Electronic Supplementary Material (ESI) for Journal of Materials Chemistry A.

Script of the Algorithm for the Calculation of the Shear Modulus

The following Mathematica®-notebook (adapted according to [20]) shows the algorithm for the calculation of the shear modulus. Comments are described by „(* Comment *)" and directions are ended by a semicolon („**;**") if their output is suppressed. Built-in Mathematica-commands have the form „Befehl[argument,...]". The symbol („*") is neglected as every space between variables is interpreted as multiplication symbol.

The solution of the algorithm includes the following main steps:

- Definition of global constants
- Import of measured data
- Exclusion from data outliers
- Determination of quartz parameter with measurement data of the unloaded quartz at argon
- Determination of electrolyte parameter with measurement data of the loaded quartz in electrolyte
- Determination of the shear modulus for the polymerization and cycling of PEDOT in monomer-free ionic liquid

```
(* start from scratch *)
Remove["Global`*"];
(* CONSTANTS *)
ρq=2.648; (* g/cm³ *)
μq= 2.957*10^11; (* g/cms² *)
Zq=Sqrt[ρq μq]; (* g/cm²s *)
K2=0.00774; (* 1 *)
L1=8.898659*10^-3; (* Vs/A = H *)
A=0.22135;(* cm² *)
F=96485.3; (*As/mol *)

(* you may want to change the following parameters *)
ηl=0.16; (* g/cms; Acetonitrile=0.00306, Water=0.01, [EMIm]AlCl4=0.16(16mPas)  *)
```

```
ρl=1.2985; (* g/cm³; Acetonitrile=0.7845, Water=1, [EMIm]AlCl4=1.2985 *)
ρf=1.334; (* PEDOT=1.334 g/cm³ *)

(* DATA FILES *)
wav=Import["C:\\WINDOWS\\Media\\notify.wav"];
SetDirectory["C:\\Users\\ts2n15\\Dropbox\\Promotion\\Experimental\\Measurements\\Ye
ar_2\\EQCM\\041918"];
fileIn0="0_Cal1.dat"; (* AIR DATA *)
fileIn1="1_Pol.dat"; (* LIQ/DEP DATA *)
fileIn2="2_Cycl1.dat"; (* CHAR. DATA *)
(* fileIn3 and fileIn4 are defined online *)

(* YOU MAY NEED TO CHANGE THE DEP START CRITERIA AND b-FACTOR RANGE IN THE
SECTIONS BELOW !!! *)
dims=-1 (* output dimensions for G: 0⍰dyn/cm² (calc.),-1⍰Pa, -5⍰N/cm^2 *);

If[dims===0, Gs="G' / dyn/cm²";Gl="G\" / dyn/cm²"];
If[dims===-1,Gs="G' / Pa";Gl="G\" / Pa"];
If[dims===-5,Gs="G' / N/cm²";Gl="G\" / N/cm²"];
(* 0. AIR DATA CALCULATIONS *)
dataIn0=Import[fileIn0];
(* get first data row *)
j=1; While[j<= Length[dataIn0],
isComment=StringMatchQ[dataIn0[[j,1]],"time"];
j++;
If[isComment===True,Break[];]; ]

(* dataAir = { t/s, f/Hz, w/Hz Subscript[}, n] *)
(* ARRAY *) datat0=Table[dataIn0[[i,1]],{i,j,Length[dataIn0]}]
(* ARRAY *) dataf0=Table[dataIn0[[i,5]],{i,j,Length[dataIn0]}]
(* ARRAY *) dataw0=Table[dataIn0[[i,6]],{i,j,Length[dataIn0]}]
(* check for outliers *)
```

```
fInt=SetPrecision[Sort[{Median[dataf0]-
