# Table of Contents



```
function SLDvfADJ_twostep_averagedactualdata_Oct16_startingcomp2

close all
for po=2:2
    outer(po)
end
disp('change composition values')

    function outer(alloytype)
```

# function to determine volume fraction change caused by a changing matrix composition

uses the actual averaged volume fraction values in: VF and alloy type out: adjusted VF How? VFin adjusts SLD of matrix, delSLD recalculated, VFout calculated with new SLD 3 iterations load('9922 st','st')

```
load('9922 st loose fit1st4','st')
disp('................start..................')
%some initialises
time=st(4).time;

time2=st(alloytype).time;
xlaves=st(4).lavesVF;


%make vf of beta a constant
meanVF=mean(st(4).betaVF');
for n=1:3
    xbeta(:,n)=meanVF(n).*ones(length(time),1);
end

%if we want beta to increase gradually
gradual_beta =0;
if gradual_beta==1
    xbeta(1,1)=xbeta(1,1)*0.5;
    xbeta(2,1)=xbeta(2,1)*0.75;
    xbeta(1,2)=xbeta(1,2)*0.75;
    xbeta(2,2)=xbeta(2,2)*0.95;
```

```
    xbeta(1,3)=xbeta(1,3);
end
%get the vf values from SANS using adjustment for ally type
xbeta=xbeta/st(alloytype).betaVFadj;
xlaves=xlaves'/st(alloytype).lavesVFadj;

xbetaold=xbeta;
xlavesold=xlaves;

%if want to plot the res
plotq=1;
```

# computing bit- given volume fraction calculates new volume fraction and SLD

```
[xbeta,xlaves,
 delSLD]=innerinner(xbeta,xlaves,xbetaold,xlavesold,time,alloytype,plotq);
[xbeta,xlaves,
 delSLD]=innerinner(xbeta,xlaves,xbetaold,xlavesold,time,alloytype,plotq);
[xbeta,xlaves, delSLD, SLD_matrixSLD,
 mATMpc ]=innerinner(xbeta,xlaves,xbetaold,xlavesold,time,alloytype,plotq);

switch alloytype
    case 1
save('lowAl vfadj
 averagedact2','xbeta','xlaves','xlavesold','xbetaold','delSLD','time','time2','st
    case 2
save('9922 vfadj
 averagedact2','xbeta','xlaves','xlavesold','xbetaold','delSLD','time','time2','st
    case 3
save('F1E vfadj
 averagedact2','xbeta','xlaves','xlavesold','xbetaold','delSLD','time','time2','st
end
```

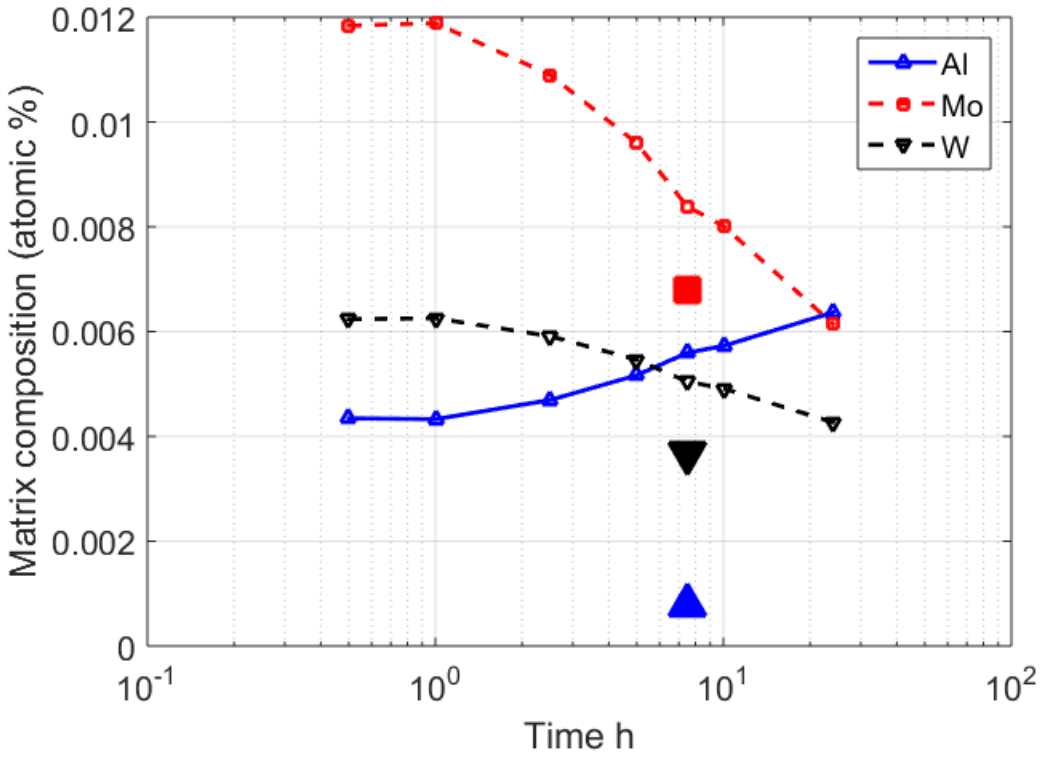
# plot the SANS composition vs the APT one

