

```

% этот скрипт собственно и является моделью изотопного состава осадков
% к нему обращается основной скрипт "SIM2023", если там нажать 1

if x == 1
    getfixeddata      % вызываем скрипт, который вытаскивает данные из "InputSIM2023"
elseif x == 2
    disp(0)
else
    disp(0)
end

% ПРОВЕРКА ВХОДНЫХ ПАРАМЕТРОВ
% температура перехода от смешанных к твердым осадкам не должна быть выше,
% чем температура перехода от жидких к смешанным;
% влажность не должна превышать 1

if T_i > T_w
    disp('Проверь параметры модели. T_i должно быть ниже, чем T_w');
    return;
end

if H_s0 > 1
    disp('влажность не должна превышать 1');
    return;
end

% БЛОК 1. ИЗОТОПНЫЙ СОСТАВ ВОДЯНОГО ПАРА В ИСТОЧНИКЕ

% считаем к-ты k* (см. предпоследнюю формулу на стр. 26 у Salamatin et al., 2004):
k18_modif = k_180 + Lambda180*(1-k_180);
kD_modif = k_D + LambdaD*(1-k_D);
k17_modif = k_170 + Lambda170*(1-k_170);

% равновесные к-ты фракц-я для 180, дейтерия и кислорода 17:
alfa18_e = exp(A18_v_l/((teta_s+273.15)^2)-B18_v_l/(teta_s+273.15)-C18_v_l);
alfaD_e = exp(AD_v_l/((teta_s+273.15)^2)-BD_v_l/(teta_s+273.15)+CD_v_l);
alfa17_e = alfa18_e^n;

% кинетические коэффициенты фракционирования при испарении влаги:
alfa18_kin = (1-k18_modif*H_s0)/(1-k18_modif);
alfaD_kin = (1-kD_modif*H_s0)/(1-kD_modif);
alfa17_kin = (1-k17_modif*H_s0)/(1-k17_modif);

% эффективные коэффициенты фракционирования при испарении влаги:
alfa18_ef = alfa18_e*alfa18_kin;
alfaD_ef = alfaD_e*alfaD_kin;
alfa17_ef = alfa17_e*alfa17_kin;

% изотопный состав морской воды в R:
R18_m = (delta18_m+1000)/1000*2005;
RD_m = (deltaD_m+1000)/1000*312;
R17_m = (delta17_m+1000)/1000*380;

```

```
% изотопный состав первичного пара в R:
```

```
R18_s = R18_m/alfa18_ef;
```

```
RD_s = RD_m/alfaD_ef;
```

```
R17_s = R17_m/alfa17_ef;
```

```
% изотопный состав первичного пара в промилле:
```

```
delta18_s = (R18_s/2005-1)*1000;
```

```
lnd180_s = log(1+delta18_s/1000)*1000;
```

```
deltaD_s = (RD_s/312-1)*1000;
```

```
lndD_s = log(1+deltaD_s/1000)*1000;
```

```
dln_s = (lndD_s)-(-0.0285*((lnd180_s)^2) + 8.47*(lnd180_s)); % dln по Uemura 2012
```

```
delta17_s = (R17_s/380-1)*1000;
```

```
dxs_s = deltaD_s-8*delta18_s; % dxs в паре
```

```
% 17O-excess в паре:
```

```
O17excess_s = (log(delta17_s/1000+1) - 0.528*log(delta18_s/1000+1))*1000000;
```

```
% БЛОК 2. РАСПРЕДЕЛЕНИЕ ТЕМПЕРАТУРЫ, ДАВЛЕНИЯ И К-ТОВ ФРАКЦИОНИРОВАНИЯ ВДОЛЬ ТРАЕКТОРИИ
```

```
% разбиваем нашу траекторию на 300 отрезков:
```

```
inc = S/300;
```

```
path = 0:inc:S;
```

```
path=path';
```

```
% считаем высоту воздушной массы, м:
```

```
path(:,2) = E_d*((1-exp(-gam*path(:,1)))/(1-exp(-gam*S)));
```

```
% считаем распределение температуры вдоль траектории, *C:
```

```
deltaT_total = teta_s - T_d;
```

```
beta_E = beta_E/1000;
```

```
deltaT_lat = deltaT_total - beta_E*E_d; % это та часть общего понижения температуры, которая  
% связана с увеличением широты
```

```
lat_grad_T = deltaT_lat/S;
```

```
path(:,3) = teta_s - path(:,2)*beta_E - lat_grad_T*path(:,1);
```

```
% распределение давления (Па) и мольной плотности (моль/м3):
```

```
path(1,5) = p_sl*1000000; % задаем начальное значение давления
```

```
path(1,4) = path(1,5)/(Rg*((path(1,3)+273.15))); % начальное значение мольной плотности
```

```
for i = 2:301
```

```
    dp = path(i-1,4)*-1*grav*Ma;
```

```
% приращение давления на 1 м
```

```
    path(i,5) = path(i-1,5)+dp*(path(i,2)-path(i-1,2));
```

```
% давление на след. шаге расчета
```

```
    path(i,4) = path(i,5)/(Rg*((path(i,3)+273.15)));
```

```
% мольная плотность
```

```
end
```

```
% давление насыщения пара, Па:
```

```
path(:,6) = 610.6*exp(A_w*path(:,3)./(B_w + path(:,3))); % над водой
```

```
path(:,7) = 610.6*exp(A_i*path(:,3)./(B_i + path(:,3))); % и надо льдом
```

```
% равновесные к-ты фракц-я пар-вода для 180 и дейтерия:
```

```
path(:,8) = exp(A18_v_1./((path(:,3)+273.15).^2)-B18_v_1./(path(:,3)+273.15)-C18_v_1);
```

```
path(:,9) = exp(AD_v_1./((path(:,3)+273.15).^2)-BD_v_1./(path(:,3)+273.15)+CD_v_1);
```

```
% к-т фракционирования для переохлажденной жидкости, только D:
```

```
path(:,10) = exp(AD_v_scl./(path(:,3)+273.15).^2 - CD_v_scl);
```

