Supporting Information 3: Simulations to aid in the design of microbes for synthesis of metallic nanomaterials

Kyle L. Naughton[1], James Q. Boedicker[1,2]

[1] Department of Physics and Astronomy, University of Southern California, Los Angeles, CA, USA.

[2] Department of Biological Sciences, University of Southern California, Los Angeles, CA, USA.

Example Code for Figure 2 in the Main Text: As(V) Reduction

```
% ================= Simulations to aid in the design of microbes for
synthesis of metallic nanomaterials ================ %
% Naughton and Boedicker 2021

% ================= ABOUT THIS CODE & THEORY ================ %
%{
Classical Nucleation Theory (CNT) dictates that NP formation can be broken up
into 3 parts:
(i) chemical reaction
(ii) nucleation
(iii) growth
The whole process is driven by the supersaturation "S" (oversaturation) of
atomic monomers that compose the material.
The Gibbs free energy, which directs nucleation, depends primarily on the
interfacial free energy "gamma"
For a detailed explanation of theory see Thanh, Maclean, & Mahiddine Chem Rev
2014, https://pubs.acs.org/doi/10.1021/cr400544s

This code appends biological perturbation to CNT
- Compartmentalization into three regions (extracellular, periplasm,
cytoplasm)
- Transport between those three regions, where transport is dicated by
Michaelis-Menton (MM) kinetics
https://en.wikipedia.org/wiki/Michaelis%E2%80%93Menten_kinetics
- Reduction of pre-cursor materials (e.g. of Pd(II) -> Pd(0)), which is
described by a 1st order rxn (d[X(0)]/dt = +n_reductases*k_reduction*[X(II)]
- Nucleation, which is driven by a reduction in the interfacial free energy
"gamma" and the availability of nucleation sites "nuc_sites" in each
  region. The variable "theta" accounts for the potential reduction of the
interfacial energy via heterogeneous nucleation on biomolecules.

One of the interesting perturbations added is that homogenous, spontaneous
nucleation of NP competes with heterogenous nucleation, which occurs on
defect or, importantly, a metal binding peptide (e.g. CDS7 from Peelle et al
Acta Biomater 2005)

Another key ingredient is the transport across the membranes. The rate is
given by "Vmax" and the rate can be controlled for each membrane for each
chemical species. That means that Zn2+ travels differently than S2- and that
both species travel differently across each membrane.
```

```
See the variable "Vmax", which has two columns: 1 for extracell-periplasm
(OM) barrier and 2 for the periplasm-cytoplasm (IM) barrier



%}

% ================== DETAILS ABOUT THIS CODE ================== %
%{
* most units are moles/L, meters, seconds, and grams

VARIABLES
h = size of time step (in seconds), "h" is from Euler forward literature
Nsteps = number of hours of the simulation
i = number of steps

l = region (extra, peri, cyto)
k = homogenous or heterogenous nucleation (1, 2)
y(i,s,l,k) = molar concentration of species 1,2,3,4,5
Nanoparticle Radius(i,s,l,k) = radius, each column is a time step containing
all radii
Nanoparticle Number(i,s,l,k) = number, each column is a time step containing
number of all radii


The code used the Euler forward method and can divided into 4 steps
Step 0: Time, Concentrations, Parameters defined
Step 1: The concentration of each species is updated, transport and chemical
reduction occur
Step 2: Nucleation occurs. Homogenous and heterogenous (heterogenous
nucleation requires unoccupied nucleation sites > 0) occur
Step 3: Nanoparticle growth occurs.


The code outputs a big array of the number of NPs "Number" in each region "l"
and the radius of NPs "Radius"
The data can be cleaned up to erase NP that are smaller than an atom or when
there is less than 1 of something
The data are binned into nbins in an array called "ed" for each region. A
histogram can be displayed (figure 696, figure 601)
A Gaussian Fit is attempted to the binned data
The concentration of each species can be displayed (Figure 3)
The nucleation rate (dN) and total number of nuclei can be displayed (Figure
76)
The code prints the key parameters used in the terminal and lets you know
what you did or did not turn of


TURNING ON or OFF PERTUBATIONS
Nucleation, transport, and single-metal reduction can be turned on or off
using logical Matlab structures
Example: "transport.true = 1" turns on transport, otherwise all the
parameters are set to 0
```

