Appendix. AD Model Builder computer code to fit the WIN model, the input data file, and the standard deviation parameter file.

Computer Code

```
  //WINTER FLOUNDER AGE-STRUCTURED MODEL
  //JON BRODZIAK NEFSC OCTOBER 2001
  //COMMENT LINES BEGIN WITH "//"
DATA_SECTION
  //READ DATA FROM INPUT FILE "WIN.DAT"
  init_int styr
  init_int endyr
  init_int nages
  init_int nselages_fish
  init_vector catch_bio(styr,endyr)
  init_int nobs_fish
  init_ivector yrs_fish(1,nobs_fish)
  init_vector nsamples_fish(1,nobs_fish)
  init_matrix obs_p_fish(1,nobs_fish,1,nages)
  init_int nobs_FALL
  init_ivector yrs_FALL(1,nobs_FALL)
  init_number zfrac_FALL
  init_vector obs_FALL(1,nobs_FALL)
  init_int nsamples_FALL
  init_matrix obs_p_FALL(1,nobs_FALL,1,nages)
  init_int nobs_SPR
  init_ivector yrs_SPR(1,nobs_SPR)
  init_number zfrac_SPR
  init_vector obs_SPR(1,nobs_SPR)
  init_int nsamples_SPR
  init_matrix obs_p_SPR(1,nobs_SPR,1,nages)
  init_vector wt(1,nages)
  init_number zfrac_spawn
  init_vector maturity(1,nages)
  init_number lambda_recruitment
  init_number lambda_fishery_age
  init_number lambda_FALL_age
  init_number lambda_biomass_index_FALL
  init_number lambda_SPR_age
  init_number lambda_biomass_index_SPR
  init_number lambda_catch_biomass
  init_number lambda_fishery_sel
  init_number lambda_f_penalty

  int styr_rec

 LOCAL_CALCS
 //COMPUTE YEAR OF FIRST RECRUITMENT DEVIATION TO BE ESTIMATED
  styr_rec=styr-nages+2;
 END_CALCS


INITIALIZATION_SECTION
  //PROVIDE INITIAL PARAMETER VALUES
  //NATURAL MORTALITY (NOT ESTIMATED)
  M 0.20

  //MEAN RECRUITMENT) IN THOUSANDS OF FISH
  mean_log_rec 8.45

  //LOG(MEAN ANNUAL FISHING MORTALITY)
  log_avg_fmort -2.5
```

```
 //FALL SURVEY INDEX PARAMETERS
 qFALL 1.
 exp_FALL 1.
 log_gamma_FALL -2.
 log_beta_FALL 0.
 log_a50_FALL 1.5

 //SPRING SURVEY INDEX PARAMETERS
 qSPR 1.
 exp_SPR 1.
 log_gamma_SPR -25.
 log_beta_SPR 0.
 log_a50_SPR 1.5

PARAMETER_SECTION
 //DECLARE MODEL PARAMETERS AND VARIABLES
 init_bounded_number M(.02,.25,-1)
 init_number mean_log_rec(1)
 init_bounded_dev_vector rec_dev(styr_rec,endyr,-15,15,3)

 init_bounded_number qFALL(.001,1000.,7)
 init_bounded_number exp_FALL(.25,4.,-1)
 init_bounded_number log_gamma_FALL(-50.,0.999,6)
 init_bounded_number log_beta_FALL(-50.,10.,6)
 init_bounded_number log_a50_FALL(0.,3.,6)

 init_bounded_number qSPR(.001,1000.,5)
 init_bounded_number exp_SPR(.25,4.,-1)
//FIX log_gamma_SPR at -25 to assume flat-topped curve
 init_bounded_number log_gamma_SPR(-50.,0.999,-1)
 init_bounded_number log_beta_SPR(-50.,10.,4)
 init_bounded_number log_a50_SPR(0.,3.,4)

 init_number log_avg_fmort(2)
 init_bounded_dev_vector fmort_dev(styr,endyr,-15,15,2)

 init_vector log_selcoffs_fish(1,nselages_fish,4)

 vector log_sel_fish(1,nages)
 vector sel(1,nages)
 vector sel_FALL(1,nages)
 vector sel_SPR(1,nages)
 number avgsel_fish

 vector rec_years(styr_rec,endyr)
 vector years(styr,endyr)
 vector ages(1,nages)

 vector totn_FALL(styr,endyr)
 vector totn_SPR(styr,endyr)
 vector popnbiom(styr,endyr)
 sdreport_vector spawnbiom(styr,endyr)
 sdreport_vector recruitment(styr,endyr)
 vector explbiom(styr,endyr)
 vector surplus_production(styr,endyr-1)
 vector pred_FALL(styr,endyr)
 vector pred_SPR(styr,endyr)
 matrix pred_p_fish(styr,endyr,1,nages)
 matrix pred_p_FALL(styr,endyr,1,nages)
 matrix pred_p_SPR(styr,endyr,1,nages)
 vector pred_catch(styr,endyr)

 vector natage_FALL(1,nages)
 vector natage_SPR(1,nages)
 vector natage_spawn(1,nages)
 matrix natage(styr,endyr,1,nages)
```

```
  matrix catage(styr,endyr,1,nages)
  matrix Z(styr,endyr,1,nages)
  matrix F(styr,endyr,1,nages)
  matrix S(styr,endyr,1,nages)

  number beta_FALL
  number gamma_FALL
  number a50_FALL
  number beta_SPR
  number gamma_SPR
  number a50_SPR

  number survival

  vector offset(1,4)
  number rec_like
  number catch_like
  vector age_like(1,4)
  vector sel_like(1,4)
  number fpen
  number FALL_like
  number SPR_like

  number rmse_catch_bio
  number rmse_FALL
  number rmse_SPR

  objective_function_value f

  sdreport_number endbiom
  sdreport_number depletion_popnbiom
  sdreport_number endspawn
  sdreport_number depspawn
  sdreport_number deppopnbiom82
  sdreport_number depspawn82
  sdreport_vector endN(1,nages)
  likeprof_number endF


RUNTIME_SECTION
  convergence_criteria 1e-6;

PRELIMINARY_CALCS_SECTION

  //SET TIME HORIZON:years