% к-ты фракционирования пар-лёд, кислород 18 и дейтерий:

```
path(:,11) = exp(A18_v_i./((path(:,3)+273.15).^2)-B18_v_i./(path(:,3)+273.15)+C18_v_i);  
path(:,12) = exp(AD_v_i./((path(:,3)+273.15).^2)-BD_v_i./(path(:,3)+273.15)+CD_v_i);
```

### % БЛОК 3. ОТ ПОВЕРХНОСТИ МОРЯ ДО НАСЫЩЕНИЯ

```
ps = path(1,6)*H_s0; % давление водяного пара в начале траектории, Па  
ys = ps/path(1,5); % мольная концентрация пара в начале траектории  
for i = 1:301  
    if ys < path(i,6) / path(i,5)  
        path(i,14) = ys; % пока не достигнуто насыщение, мольная конц-я пара равна начальной  
        k = i;  
    else  
        break  
    end  
end  
  
if ys < path(301,6) / path(301,5) % если влага так и не достигла насыщения  
    disp('Влага в воздушной массе так и не достигла насыщения, осадки не начались')  
    return % и заканчиваем расчет  
end
```

% изот. состав влаги в воздушной массе равен изот. составу начального пара

```
path(k,18) = R18_s;  
path(k,21) = (path(k,18)/2005-1)*1000; % d180  
path(k,19) = RD_s;  
path(k,22) = (path(k,19)/312-1)*1000; % dD  
path(k,20) = R17_s;  
path(k,23) = (path(k,20)/380-1)*1000; % d170
```