5MedianDeviation[dataf0],Median[dataf0]+5MedianDeviation[dataf0]}],10];
wInt=Sort[{Median[dataw0]-
5MedianDeviation[dataw0],Median[dataw0]+5MedianDeviation[dataw0]}];
For[i=1,i<=Length[datat0],i++,
If[fInt[[1]]<=dataf0[[i]]<=fInt[[2]] || wInt[[1]]<=dataw0[[i]]<=wInt[[2]],(*  do nothing *),
datat0=Delete[datat0,i];
dataf0=Delete[dataf0,i];
dataw0=Delete[dataw0,i];
i--; ]; ];

(* SKALAR *) f0=SetPrecision[Mean[dataf0],10];
(* SKALAR *) w0=Mean[dataw0];
{f0,w0}
(* SKALAR *) R1=2  π L1 w0 (* sH = sV/sA = Ω *)

(* GRAPHS *)
gfx01=ListPlot[Partition[Riffle[datat0,dataf0-dataf0[[1]]],2], PlotStyle->Blue,PlotRange->All];
gfx02=ListPlot[Partition[Riffle[datat0,dataw0-dataw0[[1]]],2], PlotStyle->Red];
Show[gfx01,gfx02, PlotRange->All, ImageSize->Small]

(* some clean up *)
ClearAll[j,i, isComment];
Remove[dataIn0,dataAir,"gfx0@"];

(* 1.a DEP DATA - LIQUID *)
dataIn1=Import[fileIn1];
(* get first data row *)
j=1; While[j<= Length[dataIn1],
isComment=StringMatchQ[dataIn1[[j,1]],"time"];
j++;
If[isComment===True,Break[];]; ]
```

```
(* ARRAY *) datat1=Table[dataIn1[[i,1]],{i,j,Length[dataIn1]}];
(* ARRAY *) dataE1=Table[dataIn1[[i,3]],{i,j,Length[dataIn1]}];
(* ARRAY *) dataf1=Table[dataIn1[[i,5]],{i,j,Length[dataIn1]}];
(* ARRAY *) dataw1=Table[dataIn1[[i,6]],{i,j,Length[dataIn1]}];

(* get first data point of electrochemistry, n - you may need to change the criteria *)
E0=0.0014; (* V *)
n=1; While[Abs[dataE1[[n]]-E0]<=0.01,n++]; n--;
n++
(* SKALAR *) R2s(* =Rs-R1 *)= 2  π L1 dataw1[[n]]-R1 (* Ω *)
(* SKALAR *) C0=Sqrt[(ρl ηl  π)/f0] 1/(8 Zq K2 R2s)  (* Sqrt[(g^2s)/(cm^4s)](cm^2sA)/gV = F
*)
(* SKALAR *) M=Nh/(8 K2 f0 C0) /.Nh->1 (* sV/As = Ω *)
(* SKALAR *) Xl=Sqrt[2 π f0 ρl ηl] (* Sqrt[1/s g/cm³ g/cms] = g/cm²s *)

ClearAll[j,isComment];
Remove[dataIn1,Nh];
0
1081.98
4.31948*10^-12
375626.
3604.64
dataw1[[n]]
19639.3

(* 1.b DEP DATA - FILM *)
(* ARRAY *) df0=dataf1-f0; (* 1/s *)
(* SKALAR *) fs=dataf1[[n]];
(* SKALAR *) ws=dataw1[[n]];
(* ARRAY *) dfs=dataf1-fs; (* 1/s *)
(* ARRAY *) dws=dataw1-ws; (* 1/s *)
```

```
(* ARRAY *) hfSB1=-(dfs/ρf) Zq/(2f0^2); (* cm *)
(* ARRAY *) Zfm=2 π f0 ρf hfSB1; (* 1/s g/cm³ cm = g/cm²s *)
(* ARRAY *) X2 (* =ω0L2 *)=-4 π L1 df0; (* H/s = Vs/As = Ω *)
(* ARRAY *) R2 (* =R-R1 *)= 2  π L1 dataw1-R1; (* Ω *)
(* SKALAR *) Xtr=Zq/M (-4 π L1 df0[[n]]-R2s) (* g/cm²s *)
(* ARRAY *) RE=Zq/M  R2/Xl;(* Ω/Ω(g/cm²s)/(g/cm²s) = 1 *)
(* ARRAY *) IM=(Zq/M X2/Xl-Xtr/Xl);(* Ω/Ω-(g/cm²s)/(g/cm²s) = 1 *)

(* GRAPHS *)
gfx1b1=ListLinePlot[Partition[Riffle[datat1,dfs],2],  AxesLabel->"df, dw",  PlotStyle-
>Red,ImageSize->Small,AxesOrigin->{0,0}];
gfx1b2=ListLinePlot[Partition[Riffle[datat1,dws],2],  AxesLabel->"dw", PlotStyle-
