Makaiko L. Chithambo

Appendix A

Examples of computer code

In this appendix, we provide two examples of computer codes used to describe the kinetic model of luminescence[1]. The codes are prepared in the software package *Mathematica*. The first example is the code used to produce figure 3 in Pagonis *et al* [1] for quartz. The second example is the code used to generate figure 10 in Pagonis *et al* [2] for α-Al_2O_3:C, taking into account the presence of shallow traps in the model.

A1 Simulation of time-resolved spectra at various stimulation temperatures in quartz

```
Remove["Global'"];
k = 8.617'/10^5; sum2List = {}; tauList = {};
temps = {20, 100, 150, 200, 250, 270};
grall1 = {}; grall2 = {}; grall3 = {}; grall4 = {}; grall5 = {}; /
aList = {}; grall6 = {}; grall10 = {}; grall7 = {}; grall11 = {}; /
grall8 = {}; grall9 = {};
A = 1.3' 10^11;
Do[Clear[w3, w3NR, m, nc, m0NR, t, gr1, gr2, gr3, gr4, sol];
  irrTemp = temps&i1;; nc0 = 0; m0 = 10^14; mNR0 = 10^14;
  n10 = 9 10^13; N1 = 10^14; An1 = 5/10^14; P = 0.2'; Ae2 = 1/10^8;
  N2 = 10^14; d2 = 10^6; Ae24 = 0.5' 10^-9; tau2 = 1/10^6; n20 = 0;

  sol = NDSolve[{Derivative[1][n1][t] == -n1[t] P +
       An1 (N1 - n1[t]) nc[t],
     Derivative[1][nc][t] ==
```

[1] Kindly provided by Dr V Pagonis, Professor of Physics at McDaniel College, MD, United States of America.

doi:10.1088/2053-2571/aae5dach7

```
     n1[t] P - An1 (N1 - n1[t]) nc[t] -
      nc[t] (N2 - n2[t] - ne[t]) Ae2,
    Derivative[1][ne][t] ==
     d2 n2[t] + nc[t] (N2 - n2[t] - ne[t]) Ae2 - ne[t] Ae24 m[t] -
      ne[t]/tau2 - ne[t] A E^(-(0.64'/(k (273 + irrTemp)))),
    Derivative[1][n2][t] == -d2 n2[t] + ne[t]/tau2,
    Derivative[1][m][t] == -ne[t] Ae24 m[t],
    Derivative[1][mNR][t] == +A E^(-(0.64'/(k (273 + irrTemp))))
       ne[t], nc[0] == nc0, ne[0] == 0, n1[0] == n10, n2[0] == n20,
    m[0] == m0, mNR[0] == mNR0}, {nc, ne, n1, n2, m, mNR}, {t, 0,
    200/10^6}];

gr1 = Plot[nc[t] /. sol, {t, 0, 200/10^6}, PlotRange -> All,
  PlotLabel -> {{nc}, irrTemp}, PlotStyle -> {Hue[i1 0.1']},
  DisplayFunction -> Identity];
gr2 = Plot[n1[t] /. sol, {t, 0, 200/10^6}, PlotRange -> All,
  PlotLabel -> {{n1}, irrTemp}, PlotStyle -> {Hue[i1 0.1']},
  DisplayFunction -> Identity];
gr4 = Plot[ne[t] Ae24 m[t] /. sol, {t, 0, 200/10^6},
  PlotRange -> All, PlotLabel -> {{TROSL}, irrTemp},
  PlotStyle -> {Hue[i1 0.1']}, DisplayFunction -> Identity];
gr3 = Plot[n2[t] /. sol, {t, 0, 200/10^6}, PlotRange -> All,
  PlotLabel -> {{n2}, irrTemp}, PlotStyle -> {Hue[i1 0.1']},
  DisplayFunction -> Identity];
gr6 = Plot[ne[t] /. sol, {t, 0, 200/10^6}, PlotRange -> All,
  PlotLabel -> {{ne}, irrTemp}, PlotStyle -> {Hue[i1 0.1']},
  DisplayFunction -> Identity];
gr5 = Plot[m[t] /. sol, {t, 0, 200/10^6}, PlotRange -> All,