### % БЛОК 4. ОТ НАСЫЩЕНИЯ ДО ПЕРВОЙ ПОРЦИИ ЖИДКОЙ ВЛАГИ

% влага начала конденсироваться, но (если  $L_0 > 0$ ) пока не выпадает из облака

```
if L0 > 0 % если L0 задан равным 0, эту секцию просто пропускаем  
    path(k+1,14) = path(k+1,6)/path(k+1,5); % концентрация пара по-прежнему равна насыщенн  
    yw = path(k,14) - path(k+1,14); % первая порция влаги (снижение кол-ва водяного пара)  
    L = yw/path(k+1,14); % относительное содержание влаги в воздухе (yw/yv) в первый момент  
    path(k+1,15) = yw;  
    path(k+1,16) = L;  
    path(k+1,18) = R18_s; % изотопный состав (180) водяного пара  
    path(k+1,21) = (path(k+1,18)/2005 - 1) * 1000; % d180_v  
    path(k+1,20) = R17_s; % то же для 170  
    path(k+1,23) = (path(k+1,20)/380 - 1) * 1000; % d170_v  
    path(k+1,19) = RD_s; % то же для дейтерия  
    path(k+1,22) = (path(k+1,19)/312 - 1) * 1000; % dD_v  
    k1 = k + 1;  
    for i = k+2:301  
        if L < L0 % пока L не достигло L0  
            path(i,14) = path(i,6) / path(i,5); % количество пара всегда равно насыщенному  
            dyw = path(i-1,14)-path(i,14); % лишний пар уходит во влагу  
            yw = yw + dyw; % влаги становится больше  
            path(i,15) = yw; % записываем yw в 15-ю колонку
```

```

        L = yw / path(i,14);      % L также подрастает
        path(i,16) = L;          % записываем L в 16-ю колонку
        k1 = i;
        path(i,18) = R18_s; % теперь считаем изотопный состав для 18O (R18_v)
        path(i,21) = (path(i,18)/2005-1)*1000; % d180 в паре (d180_v)
        path(i,20) = R17_s; % для кислорода 17
        path(i,23) = (path(i,20)/380-1)*1000; % d170 в паре
        path(i,19) = RD_s; % для дейтерия
        path(i,22) = (path(i,19)/312-1)*1000; % dD в паре
    else
        break % если L превысит L0 - останавливаем расчет
    end
end % конец расчета на 301-й строке
else
    k1 = k+1; % если L0 = 0
end

% если модель просчитала траекторию до конца, а L так и не достигло L0,
% т.е. образование осадков так и не началось:
if k1 == 301
    disp('влага достигла насыщения, но L не превысило L0, и осадки не начались');
    return % и заканчиваем расчет
end

```

#### % БЛОК 5. ЖИДКИЕ ОСАДКИ

```

if path(i,3) > T_w % пока температура превышает т-ру перехода к смешанным осадкам

for i = k1:301
    if path(i,3) > T_w % если температура воздуха выше перехода к смешанным осадкам
        path(i,14) = path(i,6)/path(i,5); % концентрация пара по-прежнему равна насыщенной
        path(i,16) = L0; % L = L0
        % относительное изменение кол-ва влаги (dyv/yv):
        dyv_rel = (path(i,14) - path(i-1,14))/path(i-1,14);
        % приращения изотопного состава (отрицательные) - формула (4) у Саламатина:
        path(i,24) = (dyv_rel * (path(i,8) - 1) / (1 + path(i,8)*L0)) * ...
            (1+path(i-1,21)/1000) * 1000; % приращение d180
        path(i,21) = path(i-1,21) + path(i,24); % оно прибавляется к d180_v
        path(i,30) = path(i,8) * (1 + L0) / (1 + path(i,8)*L0) * ...
            (path(i,21)+1000) - 1000; % d180 в осадках (нижняя часть формулы (4))
        path(i,26) = (dyv_rel * (path(i,8)^n - 1) / (1 + (path(i,8)^n)*L0)) * ...
            (1+path(i-1,23)/1000) * 1000; % приращение d170 в паре
        path(i,23) = path(i-1,23) + path(i,26); % оно прибавляется к d170_v
        path(i,32) = (path(i,8)^n) * (1 + L0) / (1 + (path(i,8)^n)*L0) * ...
            (path(i,23)+1000) - 1000; % d170 в осадках
        % для dD расчет разный для т-ры выше 0*С и ниже 0*С:
        if path(i,3) > 0
            path(i,25) = (dyv_rel * (path(i,9) - 1) / (1 + path(i,9)*L0)) * ...
                (1+path(i-1,22)/1000) * 1000; % приращение dD в паре
            path(i,22) = path(i-1,22) + path(i,25); % прибавляется к dD_v
            path(i,31) = path(i,9)*(1 + L0) / (1 + path(i,9)*L0) *...

```

```

        (path(i,22)+1000) - 1000; % dD в осадках
    else % для переохлажденной жидкости
        path(i,25) = (dyv_rel * (path(i,10) - 1) / (1 + path(i,10)*L0)) * ...
            (1+path(i-1,22)/1000) * 1000; % приращение dD в паре
        path(i,22) = path(i-1,22) + path(i,25); % прибавляется к dD_v
        path(i,31) = path(i,10)*(1 + L0) / (1 + path(i,10)*L0) * ...
            (path(i,22)+1000) - 1000; % dD в осадках
    end
    path(k1:301,33) = path(k1:301,31) - 8 * path(k1:301,30); % dxs в осадках
    path(k1:301,35) = (log(path(k1:301,32)/1000+1) - ...
        0.528*log(path(k1:301,30)/1000+1)) * 1000000; % 170-excess в осадках
    % dln по Uemura:
    path(k1:301,42) = log(1+path(k1:301,31)/1000)*1000;
    path(k1:301,43) = log(1+path(k1:301,30)/1000)*1000;
    path(k1:301,34) = path(k1:301,42)-(-0.0285*((path(k1:301,43)).^2) + ...
        8.47*path(k1:301,43));
    k2 = i;
else
    break % когда температура достигает T_w - заканчиваем расчет и
          % переходим к следующему блоку
end
end

else
    k2 = k1; % если т-ра ниже, чем T_w, то модель пропускает жидкие осадки,
            % и сразу переходит к смешанным
end

if k2 == 301 % если в конце траектории так и не случился переход к
            % смешанным осадкам
    disp('в конце траектории выпали жидкие осадки');
end

