Dataset of values of steady state dimensionless variables in food web model as a function of temperature and net primary production and Matlab program used to determine those values

Edward Laws[1], Kanchan Maiti[2]

1. Louisiana State University, College of the Coast & Environment, Department of Environmental Sciences, Baton Rouge, LA 70803, USA
2. Louisiana State University, College of the Coast & Environment, Department of Oceanography & Coastal Sciences, Baton Rouge, LA 70803, USA

Edward Laws (edlaws@lsu.edu)

This file contains the values of the five dimensionless parameters (ef, f_{2L}, f_3, f_5, and f_6) that are needed to find the steady state solutions to the food web model equations in the linked article by Laws and Maiti. The relevant equations in that article are equations 12–31. The values are given at temperatures of −1.8, 0, 5, 10, 15, 20, 25, and 30°C and net primary production (NPP) rates in a geometric series from 0.5 to 1024 mg C m^{-3} d^{-1} in multiples of 2. To interpolate to other temperatures and NPPs, it is best to use the logarithms of the NPP values in the table. There are five excel spread sheets, one for each of the dimensionless variables ef, f_{2L}, f_3, f_5, and f_6. The model is a nitrogen-based model, and all concentrations in the model have units of mg N m^{-3}. However, the NPP rates in the five excel spread sheets have units of mg C m^{-3} d^{-1}.

Keywords

model | biological pump | time lags | food webs | resilience

Specifications Table

Subject	Oceanography
Specific subject area	The data provide the information needed to find the steady state concentrations of the food web model described in the linked Deep-Sea Research paper by Laws and Maiti. Note that all concentrations in the model have units of mg N m^{-3}.

Type of data	Table
How data were acquired	The data were acquired by finding the values of the five dimensionless variables that produced a steady state solution of equations 2–11 of the linked Laws and Maiti paper that was associated with maximum resiliency to perturbations. This required setting up the community matrix and finding the eigenvalues of the community matrix. The least negative real part of the 10 eigenvalues determines the resiliency of the food web to perturbations.
Data format	Excel spreadsheet
Parameters for data collection	The temperature in degrees Celsius and primary production rate in units of mg C m^{-3} d^{-1} were specified. The five parameters were then varied systematically to find the combination that produced a steady state with maximum resiliency.
Description of data collection	The data were obtained using a program written in Matlab that may be obtained from E. Laws (edlaws@lsu.edu) upon request.
Data source location	Louisiana State University Baton Rouge, Louisiana United States

Related research article	**Laws, E. A. and Maiti, K., Temperature affects the time required to discern the relationship between primary production and export production in the ocean. Deep-Sea Research Part I.**

Value of the Data

- These data make it easy to find the values of the five dimensionless parameters that produce a steady state solution to the food web model that has maximum resiliency to perturbations. The alternative is to systematically vary the parameters to identify the combination that gives maximum resiliency, but that is very time consuming. With this table, the optimum parameters can be interpolated to any combination of temperature and primary production rate.
- Anyone with an interest in exploring the effects of temperature and photosynthetic rates on the efficiency of the biological pump in the ocean will benefit from this table.
- These data could be used to identify the timeframe over which field studies would need to be carried out to obtain meaningful estimates of the relationship between primary production and export production in the ocean.

Data Description

There are five excel spreadsheets identified as ef, f2L, f3, f5, and f6. Each spreadsheet shows the values of one of these dimensionless parameters as a function of temperature (column A) and primary production rate (row 1). In each case, these are the values of the dimensionless parameters that produced a steady state of the food web described in the Laws and Maiti manuscript with maximum resiliency to perturbations.

The ef spreadsheet shows the values of the export ratio (ef), which is the ratio of export production to primary production.

The f2L spreadsheet shows the values of the dimensionless parameter f2L, which is the growth rate of the large phytoplankton expressed as a fraction of their maximum growth rate at the temperature in column A.

The f3 spreadsheet shows the values of the dimensionless parameter f3, which is the growth rate of the flagellates expressed as a fraction of their maximum growth rate at the temperature in column A.

The f5 spreadsheet shows the values of the dimensionless parameter f5, which is the growth rate of the filter feeders expressed as a fraction of their maximum growth rate at the temperature in column A.

The f6 spreadsheet shows the values of the dimensionless parameter f6, which is the growth rate of the carnivores expressed as a fraction of their maximum growth rate at the temperature in column A.