  for (int i=styr; i<=endyr; i++)
    {
    years(i)=i;
    }

  //SET RECRUITMENT TIME HORIZON:rec_years
  for (i=styr_rec; i<=endyr; i++)
    {
    rec_years(i)=i;
    }

  //SET AGE CLASSES:ages
  for (i=1; i<=nages; i++)
    {
    ages(i)=i;
    }

  //RESCALE FALL SURVEY INDEX
  obs_FALL*=1000;

  //RESCALE SPRING SURVEY INDEX
```

```
obs_SPR*=1000;

//CHECK INPUT DATA
cout << "START YEAR: "<<styr<< endl;
cout << "END YEAR: "<<endyr<< endl;
cout << "AGE CLASSES: "<<nages<<endl;
cout << "FISHERY SELECTED AGES: "<<nselages_fish<<endl;
cout << "CATCH BIOMASS" << endl;
cout << catch_bio << endl;
cout << "FISHERY YEARS"<<endl;
cout << yrs_fish<< endl;
cout << "FALL SURVEY YEARS"<<endl;
cout << yrs_FALL<< endl;
cout << "FRACTION OF Z BEFORE FALL SURVEY"<<endl;
cout << zfrac_FALL<< endl;
cout << "FALL SURVEY INDEX"<<endl;
cout << obs_FALL<< endl;
cout << "SPRING SURVEY YEARS"<<endl;
cout << yrs_SPR<< endl;
cout << "FRACTION OF Z BEFORE SPRING SURVEY"<<endl;
cout << zfrac_SPR<< endl;
cout << "SPRING SURVEY INDEX"<<endl;
cout << obs_SPR<< endl;
cout << "FISHERY AGE COMPOSITION"<<endl;
cout << obs_p_fish<< endl;
cout << "FALL SURVEY AGE COMPOSITION"<<endl;
cout << obs_p_FALL<< endl;
cout << "SPRING SURVEY AGE COMPOSITION"<<endl;
cout << obs_p_SPR<< endl;
cout << "WEIGHT AT AGE"<<endl;
cout << wt<< endl;
cout << "FRACTION OF Z BEFORE SPAWNING"<<endl;
cout << zfrac_spawn<< endl;
cout << "MATURITY AT AGE"<<endl;
cout << maturity<< endl;
cout << "LAMBDA RECRUITMENT: " << lambda_recruitment <<endl;
cout << "LAMBDA FISHERY AGE: " <<lambda_fishery_age <<endl;
cout << "LAMBDA FALL SURVEY AGE: " <<lambda_FALL_age <<endl;
cout << "LAMBDA FALL SURVEY INDEX: " <<lambda_biomass_index_FALL <<endl;
cout << "LAMBDA SPRING SURVEY AGE: " <<lambda_SPR_age <<endl;
cout << "LAMBDA SPRING SURVEY INDEX: " <<lambda_biomass_index_SPR <<endl;
cout << "LAMBDA CATCH BIOMASS: " <<lambda_catch_biomass <<endl;
cout << "LAMBDA FISHERY SELECTIVITY: " <<lambda_fishery_sel <<endl;
cout << "LAMBDA F PENALTY: " <<lambda_f_penalty <<endl;

//COMPUTE OFFSET FOR FISHERY AGE MULTINOMIAL
  for (i=1; i <= nobs_fish; i++)
    {
    //CHECK FOR FISHERY AGE DATA IN YEAR i, -99 = MISSING DATA
    if (obs_p_fish(i,1) >= 0.0)
      obs_p_fish(i)=obs_p_fish(i)/sum(obs_p_fish(i));
    for (int j=1; j<=nages; j++)
      {
        if (obs_p_fish(i,j)>0.0)
          {
          offset(1)-=nsamples_fish(i)*obs_p_fish(i,j)*log(obs_p_fish(i,j));
          }
      }
    }
//cout << "FISHERY PROPORTION AT AGE DATA" << endl;
//cout << obs_p_fish << endl;

//COMPUTE OFFSET FOR AUTUMN SURVEY AGE MULTINOMIAL
  for (i=1; i <= nobs_FALL; i++)
    {
    //CHECK FOR AGE DATA IN YEAR i, -99 = MISSING DATA
```

```
      if (obs_p_FALL(i,1) >= 0.0)
        obs_p_FALL(i)=obs_p_FALL(i)/sum(obs_p_FALL(i));
      for (int j=1; j<=nages; j++)
        {
         if (obs_p_FALL(i,j)>0.0)
           {
           offset(2)-=nsamples_FALL*obs_p_FALL(i,j)*log(obs_p_FALL(i,j));
           }
        }
      }
  //cout << "FALL SURVEY PROPORTION AT AGE DATA" << endl;
  //cout << obs_p_FALL << endl;

  //COMPUTE OFFSET FOR SPRING SURVEY AGE MULTINOMIAL
     for (i=1; i <= nobs_SPR; i++)
      {
      //CHECK FOR AGE DATA IN YEAR i, -99 = MISSING DATA
      if (obs_p_SPR(i,1) >= 0.0)
        obs_p_SPR(i)=obs_p_SPR(i)/sum(obs_p_SPR(i));
      for (int j=1; j<=nages; j++)
        {
         if (obs_p_SPR(i,j)>0.0)
           {
           offset(3)-=nsamples_SPR*obs_p_SPR(i,j)*log(obs_p_SPR(i,j));
           }
        }
      }
  //cout << "SPRING SURVEY PROPORTION AT AGE DATA" << endl;
  //cout << obs_p_SPR << endl;

TOP_OF_MAIN_SECTION
  //ALLOCATE SPACE IN READ-WRITE MEMORY
  arrmblsize=2000000;
  gradient_structure::set_GRADSTACK_BUFFER_SIZE(2000000);
  gradient_structure::set_CMPDIF_BUFFER_SIZE(60000000);

PROCEDURE_SECTION
  //DO THE FUNCTION CALLS IN SEQUENCE
  get_selectivity();
  get_mortality();
  survival=mfexp(-1.0* M);
  get_numbers_at_age();
  get_catch_at_age();
  evaluate_the_objective_function();

FUNCTION get_selectivity
  //FISHERY SELECTIVITY ESTIMATION FOR AGES 1 TO NSELAGES_FISH
  //SET AVERAGE TO 1 AND THEN RESCALE SO MAX VALUE=1
  for (int j=1;j<=nselages_fish;j++)
    {
    log_sel_fish(j)=log_selcoffs_fish(j);
    }
  for (j=nselages_fish+1;j<=nages;j++)
    {
    log_sel_fish(j)=log_sel_fish(j-1);
    }
  avgsel_fish=log(mean(mfexp(log_selcoffs_fish)));