```

#### % БЛОК 6. СМЕШАННЫЕ ОСАДКИ

```

if k2 < 301

if path(i,3) > T_i % пока температура выше перехода к твердым осадкам

for i = k2+1:301
    if path(i,3) > T_i % если температура выше перехода к ледяным облакам
        % считаем эффективную L (формула 8 у Саламатина):
        L = (L0/(T_w-T_i)) * (path(i,3)-T_i);
        path(i,16) = L;
        % считаем эффективную сигму (та же формула 8):
        path(i,17) = sigma0 + ((1-sigma0)/(T_w-T_i)) * (path(i,3) - T_i);
        % apparent saturation vapor pressure:
        path(i,37) = path(i,17) * path(i,6) + (1 - path(i,17)) * path(i,7);
        path(i,14) = path(i,37) / path(i,5); % концентрация пара равна насыщенной
        % относительное изменение кол-ва влаги (dyv/yv):
        dyv_rel = (path(i,14) - path(i-1,14))/path(i-1,14);
    end
end
end

```

```

% supersaturation ratio Si (формула 5):
Si = 1 + path(i,17) * (path(i,6) / path(i,7) - 1);
path(i,38) = Si; % пишем Si в нашу табличку
% кинетические к-ты фракционирования (формула 6):
alfa_kin18 = 1 / (1 / (path(i,11)*Si) + Dif_180 * (1 - 1 / Si)); % кислород 18
alfa_kinD = 1 / (1 / (path(i,12)*Si) + Dif_D * (1 - 1 / Si)); % дейтерий
alfa_kin17 = (path(i,11))^n * (Si/(1+path(i,11)*(Si-1)*Dif_180))^0.518; % 170
path(i,39) = alfa_kin18; % пишем кинетические коэффициенты в табличку
path(i,40) = alfa_kinD; % то же для дейтерия
path(i,41) = alfa_kin17; % то же для кислорода 17
% всякие промежуточные функции для уравнения (9):
dL_rel = (path(i,16) - path(i-1,16)) / (1 + path(i,16)); % = dL/(1+L)
aux18 = Nu * path(i,8) + (1 - Nu) * alfa_kin18; % Nu*alfa_w+(1-Nu)*alfa_kin
auxD = Nu * path(i,10) + (1 - Nu) * alfa_kinD; % то же для дейтерия
aux17 = Nu * (path(i,8)^n) + (1 - Nu) * alfa_kin17; % то же для кислорода 17
aux18_2 = 1 + L*path(i,8); % еще один вспомогательный для кислорода 18
auxD_2 = 1 + L*path(i,10); % он же для дейтерия
aux17_2 = 1 + L*(path(i,8)^n); % он же для кислорода 17
% приращения значений изотопного состава
path(i,24) = (((alfa_kin18 + L*aux18)/aux18_2-1)*dyv_rel + ...
    dL_rel*((1+L)*aux18/aux18_2-1)) * (path(i-1,21)/1000+1) * 1000; % 180
path(i,21) = path(i-1,21) + path(i,24); % прибавляем эту добавку к d180_v
path(i,25) = (((alfa_kinD + L*auxD)/auxD_2-1)*dyv_rel + ...
    dL_rel*((1+L)*auxD/auxD_2-1)) * (path(i-1,22)/1000+1) * 1000; % приращение dD_v
path(i,22) = path(i-1,22) + path(i,25); % прибавляем эту добавку к dD_v
path(i,26) = (((alfa_kin17 + L*aux17)/aux17_2-1)*dyv_rel + ...
    dL_rel*((1+L)*aux17/aux17_2-1)) * (path(i-1,23)/1000+1) * 1000; % приращение d170_v
path(i,23) = path(i-1,23) + path(i,26); % прибавляем эту добавку к d170_v
% !!!!! ниже ПРИБЛИЗИТЕЛЬНЫЙ пересчет изотопного состава осадков, НЕ как у Саламатина
path(i,30) = ((path(i,8)*(1+L)/(1+path(i,8)*L))*(path(i,3)-T_i)/(T_w-T_i)...
    + alfa_kin18*(T_w-path(i,3))/(T_w-T_i))*(path(i,21)+1000) - 1000; % d180 в осадках
path(i,31) = ((path(i,10)*(1+L)/(1+path(i,10)*L))*(path(i,3)-T_i)/(T_w-T_i)...
    + alfa_kinD*(T_w-path(i,3))/(T_w-T_i))*(path(i,22)+1000) - 1000; % dD в осадках
path(i,32) = (((path(i,8)^n)*(1+L)/(1+(path(i,8)^n)*L))*(path(i,3)-T_i)/(T_w-T_i)...
    + alfa_kin17*(T_w-path(i,3))/(T_w-T_i))*(path(i,23)+1000) - 1000; % d170 в осадках
k3 = i;
path(k2:301,33) = path(k2:301,31) - 8 * path(k2:301,30); % dxs в осадках
% dln по Uemura:
path(k2:301,42) = log(1+path(k2:301,31)/1000)*1000;
path(k2:301,43) = log(1+path(k2:301,30)/1000)*1000;
path(k2:301,34) = path(k2:301,42)-(-0.0285*((path(k2:301,43)).^2)...
    + 8.47*path(k2:301,43)); % dln Uemura
path(k2:301,35) = (log(path(k2:301,32)/1000+1) - ...
    0.528*log(path(k2:301,30)/1000+1)) * 1000000; % 170-excess в осадках
if k3 == 301
    disp('в конце траектории выпали смешанные осадки');
end
else
    break % когда т-ра достигает T_i, переходим к расчету ледяных осадков
end
end
else