  PlotLabel -> {{m}, irrTemp}, PlotStyle -> {Hue[i1 0.1']},
  DisplayFunction -> Identity];
graphs = Show[
  GraphicsArray[{{gr1, gr2, gr3}, {gr4, gr5, gr6}},
   Spacings -> Scaled[0.03'], ImageSize -> 72 8],
  DisplayFunction -> $DisplayFunction]; AppendTo[grall6, gr1];
AppendTo[grall7, gr2]; AppendTo[grall8, gr4]; AppendTo[grall9, gr3];
 AppendTo[grall10, gr6];

Print["nc0=", N[nc0], " ne0=", N[ne0], " n10=", N[n10], " n20=",
 N[n20], " m0=", N[m0], "mNR0=", N[mNR0]];
nc0 = First[nc[200/10^6] /. sol]; ne0 = First[ne[200/10^6] /. sol];
n10 = First[n1[200/10^6] /. sol]; n20 = First[n2[200/10^6] /. sol];
m0 = First[m[200/10^6] /. sol]; mNR0 = First[mNR[200/10^6] /. sol];
P = 0;

sol2 = NDSolve[{Derivative[1][n1][t] == -n1[t] P +
      An1 (N1 - n1[t]) nc[t],
    Derivative[1][nc][t] ==
     n1[t] P - An1 (N1 - n1[t]) nc[t] -
      nc[t] (N2 - n2[t] - ne[t]) Ae2,
    Derivative[1][ne][t] ==
```

```
    d2 n2[t] + nc[t] (N2 - n2[t] - ne[t]) Ae2 - ne[t] Ae24 m[t] -
     ne[t]/tau2 - ne[t] A E^(-(0.64'/(k (273 + irrTemp)))),
   Derivative[1][n2][t] == -d2 n2[t] + ne[t]/tau2,
   Derivative[1][m][t] == -ne[t] Ae24 m[t],
   Derivative[1][mNR][t] == +A E^(-(0.64'/(k (273 + irrTemp))))
      ne[t], nc[0] == nc0, ne[0] == ne0, n1[0] == n10, n2[0] == n20,
    m[0] == m0, mNR[0] == mNR0}, {nc, ne, n1, n2, m, mNR}, {t, 0,
   200/10^6}];

gr1 = Plot[nc[t] /. sol2, {t, 0, 200/10^6}, PlotRange -> All,
  PlotLabel -> {{nc}, irrTemp}, PlotStyle -> {Hue[i1 0.1']},
  DisplayFunction -> Identity];
gr2 = Plot[n1[t] /. sol2, {t, 0, 200/10^6}, PlotRange -> All,
  PlotLabel -> {{n1}, irrTemp}, PlotStyle -> {Hue[i1 0.1']},
  DisplayFunction -> Identity];
gr4 = Plot[ne[t] Ae24 m[t] /. sol2, {t, 0, 200/10^6},
  PlotRange -> All, PlotLabel -> {{TROSL}, irrTemp},
  PlotStyle -> {Hue[i1 0.1']}, DisplayFunction -> Identity];
gr3 = Plot[n2[t] /. sol2, {t, 0, 200/10^6}, PlotRange -> All,
  PlotLabel -> {{n2}, irrTemp}, PlotStyle -> {Hue[i1 0.1']},
  DisplayFunction -> Identity];
gr6 = Plot[ne[t] /. sol2, {t, 0, 200/10^6}, PlotRange -> All,
  PlotLabel -> {{ne}, irrTemp}, PlotStyle -> {Hue[i1 0.1']},
  DisplayFunction -> Identity];
gr5 = Plot[m[t] /. sol2, {t, 0, 200/10^6}, PlotRange -> All,
  PlotLabel -> {{m}, irrTemp}, PlotStyle -> {Hue[i1 0.1']},
  DisplayFunction -> Identity];
graphs = Show[
  GraphicsArray[{{gr1, gr2, gr3}, {gr4, gr5, gr6}},
   Spacings -> Scaled[0.03'], ImageSize -> 72 8],
  DisplayFunction -> $DisplayFunction]; AppendTo[grall6, gr1];
AppendTo[grall7, gr2]; AppendTo[grall8, gr4]; AppendTo[grall9, gr3];
 AppendTo[grall10, gr6];

If[i1 == 1,
 a1 = Table[{N[t], First[Flatten[ne[t] Ae24 m[t] /. sol2]]}, {t, 0,
    200/10^6, 1/10^6}]];
If[i1 == 2,
 a2 = Table[{N[t], First[Flatten[ne[t] Ae24 m[t] /. sol2]]}, {t, 0,
    200/10^6, 1/10^6}]];
