Electronic Supplementary Material (ESI) for Physical Chemistry Chemical Physics.

Electronic Supplementary Information for the Comment on "A single level tunneling model for molecular junctions: evaluating the simulation methods" by Opodi et al

Ioan Bâldea

Theoretical Chemistry, Heidelberg University, INF 229, D-69124 Heidelberg, Germany

The reader can find below all the problematic issues related to the MATLAB code of ref. 1. Letting alone the five syntactic errors it contains (marked in **red** below in the MATLAB files **Tunneling1SyntaxErrorsCorrected.m** and **Fitting-SyntaxErrorsCorrected.m)**, the main problem is the size mismatch of the variables transmitted from the routine "Tunneling1" to the main program (see the highlighted text in **red** below).

As a possible way to overcome the aforementioned difficulties, we present a compact MATLAB code that can be flexibly adapted to any data fitting based on method 1 (see MATLAB file **IVfittingExcelInput.m** attached).

For the reader's convenience, we also attach a MATLAB code (**generateIVfitIV.m**) able to simulate I-V data using model parameter values and overimposed noise as desired (values highlighted in magenta below), used subsequetly as input for data fitting.

In all MATLAB codes below, all energies are in electronvolt and currents in Ampère.

```
ls -l Tunneling1SyntaxErrorsCorrected.m FittingSyntaxErrorsCorrected.m

-rw-r--r-- 1 ioan  984 Dec 28 00:01 FittingSyntaxErrorsCorrected.m
-rw-r--r-- 1 ioan 1106 Dec 28 00:11 Tunneling1SyntaxErrorsCorrected.m

ls -l folder.xlsx # input I-V data

-rw-r--r-- 1 ioan 16600 Dec 27 22:13 folder.xlsx

cat FittingSyntaxErrorsCorrected.m

clear;
clc;
tdata=xlsread('folder');
I=((tdata(:,2)));
V=((tdata(:,1))); % IB: syntax error corrected
betaT0=[0.3, 1e-3]; % IB: just a specific choice, NOT an error of Opodi et al
% betaT0=['input Eg g'];
opts = statset('RobustWgtFun','bisquare','MaxIter',1000);
betaT=nlinfit(V,I,@Tunneling1,betaT0,opts);
EgT=betaT(1);
GamaR=betaT(2);
betaT1=[EgT GamaR];
```

```
VV=linspace(-2,2,501);
Ifit=Tunneling1(betaT1,VV);
figure(1)
plot(VV,Ifit,'b')
hold on
plot(V,I,'r.')
xlabel('Voltage(v)')
ylabel('Current(A)')
title('Current---Voltage')
grid on
```

cat Tunneling1SyntaxErrorsCorrected.m

```
%% Definition of fitting function
function IT=Tunneling1(beta,V)%Function
%%% Constant setting
e=1.60217e-19;% electron charge
h=6.62607004e-34;%Planks constant
hbar=h/(2*pi);
% below is the calculation of the I-V
G0=(2*e*e/hbar);% conductance quantum
kT=0.025;
N=1; % IB: syntax error corrected
IV=length(V);
Eg=beta(1);%energy offset
g1=beta(2);%coupling
g2=g1;
g=g2+g1;
alpha=0.5;%voltage division factor
%Energy grid
NE=50001; % IB: syntax error corrected
E=linspace(-5,5,NE); % IB: syntax error corrected
dE=E(2)-E(1); % IB: syntax error corrected
D=(g/(2*pi))./((E.^2)+((g/2)^2));
for iV=1:IV
Vd=V(iV);
UL=(alpha*Vd);
UL2=((1-alpha)*Vd);
f1=1./(1+exp((E-(Eg+UL))./kT));
f2=1./(1+exp((E-(Eg-UL2))./kT));
IT(iV)=((N*dE*G0*(sum(D.*(f1-f2)))*(g1*g2/g)));
end
```