>Blue,ImageSize->Small,AxesOrigin->{0,0}];
gfx1b3=ListLinePlot[Partition[Riffle[datat1,Zfm],2], AxesLabel->"Zfm", PlotStyle-
>Black,ImageSize->Small,AxesOrigin->{0,0}];
gfx1b4=ListLinePlot[Partition[Riffle[datat1,R2],2],  AxesLabel->"R2, X2", PlotStyle-
>Red,ImageSize->Small,AxesOrigin->{0,0}];
gfx1b5=ListLinePlot[Partition[Riffle[datat1,X2],2],  AxesLabel->"X2", PlotStyle-
>Blue,ImageSize->Small,AxesOrigin->{0,0}];
gfx1b6=ListLinePlot[Partition[Riffle[datat1,RE],2],  AxesLabel->"RE, IM", PlotStyle-
>Red,ImageSize->Small, AxesOrigin->{0,0}];
gfx1b7=ListLinePlot[Partition[Riffle[datat1,IM],2],   AxesLabel->"IM", PlotStyle-
>Blue,ImageSize->Small,AxesOrigin->{0,0}];
GraphicsRow[{Show[gfx1b1,gfx1b2,PlotRange->All],gfx1b3, Show[gfx1b4,gfx1b5,PlotRange-
>All],Show[gfx1b6,gfx1b7,PlotRange->All]}]
-524.637

(* 1.c REVERSE SERIES FIT - b-MAXIMIZE *)
(* ARRAY *) dataG1={}; (* g/cms² = dyn/cm² = 10^5 N/cm² *)
(* ARRAY *) datab={}; (* b, stretch factor of hfSB, [b] = 1 *)
(* ARRAY *) bb={0.5,5.5}; (* range of b *)
```

```
Clear[dataGp]; dataGp={}; (* debug *)

(* SKALAR *) gs1=2800+400I; (* start value for FindRoot[] *)
fn1[g_]:=(Sqrt[I]+g/Xl Tanh[I ( b Zfm[[j]])/g])/(1+Sqrt[I] Xl/g Tanh[I ( b Zfm[[j]])/g])-(RE[[j]]+I
IM[[j]])

j=Length[datat1];While[j>=1,

run=1;While[run<=2,
(* Set b range *)
ClearAll[databb]; databb={}; (* helper array *)
If[run===1,
databb=Table[b,{b,bb[[1]],bb[[2]],0.05}],
databb=Table[b,{b,b1,b2,0.001}]; ];

(* Calculate g(b)-values *)
ClearAll[datagb];datagb={}; (* helper array *)
For[i=1,i<=Length[databb],i++,
rg1=Quiet[FindRoot[fn1[g] /.b->databb[[i]],{g,gs1}]];
AppendTo[datagb,g /.rg1]; ];

(* check for negatives and outliers *)
Glist=datagb^2/ρf;
(* For[i=1,i≤Length[databb],i++, (* remove negatives in pre-run only ... *)
If[run===1 && Re[Glist[[i]]]<0,   (* this may result in empty lists ... *)
databb=Delete[databb,i];          (* b1,b2 will then be the ones used previously *)
datagb=Delete[datagb,i];
Glist=Delete[Glist,i];
i--; ]; ]; *)
```

```
GInt=Quiet[Sort[{Median[Re[Glist]]-
5MedianDeviation[Re[Glist]],Median[Re[Glist]]+5MedianDeviation[Re[Glist]]}]];
(* For[i=1,i≤Length[databb],i++,
If[GInt[[1]]≤Re[Glist[[i]]]≤GInt[[2]] ,(*  do nothing *),
databb=Delete[databb,i];
datagb=Delete[datagb,i];
Glist=Delete[Glist,i];
i--; ]; ]; *)

(* Check for max G'(b) value *)
ClearAll[maxGp];
maxGp=Quiet[First[First[Position[Re[Glist],Max[Re[Glist]]]]]]; (* returns an index *)
(* gs1=datagb[[maxGp]]; ** results in scatter for fine run / start value issue *)
If[run===1,
AppendTo[dataGp,{maxGp,databb[maxGp],Length[Glist]}];
Which[bb[[1]]+0.1<databb[[maxGp]]<bb[[2]]-0.1,b1=databb[[maxGp]]-