If[i1 == 3,
 a3 = Table[{N[t], First[Flatten[ne[t] Ae24 m[t] /. sol2]]}, {t, 0,
    200/10^6, 1/10^6}]];
If[i1 == 4,
 a4 = Table[{N[t], First[Flatten[ne[t] Ae24 m[t] /. sol2]]}, {t, 0,
    200/10^6, 1/10^6}]];
If[i1 == 5,
 a5 = Table[{N[t], First[Flatten[ne[t] Ae24 m[t] /. sol2]]}, {t, 0,
    200/10^6, 1/10^6}]];
```

```
If[i1 == 6,
 a6 = Table[{N[t], First[Flatten[ne[t] Ae24 m[t] /. sol2]]}, {t, 0,
    200/10^6, 1/10^6}]];, {i1, 1, 6}];

For[i = 1, i <= Length[a1], i++,
  dataRow = {Table[a1[[j]], {j, i, i}], Table[a2[[j]], {j, i, i}],
    Table[a3[[j]], {j, i, i}], Table[a4[[j]], {j, i, i}],
    Table[a5[[j]], {j, i, i}], Table[a6[[j]], {j, i, i}]};
  AppendTo[aList, Flatten[dataRow]];];
SetDirectory["C://Documents and Settings//Vasilis Pagonis//Desktop"];
Export["1.txt", aList, "Table"];


bList = {a1, a2, a3, a4, a5, a6}; sum2List = {};
Do[Clear[sum2, snlog, tlList3, tau, a22, x2, f1];
  tlList3 = bList&i1;; irrTemp = temps&i1;;

  ListPlot[tlList3, PlotJoined -> True,
   AxesLabel -> {"Time,s)", "TROSL"},
   PlotStyle -> {RGBColor[1, 0, 0]}, PlotRange -> All,
   ImageSize -> 72 4]; sum2 = Plus @@ tlList3[[All, 2]];
  Print["sum2=", sum2]; AppendTo[sum2List, {irrTemp, sum2}];
  npoints = 1; tlList3 = Drop[tlList3, -npoints];
  tlList3 = Drop[tlList3, 2];
  Print[ListPlot[tlList3, ImageSize -> 72 2]];
  ListPlot[Transpose[{tlList3[[All, 1]], Log[tlList3[[All, 2]]]}],
   ImageSize -> 72 4];

  snlog = Transpose[{tlList3[[All, 1]], Log[tlList3[[All, 2]]]}];
  f1 = ListPlot[snlog, PlotRange -> All, ImageSize -> 72 2];
  a22[x2_] = Fit[snlog, {1, x2}, x2];
  Print["slope=", a22[1] - a22[0]]; tau = 10^6/(a22[1] - a22[0]);
  Print["tau=", 10^6/(a22[1] - a22[0]), " microsecs"];
  AppendTo[tauList, {irrTemp, -tau}];
  Print[irrTemp, " deg C"];, {i1, 1, 6}];

grtau = ListPlot[tauList,
   PlotRange -> {{0, Automatic}, {0, Automatic}},
   PlotLabel -> {"tau, microseconds"}, PlotJoined -> False,
   ImageSize -> 72 3, DisplayFunction -> Identity];
grsum2 = ListPlot[sum2List,
   PlotRange -> {{0, Automatic}, {0, Automatic}},
   PlotLabel -> {"area-TROSL"}, PlotJoined -> False,
   ImageSize -> 72 3, DisplayFunction -> Identity];
grcalctau =
  Plot[tauList[[1, 2]]/(
   1 + 0.26' 10^7 E^(-(0.64'/(k (273 + x))))), {x, 20, 350},
   ImageSize -> 72 3, DisplayFunction -> Identity];
```

```
calctau =
  Table[{x, tauList[[1, 2]]/(
    1 + 0.26' 10^7 E^(-(0.64'/(k (273 + x)))))}, {x, 20, 350}];

Print[Show[grall6, PlotRange -> All,
  DisplayFunction -> $DisplayFunction, ImageSize -> 72 4]]; Print[
 Show[grall7, PlotRange -> All, DisplayFunction -> $DisplayFunction,
  ImageSize -> 72 4]]; Print[
 Show[grall8, PlotRange -> All, DisplayFunction -> $DisplayFunction,
  ImageSize -> 72 4]]; Print[
 Show[grall9, DisplayFunction -> $DisplayFunction,