Experimental Design, Materials and Methods

Below is a complete listing of the Matlab program that was used to determine the steady state solutions associated with maximum resiliency of the food web. Note that there are options to check to make sure that the steady state solutions are being satisfied and that the derivatives that make up the community matrix are being correctly calculated. This option can be exercised by setting the parameter ifcheck equal to 1. If this option is exercised, it should be used to do spot checks, not for every combination of the five dimensionless variables.

```
% This program calculates results using the food web model described
% by Laws and Maiti in Water (2021)

% The differential equations describing the rate of change of each
% trophic level are as follows:

% dx1/dt = L-(1-r2l)*F2l-(1-r2s)*F2s+r3*F3+r4*F4+r5*F5+r6*F6+rb*Fb;
% dx2l/dt = q2l*F2l-F5*x2l/(x2l+x4);
% dx2s/dt = q2s*F2s-F3*x2s/(x2s+xb);
% dx3/dt = q3*F3-F4;
% dx4/dt = q4*F4-F5*x4/(x4+x2l);
% dx5/dt = q5*F5-F6;ef
% dx6/dt = q6*F6-M*x6;
% dDON/dt = s2s*F2s+s3*F3+s4*F4-Fb;
% dPON/dt = s2l*F2l+s5*F5+s6*F6+M*x6-D*PON;
% dxb/dt = qb*Fb-F3*xb/(x2s+xb);

% Box 1 is the inorganic nutrient box
% Box 2l is the large phytoplankton
% Box 2s is the small phytoplankton
% Box 3 is the flagellates, which feed on small phytoplankton and
% bacteria
% Box 4 is the ciliates, which feed on flagellates
% Box 5 is the filter feeders, which feed on the large phytoplankton
% and ciliates
% Box 6 is the carnivores, which feed on the filter feeders

% The remaining boxes are dissolved organic N (DON), particulate
```

```
% organic N detritus (PON), and bacteria (b)


clear;

t0 = clock;

% Temperature is assumed to be Celsius degrees,
% tp is entered in units of mg C per cubic meter per day
% the program converts that to mg N per cubic meter per day
% by dividing by the Redfield ratio of 5.68 grams carbon per gram nitrogen

format short g;

% In this example, the temperature is 0 Celsius, and the primary production
% rate is 512 mg C per cubic meter per day. The primary production rate is
% converted to nitrogen units by dividing by the Redfield C/N ratio of 5.68
% grams of carbon per gram of nitrogen

T=0;
tp=512/5.68;

D=1;

% D is the fraction of the nitrogen detritus that sinks out of the mixed
% layer per day. This determines the value of PON because the steady state
% solution requires that L = PON*D. The requirement for maximum resiliency
% is unaffected by the value assigned to D, and hence a value of 1 per day
% is assumed but is arbitrary.

% The q values are the fraction of ingested food converted to biomass

% For example, the small phytoplankton (X2s) are assumed to convert 70% of
% the N they take up from the water into biomass.  The q values can be
% regarded as gross growth efficiencies

q2s=0.7;
q2l=0.7;
q3=0.35;
q4=0.35;
q5=0.30;
q6=0.35;
qb=1.0;




% r is the fraction of consumed food that is respired