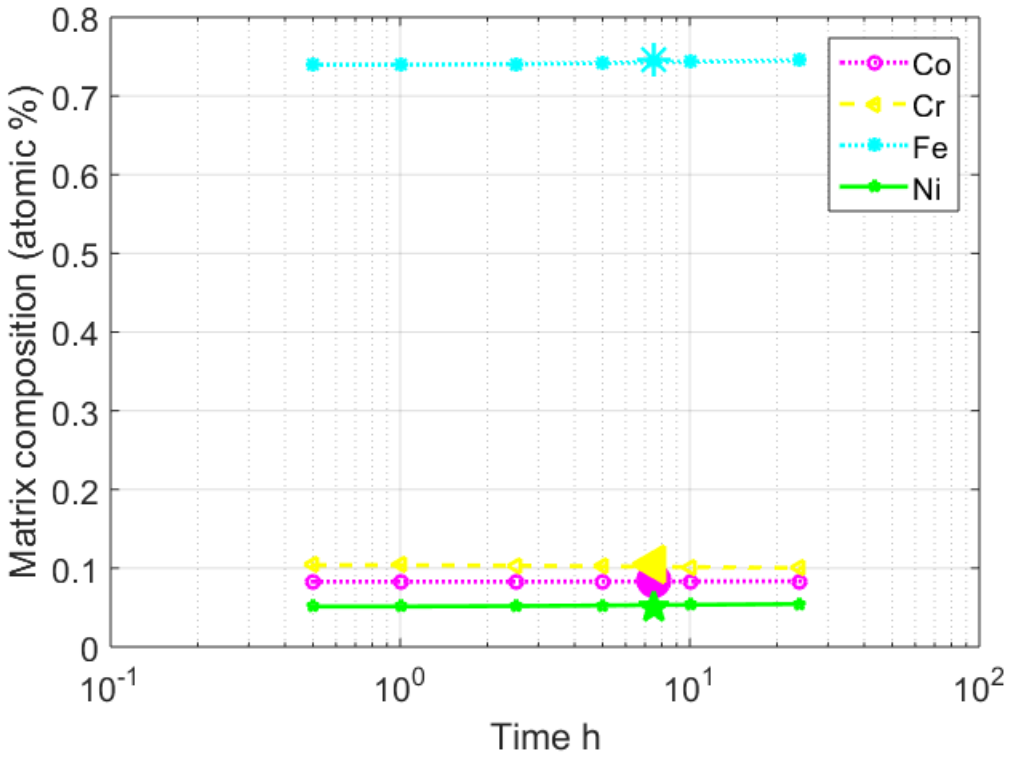
```
matcomp_meas_7p5 = [...
    5.90e-04, 0.084028981, 0.11121341, 0.747072779, ...
    0.007363787, 0.04359324, 0.003835351;...

    7.82e-04, 8.48e-02, 1.04e-01, ...
    7.47e-01, 6.79e-03, 4.97e-02, 3.67e-03; ...

    2.072e-03, 8.224e-02, 1.166e-01,   ...
    7.481e-01, 1.045e-02, 3.518e-02, 4.560e-03    ];

col={[0 0 1];[1 0 1]; [1 1 0]; [0 1 1]; [1 0 0];[0 1 0];[0 0 0]};
colt={'^-';'o:';'<--';'*:';'s--';'p-';'v--'}

switch alloytype
    case 1%lowAl
%         eleme={'Al'; 'Co'; 'Cr';
```