Example: "metal_reduction.true = 1" turns of reduction of a single metal species, like Pd(II)
Example: "hetero_nucleation.true = 1" turns on nucleating sites, which you define by changing "theta" and "nuc_sites"
* For nucleation you can set an upper limit on the number of sites available with "nucleation.enforce_number" (assume peptides keep NP attached) or allow for NP to detach, which
effectively means there is no limit on the number of nucleation sites

CORE PARAMETERS

"gamma" (interfacial energy)
"k1" (the chemical reaction rate)
"C0_sat" (the solubility of the material)
"kgr" (the basal growth rate constant of the material)
"Vmax" (the transport rate of the materials, when transport on)

the reduction rate of X to X' with "nr_X" or "kr_X" (number and rate)

CONCENTRATION
Enter the concentration in millimoles of your chemicals. code automatically converts "1.0" to "0.001" moles/L

l=1
l=2 l=3
y(time step, type of chemical, compartment) = [initial conc in extracellular space, initical Conc in periplasm, initial conc in cytoplasm]

TRANSPORT
Because the system has different volumes and concentrations, Michaelis-Menton (MM) kinetics works best.
Eqn ~ (N_trans + leaky)*Vmax*(y2-y1)/(vol(ll)*(Km+y1+y2))
The paramters are Vmax (max rate), Km (conc of substrate at half-max rate),
N_trans, the relative number of transporters = 0 -> 1
leaky, leaky transport rate = 0 -> 1, which can be turned off

NUCLEATION
A key parameter is gamma. In the presence of unoccupied nucleation sites, gamma is reduced for heterogeneous nucleation by a factor theta