0.1;b2=databb[[maxGp]]+0.1,
databb[[maxGp]]<=bb[[1]]+0.1,b1=bb[[1]];b2=bb[[1]]+0.2;,
databb[[maxGp]]>=bb[[2]]-0.1,b1=bb[[2]]-0.2;b2=bb[[2]];];
i++;,
(* else *)
G=Glist[[maxGp]];(* = gs1^2/ρf; ** see maxGp above *)
If[Re[G]>0&&Im[G]>0,AppendTo[dataG1,G];,AppendTo[dataG1,0];];
AppendTo[datab,databb[[maxGp]]];];

run++;];

(* store first and last g(b)-set for display *)
If[j===Length[datat1],databb2=databb;datagb2=datagb];
If[j===1,databb1=databb;datagb1=datagb];

j--;];
```

```
(* Fiting was done from end to start, thus: *)
dataG1=10^dims Reverse[dataG1];
datab=Reverse[datab];

(* get film properties *)
hf1=hfSB1 datab;
mf=A  ρf hf1;

(* GRAPHS *)
gfx1c1=ListLinePlot[Partition[Riffle[databb1,Re[datagb1^2/ρf]],2],AxesLabel-
>"G'(b)",PlotStyle->Green,ImageSize->Small,AxesOrigin->{bb[[1]],0}];
gfx1c2=ListLinePlot[Partition[Riffle[databb2,Re[datagb2^2/ρf]],2],PlotStyle->Red,ImageSize-
>Small];
gfx1c3=ListLinePlot[Partition[Riffle[datat1,datab],2],AxesLabel->"b",PlotStyle-
>Black,ImageSize->Small];
gfx1c4=ListLinePlot[Partition[Riffle[datat1,Re[dataG1]],2],AxesLabel->"G', G\"",PlotStyle-
>Red,ImageSize->Small];
gfx1c5=ListLinePlot[Partition[Riffle[datat1,Im[dataG1]],2],AxesLabel->"G\"",PlotStyle-
>Blue,ImageSize->Small];
GraphicsRow[{Show[gfx1c1,gfx1c2,PlotRange->{{bb[[1]],bb[[2]]},{-5
10^6,Automatic}}],gfx1c3,Show[gfx1c4,gfx1c5,PlotRange->{Automatic,{0,3*10^6}}]}]
(*
ClearAll[j,i,run,(**)E0,n,(**)Zfm,X2,R2,RE,IM,(**)gs1,rg1,maxGp,b,databb,databb1,databb2,
datagb,datagb1,datagb2,G,Glist,GInt,b1,b2];
Remove[dataE1,dataf1,dataw1,(**)df0,hfSB1,"gfx1@"]; *)
EmitSound[wav]

(* 1.d EXPORT RESULTS OF FITTING *)
fileOut=StringSplit[fileIn0,"."];
fileOut=fileOut[[1]]<>"_inf."<>fileOut[[2]];
tableOut={{"ηl",ηl},{"ρl",ρl},{"ρf",ρf},
{"f0",f0},{"w0",w0},
```

```
{"C0",C0},{"R1",R1},{"R2s",R2s},{"Xl",Xl},{"Xtr",Xtr},
{"fs",fs},{"ws",ws},{"b",bb}};
Export[fileOut,tableOut];
Remove[fileOut,tableOut];

fileOut=StringSplit[fileIn1,"."];
fileOut=fileOut[[1]]<>"_fit."<>fileOut[[2]];
tableHead={"df","dw","hf / m","b","mf / g",Gs,Gl};
tableOut=Table[{dfs[[i]],dws[[i]],hf1[[i]]/100,datab[[i]],mf[[i]],Re[dataG1[[i]]],Im[dataG1[[i]]]
},{i,1,Length[datat1]}];
tableOut=Prepend[tableOut,tableHead];
Export[fileOut,tableOut]
ClearAll[dfs,dws,hf1,mf,datab,dataG1];
Remove[fileOut,tableHead,tableOut,datat1];
1_Pol_fit.dat
ClearAll[i]
(* memory: *){ρq,μq,Zq,K2,L1,A,ηl,ρl,ρf,dims,(**) R2s,C0,M,Xl,Xtr,(**) fs,ws,(**)fn1,(**)
Gs,Gl, i};
?Global`*
 Global`

(* 2.a Cycling in monomer-free IL *)
(* !!! TROUBLE IN PARADISE if solvent (e.g. ρf and ηf) changes !!! *)
(* fileIn2="070110a01b.dat"; *)
dataIn2=Import[fileIn2];

(* get first data row *)