  log_sel_fish-=log(mean(exp(log_sel_fish)));
  sel=mfexp(log_sel_fish);
  sel/=max(sel);
  //cout<<"FISHERY SELECTIVITY"<<endl;
  //cout<<sel<<endl;
  //cout<<"MAXIMUM VALUE: "<<max(sel)<<endl;

  //AUTUMN SURVEY SELECTIVITY ESTIMATION VIA THOMPSON MODEL
  beta_FALL=mfexp(log_beta_FALL);
```

```
  gamma_FALL=mfexp(log_gamma_FALL);
  a50_FALL=mfexp(log_a50_FALL);
  for (j=1; j<=nages; j++)
   {
   sel_FALL(j)=(1./(1.-gamma_FALL))*pow((1.-gamma_FALL)/gamma_FALL,
             gamma_FALL)*(exp(beta_FALL*gamma_FALL*(a50_FALL-
             double(j)))/(1+exp(beta_FALL*(a50_FALL-double(j)))));
   }
  sel_FALL/=max(sel_FALL);
  //cout<<"FALL SURVEY SELECTIVITY"<<endl;
  //cout<<sel_FALL<<endl;
  //cout<<"MAXIMUM VALUE: "<<max(sel_FALL)<<endl;

  //SPRING SURVEY SELECTIVITY ESTIMATION VIA THOMPSON MODEL
  beta_SPR=mfexp(log_beta_SPR);
  gamma_SPR=mfexp(log_gamma_SPR);
  a50_SPR=mfexp(log_a50_SPR);
  for (j=1; j<=nages; j++)
   {
   sel_SPR(j)=(1./(1.-gamma_SPR))*pow((1.-gamma_SPR)/gamma_SPR,
             gamma_SPR)*(exp(beta_SPR*gamma_SPR*(a50_SPR-
             double(j)))/(1+exp(beta_SPR*(a50_SPR-double(j)))));
   }
  sel_SPR/=max(sel_SPR);
  //cout<<"SPRING SURVEY SELECTIVITY"<<endl;
  //cout<<sel_SPR<<endl;
  //cout<<"MAXIMUM VALUE: "<<max(sel_SPR)<<endl;

  //cout << "END OF GET SELECTIVITY" << endl;


FUNCTION get_mortality
  //COMPUTE TOTAL MORTALITY BY YEAR AND AGE
  //COMPUTE FISHING MORTALITY MATRIX
  for (int i=styr;i<=endyr;i++)
   {
   for (int j=1;j<=nages;j++)
     {
     F(i,j)=sel(j)*mfexp(log_avg_fmort + fmort_dev(i));
     }
   }

  //COMPUTE TOTAL MORTALITY MATRIX
   Z=F+M;

  //COMPUTE SURVIVAL MATRIX
   S=mfexp(-1.0*Z);

  //cout << "END OF GET MORTALITY" << endl;

FUNCTION get_numbers_at_age
  //COMPUTE NUMBERS AT AGE MATRIX
  int itmp;

  //COMPUTE NUMBERS AT AGE IN INITIAL YEAR
  for (int j=1;j<nages;j++)
   {
    itmp=styr+1-j;
    natage(styr,j)=mfexp(mean_log_rec-M*double(j-1)+rec_dev(itmp));
   }
   natage(styr,nages)=mfexp(mean_log_rec-M*(nages-1))/
                 (1. - survival);

  //COMPUTE RECRUITMENT IN SUBSEQUENT YEARS
  for (int i=styr+1;i<=endyr;i++)
   {
```

```
     natage(i,1)=mfexp(mean_log_rec+rec_dev(i));
    }

  //COMPUTE NUMBERS AT AGES 2 TO PLUS-GROUP VIA FORWARD PROJECTION
  for (i=styr;i< endyr;i++)
    {
      for (j=2;j<=nages;j++)
        {
        natage(i+1)(j)=natage(i)(j-1)*S(i)(j-1);
        }
      natage(i+1,nages)+=natage(i,nages)*S(i,nages);
    }

  //COMPUTE VARIABLES DERIVED FROM NUMBERS AT AGE MATRIX
  for (i=styr;i<=endyr;i++)
    {

    //COMPUTE PREDICTED FALL SURVEY INDEX AND AGE COMPOSITION
     natage_FALL=elem_prod(natage(i),mfexp(-zfrac_FALL*Z(i)));
     totn_FALL(i)=(natage_FALL*sel_FALL);
     pred_FALL(i)=qFALL*pow((natage_FALL*elem_prod(sel_FALL,wt)),exp_FALL);
     pred_p_FALL(i)=elem_prod(sel_FALL,natage_FALL)/totn_FALL(i);

    //COMPUTE PREDICTED SPRING SURVEY INDEX AND AGE COMPOSITION
     natage_SPR=elem_prod(natage(i),mfexp(-zfrac_SPR*Z(i)));
     totn_SPR(i)=(natage_SPR*sel_SPR);
     pred_SPR(i)=qSPR*pow((natage_SPR*elem_prod(sel_SPR,wt)),exp_SPR);
     pred_p_SPR(i)=elem_prod(sel_SPR,natage_SPR)/totn_SPR(i);

     //COMPUTE POPULATION AND SPAWNING AND EXPLOITABLE BIOMASS
     popnbiom(i)=natage(i)*wt;
     natage_spawn=elem_prod(natage(i),mfexp(-zfrac_spawn*Z(i)));
     spawnbiom(i)=natage_spawn*elem_prod(maturity,wt);
     explbiom(i)=natage(i)*elem_prod(sel,wt);

     //COMPUTE RECRUITMENT
     recruitment(i)=mfexp(mean_log_rec+rec_dev(i));

    }

    //COMPUTE ANNUAL SURPLUS PRODUCTION
    for (i=styr;i<endyr;i++)
      {
      surplus_production(i)=explbiom(i+1)-explbiom(i)+catch_bio(i);
      }

    //COMPUTE DEPLETION RATIOS FOR POPULATION AND SPAWNING BIOMASS
    depletion_popnbiom=popnbiom(endyr)/popnbiom(styr);
    depspawn=spawnbiom(endyr)/spawnbiom(styr);
    deppopnbiom82=popnbiom(endyr)/popnbiom(1982);
    depspawn82=spawnbiom(endyr)/spawnbiom(1982);

    //COMPUTE POPULATION AND SPAWNING BIOMASS IN ENDING YEAR
    endbiom=popnbiom(endyr);
    endspawn=spawnbiom(endyr);

    //COMPUTE F AND NUMBERS AT AGE IN ENDING YEAR
    endF=mfexp(log_avg_fmort+fmort_dev(endyr));
    endN=natage(endyr);

  //cout << "END OF GET NUMBERS AT AGE" << endl;

FUNCTION get_catch_at_age
  //COMPUTE CATCH NUMBERS BY YEAR AND AGE
  for (int i=styr; i<=endyr; i++)
   {
```

```
  pred_catch(i)=0.;

  //APPLY THE CATCH EQUATION
