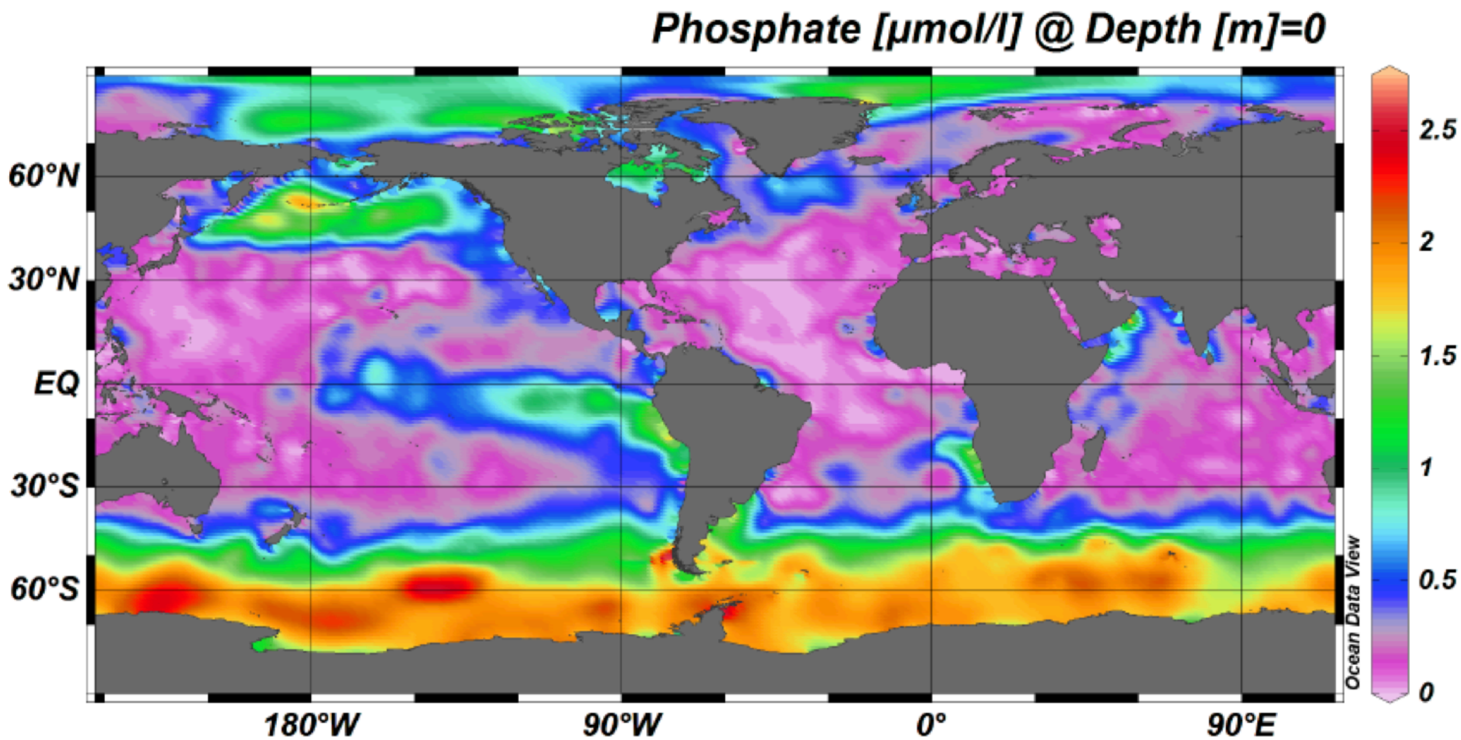
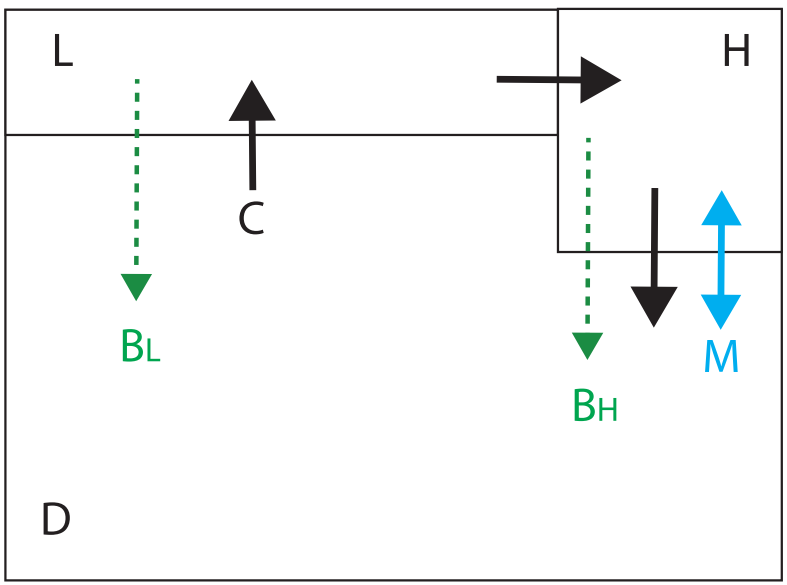
Surface nutrient distribution