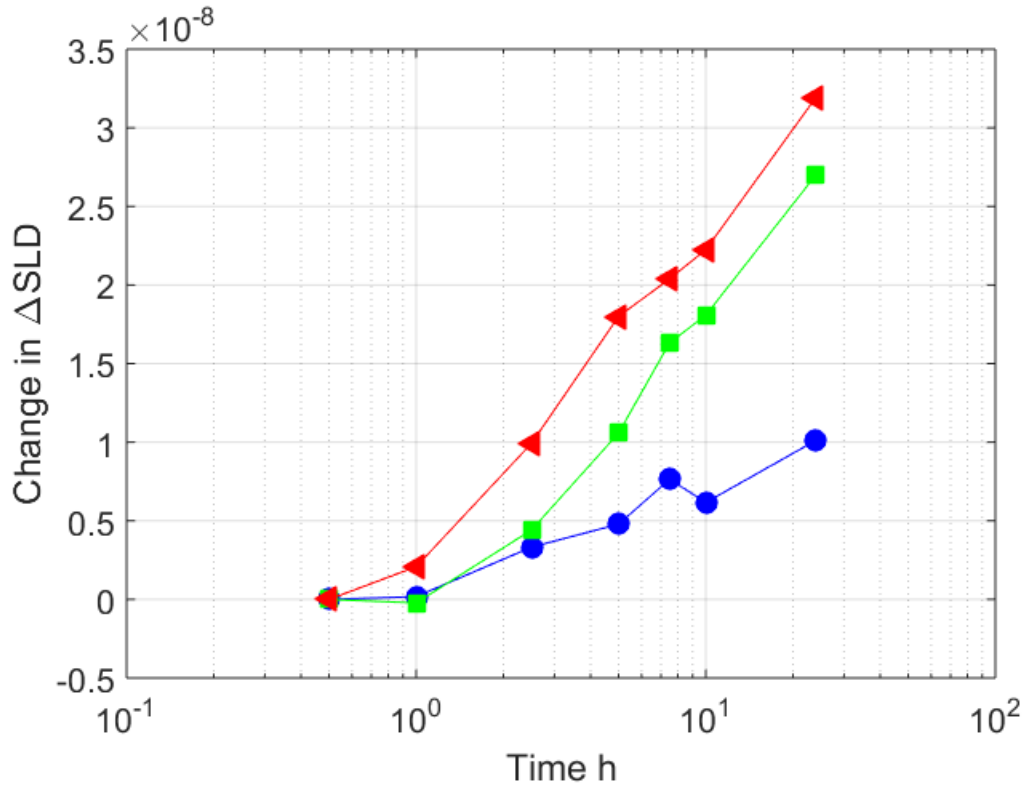
```
%               'Fe'; 'Mo'; 'Ni';'W'};
ATMpc(1)=2.56/100;ATMpc(2)=7.96/100;ATMpc(3)=10.91/100;
ATMpc(4)=69.35/100;ATMpc(5)=1.62/100;ATMpc(6)=6.85/100;ATMpc(7)=0.75/100;
    case 2%9922
ATMpc(1)=3.58/100;ATMpc(2)=7.91/100;ATMpc(3)=9.72/100;
ATMpc(4)=68.31/100;ATMpc(5)=1.17/100;ATMpc(6)=8.71/100;ATMpc(7)=0.60/100;
    case 3%F1E
ATMpc(1)=3.66/100;ATMpc(2)=7.95/100;ATMpc(3)=10.74/100;
ATMpc(4)=68.49/100;ATMpc(5)=1.62/100;ATMpc(6)=6.80/100;ATMpc(7)=0.75/100;
end

mATMpc_(:,:,1) = [ATMpc ; mATMpc(:,:,1)];
mATMpc_(:,:,2) = [ATMpc ; mATMpc(:,:,2)];
mATMpc_(:,:,3) = [ATMpc ; mATMpc(:,:,3)];

mATMpc = mATMpc_;

whchplot = [0 1 1 2 0 1 0];
for gg=1:7
    if whchplot(gg)==0
        figure(26)
    semilogx([0 0.5 1 2.5 5 7.5 10 24 ],...
    mATMpc(:,gg,2),colt{gg},'linewidth',2,'color',col{gg}),hold on
    else
         figure(27)
    semilogx([0 0.5 1 2.5 5 7.5 10 24 ],...
    mATMpc(:,gg,2),colt{gg},'linewidth',2,'color',col{gg}),hold on
    end
end

for gg=1:7
    if whchplot(gg)==0
        figure(26)
    semilogx( 7.5 ,matcomp_meas_7p5(alloytype,gg),colt{gg},...

  'linewidth',2,'color',col{gg},'markersize',15,'markerfacecolor',col{gg})
    hold on
    else
         figure(27)
     semilogx( 7.5 ,matcomp_meas_7p5(alloytype,gg),colt{gg},...

  'linewidth',2,'color',col{gg},'markersize',15,'markerfacecolor',col{gg})
    hold on
    end
end
h=figure(26)
lega = {'Al'; 'Co'; 'Cr';'Fe'; 'Mo'; 'Ni';'W'};
legend(lega(whchplot==0))
grid
xlabel('Time h')
ylabel('Matrix composition (atomic %)')
set(gca,'fontsize',16)
set(h,'position',[150 50 700 500])
```

```
h=figure(27)
legend(lega(whchplot~=0))
