```
function dcdt = Equations_uncoupled(t,c)

%   This script uses 'Definitions_uncoupled.m' and
'Solve_uncoupled.m' to
%   then solve for the time dependent solution of sulfur and carbon
in the whole ocean
%   based on equations that have:
%   1. CaCO3 output based on a first order rate law with DIC
%   2. The Org C output value based on the fractionation of DIC-Corg
using modernish
%   values of 27 permil.
%   3. The reduced sulfur output is tied to the organic carbon
output via
%   the classic Berner ratio of 1-3 wt %
%   4. The oxidized sulfur output (evaporites) is a constant (steady
state solution is
%       based on a first order
%   rate law with [SO4]).

global Stot_0 F_32_in del_in_S;

Definitions_uncoupled;

F_C12_red = interp1(timearray, Fredarray, t);
CS_red_ratio = interp1(CS_timearray, CS_array,t);
R =  interp1(R_timearray, R_array,t);
F_pyr = interp1(timearray, Fpyarray, t);

ocean3432 = c(S_34)./c(S_32);
ocean32 = c(S_32)./(c(S_34)+c(S_32));
ocean34 = c(S_34)./(c(S_34)+c(S_32));
ocean1312 = c(C_13)./c(C_12);
delSocean = (ocean3432./stdratio_S-1).*1000;
delCocean = (ocean1312./stdratio_C-1).*1000;

F_34_in = ((del_in_S./1000)+1)*stdratio_S.*F_32_in;
%eps_S_red = ((del_in_S - delS_0).*(F_32_in+ F_34_in))./
((F_C12_red_start./CS_red_ratio)+F_C12_red_start./
CS_red_ratio.*(((delS_0-30)./1000)+1).*stdratio_S);
%k_evapnew = (F_32_in + F_34_in - F_C12_red_start./CS_red_ratio -
F_C12_red_start./CS_red_ratio.*(((delS_0+eps_0)./
1000)+1).*stdratio_S)./(Stot_0.*VolOcean);
F_Evap = F_32_in + F_34_in - Fpy_start./CS_0 - Fpy_start./
CS_0.*(((delSocean + eps_ave)./1000)+1).*stdratio_S;

% delC_0 = del_C_foram_start + d13C_offset;                    %
Initial isotope ratio of ocean in permil
% RatioC_0 = ((delC_0./1000)+1)*stdratio_C;  % Initial isotope ratio
in mol_34/mol_32
%
% Ctot_0 = dic.*1029;          % Initial total [DIC] in ocean in
mol/m^3
% c0(C_12) = Ctot_0./(1+RatioC_0);         % Initial C-12 [DIC] of
ocean in mol/m^3
```

```
% c0(C_13) = RatioC_0.*c0(C_12);

F_pyr_mar = R.*F_C12_red./CS_red_ratio;



%%%%%%%%    C13 and C12 Carbon Mass Balance    %%%%%%%%%%%%%%

dcdt(C_12) = (F_12_in - (F_C12_red + k_caco3.*c(C_12).*VolOcean))./
VolOcean;
dcdt(C_13) = (F_13_in - (k_caco3.*c(C_12).*VolOcean.*ocean1312 +
F_C12_red.*...
    (((delCocean+eps_C)./1000)+1).*stdratio_C))./VolOcean;

%%%%%%%%    S32 and S34 Sulfur Mass Balance    %%%%%%%%%%%%%%

% dcdt(S_32) = (F_32_in - (R.*F_C12_red_start./CS_0 +(1-
R).*F_C12_red_start./CS_0 + F_Evap.*ocean32))./VolOcean;
% dcdt(S_34) = (F_34_in - (F_Evap.*ocean34 +...
%     R.*F_C12_red_start./CS_0.*(((delSocean+eps_S_mar)./
1000)+1).*stdratio_S +(1-R).*F_C12_red_start./CS_0).*(((delSocean
+eps_S_ter)./1000)+1).*stdratio_S)./VolOcean;

dcdt(S_32) = (F_32_in - (R.*F_pyr./CS_0 + (1-R).*F_pyr./CS_0 +
F_Evap.*ocean32))./VolOcean;
dcdt(S_34) = (F_34_in - (F_Evap.*ocean34 +...
    R.*F_pyr./CS_0.*(((delSocean+eps_S_mar)./1000)+1).*stdratio_S
+(1-R).*F_pyr./CS_0.*(((delSocean+eps_S_ter)./
1000)+1).*stdratio_S))./VolOcean;



dcdt = dcdt';
```