  for (int j = 1 ; j<= nages; j++)
    {
    catage(i,j) = natage(i,j)*F(i,j)*(1.-S(i,j))/Z(i,j);

    //COMPUTE PREDICTED CATCH BIOMASS
    pred_catch(i)+=catage(i,j)*wt(j);
    }

    //COMPUTE PREDICTED FISHERY AGE COMPOSITION
    pred_p_fish(i)=catage(i)/sum(catage(i));
  }
 //cout << "END OF GET CATCH AT AGE" << endl;


FUNCTION evaluate_the_objective_function
 //COMPUTE THE MODEL LIKELIHOOD (f)
 f=.0;

 //DO THIS WHEN RECRUITMENT DEVIATIONS ARE ESTIMATED (PHASE>2)
 if (active(rec_dev))
  {
  age_like=0.;
  int ii;

  //COMPUTE RECRUITMENT LIKELIHOOD COMPONENT
  rec_like=norm2(rec_dev);
  f+=lambda_recruitment*rec_like;

  //COMPUTE AGE COMPOSITION LIKELIHOODS
  //FISHERY COMPONENT
  for (int i=1; i <= nobs_fish; i++)
    {
    ii=yrs_fish(i);
    for (int j=1; j<=nages; j++)
      {
      if (obs_p_fish(i,1) >= 0.0)
        age_like(1)-=nsamples_fish(i)*obs_p_fish(i,j)*log(pred_p_fish(ii,j)+1.e-13);
     //cout << "FISHERY AGE: "<<age_like(1) << " " << i << " "<<j<< endl;
      }
    }
    age_like(1)-=offset(1);
    age_like(1)*=lambda_fishery_age;

  //AUTUMN SURVEY COMPONENT
  for (i=1; i <= nobs_FALL; i++)
    {
    ii=yrs_FALL(i);
    for (int j=1; j<=nages; j++)
      {
      if (obs_p_FALL(i,1) >= 0.0)
        age_like(2)-=nsamples_FALL*obs_p_FALL(i,j)*log(pred_p_FALL(ii,j)+1.e-13);
     //cout << "FALL SURVEY AGE: " << age_like(2) << " " << i << " "<<j<< endl;
      }
    }
    age_like(2)-=offset(2);
    age_like(2)*=lambda_FALL_age;

  //SPRING SURVEY COMPONENT
  for (i=1; i <= nobs_SPR; i++)
    {
    ii=yrs_SPR(i);
    for (int j=1; j<=nages; j++)
      {
```

```
      if (obs_p_SPR(i,1) >= 0.0)
        age_like(3)-=nsamples_SPR*obs_p_SPR(i,j)*log(pred_p_SPR(ii,j)+1.e-13);
     //cout << "SPRING SURVEY AGE: " << age_like(3) << " " << i << " "<<j<< endl;
      }
    }
    age_like(3)-=offset(3);
    age_like(3)*=lambda_SPR_age;

    f+=sum(age_like);
  }

//COMPUTE AUTUMN SURVEY INDEX LIKELIHOOD (LOGNORMAL)
FALL_like=norm2(log(obs_FALL+0.001)-log(pred_FALL(yrs_FALL)+0.001));
FALL_like*=0.5*double(size_count(obs_FALL));
f+=lambda_biomass_index_FALL*FALL_like;

//COMPUTE ROOT MEAN SQUARED ERROR FOR FALL SURVEY INDEX FIT
rmse_FALL=norm(log(obs_FALL+0.001)-log(pred_FALL(yrs_FALL)+0.001));
rmse_FALL*=1.0/sqrt(double(size_count(obs_FALL)));

//COMPUTE SPRING SURVEY INDEX LIKELIHOOD (LOGNORMAL)
SPR_like=norm2(log(obs_SPR+0.001)-log(pred_SPR(yrs_SPR)+0.001));
SPR_like*=0.5*double(size_count(obs_SPR));
f+=lambda_biomass_index_SPR*SPR_like;

//COMPUTE ROOT MEAN SQUARED ERROR FOR SPRING SURVEY INDEX FIT
rmse_SPR=norm(log(obs_SPR+0.001)-log(pred_SPR(yrs_SPR)+0.001));
rmse_SPR*=1.0/sqrt(double(size_count(obs_SPR)));

//COMPUTE CATCH BIOMASS LIKELIHOOD
catch_like=norm2(log(catch_bio+0.000001)-log(pred_catch+0.000001));
catch_like*=0.5*double(size_count(catch_bio));
f+=lambda_catch_biomass*catch_like;

//COMPUTE ROOT MEAN SQUARED ERROR FOR CATCH BIOMASS FIT
rmse_catch_bio=norm(log(catch_bio+0.000001)-log(pred_catch+0.000001));
rmse_catch_bio*=1.0/sqrt(double(size_count(catch_bio)));

//COMPUTE SELECTIVITY LIKELIHOODS
//FISHERY COMPONENT
sel_like(1)=norm2(first_difference(first_difference(log_sel_fish)));
f+=lambda_fishery_sel*square(avgsel_fish);

//SURVEY COMPONENTS (PLACEHOLDERS FOR FUTURE USE)
sel_like(2)=0.;
sel_like(3)=0.;
sel_like(4)=0.;

f+=lambda_fishery_sel*sel_like(1);

//COMPUTE F PENALTY LIKELIHOOD CONSTRAINT
//HIGH PENALTY IF ESTIMATION PHASE < 3
//LOW PENALTY IF ESTIMATION PHASE >= 3
if (current_phase()<3)
  {
   fpen=10.*norm2(mfexp(fmort_dev+log_avg_fmort)-.1);
  }
else
  {
   fpen=0.001*norm2(mfexp(fmort_dev+log_avg_fmort)-.1);
  }
if (active(fmort_dev))
  {
   fpen+=norm2(fmort_dev);
  }
f+=lambda_f_penalty*fpen;
```

```
REPORT_SECTION
  //OUTPUT RESULTS TO FILE "WIN.REP"

  report << "Winter Flounder Age-structured Model WIN" << endl;

  report << "Estimated Numbers (000s) of Fish at Age (year,age)" << endl;
  report << natage << endl;
  report << "Estimated Fishing Mortality (year,age)" << endl;
  report << F << endl;

  report << "Observed FALL SURVEY Biomass Index (year)" << endl;
  report << yrs_FALL << endl;
  report << obs_FALL << endl;
  report << "Predicted FALL SURVEY Biomass Index (year)" << endl;
  report << pred_FALL << endl;
  report << "Residuals for FALL SURVEY Biomass Index (year)" << endl;
  report << obs_FALL - pred_FALL(yrs_FALL) << endl;