Same error message generated by using matlab/R2020a and matlab/R2021a:

```
matlab -nodisplay -singleCompThread -logfile log -batch FittingSyntaxErrorsCorrected

Error using nlinfit (line 219)
MODELFUN must be a function that returns a vector of fitted values the same
size as Y (501-by-1).  The model function you provided returned a result that
was 1-by-501.

One common reason for a size mismatch is using matrix operators (*, /, ^) in
your function instead of the corresponding elementwise operators (.*, ./, .^).

Error in FittingSyntaxErrorsCorrected (line 9)
betaT=nlinfit(V,I,@Tunneling1,betaT0,opts);
```

```
ls -l generateIVfitIV.m IVfittingExcelInput.m

-rw-r--r-- 1 ioan 1224 Dec 28 00:33 generateIVfitIV.m
-rw-r--r-- 1 ioan 1391 Dec 28 00:34 IvfittingExcelInput.m

ls -l IVcurve.xlsx # I-V data underlying red curve in Fig. 3

-rwxr-xr-x 1 ioan 5958 Dec 27 23:22 Ivfolder.xlsx

cat IVfittingExcelInput.m

clear;
clc;
Ivdata=xlsread('Ivcurve.xlsx');% Import IV data to be fitted from an EXCEL file
I=((IVdata(:,2)));
V=((IVdata(:,1)));

G0 = 77.48e-6;
kT = 0.0257;
N = 1;
e0i = 0.3; Deltai = 1e-3; % Initial guess

betaT0 = [e0i Deltai];
transm = @(x, e0, GammaGeom, GammaArithm) GammaGeom.^2./( (x-e0).^2 +
GammaArithm.^2);
fermi  = @(x, kT) 1./(1 + exp(x./kT));
integrand = @(x, e0, GammaGeom, GammaArithm, kT, V) (fermi(x-V./2, kT) - fermi(x
+ V./2, kT)) .* transm(x, e0, GammaGeom, GammaArithm);
current = @(V, e0, GammaGeom, GammaArithm, kT) integral(@(x) integrand(x, e0,
GammaGeom, GammaArithm, kT, V), -inf, inf, 'ArrayValued', true);
jT = @(beta,V) N .* G0 .* current(V, beta(1), beta(2), beta(2), kT);

% Specify nonlinear fitting model:
opts = statset('RobustWgtFun','bisquare','MaxIter',1000);
betaBestFit=nlinfit(V,I, @(beta,V) jT(beta, V), [e0i Deltai], opts);

e0BestFit=betaBestFit(1); %fitting result of energy barrier
GammaTildeBestFit=betaBestFit(2); %fitting result of coupling strength

fprintf(' Best Fit: e0= %12.9g GammaTilde= %12.9g  Gamma= %12.9g\n', [e0BestFit;
GammaTildeBestFit; 2.* GammaTildeBestFit])

Ifit = jT(betaBestFit, V);

% Write fitted I-V curve in ASCII format to plot using e.g. gnuplot or xmgrace:
for iV=1:length(V)
  fprintf(' V= %12.9g  Iexp= %12.9g Ifit= %12.9g \n', [V(iV); I(iV); Ifit(iV)]);
end


matlab -nodisplay -singleCompThread -logfile log -batch IVfittingExcelInput

 Best Fit: e0=  0.269970303 GammaTilde=  0.035484049  Gamma=  0.070968098

 V=         -0.7  Iexp= -6.94996305e-06 Ifit= -6.96519544e-06
 V=      -0.6972  Iexp= -6.91669218e-06 Ifit= -6.93696214e-06
 V=      -0.6944  Iexp= -6.93058737e-06 Ifit= -6.90804291e-06
.....
.....
 V=       0.6916  Iexp= 6.89873492e-06 Ifit= 6.87842487e-06
 V=       0.6944  Iexp= 6.89666924e-06 Ifit= 6.90804291e-06
 V=       0.6972  Iexp= 6.94525668e-06 Ifit= 6.93696214e-06