  ImageSize -> 72 4]]; Print[
 Show[grall10, DisplayFunction -> $DisplayFunction,
  ImageSize -> 72 4]];
Print[Show[{grcalctau, grtau}, PlotLabel -> {"tau vs temp"},
   ImageSize -> 72 3, DisplayFunction -> $DisplayFunction]];

grcalcsum2 =
  Plot[{x, sum2List[[1, 2]]/(
    1 + 0.26' 10^7 E^(-(0.64'/(k (273 + x)))))}, {x, 20, 350},
   ImageSize -> 72 3, DisplayFunction -> Identity];
calcsum2 =
  Table[{x, sum2List[[1, 2]]/(
    1 + 0.26' 10^7 E^(-(0.64'/(k (273 + x)))))}, {x, 20, 350}];
Show[{grcalcsum2, grsum2}, PlotLabel -> {"TROSL-sum vs temp"},
 ImageSize -> 72 3, DisplayFunction -> $DisplayFunction]

Export["tau.txt", tauList, "Table"];
Export["calctau.txt", calctau, "Table"];
Export["sum2.txt", sum2List, "Table"];
Export["calcs.txt", calcsum2, "Table"];
```

A2 Simulations of time-resolved spectra in α-Al_2O_3:C inclusive of shallow traps

```
Remove["Global'*"];
programMain := (k1 = 8.617'/10^5; f = 10^10; E1 = 1.3'; E2 = .8;
  s1 = 10^13;
  s2 = 10^13; /[Alpha] = 1/10^14; /[Delta]1 = 1/10^12; /[Delta]2 = 1/
   10^14; /[Gamma] = 1/10^13; N1 = 10^13; N2 = 1*5.5*10^12;
  M1 = 10^14; M2 = 10^14; W = 1.0; C1 = 1.2*10^13; w1 = 1; w2 = 3000;
  w3 = 29; A1 = 1/10^10;

  solveDiffeq[n10_, n20_, m10_, m20_, nF0_, n3P0_, n1P0_, nc0_, R_,
    tfinal_, /[Beta]heat_, irrTemp_] :=
```

```
 Module[{t},
  sol = NDSolve[{Derivative[1][n1][
       t] == /[Alpha] (N1 - n1[t]) nc[t] -
       n1[t] s1 E^(-(E1/(k1 (273 + irrTemp + /[Beta]heat t)))),
     Derivative[1][n2][t] ==
      A1 (N2 - n2[t]) nc[t] -
       n2[t] s2 E^(-(E2/(k1 (273 + irrTemp + /[Beta]heat t)))),
     Derivative[1][m1][t] == /[Delta]1 (M1 - m1[t]) nc[t],
     Derivative[1][m2][t] == /[Delta]2 (M2 - m2[t]) nc[t],
     Derivative[1][nF][t] ==
      n3P[t] C1 E^(-(W/(
         k1 (273 + irrTemp + /[Beta]heat t)))) - /[Gamma] nF[t] nc[
         t] + w1 n1P[t],
     Derivative[1][n3P][t] ==
      w2 n1P[t] -
       n3P[t] C1 E^(-(W/(k1 (273 + irrTemp + /[Beta]heat t)))) -
       w3 n3P[t],
     Derivative[1][n1P][t] ==
      f + /[Gamma] nF[t] nc[t] - w1 n1P[t] - w2 n1P[t],
     Derivative[1][nc][t] == -Derivative[1][n1][t] - n2'[t] -
       Derivative[1][m1][t] - Derivative[1][m2][t] +
       Derivative[1][nF][t], n1[0] == n10, n2[0] == n20,
     m1[0] == m10, m2[0] == m20, nF[0] == nF0, n3P[0] == n3P0,
     n1P[0] == n1P0, nc[0] == nc0}, {n1, n2, m1, m2, nF, n3P, n1P,
     nc}, {t, 0, tfinal}, MaxSteps -> 50000]];

graphAllEleven[a_, tfinal_] :=
 Module[{n1graph, n3graph, n4graph, n5graph, n6graph, n7graph,
   n8graph, n9graph, ncgraph, graphs},
  ncgraph =
   Plot[Evaluate[nc[t] /. a], {t, 0, tfinal},
    AxesLabel -> {"t(s)", "nc(t)"},
    PlotStyle -> {RGBColor[1, 0, 0]},
    DisplayFunction -> $DisplayFunction, PlotRange -> All];
  n1graph =