  report << "Observed SPRING SURVEY Biomass Index (year)" << endl;
  report << yrs_SPR << endl;
  report << obs_SPR << endl;
  report << "Predicted SPRING SURVEY Biomass Index (year)" << endl;
  report << pred_SPR << endl;
  report << "Residuals for SPRING SURVEY Biomass Index (year)" << endl;
  report << obs_SPR - pred_SPR(yrs_SPR) << endl;

  report << "Fishery age composition effective sample size (year)" << endl;
  report << yrs_fish << endl;
  report << nsamples_fish << endl;

  report << "Observed Fishery Proportion at Age (year,age)" << endl;
  report << obs_p_fish << endl;
  report << "Predicted Fishery Proportion at Age (year,age)" << endl;
  report << pred_p_fish << endl;

  report << "Observed FALL SURVEY Proportion at Age (year,age)" << endl;
  report << obs_p_FALL<< endl;
  report << "Predicted FALL SURVEY Proportion at Age (year,age)" << endl;
  report << pred_p_FALL<< endl;

  report << "Observed SPRING SURVEY Proportion at Age (year,age)" << endl;
  report << obs_p_SPR<< endl;
  report << "Predicted SPRING SURVEY Proportion at Age (year,age)" << endl;
  report << pred_p_SPR<< endl;

  report << "Population Biomass (mt) by Year"<< endl;
  report << years << endl;
  report << popnbiom << endl;
  report << "Population Biomass in 2000" << endl;
  report << endbiom << endl;
  report << "Depletion ratio in 2000 for population biomass" << endl;
  report << depletion_popnbiom << endl;
  report << "Depletion ratio in 2000 relative to 1982 population biomass" << endl;
  report << deppopnbiom82 << endl;

  report << "Spawning Biomass (mt) by Year" << endl;
  report << years << endl;
  report << spawnbiom << endl;
  report << "Spawning Biomass in 2000" << endl;
  report << endspawn << endl;
  report << "Depletion ratio in 2000 for spawning biomass" << endl;
  report << depspawn << endl;
  report << "Depletion ratio in 2000 relative to 1982 spawning biomass" << endl;
  report << depspawn82 << endl;
```

```
report << "Exploitable Biomass (mt) by Year"<< endl;
report << years << endl;
report << explbiom << endl;

report << "Population numbers at age (thousands) in 2000" << endl;
report << ages << endl;
report << endN << endl;

report << "Mean Recruitment (thousands of age-1 recruits)" << endl;
report << mfexp(mean_log_rec) << endl;

report << "Recruitment (thousands of age-1 recruits) by Year" << endl;
report << rec_years << endl;
report << mfexp(mean_log_rec+rec_dev) << endl;

report << "Observed Catch Biomass (mt) by Year" << endl;
report << years << endl;
report << catch_bio << endl;
report << "Predicted Catch Biomass (mt) by Year" << endl;
report << pred_catch << endl;
report << "Residuals for Catch Biomass (year)" << endl;
report << catch_bio - pred_catch << endl;

report << "Annual Surplus Production (mt)" << endl;
report << years << endl;
report << surplus_production << endl;

report << "Estimated Average Annual Fishing Mortality by Year" << endl;
report << years << endl;
report << mfexp(log_avg_fmort+fmort_dev) << endl;
report << "Fishing Mortality in 2000" << endl;
report << endF << endl;

report << "Fishery Selectivity by Age" << endl;
report << ages << endl;
report << sel << endl;

report << "FALL SURVEY Selectivity by Age" << endl;
report << ages << endl;
report << sel_FALL << endl;

report << "SPRING SURVEY Selectivity by Age" << endl;
report << ages << endl;
report << sel_SPR << endl;

report << "OBJECTIVE FUNCTION VALUE: " << f << endl;

report << "LIKELIHOOD EMPHASIS FACTORS" << endl;
report<< "RECRUITMENT::FISHERY AGE::FALL SURVEY AGE::SPRING SURVEY AGE::F PENALTY"<<endl;
report << lambda_recruitment<<"  "<< lambda_fishery_age<< " "<<lambda_FALL_age<<" "<<lambda_SPR_age<<"
"<<lambda_f_penalty<<endl;
report<< "FISHERY SELECTIVITY::CATCH BIOMASS::FALL SURVEY INDEX::SPRING SURVEY INDEX"<<endl;
report << lambda_fishery_sel<<" "<<lambda_catch_biomass<<" "<<lambda_biomass_index_FALL<<"
"<<lambda_biomass_index_SPR<<endl;

report << "LIKELIHOOD COMPONENTS" << endl;
report<< "RECRUITMENT::FISHERY AGE::FALL SURVEY AGE::SPRING SURVEY AGE::F PENALTY"<<endl;
report << rec_like<<"  "<< age_like<< " "<<fpen<<endl;
report<< "FISHERY SELECTIVITY::CATCH BIOMASS::FALL SURVEY INDEX::SPRING SURVEY INDEX"<<endl;
report << sel_like(1)+square(avgsel_fish)<<" "<<catch_like<<" "<<FALL_like<<" "<<SPR_like<<endl;

report << "ROOT MEAN SQUARE ERRORS" << endl;
report << "CATCH BIOMASS: " << rmse_catch_bio << endl;