 V=          0.7  Iexp= 6.97180084e-06 Ifit= 6.96519544e-06
```

```
cat generateIVfitIV.m

kT = 0.0257; % Value for room temperature
G0 = 77.48e-6; % Conductance quantum
N = 1; % Number of molecules per junction

% Start I-V data simulation, to be subsequently fitted
transm = @(x, e0, GammaGeom, GammaArithm) GammaGeom.^2./( (x-e0).^2 +
GammaArithm.^2); % transmission function
fermi  = @(x, kT) 1./(1 + exp(x./kT));
integrand = @(x, e0, GammaGeom, GammaArithm, kT, V) (fermi(x-V./2, kT) - fermi(x
+ V./2, kT)) .* transm(x, e0, GammaGeom, GammaArithm);
current = @(V, e0, GammaGeom, GammaArithm, kT) integral(@(x) integrand(x, e0,
GammaGeom, GammaArithm, kT, V), -inf, inf, 'ArrayValued', true);
jT = @(beta,V) G0 .* current(V, beta(1), beta(2), beta(2), kT); % T-dependent
current

% Using model parameters as in Fig. 3
e0=0.27; Delta = 0.071 ./ 2; GammaGeom = Delta; GammaArithm = Delta;
NVg=501; % Generate I-V curves with 501 points
Vg=linspace(-0.7, 0.7,NVg);

noiseAmplitude = 0.02;
for iV=1:NVg
      % Below, 'rand' is used to add a bit disorder
      Ig(iV) = G0 .* current(Vg(iV), e0, GammaGeom, GammaArithm, kT) + G0 .*
GammaGeom.^2/GammaArithm .* noiseAmplitude .* (rand - 0.5);
      IVsimulated(iV, 1)=Vg(iV); IVsimulated(iV, 2)=Ig(iV);
end

writematrix(IVsimulated, 'IVcurve.xlsx'); % Export I-V curve to an EXCEL file
% End I-V data simulation

% Start data fitting
% Import IV data to be fitted from the EXCEL file
IVdata=xlsread('IVcurve.xlsx');
I=((IVdata(:,2)));
V=((IVdata(:,1)));

e0i = 0.3; Deltai = 0.03; % Initial guess
opts = statset('RobustWgtFun','bisquare','MaxIter',1000);
betaBestFit=nlinfit(V,I, @(beta,V) jT(beta, V), [e0i Deltai], opts);

e0BestFit=betaBestFit(1);
GammaTildeBestFit=betaBestFit(2);

fprintf(' Best Fit: e0= %12.9g GammaTilde= %12.9g  Gamma= %12.9g\n', [e0BestFit;
GammaTildeBestFit; 2.* GammaTildeBestFit])

Ifit = jT(betaBestFit, V);

% Write fitted I-V curve in ASCII format to plot using e.g. gnuplot or xmgrace:
for iV=1:length(V)
  fprintf(' V= %12.9g  Iexp= %12.9g Ifit= %12.9g \n', [V(iV); I(iV); Ifit(iV)]);
end


matlab -nodisplay -singleCompThread -logfile log -batch generateIVfitIV
```

```
 Best Fit: e0=  0.269970303 GammaTilde=  0.035484049  Gamma=  0.070968098

 V=         -0.7  Iexp= -6.94996305e-06 Ifit= -6.96519544e-06
 V=      -0.6972  Iexp= -6.91669218e-06 Ifit= -6.93696214e-06
 V=      -0.6944  Iexp= -6.93058737e-06 Ifit= -6.90804291e-06
 V=      -0.6916  Iexp= -6.85768076e-06 Ifit= -6.87842487e-06
 V=      -0.6888  Iexp= -6.8427811e-06 Ifit= -6.84809535e-06
 V=       -0.686  Iexp= -6.84111897e-06 Ifit= -6.8170419e-06
 V=      -0.6832  Iexp= -6.79934487e-06 Ifit= -6.78525231e-06
 V=      -0.6804  Iexp= -6.75201301e-06 Ifit= -6.7527147e-06
 V=      -0.6776  Iexp= -6.69609636e-06 Ifit= -6.71941751e-06
