(\*The following program is based on a global optimization algorithm, using reverse fitting of the experimentally obtained PA power-dependence curve data to determine the true PA response curve of prepared nanoparticles. The software used is 1stOpt 10.0 from 7D-soft high technology Inc.\*)

//Algorithm=UGO1;

//QuickReg = 2;

Parameter a1=[-50,-20],a2=[-400,-300],a3=[-20,0],b1=[5,15],b2=[100,160],b3=[2,20],con1=[0,1];

Variable x, y;

Constant areaofspot=2.826E-9,con2=171394,

ConstStr p1=(a2-a1)/(b1-b2),

p2=(a3-a2)/(b2-b3),

ppp=log(100000\*x\*areaofspot\*(1E+9)\*exp(-0.000020989278\*(r^2))/con2),

realpowerdependence=if(ppp<p1,a1+b1\*ppp,if(ppp>=p2,a3+b3\*ppp,a2+b2\*ppp));

Function y=log(int(con1\*(10^realpowerdependence)\*6.28\*r,r=0,2000));

2<p2<3;

2<p1<3;

p1<p2;

Data;

// x y

534.65525 4.32222

543.24325 4.39794

551.3705 4.47712

556.4055 4.50515

561.5925 4.54407

564.224 4.60206

570.41325 4.65321

572.1185 4.69897

584.50175 4.81291

589.38 4.8451

591.47 4.90309

593.42 5.0499

595.59775 5.20412

597.3 5.353

598.96075 5.47712

600.154 5.584

601.352 5.7031

602.445 5.79758

603.426 5.92

604.532 6.0436

608.42275 6.25836

612.06125 6.39244

617.01075 6.45332

627.38475 6.56348

650.218 6.70672

658.35 6.75051

673.28875 6.81657

686.603 6.87674

704.63875 6.96095

723.05925 7.05154

741.40375 7.12767

759.981 7.20041

773.68475 7.25091

790.172 7.33385

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(\*Based on the intrinsic response curve of PA nanoparticles obtained from the above program, the following program uses the Monte Carlo method to simulate the process of Gaussian beam scanning for super-resolution imaging of PA nanoparticles. The software used is Mathematica 12.0 from Wolfram Research\*)

ClearAll["Global`\*"]

wavelength=1064;

NA=1.45;

areaofspot=0.2826\*10^(-8)\*(wavelength/1064)^2;(\*cm^2\*)

powerdensity=655.1;(\*kW/cm2\*)

power=powerdensity\*areaofspot\*10^9;(\*uW, real inputpower\*)

laserprofileorigin[r\_]:=(2\*BesselJ[1,2\*Pi\*NA\*r/wavelength]/(2\*Pi\*NA\*r/wavelength))^2;

data=Table[{i,laserprofileorigin[i]},{i,1,5000,1}];(\*get data of laser profile\*)

gaussianModel=a Exp[-(x^2)/(2 b^2)];

fit=NonlinearModelFit[data,gaussianModel,{a,b},x];

laserprofile[x\_]:=fit[x]/fit[0];(\*the profile was fitted with real gaussian distribution\*)

constant=Integrate[laserprofileorigin[r]\*2Pi\*r,{r,0,50000000}];

powerdensityprofile[rr\_]:=100000\*power\*laserprofile[rr]/constant;(\*uW/nm^2 to kW/cm^2\*)

a1=-32.586;

a2=-412.245;

a3=-17.017;

b1=11.259;

b2=138.539;

b3=7;

con1=0.0976386019;

p1=(a2-a1)/(b1-b2);

p2=(a3-a2)/(b2-b3);

p1

p2

f[x\_]:=Piecewise[{{a1+b1\*x,x<=p1},{a2+b2\*x,p1<x<=p2},{a3+b3\*x,x>=p2}}];

Plot[f[x],{x,2.5,4}];

powerdependentdatareal=Table[{10^x,10^f[x]},{x,1,4,0.01}];

ListLogLogPlot[powerdependentdatareal]

powerdependentdatafunction=Interpolation[powerdependentdatareal,InterpolationOrder->1];

nonlinearemissionfunction[r\_]:=powerdependentdatafunction[powerdensityprofile[r]];

LogLogPlot[powerdependentdatafunction[p],{p,100,10000}];

Plot[nonlinearemissionfunction[r],{r,0,1000}];

Plot[powerdependentdatafunction[p],{p,0,1000}];

Plot[powerdensityprofile[r],{r,0,1000}]

Plot[laserprofile[r],{r,0,1000}];

Plot[{laserprofileorigin[r],laserprofile[r]},{r,0,1000},PlotRange->Full];

power;

NIntegrate[powerdensityprofile[r]\*2Pi\*r/100000,{r,0,500}];

Plot[powerdensityprofile[r],{r,0,500}];

Plot[nonlinearemissionfunction[r],{r,0,100}];

LogPlot[powerdependentdatafunction[p],{p,400,5000}]

power/areaofspot;

Integrate[(powerdensityprofile[r]/100000)\*2\*Pi\*r,{r,0,3000}];

centerofparticle1={55,-33};

centerofparticle2={-66,45};

particlediameter=27;(\*nm\*)

points1=200;

plotrange={{-150,150},{-150,150}};

region1=RegionUnion[Disk[centerofparticle1,particlediameter/2],Disk[centerofparticle2,particlediameter/2]];

pts1=RandomPoint[region1,points1];

Graphics[Point[pts1],Frame->True,AxesOrigin->{0,0},Axes->True,PlotRange->plotrange]

region2=Rectangle[{plotrange[[1,1]],plotrange[[2,1]]},{plotrange[[1,2]],plotrange[[2,2]]}];

points2=10000;(\*Scanning points\*)

pts2=RandomPoint[region2,points2];

Graphics[Point[pts2],Frame->True,AxesOrigin->{0,0},Axes->True];

intensitymapping=Table[{pts2[[j]],Total[Table[nonlinearemissionfunction[RegionDistance[Point[pts2[[j]]],pts1[[i]]]],{i,1,points1}]]}//Flatten,{j,1,points2}];

customColorFunction[z\_] := Blend[{{0, RGBColor[26/255, 40/255, 71/255]}, {0.7, RGBColor[33/255, 166/255, 117/255]}, {1, RGBColor[255/255, 241/255, 67/255]}}, z];

ListDensityPlot[intensitymapping, PlotRange -> Full, ColorFunction -> customColorFunction, ColorFunctionScaling -> True]