j=1; While[j<= Length[dataIn2],
isComment=StringMatchQ[dataIn2[[j,1]],"time"];
j++;
If[isComment===True,Break[];]; ]
```

```
(* ARRAY *) datat2=Table[dataIn2[[i,1]],{i,j,Length[dataIn2]}];
(* ARRAY *) dataE2=Table[dataIn2[[i,3]],{i,j,Length[dataIn2]}];
(* ARRAY *) dataf2=Table[dataIn2[[i,5]],{i,j,Length[dataIn2]}];
(* ARRAY *) dataw2=Table[dataIn2[[i,6]],{i,j,Length[dataIn2]}];

(* start evaluation right from the beginning - no n specification needed here *)
(* ARRAY *) dff0=dataf2-f0; (* 1/s *)
(* ARRAY *) dffs=dataf2-fs; (* 1/s *)
(* ARRAY *) dwfs=dataw2-ws ;(* 1/s *)

(* ARRAY *) hfSB2=-(dffs/ρf) Zq/(2f0^2); (* cm *)
(* ARRAY *) Zfm=2 π f0 ρf hfSB2 ;(* 1/s g/cm³ cm = g/cm²s *)
(* ARRAY *) X2 (* =ω0L2 *)=-4 π L1 dff0 ;(* H/s = Vs/As = Ω *)

(* ARRAY *) R2 (* =R-R1 *)= 2  π L1 dataw2-R1; (* Ω *)
(* ARRAY *) RE=Zq/M R2/Xl;(* Ω/Ω(g/cm²s)/(g/cm²s)=1 *)
(* ARRAY *) IM=(Zq/M 1/Xl (X2-Xtr));(* Ω/Ω-(g/cm²s)/(g/cm²s) = 1 *)

(* GRAPHS *)
gfx2a1=ListLinePlot[Partition[Riffle[datat2,dffs],2],  AxesLabel->"df, dw",  PlotStyle-
>Red,ImageSize->Small,AxesOrigin->{0,0}];
gfx2a2=ListLinePlot[Partition[Riffle[datat2,dwfs],2],  AxesLabel->"dw", PlotStyle-
>Blue,ImageSize->Small,AxesOrigin->{0,0}];
gfx2a3=ListLinePlot[Partition[Riffle[datat2,Zfm],2], AxesLabel->"Zfm", PlotStyle-
>Black,ImageSize->Small,AxesOrigin->{0,0}];
gfx2a4=ListLinePlot[Partition[Riffle[datat2,R2],2],  AxesLabel->"R2, X2", PlotStyle-
>Red,ImageSize->Small,AxesOrigin->{0,0}];
gfx2a5=ListLinePlot[Partition[Riffle[datat2,X2],2],  AxesLabel->"X2", PlotStyle-
>Blue,ImageSize->Small,AxesOrigin->{0,0}];
gfx2a6=ListLinePlot[Partition[Riffle[datat2,RE],2],  AxesLabel->"RE, IM", PlotStyle-
>Red,ImageSize->Small, AxesOrigin->{0,0}];
```

```
gfx2a7=ListLinePlot[Partition[Riffle[datat2,IM],2],   AxesLabel->"IM", PlotStyle->Blue,ImageSize->Small,AxesOrigin->{0,0}];
GraphicsGrid[{{Show[gfx2a1,gfx2a2,PlotRange->All],gfx2a3, Show[gfx2a4,gfx2a5,PlotRange->All],Show[gfx2a6,gfx2a7,PlotRange->All]}}]
ClearAll[j,isComment];
Remove[dataIn2];

(* 2.b REVERSE SERIES FIT - b-MAXIMIZE *)
(* ARRAY *) dataG1={};
(* ARRAY *) datab={};
(* ARRAY *)(* bb={0.5,1.5}; *)(* range of b *)

(* SKALAR *) gs1=2400+800I; (* start value for FindRoot[] *)

j=Length[datat2];While[j>=1,
run=1;While[run<=2,
(* Set b range *)
ClearAll[databb];
If[run===1,
databb=Table[b,{b,bb[[1]],bb[[2]],0.025}],
databb=Table[b,{b,b1,b2,0.001}]; ];

(* Calculate g(b)-values *)
ClearAll[datagb];datagb={};
For[i=1,i<=Length[databb],i++,
rg1=Quiet[FindRoot[fn1[g] /.b->databb[[i]],{g,gs1}]];
AppendTo[datagb,g /.rg1]; ];

(* check for outliers *)
Glist=datagb^2/ρf;