 V=      -0.6748  Iexp= -6.66159136e-06 Ifit= -6.68534956e-06
 V=       -0.672  Iexp= -6.67111948e-06 Ifit= -6.65050014e-06
 V=      -0.6692  Iexp= -6.59072399e-06 Ifit= -6.61485898e-06
 V=      -0.6664  Iexp= -6.55498797e-06 Ifit= -6.57841635e-06
 V=      -0.6636  Iexp= -6.54365611e-06 Ifit= -6.54116313e-06
 V=      -0.6608  Iexp= -6.48822807e-06 Ifit= -6.50309079e-06
 V=       -0.658  Iexp= -6.4855148e-06 Ifit= -6.46419152e-06
 V=      -0.6552  Iexp= -6.43035235e-06 Ifit= -6.42445823e-06
 V=      -0.6524  Iexp= -6.36257161e-06 Ifit= -6.38388462e-06
 V=      -0.6496  Iexp= -6.32791424e-06 Ifit= -6.34246528e-06
 V=      -0.6468  Iexp= -6.27640856e-06 Ifit= -6.30019565e-06
 V=       -0.644  Iexp= -6.24996101e-06 Ifit= -6.25707217e-06
 V=      -0.6412  Iexp= -6.24005562e-06 Ifit= -6.21309229e-06
 V=      -0.6384  Iexp= -6.15043723e-06 Ifit= -6.16825452e-06
 V=      -0.6356  Iexp= -6.10003887e-06 Ifit= -6.12255851e-06
 V=      -0.6328  Iexp= -6.06749347e-06 Ifit= -6.07600508e-06
 V=        -0.63  Iexp= -6.01570464e-06 Ifit= -6.02859629e-06
.....
.....
.....
 V=          0.63  Iexp= 6.03407547e-06 Ifit= 6.02859629e-06
 V=        0.6328  Iexp= 6.08741279e-06 Ifit= 6.07600508e-06
 V=        0.6356  Iexp= 6.12647631e-06 Ifit= 6.12255851e-06
 V=        0.6384  Iexp= 6.16555739e-06 Ifit= 6.16825452e-06
 V=        0.6412  Iexp= 6.22246066e-06 Ifit= 6.21309229e-06
 V=         0.644  Iexp= 6.266649e-06 Ifit= 6.25707217e-06
 V=        0.6468  Iexp= 6.31153346e-06 Ifit= 6.30019565e-06
 V=        0.6496  Iexp= 6.35145853e-06 Ifit= 6.34246528e-06
 V=        0.6524  Iexp= 6.40993094e-06 Ifit= 6.38388462e-06
 V=        0.6552  Iexp= 6.41003666e-06 Ifit= 6.42445823e-06
 V=         0.658  Iexp= 6.47732744e-06 Ifit= 6.46419152e-06
 V=        0.6608  Iexp= 6.49023658e-06 Ifit= 6.50309079e-06
 V=        0.6636  Iexp= 6.52191429e-06 Ifit= 6.54116313e-06
 V=        0.6664  Iexp= 6.58603997e-06 Ifit= 6.57841635e-06
 V=        0.6692  Iexp= 6.61386871e-06 Ifit= 6.61485898e-06
 V=         0.672  Iexp= 6.65001395e-06 Ifit= 6.65050014e-06
 V=        0.6748  Iexp= 6.69607396e-06 Ifit= 6.68534956e-06
 V=        0.6776  Iexp= 6.7361328e-06 Ifit= 6.71941751e-06
 V=        0.6804  Iexp= 6.74635238e-06 Ifit= 6.7527147e-06
 V=        0.6832  Iexp= 6.79607216e-06 Ifit= 6.78525231e-06
 V=         0.686  Iexp= 6.81436716e-06 Ifit= 6.8170419e-06
 V=        0.6888  Iexp= 6.86887208e-06 Ifit= 6.84809535e-06
 V=        0.6916  Iexp= 6.89873492e-06 Ifit= 6.87842487e-06
 V=        0.6944  Iexp= 6.89666924e-06 Ifit= 6.90804291e-06
 V=        0.6972  Iexp= 6.94525668e-06 Ifit= 6.93696214e-06
 V=           0.7  Iexp= 6.97180084e-06 Ifit= 6.96519544e-06
```

(Notice that fitting results may slighltly differ from run to run because of the small amount disorder included via random number generation in data simulation)