The 3 box model



Model equation: a system of 3 ODEs

$$
\begin{aligned}
V_L \frac{dP_L}{dt} &= C(P_D - P_L) - B_L \\
V_H \frac{dP_H}{dt} &= C(P_L - P_H) + M(P_D - P_H) - B_H \\
V_D \frac{dP_D}{dt} &= (C + M)(P_H - P_D) + B_L + B_H
\end{aligned}
$$

Model equation in vector-matrix formulation

$$
\frac{d}{dt}\mathbf{P} = \mathbf{TP}
$$

Solution methods

- Analytic

$$\mathbf{P}(t) = e^{\mathbf{T}t}\mathbf{P}(0)$$

MATLAB

```
>> P(n) = expm(T*t(n))*P(1);
```

- Numerical (Euler forward)

$$\mathbf{P}_{n+1} = (\mathbf{I} + \Delta t\mathbf{T})\mathbf{P}_n$$

```
>> P(n+1) = (eye(3) + dt*T)*P(n);
```

- Numerical (Euler backward)

$$\mathbf{P}_{n+1} = (\mathbf{I} - \Delta t\mathbf{T})^{-1}\mathbf{P}_n$$

>> P(n+1) = (eye(3) - dt*T)$^{-1}$*P(n);

Practical recommendations

- Euler forward scheme can be unstable. Choose Euler backward if the situation allows (it requires matrix inverse calculation).

- If analytic solution is available, use it to check your numerical solution.

- Despite my earlier suggestion, Euler forward scheme can often produce acceptable results if a short enough timestep is used.

The code

week10_exercise.m

```
% PO4 box model

% clean first
clear all;
close all;

% 1. set up model parameter
H = 4000;  % full ocean depth, m
R = 6.3e6; % radius of the Earth, m
Area = 4*pi*R^2*0.8; % total surface area, m2
fL = 0.8;  % fraction of low lat ocean
V(1) = Area*fL*100; % surface low lat ocean vol, m3
V(2) = Area*(1-fL)*300; % surface high lat ocean vol, m3
V(3) = Area*H-V(1)-V(2); % deep ocean vol, m3;
C = 10*1e6;  % circulation rate (m3/s);
M = 40*1e6;  % mixing rate, (m3/s);
lam = 1/(60*60*24*365); % biological P consumption rate, 1/s
```

First set up the model parameters.

The code

```
% 2. set up model matrix
T = [ -(C+lam*V(1))/V(1)       0                    C/V(1); ... % first row
          C/V(2)        -(C+M+lam*V(2))/V(2)  M/V(2); ... % second row
          lam*V(1)/V(3)   (C+M+lam*V(2))/V(3)  -(C+M)/V(3)]; % third row
```

When you code the matrix manually, you enter row vectors in sequence. Each row is separated by semicolon (;)

The code

```
% 3. set initial condition & time step
dt = 60*60*24;  % one day timestep
N = 365*10;     % 10 year integration
P = zeros(3,N); % preparing the solution storage
Pa= zeros(3,N); % preparing the analytic solution storage
Pb= zeros(3,N); % preparing the Euler backward storage
P(:,1) = [2e-3; 2e-3; 2e-3]; % initial P conc. = uniform 2 mmolP/m3
Pa(:,1)= P(:,1); % initial cond for analytic sol.
Pb(:,1)= P(:,1); % initial cond for euler backward sol.
time=0:dt:(N-1)*dt; % set up time array
```

In preparation for the numerical integration, prepare arrays for storing solutions. It makes the calculation faster.