grid
xlabel('Time h')
ylabel('Matrix composition (atomic %)')
set(gca,'fontsize',16)
set(h,'position',[150 50 700 500])

disp({'Al'; 'Co'; 'Cr';'Fe'; 'Mo'; 'Ni';'W'}')
APTmeasured = matcomp_meas_7p5(alloytype,:)*100
SANSpredicted =  mATMpc(5,:)*100

col={'ob';'sg';'<r'};
h = figure
for n=1:3
semilogx(time, delSLD(:,1,n)-delSLD(1,1,n),'marker',col{n}
(1),'color',col{n}(2),'markerfacecolor',col{n}(2),'markersize',10)
hold on
end
grid
xlabel('Time h')
ylabel('Change in \DeltaSLD')
set(gca,'fontsize',16)
set(h,'position',[150 50 700 500])


colt =

    '^-'
    'o:'
    '<--'
    '*:'
    's--'
    'p-'
    'v--'


h =

  Figure (26) with properties:

      Number: 26
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [360 198 560 420]
       Units: 'pixels'

  Use GET to show all properties


h =

  Figure (27) with properties:
```

```
      Number: 27
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [360 198 560 420]
       Units: 'pixels'

  Use GET to show all properties

    'Al'    'Co'    'Cr'    'Fe'    'Mo'    'Ni'    'W'


APTmeasured =

    0.0782    8.4800   10.4000   74.7000    0.6790    4.9700    0.3670


SANSpredicted =

  Columns 1 through 7

    0.3182    8.2994   10.3675   74.2912    1.1149    5.0079    0.6008

  Columns 8 through 14

    0.5167    8.2990   10.2394   74.1838    0.9596    5.2553    0.5462

  Columns 15 through 21

    0.6740    8.3111   10.0901   74.2370    0.7510    5.4632    0.4736


h =

  Figure (2) with properties:

      Number: 2
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [360 198 560 420]
       Units: 'pixels'

  Use GET to show all properties
```