report << "FALL SURVEY INDEX: " << rmse_FALL << endl;
report << "SPRING SURVEY INDEX: " << rmse_SPR << endl;
```

//END OF MODEL

Input Data File

```
# Styr endyr
1964 2000
# Number of age classes
7
# Number of age classes for selectivity estimation
6
# Catch biomass:1964 to 2000, n=37
1517	1687	2197	2349	1999	2518	2716	4183	4512	2976	2218	2937	1893
	3594	3250	3064	3975	4012	2980	3908	3931	2163	1787	2669	2859
	1891	1953	1828	1849	1683	972	760	1336	1430	1335	1042	1839
# Number years of fishery age data 1982-2000
19
# Years of age fishery data
1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994
	1995	1996	1997	1998	1999	2000
# Number of age samples in fishery (nsamples_fish)
200 200 200 200 200 200 200 200 200 200 200 100 100 100 100 100 50 50 100
# Fishery age composition data 1 2 3 4 5 6 7+
0.00000	0.08486	0.41064	0.25206	0.12279	0.06206	0.06759
0.00157	0.12224	0.45068	0.22582	0.08562	0.03200	0.08208
0.00000	0.05432	0.10992	0.26437	0.27156	0.12245	0.17738
0.00647	0.26544	0.22862	0.26775	0.16188	0.03678	0.03306
0.00000	0.24850	0.49630	0.08792	0.08545	0.04886	0.03297
0.00000	0.30680	0.39872	0.21318	0.03159	0.02101	0.02870
0.00000	0.17289	0.57407	0.17440	0.04079	0.01854	0.01931
0.00000	0.40298	0.35663	0.14864	0.04296	0.03114	0.01766
0.00000	0.09098	0.62677	0.20613	0.05692	0.01404	0.00515
0.00000	0.19447	0.41678	0.31196	0.04457	0.01240	0.01982
0.00000	0.27448	0.26059	0.25070	0.16133	0.03178	0.02113
0.01507	0.12209	0.46449	0.18316	0.12986	0.06627	0.01906
0.00000	0.34688	0.37904	0.16016	0.04382	0.03691	0.03320
0.17432	0.44893	0.17637	0.12455	0.04998	0.01249	0.01335
0.00000	0.39007	0.30046	0.12727	0.08299	0.05508	0.04412
0.00000	0.20124	0.46752	0.24724	0.05527	0.01459	0.01413
0.00359	0.04938	0.62445	0.27636	0.03345	0.00860	0.00418
0.01678	0.31642	0.42977	0.14492	0.07181	0.01608	0.00423
0.00000	0.16969	0.44941	0.16627	0.10002	0.07469	0.03991
# Number of years of FALL SURVEY data 1964-2000
37
# Years of FALL SURVEY data
1964	1965	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976
	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988
	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000
# Fraction of Z Prior to FALL SURVEY (fraction of year)
0.75
# Untransformed FALL SURVEY biomass index
1.822	2.05	5.655	2.074	1.072	2.385	6.49	1.259	1.58	1.195	1.464	2.061	3.925
	3.992	3.1	3.829	1.865	2.434	2.692	2.363	2.445	1.119	2.178	0.889	1.273
	1.051	0.346	0.136	0.384	0.663	0.578	1.337	1.756	1.534	1.565	2.641	2.66
# Number of age samples in FALL SURVEY (nsamples_FALL)
100
# FALL SURVEY age composition data
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
```

```
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
0.39119	0.42743	0.08732	0.06710	0.02423	0.00273	0.00000
0.02611	0.22086	0.42943	0.18562	0.02169	0.07220	0.04409
0.17857	0.26801	0.24763	0.21944	0.06231	0.01591	0.00813
0.17298	0.53288	0.22308	0.04219	0.01444	0.01444	0.00000
0.32442	0.46462	0.10849	0.06005	0.01420	0.00714	0.02107
0.06907	0.26720	0.28637	0.15901	0.00000	0.10360	0.11476
0.72489	0.15738	0.09565	0.00979	0.00000	0.00615	0.00615
0.06271	0.69030	0.04701	0.09227	0.04721	0.03744	0.02305
0.16881	0.12058	0.60812	0.00000	0.10249	0.00000	0.00000
0.28464	0.00000	0.38702	0.32834	0.00000	0.00000	0.00000
0.03443	0.68125	0.23156	0.01389	0.03887	0.00000	0.00000
0.50592	0.11283	0.21099	0.14772	0.02255	0.00000	0.00000
0.18945	0.49293	0.18186	0.09771	0.03805	0.00000	0.00000
0.41367	0.38391	0.15375	0.02052	0.02044	0.00000	0.00770
0.07968	0.21967	0.40560	0.15832	0.03574	0.06070	0.04029
0.04531	0.39693	0.33262	0.17132	0.03563	0.01640	0.00180
0.15952	0.14311	0.36919	0.20744	0.09460	0.01062	0.01552
0.14880	0.30322	0.13468	0.12637	0.23342	0.03395	0.01956
0.02297	0.21601	0.21592	0.13434	0.20023	0.09406	0.11647
# Number of years of SPRING SURVEY data 1968-2000
33
# SPRING SURVEY years
1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980
	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992
	1993	1994	1995	1996	1997	1998	1999	2000
# Fraction of Z Prior to SPRING SURVEY (fraction of year)
0.25
# SPRING SURVEY biomass index
3.114	4.29	2.294	2.168	5.321	3.507	5.782	1.407	3.012	1.58	5.055	2.206	2.801
	3.749	1.523	7.111	5.604	2.65	1.214	1.247	1.648	0.757	1.573	1.319	0.898
	0.57	0.578	1.489	1.504	1.192	0.722	3.479	3.693
# Number of age samples in SPRING SURVEY (nsamples_SPR)
100
# SPRING SURVEY age composition data
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