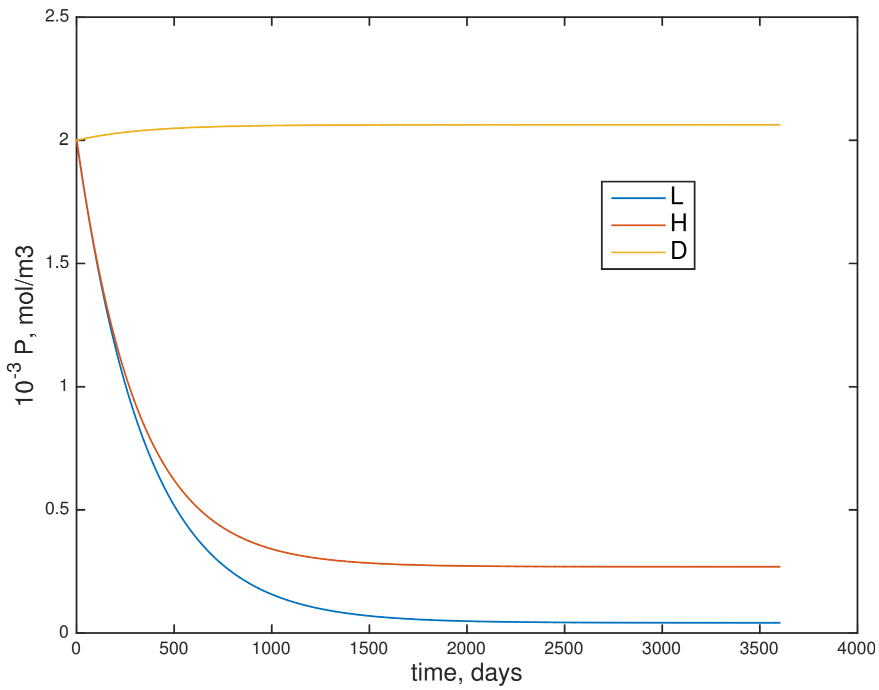
The code

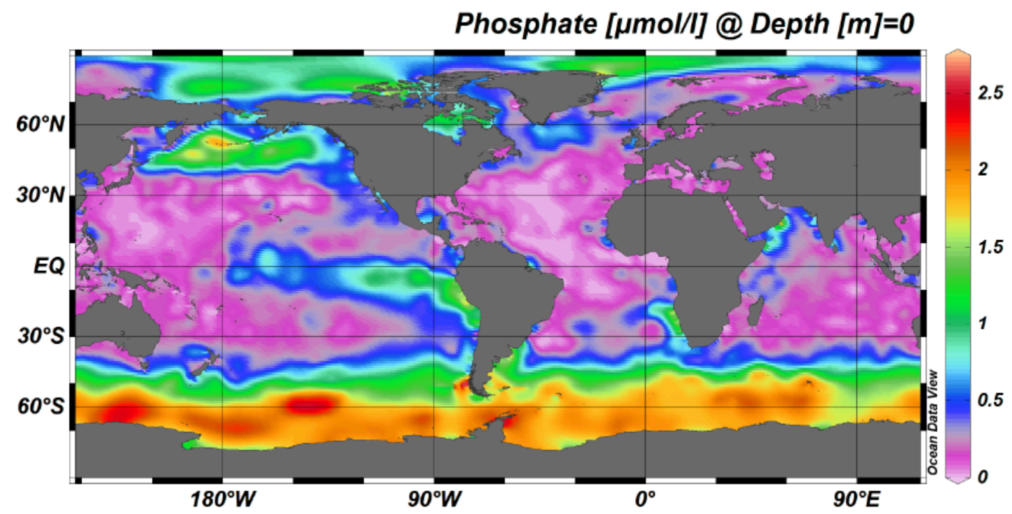
```
% 4. execute numerical integration
G = eye(3,3) + dt*T; % Euler forward
H = inv(eye(3,3) - dt*T); % Euler backward

for n=1:(N-1)
    P(:,n+1) = G*P(:,n); % Euler step forward
    Pb(:,n+1) = H*Pb(:,n); % Euler backward
    Pa(:,n+1)= expm(T*time(n+1))*Pa(:,1); % analytic solution
end
```