% Bacteria are assumed not to excrete dissolved organics;
% hence rb = 1 - qb

r2s=0.0;
r2l=0.0;
r3=0.3;
r4=0.3;
rb=1-qb;
r5=0.3;
r6=0.5;

% s is the fraction of consumed food excreted as dissolved organics
% By mass balance, q + r + s = 1.
```

```
s2s=1-q2s-r2s;
s2l=1-q2l-r2l;
s3=1-q3-r3;
s4=1-q4-r4;
s5=1-q5-r5;
s6=1-q6-r6;

% Nutrient-saturated growth rates are assumed to be identical for the
% large and small phytoplankton are equal to 1.2 per day at a
% temperature of 25oC.  The temperature dependence of the nutrient-saturated &
growth rate is taken from Eppley, Fish. Bull. volume 70: 1063-1085 (1972).
% The a values are maximum ingestion rates per day

a2s=(1.2/q2s)*exp(0.0633*(T-25));
a2l=(1.2/q2l)*exp(0.0633*(T-25));

% For bacteria, the substrate-saturated growth rate is assumed 1.2 per day at
25 degrees Celsius.

% For references on heterotrophic bacterial growth rates, see the following:
% Shiah and Ducklow, Limnol. Oceanogr. 39: 1243-1258 (1994)
% Shiah and Ducklow, Mar. Ecol. Prog. Ser. 103: 297-308 (1994)
% Shiah and Ducklow, Limnol. Oceanogr. 40: 55-66 (1995)
% White, Kalff, Rasmussen, and Gasol, Microbiol. Ecol. 21: 99-118
% Hobbie and Cole, Bull. Mar. Sci. 35: 357-363 (1984)

ab=(1.2/qb)*exp(0.0633*(T-25));

% The maximum growth rates of the flagellates and ciliates is assumed
% to be 2.4 per day at 25oC based on Fenchel and Finlay, Microb. Ecol
%, vol. 9: 99-122 (1983).  These authors estimate the net growth
% efficiency (production divided by production + respiration)of
% flagellates and ciliates to be 40-60%. In this model, the net growth
% efficiency of the flagellates and ciliates is 0.55 = 0.35/(1 - 0.35),
% since 35% of ingested food is excreted, 35% goes to biomass, and
% 30% is respired.

% The temperature dependence of these rates is taken from Huntley and
% Lopez, American Naturalist, vol. 140: 201-242 (1992).

a3=2.4/q3*exp(0.1*(T-25));
a4=2.4/q4*exp(0.1*(T-25));

% Maximum ingestion rates for the other organisms are assumed to be
% 0.5 per day at a temperature of 25oC.

a5=(.5)*exp(0.1*(T-25));
a6=(.5)*exp(0.1*(T-25));

% P values for large and small phytoplankton are 75 nM and 7.5 nM of
% inorganic N, respectively
% Units are converted to mg per square meter by multiplying by 14
% times the mixed layer thickness and dividing by 1000


p2s=14*7.5/1000; % value of p2s is set to 7.5 nanomolar
p2l=10*p2s;

% The p value for bacteria is 7.5 nM

pb=p2s;
```

```
% f values are relative growth rates, i.e., growth rate divided by the
% nutrient- or food-saturated growth rate.  These relative growth rates
% are all in the range [0 1]
% Assume that the small phytoplankton are growing at some fraction of
% their maximum rate This is arbitrary, and together with p2s
% determines the steady state value of x1

fmax=s2l+q2l*(s5+q5*(s6+q6));  % This is the maximum possible value of the ef
% ratio
fmin=(q2s+qb*s2s)*q4*q3*(s5+q5*(s6+q6))/(1-qb*(s3+s4*q3));  % This is the
% minimum value of the ef ratio