-99	-99	-99	-99	-99	-99	-99
0.03256445	0.348646378	0.170123409	0.26361698	0.07734057	0.066227305
0.041480907
0.00312664	0.122095306	0.372834492	0.188206379	0.07976407	0.08290808
0.151065033
0.006363204	0.025637641	0.3457253	0.277974336	0.082880076	0.098669272
0.162750172
0.0	0.482292237	0.161910718	0.163787038	0.103484221	0.057488338	0.031037448
0.125667782	0.33042089	0.368865146	0.057866094	0.07983424	0.037345849	0
```

```
0.057474135	0.5882269	0.208669283	0.104851944	0.032215483	0	0.008562255
0.024786325	0.183042735	0.490735043	0.232444444	0.039692308	0.01408547
0.015213675
0.037182448	0.408929946	0.205773672	0.17382602	0.119938414	0.013856813
0.040492687
0.045308598	0.217909383	0.557367107	0.118444524	0.034819836	0.026150553	0
0.112447459	0.143410046	0.343043822	0.241458238	0.114528278	0.015023513
0.030088643
0.050995903	0.423788671	0.211046758	0.09824834	0.10312191	0.077270801
0.035527617
0.167436376	0.267957158	0.326422325	0.151321608	0	0.045494743	0.04136779
0.097391035	0.445382055	0.316095069	0.079043121	0.027792831	0.034295889	0
0.056916482	0.299203858	0.481512669	0.112340198	0.038850188	0.011176606	0
0.016079533	0.526950508	0.186384266	0.211281608	0.029781716	0.011368057
0.018154312
0.014977317	0.119507799	0.332732583	0.415698216	0.070784911	0.014977317
0.031321857
0.0	0.030569404	0.212673839	0.559564419	0.162686959	0	0.034505379
0.057981987	0.141130401	0.160579559	0.338911369	0.231327503	0.051638167
0.018431014
0.004073781	0.138915922	0.230689148	0.140364377	0.256919769	0.148874052
0.080162951
# Weight at age 1 2 3 4 5 6 7+
0.207	0.371	0.511	0.736	0.997	1.281	1.648
# Fraction of Z Prior to Spawning Season (fraction of year)
0.083
# Maturity at age 1 2 3 4 5 6 7+
0.05	0.62	0.98	1.00	1.00	1.00	1.00
# Likelihood emphasis: recruitment
10.0
# Likelihood emphasis: fishery age composition
1.0
# Likelihood emphasis: FALL SURVEY age composition
1.0
# Likelihood emphasis: FALL SURVEY biomass index
10.0
# Likelihood emphasis: SPRING SURVEY age composition
1.0
# Likelihood emphasis: SPRING SURVEY biomass index
10.0
# Likelihood emphasis: catch biomass
10.0
# Likelihood emphasis: fishery selectivity
10.0
# Likelihood emphasis: F penalty
1.0
```

Standard Deviation Parameter File

index	name	value	std dev
1	mean_log_rec	8.6215e+00	1.7002e-02
2	rec_dev	-8.2050e-01	1.6275e-01
3	rec_dev	-1.2486e+00	1.3983e-01
4	rec_dev	-1.0944e+00	1.5023e-01
5	rec_dev	7.4126e-01	6.7549e-02
6	rec_dev	-8.4282e-01	1.6241e-01
7	rec_dev	-1.3451e+00	1.4344e-01
8	rec_dev	-9.8949e-01	1.5730e-01
9	rec_dev	1.0137e+00	6.1918e-02
10	rec_dev	-5.8837e-01	1.7445e-01
11	rec_dev	-2.1967e-01	1.7382e-01
12	rec_dev	-1.4790e-01	1.9466e-01
13	rec_dev	5.0998e-01	1.8554e-01
14	rec_dev	7.2459e-02	2.1847e-01

15	rec_dev	1.0505e+00	1.0822e-01
16	rec_dev	1.2788e-01	2.0868e-01
17	rec_dev	4.0319e-02	2.0795e-01
18	rec_dev	1.1865e+00	9.6898e-02
19	rec_dev	-2.2321e-01	1.9630e-01
20	rec_dev	5.1868e-01	1.2550e-01
21	rec_dev	4.6350e-01	1.0412e-01
22	rec_dev	5.0984e-01	7.4753e-02
23	rec_dev	5.4443e-01	5.7669e-02
24	rec_dev	8.1636e-01	4.9546e-02
25	rec_dev	4.1616e-01	5.8773e-02
26	rec_dev	-4.3253e-01	7.1735e-02
27	rec_dev	2.1871e-01	4.9439e-02
28	rec_dev	1.5614e-01	5.2478e-02
29	rec_dev	3.2470e-01	5.1051e-02
30	rec_dev	-4.0341e-01	6.2112e-02
31	rec_dev	4.5593e-01	4.3474e-02
32	rec_dev	-3.1633e-01	5.6364e-02
33	rec_dev	-5.5266e-01	6.0753e-02
34	rec_dev	-4.8512e-02	4.7871e-02
35	rec_dev	-5.6973e-01	6.3774e-02
36	rec_dev	-2.3163e-02	4.8388e-02
37	rec_dev	3.7160e-02	5.0147e-02
38	rec_dev	3.3298e-01	4.5074e-02
39	rec_dev	1.2184e-02	5.5837e-02
40	rec_dev	-3.2015e-01	8.0130e-02
41	rec_dev	5.4326e-01	7.4621e-02
42	rec_dev	4.4625e-01	1.1027e-01
43	rec_dev	-3.5234e-01	1.8716e-01
44	qFALL	7.8547e-01	3.8040e-02
45	log_gamma_FALL	-1.8483e-01	1.0644e-02
46	log_beta_FALL	1.2829e+00	4.4374e-02
47	log_a50_FALL	1.7234e+00	7.0717e-03
48	qSPR	3.3045e-01	1.2884e-02
49	log_beta_SPR	9.6575e-01	5.9038e-02
50	log_a50_SPR	7.0653e-01	2.6172e-02
51	log_avg_fmort	-6.4774e-01	4.0836e-02
52	fmort_dev	-1.3730e+00	6.4691e-02
53	fmort_dev	-1.2356e+00	5.9459e-02
54	fmort_dev	-7.6300e-01	6.2047e-02
55	fmort_dev	-3.7726e-01	6.8532e-02
56	fmort_dev	-7.5135e-01	5.7105e-02
57	fmort_dev	-6.3439e-01	5.2751e-02
58	fmort_dev	-4.1164e-01	5.6960e-02
59	fmort_dev	2.7401e-01	6.3205e-02
60	fmort_dev	5.1091e-01	5.6296e-02
61	fmort_dev	4.1293e-01	6.1328e-02
62	fmort_dev	-3.0238e-02	5.9496e-02