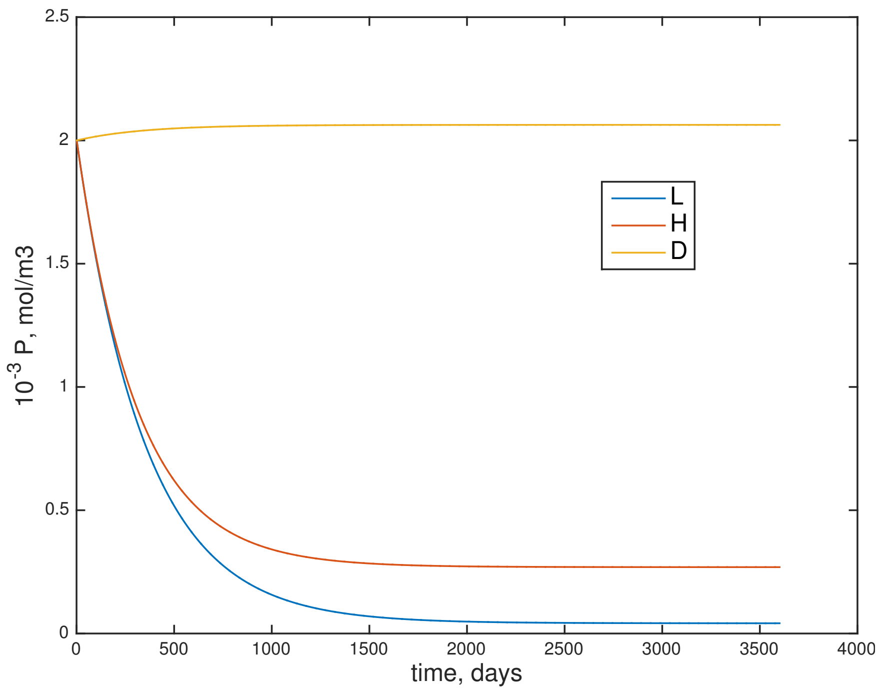
This is the core of the model integration. We apply 3 different methods for comparison. Plotting script follows this section.

The results


2.5
2
1.5
1
0.5
0
10^{-3} P, mol/m3
0
500
1000
1500
2000
2500
3000
3500
4000
time, days
L
H
D


Phosphate [µmol/l] @ Depth [m]=0
60°N
30°N
EQ
30°S
60°S
180°W
90°W
0°
90°E
2.5
2
1.5
1
0.5
0
Ocean Data View

Questions



This system tool approximately 3 years to reach the steady state.

Why 3 years?

Why did the value of deep ocean box hardly change?

Why 3 years?

- Look at the eigenvalue
- Negative eigenvalues indicate the rate of exponential decay
- The inverse of eigenvalues gives the timescale that the initial condition decays.

```
>> [Q,D]=eig(T)

Q =

    0.0098    0.9777   -0.0027
    0.0495    0.2088   -0.9999
    0.9987   -0.0235    0.0156


D =

   1.0e-07 *

         0         0         0
         0   -0.3203         0
         0         0   -0.3382
```

Why 3 years?

- Look at the eigenvalue
- Negative eigenvalues indicate the rate of exponential decay
- The inverse of eigenvalues gives the timescale that the initial condition decays.

```
>> [Q,D]=eig(T)

Q =

    0.0098    0.9777   -0.0027
    0.0495    0.2088   -0.9999
    0.9987   -0.0235    0.0156


D =

   1.0e-07 *

         0         0         0
         0   -0.3203         0
         0         0   -0.3382
```

Constant deep ocean?

- Look at the eigenvalue
- Negative eigenvalues indicate the rate of exponential decay
- The inverse of eigenvalues gives the timescale that the initial condition decays.

```
>> [Q,D]=eig(T)

Q =

    0.0098    0.9777   -0.0027
    0.0495    0.2088   -0.9999
    0.9987   -0.0235    0.0156


D =

   1.0e-07 *

         0         0         0
         0   -0.3203         0
         0         0   -0.3382
```