disp(['fmin ' num2str(fmin)]);
disp(['fmax ' num2str(fmax)]);

DN=20; % DN determines the number of intervals of the five
% dimensionless parameters that will be searched

efcenter=0.375; % in this example, the ef search is centered on 0.375
% The ef search can be centered on other values if you expect that the
% optimum value will be higher or lower
deltaef=0.1; % this determines the upper and lower bounds of the searches
% of the 5 dimensionless parameters. You can search over a wide or narrow %
region by adjusting this parameter

% in this example, the search is between 90% and 110% of the center value

deltaf=deltaef;
deltaef=efcenter*(1-deltaef);
efmax=efcenter*(1+deltaef);
efmin=efcenter*(1-deltaef);
if efmin<fmin; % efmin cannot be less than fmin
    efmin=fmin;
end;
if efmax>fmax; % efmax cannot be greater than fmax
    efmax=fmax;
end;

def=(efmax-efmin)/DN;

f2lcenter=0.28942; % center of search for f2L
f3center=0.19184; % center of search for f3
f5center=0.84469; % center of search for f5
f6center=0.4928; % center of search for f6

f2lmin=f2lcenter*(1-deltaf);f2lmax=f2lcenter*(1+deltaf);
f3min=f3center*(1-deltaf);f3max=f3center*(1+deltaf);
f5min=f5center*(1-deltaf);f5max=f5center*(1+deltaf);
f6min=f6center*(1-deltaf);f6max=f6center*(1+deltaf);

df2l=(f2lmax-f2lmin)/DN;
df3=(f3max-f3min)/DN;
df5=(f5max-f5min)/DN;
df6=(f6max-f6min)/DN;

BIGEIG=1000000; % the program will search for the solution for which the real
% part of the least negative eigenvalue is the most negative. Initially this
% value is set to a large positive number.

efm=efmin:def:efmax;
   nef=length(efm);
```

```
   ncount=zeros(1,nef);
for k=1:nef;
ef=efm(k);
bigeig=0;
L=tp*ef;

% The product of D and PON must balance L in steady state

PON=L/D;

F2s=tp*(fmax-ef)/(fmax-fmin);
F2l=tp*(ef-fmin)/(fmax-fmin);
F3=F2s*(q2s+qb*s2s)/(1-qb*(s3+s4*q3));
F4=q3*F3;
F5=q2l*F2l+q4*F4;
F6=q5*F5;
Fb=s2s*F2s+s3*F3+s4*F4;


j=0;
clear bsum;

 for f2l=f2lmin:df2l:f2lmax;

% Require that growth rates of ciliates and large phytoplankton be
% identical.  This is necessary to achieve steady state if the filter
% feeders are assumed to graze large phytoplankton and ciliates in proportion
% to x2l/(x2l + x4) and x4/(x2l + x4), respectively.

f4=q2l*a2l*f2l/(q4*a4);
f2s=1-(1-f2l)*p2s/p2l;
fb=q2s*a2s*f2s/(qb*ab);

% Note that under steady state conditions (1 - f2l)/(1 - f2s) = p2l/p2s
% Require that growth rates of small phytoplankton and bacteria be
% identical.  This is necessary to achieve steady state if the
% ciliates are assumed to graze small phytoplankton and bacteria in
% proportion to x2s/(xb + x2s) and xb/(xb + x2s), respectively.

x1=p2s/(1-f2s);
DON=pb/(1-fb);

    for f3=f3min:df3:f3max;
    for f5=f5min:df5:f5max;
        for f6=f6min:df6:f6max;

x2l=F2l/(a2l*f2l);
x2s=F2s/(a2s*f2s);
x3=F3/(a3*f3);
x4=F4/(a4*f4);
x5=F5/(a5*f5);
x6=F6/(a6*f6);
xb=Fb/(ab*fb);


biom=[x1 x2l x2s x3 x4 x5 x6 DON PON xb];
if min(biom) > 0;  % Check to make sure all biomasses are positive