63	fmort_dev	3.7326e-02	5.7395e-02
64	fmort_dev	-4.9396e-01	5.9352e-02
65	fmort_dev	-1.4176e-01	5.0880e-02
66	fmort_dev	-1.2949e-01	5.4191e-02
67	fmort_dev	-1.2996e-01	5.4272e-02
68	fmort_dev	1.3864e-01	4.9948e-02
69	fmort_dev	1.4762e-01	4.7535e-02
70	fmort_dev	-2.1702e-01	4.8527e-02
71	fmort_dev	3.4252e-02	4.8676e-02
72	fmort_dev	6.2844e-01	4.6381e-02
73	fmort_dev	5.1043e-01	5.3372e-02
74	fmort_dev	1.7459e-01	5.2940e-02
75	fmort_dev	5.9198e-01	5.0505e-02
76	fmort_dev	8.3680e-01	4.6599e-02
77	fmort_dev	7.3242e-01	5.1137e-02
78	fmort_dev	9.8974e-01	4.7055e-02
79	fmort_dev	9.9438e-01	4.6675e-02
80	fmort_dev	8.2763e-01	4.6682e-02

81	fmort_dev	8.0826e-01	4.7569e-02
82	fmort_dev	2.9701e-01	5.4910e-02
83	fmort_dev	-2.9966e-01	5.6301e-02
84	fmort_dev	6.5725e-02	5.5130e-02
85	fmort_dev	-9.7817e-02	5.6121e-02
86	fmort_dev	-5.2763e-01	5.2924e-02
87	fmort_dev	-9.0487e-01	5.3794e-02
88	fmort_dev	-4.9443e-01	5.9806e-02
89	log_selcoffs_fish	-3.7076e+00	2.5569e-01
90	log_selcoffs_fish	-1.0016e+00	2.2794e-01
91	log_selcoffs_fish	2.7917e-01	2.2589e-01
92	log_selcoffs_fish	5.7593e-01	2.2597e-01
93	log_selcoffs_fish	4.1973e-01	2.2705e-01
94	log_selcoffs_fish	-1.4280e-02	2.3325e-01
95	spawnbiom	2.1676e+04	4.8604e+02
96	spawnbiom	1.8943e+04	4.9879e+02
97	spawnbiom	1.6296e+04	4.8540e+02
98	spawnbiom	1.5627e+04	5.0103e+02
99	spawnbiom	1.4772e+04	5.3400e+02
100	spawnbiom	1.3455e+04	5.2512e+02
101	spawnbiom	1.2154e+04	5.2700e+02
102	spawnbiom	1.1504e+04	4.2093e+02
103	spawnbiom	9.8498e+03	3.6567e+02
104	spawnbiom	9.1607e+03	2.9671e+02
105	spawnbiom	9.0487e+03	3.1476e+02
106	spawnbiom	8.8685e+03	3.4346e+02
107	spawnbiom	1.0359e+04	3.3947e+02
108	spawnbiom	1.2052e+04	3.8969e+02
109	spawnbiom	1.1878e+04	3.8176e+02
110	spawnbiom	1.1649e+04	3.8238e+02
111	spawnbiom	1.1630e+04	3.7208e+02
112	spawnbiom	1.1150e+04	3.6392e+02
113	spawnbiom	1.1211e+04	3.3988e+02
114	spawnbiom	1.1936e+04	3.3971e+02
115	spawnbiom	1.0157e+04	3.0797e+02
116	spawnbiom	6.9190e+03	2.3348e+02
117	spawnbiom	6.0976e+03	2.1309e+02
118	spawnbiom	6.3685e+03	1.9874e+02
119	spawnbiom	5.1865e+03	1.6603e+02
120	spawnbiom	4.3905e+03	1.2716e+02
121	spawnbiom	4.0780e+03	1.1806e+02
122	spawnbiom	2.7955e+03	8.4724e+01
123	spawnbiom	2.4671e+03	6.8815e+01
124	spawnbiom	2.5001e+03	6.9560e+01
125	spawnbiom	2.6958e+03	7.2263e+01
126	spawnbiom	3.6855e+03	9.6371e+01
127	spawnbiom	5.2631e+03	1.3767e+02
128	spawnbiom	5.9912e+03	1.7974e+02
129	spawnbiom	6.2764e+03	2.1396e+02
130	spawnbiom	7.7520e+03	2.6217e+02
131	spawnbiom	9.8740e+03	3.7012e+02
132	recruitment	1.4459e+03	2.1202e+02
133	recruitment	2.0632e+03	3.3071e+02
134	recruitment	1.5293e+04	9.0035e+02
135	recruitment	3.0814e+03	5.4537e+02
136	recruitment	4.4552e+03	7.8581e+02
137	recruitment	4.7868e+03	9.4116e+02
138	recruitment	9.2418e+03	1.6688e+03
139	recruitment	5.9668e+03	1.3215e+03
140	recruitment	1.5867e+04	1.6339e+03
141	recruitment	6.3068e+03	1.3394e+03
142	recruitment	5.7781e+03	1.2133e+03
143	recruitment	1.8179e+04	1.6815e+03
144	recruitment	4.4395e+03	8.8476e+02
145	recruitment	9.3225e+03	1.1728e+03
146	recruitment	8.8220e+03	9.1351e+02

147	recruitment	9.2405e+03	6.8942e+02
148	recruitment	9.5657e+03	5.4987e+02
149	recruitment	1.2555e+04	6.0146e+02
150	recruitment	8.4142e+03	4.9152e+02
151	recruitment	3.6011e+03	2.5964e+02
152	recruitment	6.9065e+03	3.3611e+02
153	recruitment	6.4876e+03	3.3295e+02
154	recruitment	7.6788e+03	3.7497e+02
155	recruitment	3.7074e+03	2.2646e+02
156	recruitment	8.7556e+03	3.5621e+02
157	recruitment	4.0448e+03	2.2319e+02
158	recruitment	3.1934e+03	1.9114e+02
159	recruitment	5.2870e+03	2.4584e+02
160	recruitment	3.1394e+03	2.0001e+02
161	recruitment	5.4227e+03	2.5661e+02
162	recruitment	5.7599e+03	2.8643e+02
163	recruitment	7.7426e+03	3.4616e+02
164	recruitment	5.6178e+03	3.1303e+02
165	recruitment	4.0294e+03	3.2557e+02
166	recruitment	9.5546e+03	7.2283e+02
167	recruitment	8.6711e+03	9.7608e+02
168	recruitment	3.9017e+03	7.4857e+02
169	endbiom	1.2042e+04	4.7784e+02
170	depletion_popnbiom	5.2754e-01	1.7868e-02
171	endspawn	9.8740e+03	3.7012e+02
172	depspawn	4.5552e-01	1.4254e-02
173	deppopnbiom82	8.1339e-01	3.0019e-02
174	depspawn82	8.8073e-01	2.9353e-02
175	endN	3.9017e+03	7.4857e+02
176	endN	7.0786e+03	7.9681e+02
177	endN	6.1046e+03	4.6207e+02
178	endN	1.7612e+03	1.4514e+02
179	endN	1.4610e+03	9.3273e+01
180	endN	1.0872e+03	6.9742e+01
181	endN	8.1489e+02	5.9018e+01
182	endF	3.1912e-01	2.3925e-02