REDUCTION
Pre-precursors are y(:,4,:) and y(:,5,:), which can be reduced to y(:,1,:) and y(:,2,:) respectively
y4 -> y1: driven by Vmax(6), Km(6), placed in the periplasm by writing dY/dt = ... + Vmax(6)*y(i,4,2)/(Km(6)+y(i,4,2)) where "y(i,4,2)" refers to the i-th time steps, species 4 (e.g. As(V)) and region 2, periplasm

```
TITRATION/DROP
In real experiments you can add chemicals at different times
    if titration.true = 1 then you can add a chemical species at some time
along the simulation, which you define.

After everything is calculated you can check the Analysis section for default
figures and other figures
To uncomment something in MatLab, delete the "{" after the % in the green
part

%}

% ================== EXAMPLE: As2S3 NP in the Periplasm from Figure 2G
================== %
%{
In this example, a cell in stationary phase which does not grow synthesized
As2S3 nanoparticles in the periplasm through the expression of As(V)
reductases, outermembrane transporters for As(V), and the expression of a
putative nucleating peptide that binds As2S3.
The reaction takes place at temperature T=37C, the time steps are h=10s, and
the reaction occurs for 16 hours.

The parameters for the material are as follows for the simulation
    - volume of As2S3 v0 = 11.910*10^-29 m^3
    - interfacial energy gamma = 0.125 J/m^2
    - basal growth rate kgr = 1*10^-5 m/s
    - solubility C0_sat = 5*10^-8 moles/L

Cellular pertubrations that are ON
    - Transport of As(V), As(III), and S2- acros the outter and inner
membrane
    - Expression of 10^4 nucleating peptides in the periplasm that reduce the
barrier to nulceation
    - Expression of periplasmic, innermembrane-bound As(V) reductase that
converts As(V) to As(III)
Expected outputs are 3 figures with the Radius probability distribution,
number of nanoparticles, and concentrations of reactants

%}

% ----- Start of Code -----

close all  % closes all figures

% --------- DEFINE TIME STEPS  ---------
h = 10; % time step in seconds
Nsteps = 16; % number of hours of simulated time
Nsteps = Nsteps*3600/h; % convert to steps
time = (1:Nsteps)*h/3600; % time array in hours units for plotting
u = 'Time [hr]'; % for labeling plots when they are output
formatSpec = '%.1e'; % number of signifigant figures to display in Command
Window
```

```
% --------- INITIALIZE EMPTY ARRAYS  ---------
Radius = zeros(round(Nsteps),round(Nsteps),3,2); % big array with all the
radii at all time
Number = zeros(round(Nsteps),round(Nsteps),3,2); % array with Number at all
time
totNumber = zeros(round(Nsteps),3,2); % array with Number at all time
dN = zeros(round(Nsteps),3,2); %
rcrit = zeros(round(Nsteps),3,2);
y = zeros(round(Nsteps),5,3); % molar concentration of precursor, monomer,
monomer to nuclei, monomer to growth
dydx = zeros(round(Nsteps),5,3); % rate of consumption of chemicals
drdt = zeros(round(Nsteps),round(Nsteps),3,2); % rate of change of radius
S = zeros(round(Nsteps),3); % oversaturation at each time step
drop = struct;
transport = struct;
nucleation = struct;

% --------- BASIC PARAMETERS ---------
% Volume of extracellular, periplasmic, cytoplasmic space. Current value for
cell culture at OD ~ 1.0 , Volume of cells ~1fL and 1/3 of that is
periplasmic BNID 104052, PMID 824274 p.334
vol = [100e-15; 1/3*1e-15; 2/3*1e-15]; % in L units

% Chemical rxn rate
k1 = 10*10^-3; %  L/mole*s units

% Volume of atomic monomer, user can select different materials (or add them)
type = 4; % Type of material being synthesized: 1= ZnS 2=CdS 3=Pd 4=As2S3
species_volume = [3.9500*10^-29;  % ZnS
                  4.9769*10^-29;  % CdS
                  1.8141*10^-29;  % Pd
                  11.910*10^-29]; % As2S3 246 g/mole, density = 3.43 g/cm^3
v0 = species_volume(type);

% Radius of atomic monomer
r0 = (3*v0/(4*pi))^(1/3); % radius of ZnS "atom" (bond length ZnS is 2.3≈,
cubic face of Zn blende is 5.4≈

% Temperature of reaction (input in C, converted to K, then kT calculated)
T = 37; % temperature in celsius
T = T + 273.15; % temperature in Kelvin 298 is room temp, 310.15 is 37
kT = 1.38*10^-23 * T; % Boltzman's constant * temperature in Kelvin

% Diffussion coefficient of atomic monomer