% Mortality rate of carnivores must balance their production in steady state

M=q6*a6*f6;
```

```
p3=(xb+x2s)*(1-f3);
p4=x3*(1-f4);
p5=(x4+x2l)*(1-f5);
p6=x5*(1-f6);

ifcheck=0; % used if you want to check that the right-hand-sides of the
% equations equal zero

if ifcheck==1;
% Evaluate right-hand side of each equation

rhs1=L-(1-r2l)*F2l-(1-r2s)*F2s+r3*F3+r4*F4+r5*F5+r6*F6+rb*Fb;
rhs2=q2l*F2l-F5*x2l/(x2l+x4);
rhs3=q2s*F2s-F3*x2s/(x2s+xb);
rhs4=q3*F3-F4;
rhs5=q4*F4-F5*x4/(x4+x2l);
rhs6=q5*F5-F6;
rhs7=q6*F6-M*x6;
rhs8=s2s*F2s+s3*F3+s4*F4-Fb;
rhs9=s2l*F2l+s5*F5+s6*F6+M*x6-D*PON;
rhs10=qb*Fb-F3*xb/(x2s+xb);

% Check to be sure that equations have been solved

disp('Right-hand side of equations');
disp([rhs1 rhs2 rhs3 rhs4 rhs5 rhs6 rhs7 rhs8 rhs9 rhs10]);
end;  % end of ifcheck loop



% Calculate turnover times of state variables

tt=1./[q2s*a2s*f2s q2l*a2l*f2l q3*a3*f3 q4*a4*f4 q5*a5*f5 q6*a6*f6 qb*ab*fb];
tmax=max(tt);


% Set up community matrix
% First set up table of derivatives of F functions

d=zeros(7,10);
d(1,1)=a2l*x2l*p2l/(x1*x1);
d(1,2)=F2l/x2l;
d(2,1)=a2s*x2s*p2s/(x1*x1);
d(2,3)=F2s/x2s;
d(3,3)=a3*x3*p3/(xb+x2s)^2;
d(3,4)=F3/x3;
d(3,10)=d(3,3);
d(4,4)=a4*x4*p4/(x3*x3);
d(4,5)=F4/x4;
d(5,2)=a5*x5*p5/(x4+x2l)^2;
d(5,5)=d(5,2);
d(5,6)=F5/x5;
d(6,6)=a6*x6*p6/(x5*x5);
d(6,7)=F6/x6;
d(7,8)=ab*xb*pb/(DON*DON);
d(7,10)=Fb/xb;

% Now set up community matrix

A=zeros(10,10);
```

```
A(1,1)=(r2l-1)*(1-f2l)*a2l*x2l/x1+(r2s-1)*(1-f2s)*a2s*x2s/x1;
A(1,2)=-(1-r2l)*d(1,2)+r5*d(5,2);
A(1,3)=-(1-r2s)*d(2,3)+r3*d(3,3);
A(1,4)=r3*d(3,4)+r4*d(4,4);
A(1,5)=r4*d(4,5)+r5*d(5,5);
A(1,6)=r5*d(5,6)+r6*d(6,6);
A(1,7)=r6*d(6,7);
A(1,8)=rb*d(7,8);
A(1,10)=rb*d(7,10)+r3*d(3,10);

A(2,1)=q2l*d(1,1);
A(2,2)=(2*f5-1)*a5*x5*x2l/(x2l+x4)^2;
A(2,5)=F5*x2l/(x2l+x4)^2-d(5,5)*x2l/(x2l+x4);
A(2,6)=-q2l*F2l*d(5,6)/(q2l*F2l+q4*F4);

A(3,1)=q2s*d(2,1);
A(3,3)=(2*f3-1)*a3*x3*x2s/(x2s+xb)^2;
A(3,4)=-q2s*F2s*d(3,4)/(q2s*F2s+qb*Fb);
A(3,10)=q2s*F2s*(-d(3,10)/(q2s*F2s+qb*Fb)+F3*qb*d(7,10)/(q2s*F2s+qb*Fb)^2);

A(4,3)=q3*d(3,3);
A(4,4)=(2*f4-1)*a4*x4/x3;
A(4,5)=-d(4,5);
A(4,10)=q3*d(3,10);

A(5,2)=-d(5,2)*x4/(x2l+x4)+F5*x4/(x2l+x4)^2;
A(5,4)=q4*d(4,4);

A(5,5)=(2*f5-1)*a5*x5*x4/(x2l+x4)^2;

A(5,6)=-q4*F4*d(5,6)/(q2l*F2l+q4*F4);
A(6,2)=q5*d(5,2);
A(6,5)=q5*d(5,5);

A(6,6)=(2*f6-1)*a6*x6/x5;

A(6,7)=-d(6,7);
A(7,6)=q6*d(6,6);

A(8,1)=s2s*d(2,1);
A(8,3)=s2s*d(2,3)+s3*d(3,3);
A(8,4)=s3*d(3,4)+s4*d(4,4);
A(8,5)=s4*d(4,5);

A(8,8)=(fb-1)*ab*xb/DON;

A(8,10)=s3*d(3,10)-d(7,10);
A(9,1)=s2l*d(1,1);
A(9,2)=s2l*d(1,2)+s5*d(5,2);
A(9,5)=s5*d(5,5);
A(9,6)=s5*d(5,6)+s6*d(6,6);
A(9,7)=s6*d(6,7)+M;
A(9,9)=-D;
A(10,3)=qb*Fb*(-d(3,3)/(q2s*F2s+qb*Fb)+F3*q2s*d(2,3)/(q2s*F2s+qb*Fb)^2);
A(10,4)=-qb*Fb*d(3,4)/(q2s*F2s+qb*Fb);
A(10,8)=qb*d(7,8);

A(10,10)=(2*f3-1)*a3*x3*xb/(x2s+xb)^2;

if ifcheck==1;
    % use finite differences to check that elements of community matrix are
```

```
% correct

dp=0.000001;

  for k=1:10;