```

```

    k3 = k2;    % если температура ниже T_i, то этот блок пропускаем и переходим
                % к ледяным осадкам
end

else % если k2 = 301, пропускаем этот блок
    k3 = k2;
end

```

#### % БЛОК 7. ЛЕДЯНЫЕ ОСАДКИ

```

if k3 < 301

for i = k3+1:301
    % apparent saturation vapor pressure:
    path(i,37) = sigma0 * path(i,6) + (1 - sigma0) * path(i,7);
    path(i,14) = path(i,37) / path(i,5);    % концентрация пара равна насыщенной
    % относительное изменение кол-ва влаги (dyv/yv)
    dyv_rel = (path(i,14) - path(i-1,14))/path(i-1,14);
    Si = 1 + sigma0 * (path(i,6) / path(i,7) - 1);    % supersaturation ratio Si (формула 5)
    path(i,38) = Si;    % пишем Si в нашу табличку
    % кинетические к-ты фракционирования (формула 6):
    alfa_kin18 = 1 / (1 / (path(i,11)*Si) + Dif_180 * (1 - 1 / Si));    % кислород 18
    alfa_kinD = 1 / (1 / (path(i,12)*Si) + Dif_D * (1 - 1 / Si));    % дейтерий
    alfa_kin17 = (path(i,11))^n * (Si/(1+path(i,11)*(Si-1)*Dif_180))^0.518;    % 17O
    path(i,39) = alfa_kin18;    % пишем кинетические коэффициенты в табличку
    path(i,40) = alfa_kinD;    % то же для дейтерия
    path(i,41) = alfa_kin17;    % то же для кислорода 17
    path(i,24) = dyv_rel * (alfa_kin18 - 1) * (path(i-1,21)/1000+1) ...
        * 1000; % приращение d180_v
    path(i,21) = path(i-1,21) + path(i,24);    % приращение добавляется к d180_v
    path(i,25) = dyv_rel * (alfa_kinD - 1) * (path(i-1,22)/1000+1)...
        * 1000; % приращение dD_v
    path(i,22) = path(i-1,22) + path(i,25);    % приращение добавляется к dD_v
    path(i,26) = dyv_rel * (alfa_kin17 - 1) * ...
        (path(i-1,23)/1000+1) * 1000; % приращение d170_v
    path(i,23) = path(i-1,23) + path(i,26);    % приращение добавляется к d170_v
    path(i,30) = alfa_kin18 * (path(i,21)+1000) - 1000;    % d180 в осадках
    path(i,31) = alfa_kinD * (path(i,22)+1000) - 1000;    % dD в осадках
    path(i,32) = alfa_kin17 * (path(i,23)+1000) - 1000;    % d170 в осадках

end

else
end

path(k3:301,33) = path(k3:301,31) - 8 * path(k3:301,30); % dxs в осадках
% dln по Uemura:
path(k3:301,42) = log(1+path(k3:301,31)/1000)*1000;
path(k3:301,43) = log(1+path(k3:301,30)/1000)*1000;
path(k3:301,34) = path(k3:301,42) - (-0.0285*((path(k3:301,43)).^2) ...