   Plot[Evaluate[n1[t] /. a], {t, 0, tfinal},
    AxesLabel -> {"t(s)", "n1(t)"},
    PlotStyle -> {RGBColor[1, 0, 0]},
    DisplayFunction -> $DisplayFunction, PlotRange -> All];
  n3graph =
   Plot[Evaluate[m1[t] /. a], {t, 0, tfinal},
    AxesLabel -> {"t(s)", "m1(t)"},
    PlotStyle -> {RGBColor[1, 0, 0]},
    DisplayFunction -> $DisplayFunction, PlotRange -> All];
  n4graph =
   Plot[Evaluate[m2[t] /. a], {t, 0, tfinal},
    AxesLabel -> {"t(s)", "m2(t)"},
    PlotStyle -> {RGBColor[1, 0, 0]},
    DisplayFunction -> $DisplayFunction, PlotRange -> All];
```

```
  n5graph =
   Plot[Evaluate[nF[t] /. a], {t, 0, tfinal},
    AxesLabel -> {"t(s)", "nF(t)"},
    PlotStyle -> {RGBColor[1, 0, 0]},
    DisplayFunction -> $DisplayFunction, PlotRange -> All];
  n6graph =
   Plot[Evaluate[n3P[t] /. a], {t, 0, tfinal},
    AxesLabel -> {"t(s)", "n3P(t)"},
    PlotStyle -> {RGBColor[1, 0, 0]},
    DisplayFunction -> $DisplayFunction, PlotRange -> All];
  n7graph =
   Plot[Evaluate[n1P[t] /. a], {t, 0, tfinal},
    AxesLabel -> {"t(s)", "n1P(t)"},
    PlotStyle -> {RGBColor[1, 0, 0]},
    DisplayFunction -> $DisplayFunction, PlotRange -> All];
graphs =
 Print[Show[
   GraphicsGrid[{{n6graph}}, Spacings -> Scaled[0.03'],
    ImageSize -> 75 4], DisplayFunction -> $DisplayFunction]]];
;

graphTL[c_, e_] :=
 Module[{},
  tlList3 =
   Table[{t, First[Evaluate[n3P[t] w3 /. c]]}, {t, 0, e, 0.0001'}];
  tlList4 =
   Table[{First[Evaluate[n3P[t] w3 /. c]]}, {t, 0, e, 0.0001'}];
  ListPlot[tlList3, PlotJoined -> True,
   AxesLabel -> {"Time,sec", "TL(t)"},
   PlotStyle -> {RGBColor[1, 0, 0]}, PlotRange -> All,
   ImageSize -> 50 5, PlotLabel -> "TL vs Temperature"]];

initValues[b_, d_] :=
 Module[{}, n10 = Last[n1[d] /. b]; n20 = Last[n2[d] /. b];
  m10 = Last[m1[d] /. b]; m20 = Last[m2[d] /. b];
  nF0 = Last[nF[d] /. b]; n1P0 = Last[n1P[d] /. b];
  n3P0 = Last[n3P[d] /. b]; nc0 = Last[nc[d] /. b]];

irrTime = .2; f = 10^10; tfinal9 = irrTime; /[Beta]heat = 0;
solveDiffeq[n10, n20, m10, m20, nF0, n3P0, n1P0, nc0, f,
 tfinal9, /[Beta]heat, irrTemp]; sol9 = sol; graphTL[sol9, tfinal9];
 sn = Last[tlList4];

initValues[sol9, tfinal9];
irrTime = .2; f = 0; tfinal10 = irrTime; /[Beta]heat = 0;
solveDiffeq[n10, n20, m10, m20, nF0, n3P0, n1P0, nc0, f,
 tfinal10, /[Beta]heat, irrTemp]; sol10 = sol;
graphTL[sol10, tfinal10];
Print["....................", tloop];)
```

```
(*  ———————————————————————————————————* )

tstart = 0; tend = 300; tstep = 30;
n1List1 = {}; m1List1 = {}; m2List1 = {}; nFList1 = {}; n3PList1 = /
{}; n1PList1 = {}; ncList1 = {}; snList1 = {}; logsnList1 = {};

For [tloop = tstart, tloop <= tend, tloop += tstep,
 n10 = 0; n20 = .1*  N2; m10 = 0.1*  M1; nc0 = 0; m20 = 0;
 nF0 = m10 + m20 + n10 + n20 + nc0; n3P0 = 0; n1P0 = 0;