(* GInt=Sort[{Median[Re[Glist]]-5MedianDeviation[Re[Glist]],Median[Re[Glist]]+5MedianDeviation[Re[Glist]]}];
```

```
j--;];
```

```
For[i=1,i≤Length[databb],i++,
If[GInt[[1]]≤Re[Glist[[i]]]≤GInt[[2]],(*  do nothing *),
databb=Delete[databb,i];
datagb=Delete[datagb,i];
Glist=Delete[Glist,i];
i--; ]; ]; *)

(* Check for max G'(b) value *)
maxGp=First[First[Position[Re[Glist],Max[Re[Glist]]]]]; (* returns an index *)
(* gs1=datagb[[maxGp]]; ** results in scatter for fine run / start value issue *)

If[run===1,
Which[bb[[1]]+0.1<databb[[maxGp]]<bb[[2]]-0.1,b1=databb[[maxGp]]-
0.1;b2=databb[[maxGp]]+0.1,
databb[[maxGp]]<=bb[[1]]+0.1,b1=bb[[1]];b2=bb[[1]]+0.2;,
databb[[maxGp]]>=bb[[2]]-0.1,b1=bb[[2]]-0.2;b2=bb[[2]];];
i++;,
(* else *)
G=Glist[[maxGp]];(* = gs1^2/ρf; ** see maxGp above *)
If[Re[G]>0&&Im[G]>0,AppendTo[dataG1,G];,AppendTo[dataG1,0];];
AppendTo[datab,databb[[maxGp]]];];

run++;];

(* store first and last g(b)-set for display *)
If[j===Length[datat2],databb2=databb;datagb2=datagb];
If[j===1,databb1=databb;datagb1=datagb];

j--;];

(* Fiting was done from end to start, thus: *)
dataG1=10^dims Reverse[dataG1];
```

```
datab=Reverse[datab];

(* mass loss of the film [dopant mass] *)
hf2=hfSB2 datab;
mf=A  ρf hfSB2*datab;

(* GRAPHS *)
gfx2b1=ListLinePlot[Partition[Riffle[databb1,Re[datagb1^2/ρf]],2],AxesLabel-
>"G'(b)",PlotStyle->Green,ImageSize->Small];
gfx2b2=ListLinePlot[Partition[Riffle[databb2,Re[datagb2^2/ρf]],2],PlotStyle->Red,ImageSize-
>Small];
gfx2b3=ListLinePlot[Partition[Riffle[datat2,datab],2],AxesLabel->"b",PlotRange-
>{Automatic,{bb[[1]],bb[[2]]}},ImageSize->Small];
gfx2b4=ListLinePlot[Partition[Riffle[datat2,Re[dataG1]],2],AxesLabel->"G', G\"",PlotStyle-
>Red,ImageSize->Small];
gfx2b5=ListLinePlot[Partition[Riffle[datat2,Im[dataG1]],2],AxesLabel->"G\"",PlotStyle-
>Blue,ImageSize->Small];
GraphicsRow[{Show[gfx2b1,gfx2b2,PlotRange->All],gfx2b3,Show[gfx2b4,gfx2b5,PlotRange-
>Automatic]}]
(*ClearAll[j,i,run,(**)Zfm,X2,R2,RE,IM,(**)gs1,rg1,maxGp,b,databb,databb1,databb2,datagb,
datagb1,datagb2,G,Glist,GInt,b1,b2];
Remove[dataE2,dataf2,dataw2,(**)dff0,hfSB2,"gfx2@"]; *)
EmitSound[wav]

(* 2.c EXPORT RESULTS OF FITTING *)
fileOut=StringSplit[fileIn2,"."];
fileOut=fileOut[[1]]<>"_fit."<>fileOut[[2]];
tableHead={"df","dw","hf / m","b","mf / g",Gs,Gl};
tableOut=Table[{dffs[[i]],dwfs[[i]],hf2[[i]]/100,datab[[i]],mf[[i]]
,Re[dataG1[[i]]],Im[dataG1[[i]]]},{i,1,Length[datat2]}];
tableOut=Prepend[tableOut,tableHead];
Export[fileOut,tableOut]
```

```
(*ClearAll[dffs,dwfs,hf2,mf,datab,dataG1];*)
Remove[fileOut,tableHead,tableOut,datat2];
2_Cycl1_fit.dat
(* memory: *){ρq,μq,Zq,K2,L1,A,ηl,ρl,ρf,dims,(**)R2s,C0,M,Xl,(**) fs,ws,Xtr,(**)fn1,(**)
Gs,Gl};
?Global`*
```