0.012
0.01
0.008
0.006
0.004
0.002
0
Matrix composition (atomic %)
$10^{-1}$
$10^{0}$
$10^{1}$
$10^{2}$
Time h
Al
Mo
W


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Matrix composition (atomic %)
$10^{-1}$
$10^{0}$
$10^{1}$
$10^{2}$
Time h
Co
Cr
Fe
Ni




```
    end

.................start.................
change composition values

    function [xbeta,xlaves, delSLD, SLD_matrixdelSLD,
 mATMpc ]=innerinner(xbeta,xlaves,xbetaold,xlavesold,time,alloytype,plotq)
```

# run cal for each time step

```
        for ji=1:3
        for n=1:length(time)

    [delSLD(n,:,ji),
 SLD_matrixdelSLD(n,ji),mATMpc(n,:,ji) ]=inner(xbeta(n,ji),xlaves(n,ji),alloytype)
 2=9922, 3=F1E

        end

        end
```

# Outputs

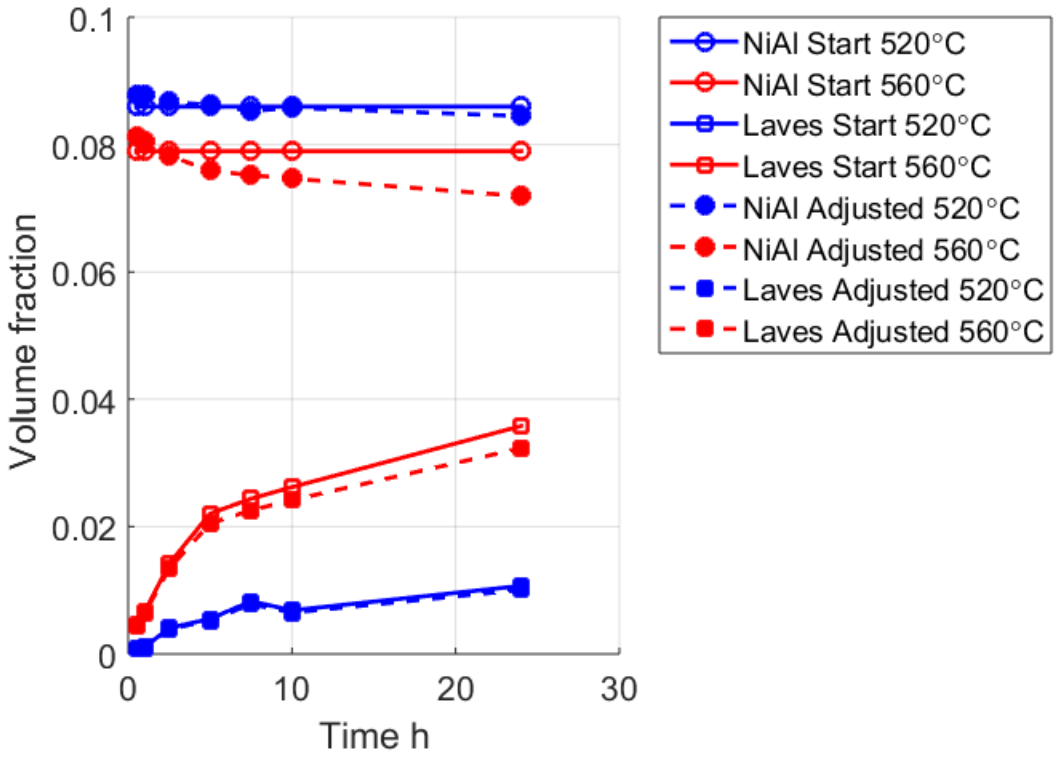
```
if plotq==1

h=figure(1);
```

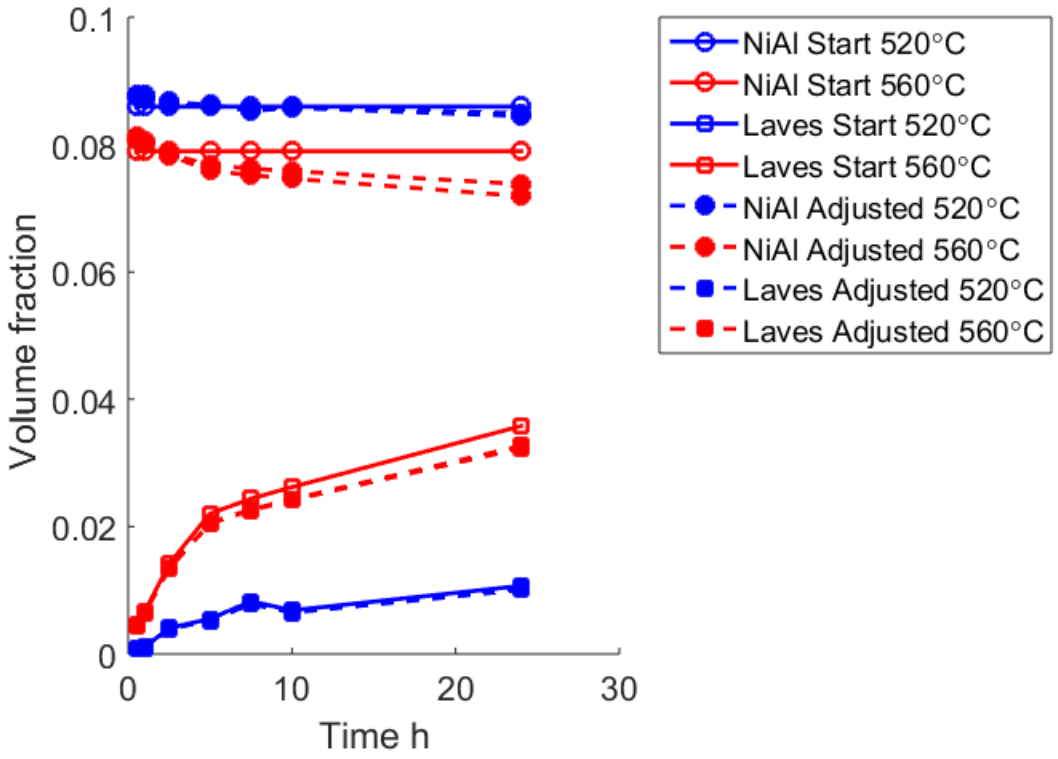
```
end

[delSLDstart]=inner(0.08,.04,alloytype);

for ji=1:3
vfadj1(:,ji)=(delSLDstart(1)./delSLD(:,1,ji)).^2;
vfadj2(:,ji)=(delSLDstart(2)./delSLD(:,2,ji)).^2;
end

wanttocomparewithassstart=0;
if wanttocomparewithassstart==1
vfadj1=(delSLD(1,1)./delSLD(:,1)).^2;
vfadj2=(delSLD(1,2)./delSLD(:,2)).^2;
end
```