```



```

+ 8.47*path(k3:301,43)); % dln Uemura
path(k3:301,35) = (log(path(k3:301,32)/1000+1) - ...
0.528*log(path(k3:301,30)/1000+1)) * 1000000; % 170-excess в осадках

```

#### % БЛОК 8. СТРОИМ ДИАГРАММЫ И ЭКСПОРТИРУЕМ ДАННЫЕ

```

SIMresults = table(path(:,1), path(:,2),path(:,3),path(:,5),path(:,21),path(:,22),path(:,23),...
path(:,30),path(:,31),path(:,32),path(:,33),path(:,34),path(:,35),'VariableNames',...
{'Расстояние, км','Высота над у.м., м','Температура, *C','Давление, Па',[ 'd180 в паре, ' ...
'промилле'],'dD в паре, промилле','d170 в паре, промилле','d180 в осадках, промилле',...
'dD в осадках, промилле', 'd170 в осадках, промилле','dxs в осадках, промилле', ...
'dln в осадках, промилле','170-excess в осадках, per meg'});
writetable(SIMresults,"SIMresults.xlsx");
disp('результаты расчетов в файле "SIMresults.xlsx"');

```

результаты расчетов в файле "SIMresults.xlsx"

```

subplot(2,2,1);
plot(path(k1:301,3),path(k1:301,30),"k");
title('Зависимость d180 от температуры');
xlabel('Температура, *C');
ylabel('d180, промилле');

subplot(2,2,2);
plot(path(k1:301,30),path(k1:301,31));
title('MWL');
xlabel('d180, промилле');
ylabel('dD, промилле');

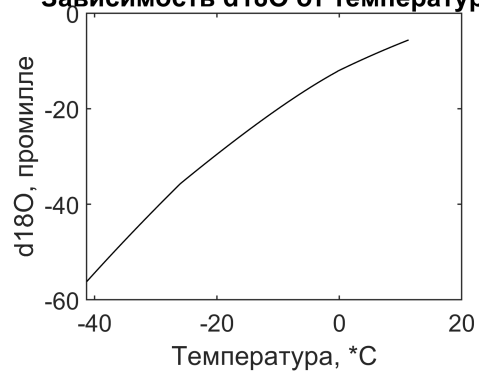
subplot(2,2,3);
plot(path(k1:301,30),path(k1:301,33),"r");
hold on
plot(path(k1:301,30),path(k1:301,34),"b");
hold off
title('Зависимость dxs и dln от d180');
xlabel('d180, промилле');
ylabel('dxs, промилле');

subplot(2,2,4);
plot(path(k1:301,30),path(k1:301,35),"m");
title('Зависимость 170-excess от d180');
xlabel('d180, промилле');
ylabel('170-excess, per meg');

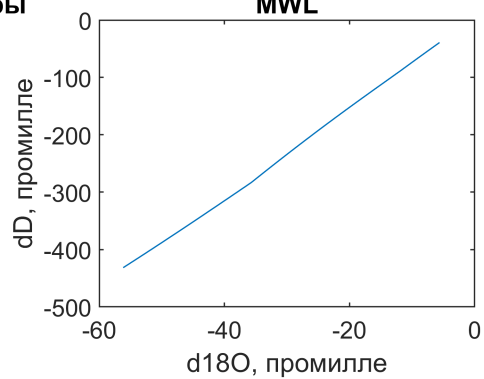
```



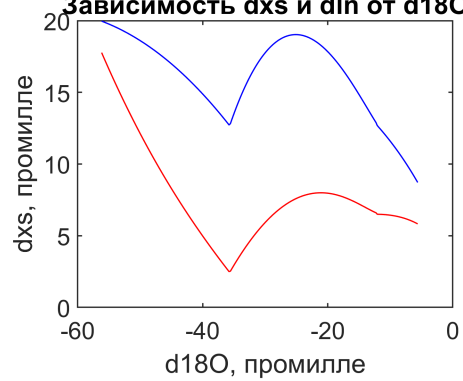
**Зависимость d18O от температуры**



**MWL**



**Зависимость dxs и dln от d18O**



**Зависимость 17O-excess от d18O**