x1p=x1;x2lp=x2l;x2sp=x2s;x3p=x3;x4p=x4;x5p=x5;x6p=x6;DONp=DON;PONp=PON;xbp=xb;
      if k==1;
          x1p=x1*(1+dp);
      end;
       if k==2;
          x2lp=x2l*(1+dp);
      end;
      if k==3;
          x2sp=x2s*(1+dp);
      end;
      if k==4;
          x3p=x3*(1+dp);
      end;
      if k==5;
          x4p=x4*(1+dp);
      end;
      if k==6;
          x5p=x5*(1+dp);
      end;
      if k==7;
          x6p=x6*(1+dp);
      end;
      if k==8;
          DONp=DON*(1+dp);
      end;
      if k==9;
          PONp=PON*(1+dp);
      end;
      if k==10;
          xbp=xb*(1+dp);
      end;
 % now calculate perturbed values of equations

 F2lp=a2l*x2lp*(1-p2l/x1p);
 F2sp=a2s*x2sp*(1-p2s/x1p);
 F3p=a3*x3p*(1-p3/(x2sp+xbp));
 F4p=a4*x4p*(1-p4/x3p);
 F5p=a5*x5p*(1-p5/(x2lp+x4p));
 F6p=a6*x6p*(1-p6/x5p);
 Fbp=ab*xbp*(1-pb/DONp);

rhs1p=L-(1-r2l)*F2lp-(1-r2s)*F2sp+r3*F3p+r4*F4p+r5*F5p+r6*F6p+rb*Fbp;

rhs2p=q2l*F2lp-F5p*x2lp/(x2lp+x4p);

rhs3p=q2s*F2sp-F3p*x2sp/(x2sp+xbp);
rhs4p=q3*F3p-F4p;

rhs5p=q4*F4p-F5p*x4p/(x4p+x2lp);

rhs6p=q5*F5p-F6p;

rhs7p=q6*F6p-M*x6p;
rhs8p=s2s*F2sp+s3*F3p+s4*F4p-Fbp;

rhs9p=s2l*F2lp+s5*F5p+s6*F6p+M*x6p-D*PONp;
```

```
rhs10p=qb*Fbp-F3p*xbp/(x2sp+xbp);

dr=[rhs1p rhs2p rhs3p rhs4p rhs5p rhs6p rhs7p rhs8p rhs9p rhs10p]-[rhs1 rhs2
rhs3 rhs4 rhs5 rhs6 rhs7 rhs8 rhs9 rhs10];
dr=dr';
if k==1;
    AA(:,1)=dr/(x1*dp);
end;
if k==2;
    AA(:,2)=dr/(x2l*dp);
end;
if k==3;
    AA(:,3)=dr/(x2s*dp);
end;
if k==4;
    AA(:,4)=dr/(x3*dp);
end;
if k==5;
    AA(:,5)=dr/(x4*dp);
end;
if k==6;
    AA(:,6)=dr/(x5*dp);
end;
if k==7;
    AA(:,7)=dr/(x6*dp);
end;
if k==8;
    AA(:,8)=dr/(DON*dp);
end;
if k==9;
    AA(:,9)=dr/(PON*dp);
end;
if k==10;
    AA(:,10)=dr/(xb*dp);
end;
  end;
  format short;

 disp(A-AA);

end; % end of second ifcheck loop

% Calculate eigenvalues of community matrix

eigv=eig(A);
eigr=real(eigv);
eiga=abs(eigv);
teig=-2*eigr./(eiga.^2);
mineig=max(eigr);

    if mineig<bigeig;
        bigeig=mineig;
        fmat=[f2s f2l f3 fb f4 f5 f6];
        xmat=[x1 x2l x2s x3 x4 x5 x6 DON PON xb];
        if bigeig<BIGEIG;
            BIGEIG=bigeig;
             FMAT=fmat;
             XMAT=xmat;
             EFBIG=ef;
        end;
        end;
```

```
end; % end of biom > 0 check
end; % end of f6 integration
end; % end of f5 integration
end; % end of f3 integration
end; % end of f2l integration

bdata(k)=bigeig;
%disp(ef);

end; % end of ef integration

clf;
plot(efm,bdata,'k');
xlabel('ef ratio');
ylabel('Most negative eigenvalue');

disp([efm;bdata]');

[bigeig,I]=min(bdata);
efbig=efm(I);
disp(['temperature ' num2str(T) ' total production ' num2str(tp*5.68)]);

format long g;
disp([EFBIG BIGEIG]);

format long g;
disp(FMAT');
format long g;
disp(XMAT');

disp([FMAT(2) f2lmin f2lmax;FMAT(3) f3min f3max;FMAT(6) f5min f5max;FMAT(7)
f6min f6max]);
disp(['best ef ratio ' num2str(EFBIG) ' eigenvalue ' num2str(BIGEIG)]);
timeused=etime(clock,t0)/60; % elapsed time in minutes
disp(['elspsed time ' num2str(timeused) ' minutes']);
```