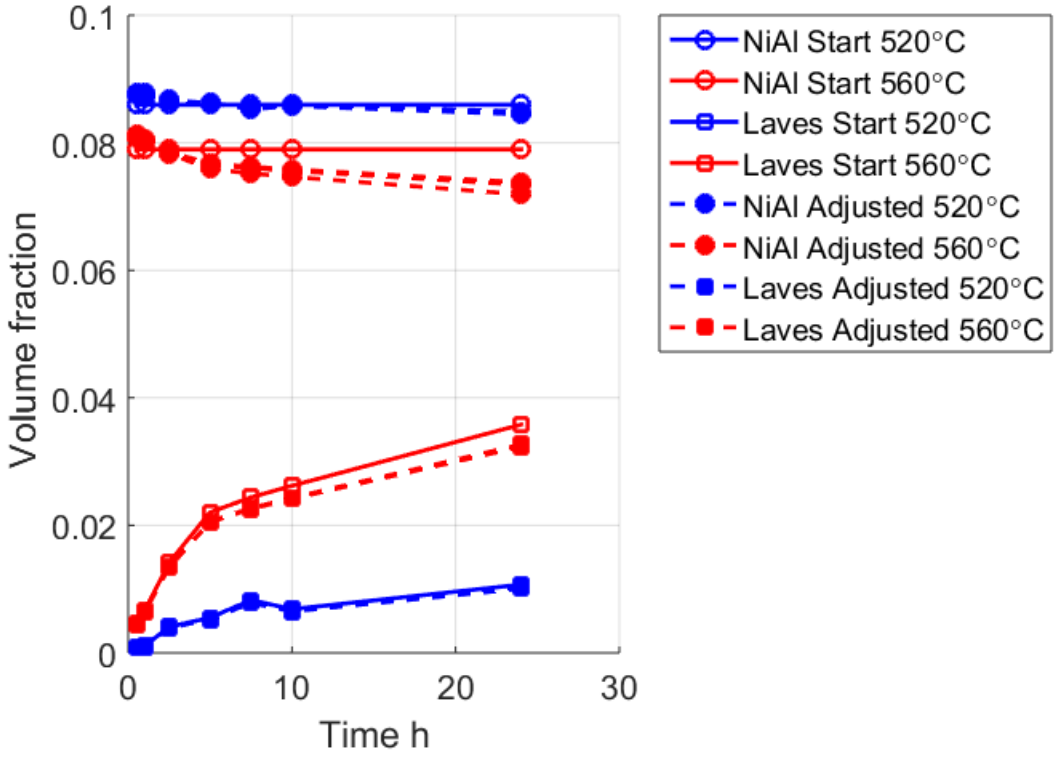
for m=1:3

```
    vbeta=vfadj1.*xbetaold(:,:);
    vlaves=vfadj2.*xlavesold(:,:);
% end

vmatrix=ones(size(vbeta));
vtot=vbeta+vlaves+vmatrix;
for n=1:length(vbeta);
    vbeta(n,:)=vbeta(n,:)./vtot(n,:);
    vlaves(n,:)=vlaves(n,:)./vtot(n,:);
end


if plotq==1
    hold on
plot(time,xbetaold(:,[1]),'b-o','linewidth',2,'markersize',8),
plot(time,xbetaold(:,[3]),'r-o','linewidth',2,'markersize',8),
plot(time,xlavesold(:,[1 ]),'b-s','linewidth',2,'markersize',8)
plot(time,xlavesold(:,[3 ]),'r-s','linewidth',2,'markersize',8)
semilogx(time,vbeta(:,[1]),'b--
o','linewidth',2,'markersize',8,'markerfacecolor','b'),
plot(time,vbeta(:,[3]),'r--
o','linewidth',2,'markersize',8,'markerfacecolor','r'),

plot(time,vlaves(:,[1]),'b--
s','linewidth',2,'markersize',8,'markerfacecolor','b')
plot(time,vlaves(:,[3]),'r--
s','linewidth',2,'markersize',8,'markerfacecolor','r')


grid
xlabel('Time h')
ylabel('Volume fraction ')
set(gca,'fontsize',16)
set(h,'position',[150 50 700 500])
legend('NiAl Start 520\circC','NiAl Start 560\circC',...
 'Laves Start 520\circC','Laves Start 560\circC',...
 'NiAl Adjusted 520\circC','NiAl Adjusted 560\circC',...
```

```
'Laves Adjusted 520\circC','Laves Adjusted
 560\circC','Location','northeastoutside')
end
xbeta=vbeta;
xlaves=vlaves;
delSLDold=delSLD;
```




0.1
0.08
0.06
0.04
0.02
0
Volume fraction
0
10
20
30
Time h
NiAl Start 520°C
NiAl Start 560°C
Laves Start 520°C
Laves Start 560°C
NiAl Adjusted 520°C
NiAl Adjusted 560°C
Laves Adjusted 520°C
Laves Adjusted 560°C


0.1
0.08
0.06
0.04
0.02
0
Volume fraction
0
10
20
30
Time h
NiAl Start 520°C
NiAl Start 560°C
Laves Start 520°C
Laves Start 560°C
NiAl Adjusted 520°C
NiAl Adjusted 560°C
Laves Adjusted 520°C
Laves Adjusted 560°C

```
end
```