 irrTemp = tloop;

 programMain;
 AppendTo[snList1, {273 + irrTemp, First[sn]}];

 If[tloop == 0, a1 = tlList3]; If[tloop == 30, a2 = tlList3];
 If[tloop == 60, a3 = tlList3]; If[tloop == 90, a4 = tlList3];
 If[tloop == 120, a5 = tlList3]; If[tloop == 150, a6 = tlList3];
 If[tloop == 180, a7 = tlList3]; If[tloop == 210, a8 = tlList3];
 If[tloop == 240, a9 = tlList3]; If[tloop == 270, a10 = tlList3];
 If[tloop == 300, a11 = tlList3];

 ]

ListPlot[snList1, PlotJoined -> True,
 PlotRange -> {{300, 600}, {0, Automatic}}]
SetDirectory["C://Documents and Settings//Vasilis Pagonis//Desktop"];
Export["1.txt", snList1, "Table"];
data = {};
For[i = 1, i <= Length[a1], i++,
  dataRow = {Table[a1[[j]], {j, i, i}], Table[a2[[j]], {j, i, i}],
     Table[a3[[j]], {j, i, i}], Table[a4[[j]], {j, i, i}],
     Table[a5[[j]], {j, i, i}], Table[a6[[j]], {j, i, i}],
     Table[a7[[j]], {j, i, i}], Table[a8[[j]], {j, i, i}],
     Table[a9[[j]], {j, i, i}], Table[a10[[j]], {j, i, i}],
     Table[a11[[j]], {j, i, i}]}; AppendTo[data, Flatten[dataRow]];];
SetDirectory["C://Documents and Settings//Vasilis Pagonis//Desktop"];
Export["data.txt", data, "Table"];

bList = {a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11};

tauList = {};
Do[Clear[sum2, snlog, tlList3, tau, a22, x2, f1];
  tlList3 = bList&i1;;
  irrTemp = 30*  (i1 - 1);


  tlList3 = Drop[tlList3, 10];
  tlList3 = Drop[tlList3, -1950];
```

```
  ListPlot[tlList3, PlotRange -> All,
   PlotJoined -> True, ImageSize -> 72 3, PlotLabel -> {"t, TROSL"}];
  ListPlot[Transpose[{tlList3[[All, 1]], Log[tlList3[[All, 2]] ]}],
   ImageSize -> 72 5, PlotLabel -> {"t, Log(TROSL"}];
  (*  next line takes log of data* )
  snlog = Transpose[{tlList3[[All, 1]], Log[tlList3[[All, 2]] ]}];
  (*  next line fits a straight line of the data* )
  a22[x2_] = Fit[snlog, {1, x2}, x2]; Print["slope=", a22[1] - a22[0]];
  (*  next line calculates tau from the slope of the data* )
  tau = 10^3/(a22[1] - a22[0]);
  Print["tau=", 10^3/(a22[1] - a22[0]), " millisecs"];
  (*  next line adds tau value to a list, so it can be plotted* )
  AppendTo[tauList, {irrTemp, -tau}];
  Print[irrTemp, "deg C"];, {i1, 1, 11}];
grtau = ListPlot[tauList,
   PlotRange -> {{0, Automatic}, {0, Automatic}},
   PlotLabel -> {"tau, milliseconds"}, PlotJoined -> False,
   ImageSize -> 72 3];
Export["tau.txt", tauList, "Table"];
Null
```

References

[1] Pagonis V, Ankjærgaard C, Murray A S, Jain M, Chen R, Lawless J and Greilich S 2010 Modelling the thermal quenching mechanism in quartz based on time-resolved optically stimulated stimulated luminescence *J. Lumin* **130** 902–9

[2] Pagonis V, Chen R, Maddrey J W and Sapp B 2011 Simulations of time-resolved photo-luminescence experiments in α-Al_2O_3:C *J. Lumin.* **131** 1086–94