans -= dnorm( ln\_ydev(y), Type(0.), sd\_ydev, true ); }

}

vector<Type> isYearUsedTF(Nyears);

isYearUsedTF.setZero();

// Area effects

int Narea = ln\_xdev.size();

int area;

Type sd\_xdev = exp(ln\_sd\_xdev);

// Random effect probability of each area

if (Options(1) == 1)

{

for (int a = 0; a < Narea; a++) { ans -= dnorm( ln\_xdev(a), Type(0.), sd\_xdev, true ); }

}

// Loop over each individual in the data set

for (int i = 0; i < Nindiv; i++)

{

sex = Sex(i) - 1; // Sex of the individual

year = Year0(i); // The (index) year that the individual was born

area = Area1(i) - 1; // Location of the individual at time 1

b\_indiv(i) = bmean(sex) \* exp(ln\_bdev(i)); // Value for b\_indiv

a\_indiv(i) = gamma(sex) \* pow(b\_indiv(i), psi); // Derived value for a\_indiv

// Random effect probability of bdevs

if (Options(2) == 1)

{

ans -= dnorm( ln\_bdev(i), Type(0.), sd\_bdev(sex), true );

}

sumj = Type(0.);

for (int j = 0; j < (iAge1(i)-1); j++)

{

// This piece of code gives us the year (from 0 to 39, ref to the years

// 1973/74 to 2012/13) that the fish is in at each time-step. This is

// required because if a time-step other than annual is used then we need

// to be referencing the correct year-effect parameter.

time = Time0(i) + j;

if ( fmod (time, time\_step) == 0. ) { year += 1; }

sumj += gamma(sex) \* exp(-b\_indiv(i) \* j) \* exp(ln\_ydev(year)) \* exp(ln\_xdev(area));

isYearUsedTF( year ) = 1;

}

Length1\_hat(i) = ( L0(sex) \* exp(-b\_indiv(i) \* iAge1(i)) ) + ( pow(b\_indiv(i), psi-1) \* (1-exp(-b\_indiv(i))) \* sumj );

Length1\_hat(i) += z1(i);

// Time-variation probability from birth to first capture

sumj = Type(0.0001);

for (int j = 0; j < (iAge1(i)-1); j++) { sumj += exp(Type(2.0) \* -b\_indiv(i) \* j); }

sd\_z1(i) = sd\_z \* (pow(b\_indiv(i), psi-1) \* (1-exp(-b\_indiv(i)))) \* pow(sumj,0.5);

if (Options(3) == 1)

{

ans -= dnorm( z1(i), Type(0.), sd\_z1(i), true );

}

ans -= dnorm( Length1(i), Length1\_hat(i), sd\_obs \* Length1\_hat(i), true );

// Probability of second length measurement

year = Year1(i); // The (index) year that the individual was first captured

sumj = Type(0.);

for (int j = 0; j < (iLiberty(i)-1); j++)

{

time = Time1(i) + j;

if ( fmod (time, time\_step) == 0. ) { year += 1; }

sumj += gamma(sex) \* exp(-b\_indiv(i) \* j) \* exp(ln\_ydev(year)) \* exp(ln\_xdev(area));

isYearUsedTF( year ) = 1;

}

Length2\_hat(i) = ( Length1\_hat(i) \* exp(-b\_indiv(i) \* iLiberty(i)) ) + ( pow(b\_indiv(i), psi-1) \* (1 - exp(-b\_indiv(i))) \* sumj );

Length2\_hat(i) += z2(i);

// Time-variation probability from first capture to second capture

sumj = Type(0.0001);

for (int j = 0; j < (iLiberty(i)-1); j++) { sumj += exp(Type(2.0) \* -b\_indiv(i) \* j); }

sd\_z2(i) = sd\_z \* (pow(b\_indiv(i), psi-1) \* (1-exp(-b\_indiv(i)))) \* pow(sumj,0.5);

if (Options(3) == 1)

{

ans -= dnorm( z2(i), Type(0.), sd\_z2(i), true );

}

ans -= dnorm( Length2(i), Length2\_hat(i), sd\_obs \* Length2\_hat(i), true );

}

// REPORT DIAGNOSTICS

REPORT( b\_indiv );

REPORT( a\_indiv );

REPORT( Linf );

// Append outputs to report

ADREPORT( gamma );

ADREPORT( psi );

ADREPORT( L0 );

ADREPORT( amean );

ADREPORT( bmean );

ADREPORT( sd\_bdev );

ADREPORT( sd\_obs );

ADREPORT( sd\_z );

ADREPORT( sd\_ydev );

ADREPORT( sd\_xdev );

ADREPORT( Length1\_hat );

ADREPORT( Length2\_hat );

ADREPORT( a\_indiv );

ADREPORT( b\_indiv );

ADREPORT( ln\_bdev );

ADREPORT( sd\_z1 );

ADREPORT( sd\_z2 );

ADREPORT( ln\_ydev );

ADREPORT( ln\_xdev );

return ans;

}