# calculating bit

```
function [delSLD ,SLD_matrix, mATMpc]=inner(vf1,vf2,alloy)

    % get precip atm % and SLD
[p1ATMpc, pcell_det]=precip_ATMpc(1);%1= NiAl
SLD_p1=SLDcalc(p1ATMpc,pcell_det);
[p2ATMpc ,pcell_det]=precip_ATMpc(2);%2=Laves
SLD_p2=SLDcalc(p2ATMpc,pcell_det);

    % get matrix atm% and SLD requires the precipitate compositions
[mATMpc ,cell_det]=matrix_ATMpc(vf1,vf2,p1ATMpc,p2ATMpc,alloy);
SLD_matrix=SLDcalc(mATMpc,cell_det);

    % calculate deltaSLD
delSLD=[abs( SLD_matrix - SLD_p1 ),abs( SLD_matrix - SLD_p2 )];

end

function [ATMpc_,
 cell_det]=matrix_ATMpc(vf1,vf2,p1ATMpc,p2ATMpc,alloy)

switch alloy
    case 1%lowAl
%         eleme={'Al'; 'Co'; 'Cr';'Fe'; 'Mo'; 'Ni';'W'};

ATMpc(1)=2.56/100;ATMpc(2)=7.96/100;ATMpc(3)=10.91/100;
ATMpc(4)=69.35/100;ATMpc(5)=1.62/100;ATMpc(6)=6.85/100;ATMpc(7)=0.75/100;
    case 2%9922
ATMpc(1)=3.58/100;ATMpc(2)=7.91/100;ATMpc(3)=9.72/100;
ATMpc(4)=68.31/100;ATMpc(5)=1.17/100;ATMpc(6)=8.71/100;ATMpc(7)=0.60/100;
    case 3%F1E
ATMpc(1)=3.66/100;ATMpc(2)=7.95/100;ATMpc(3)=10.74/100;
ATMpc(4)=68.49/100;ATMpc(5)=1.62/100;ATMpc(6)=6.80/100;ATMpc(7)=0.75/100;
end
cell_det(1)=2.875^3*1e-24;%vol unit cell
cell_det(2)=2;%atoms/cell

vfM=1-vf1-vf2;
ATMpc_ = (ATMpc - p1ATMpc*vf1 - p2ATMpc*vf2)/vfM ;


for n=1:length(ATMpc)
    if ATMpc_(n)<0
        eleme={'Al'; 'Co'; 'Cr';'Fe'; 'Mo'; 'Ni';'W'};

        disp(['laves error <0% ', eleme{n},' vf: ',num2str(vf2),'
 ',num2str(100*ATMpc_)])
        ATMpc_(n)=0;
    end
end
ATMpc_=ATMpc_/sum(ATMpc_);

end
```

```
function [ATMpc, cell_det]=precip_ATMpc(Ni_lav)
```

# given precipitate type gets atomic % and unit cell

```
if Ni_lav==1%element order eleme={'1)Al, 2)Co 3)Cr 4)Fe 5)Mo 6)Ni
 7)W'}
%NiAl
ATMpc(1)=38/100;ATMpc(2)=4.07/100;ATMpc(3)=2.28/100;
ATMpc(4)=8.0/100;ATMpc(5)=0.49/100;ATMpc(6)=48/100;ATMpc(7)=0.16/100;

cell_det(1)=2.875^3*1e-24;%vol unit cell
cell_det(2)=2;%atoms/cell
else
%laves      1     2     3   4       5   6   7
% legend({'Al'; 'Co'; 'Cr';'Fe'; 'Mo'; 'Ni';'W'})
ATMpc(1)=2.7/100;ATMpc(2)=4.95/100;ATMpc(3)=21.6/100;
ATMpc(4)=36.05/100;ATMpc(5)=22.1/100;ATMpc(6)=4.8/100;
ATMpc(7)=7.8/100;
cell_det(1)=1.589e-22;%vol unit cell
cell_det(2)=12;%atoms/cell
end

end

function SLD=SLDcalc(ATMpc,cell_det)
```

# given atomic % and cell get SLD

element order eleme={'1)Al, 2)Co 3)Cr 4)Fe 5)Mo 6)Ni 7)W'}

```
Vcell=cell_det(1);
SLD_element=[34.49 24.9 36.35 94.5 67.15 103 47.55];
% mass_atom=[4.51298E-26 9.8573E-26 8.69698E-26 9.34076E-26
 1.60471E-25 9.81719E-26 3.07495E-25];
SLDi=cell_det(2)*sum(SLD_element.*ATMpc);%sum of sld elements
% masscell=sum(cell_det(2)*mass_atom.*ATMpc);%mass of 1 cell
% dens = 1e3*masscell/cell_det(1);
SLD=1e-30*SLDi/Vcell;

end

end
```

*Published with MATLAB® R2015a*