D = 10^-9; % esitmated diffussion coefficient m^2/s of typical O2,C02,KCl in
H20, BioNumber 104440,102625 http://book.bionumbers.org/what-are-the-time-
scales-for-diffusion-in-cells/

% Avogadro's Number
Na = 6.022*10^23; % Avogadros number, atoms/mole

% Viscosity of medium and its temperatue dependence
viscosity = (2.414e-5)*10^(247.8/(T-140)); % viscocity versus temperature
```

```
% parameter from Talapin et al 2002 that appears in dR/dt growth equation
alpha = 0.5; % for growth rate from Talapin et al 2002

% Basal growth rate of material. NOTE: You can change for homo and
heterogenous nucleation+growth
kgr = [1e-5; 1e-5]; % growth rate for homogenous and heterogenous nucleation
m/s, value similar to that used in Chen et al 2015

% Interfacial energy of C0
gamma = 0.125; % used in Talapin et al 2002

% Solubility of C0
C0_sat = 5e-8; % Saturation conc./ solubility of ZnS
%}


% --------------- PRECURSOR PARAMETERS in milli-moles/L ---------------
y(1,1,:) = [0.0; 0.0; 0.0]*1e-3; % moles of precursor y1, e.g. Zn2+,
y(1,2,:) = [1.0; 0.0; 0.0]*1e-3; % moles of precursor y2, e.g. S2-
y(1,3,:) = [0.0; 0.0; 0.0]*1e-3; % moles of monomer, e.g. ZnS0
y(1,4,:) = [1.0; 0.0; 0.0]*1e-3; % moles of pre-precursor
y(1,5,:) = [0.0; 0.0; 0.0]*1e-3; % moles of pre-precursor


% --------------- TRANSPORT PARAMETERS ---------------
transport.true = 1; % make == 1 to allow metal ion transport
transport.leaky = 0; % make == 1 to allow leaky transport in the absence of
N_trans, the number of transport proteins.
if transport.true == 1
     % relative number of transporters per membrane (0 to 1 for min to max)
     % Column 1 is outer membrane, column 2 is inner memrane
     N_trans = [1 0.2; % for Species 1
                1 0.8; % for Species 2
                0 0; % for Species 3
                1 0.2; % for Species 4
                1 0.8]; % for Species 5
% Example: N_trans(5,1) is the number/max for Species 5 bt extracell and
periplasm on outer membrane
else
     N_trans = zeros(5,2); % transport off (leaky still possible)
end

% Assume Michaelis-Menton Kinetics, which uses Vmax (max rate) and Km (conc.
of half-maximum rate of substrate/ precursor)
Vmax = [1.995e-23; % ion transport rate Y1, e.g. Zn2+ or Cd2+
        8e-23; % ion transport rate Y2, e.g. S2-
        0; % ion transport Y3, the insoluble metal species
        2e-23; % ion transport pre-precursor of Y4, e.g. As(V)
        6e-22]; % ion transport pre-precursor of Y5, e.g. thiosulfate

Km = [0.71e-6; % mole/L
      1e-6; % S2-
      0; % insoluble monomers of, for example, ZnS[0] or Pd[0]
      0.71e-6; % e.g. As(V)
      0.71e-6]; % e.g. thiosulfate
```

```
% LEAKY TRANSPORT PARAMETERS ---------------
% Note: In general, transport can be leaky despite no transport proteins
being expressed. This gives the user the ability to allow leaky transport
% in addition to allowing transport proteins to be expressed.
if transport.leaky == 1
     leaky = [0 0; % leaky transport for Species 1
              0 0; % leaky transport for Species 2
              0 0; % leaky transport for Species 3
              0 0; % leaky transport for Species 4
              0 0]; % leaky transport for Species 5
% Example: leaky(1,2) is the leaky transport rate (as % of maximum rate) for
Species 1 into cytoplasm from periplasm acoss inner membrane
else
    leaky = zeros(5,2);
end


% ---------------- METAL REDUCTION PARAMETERS ---------------
% THIOSULFATE REDUCTION PARAMETERS ---------------
nred_S = [0; % empty
          1]; % relative expression of reducatse for control of thiosulfate
reduction (default = 1)
kred_S = [0; % empty
          1e-2]; % rate of thiosulfate reduction


% ARENSITE REDUCTION PARAMETERS ---------------
% number of reducatses, set 0 to 1 where 1 is maximum and >1 is over
expression
nred_As = [0;  % As(V) reductase in extracellular space
           1;  % As(V) reductase in periplasm
           0]; % As(V) reductase in cytoplasm
kred_As = [0; % reduction outside
           25e-5; % reduction periplasmic
           0];% reduction cytoplasmic

% PALLADIUM REDUCTION PARAMETERS ---------------
metal_reduction.true = 0;
% automatically turn off if youre not dealing w Pd
if type ~= 3
    metal_reduction.true = 0;
end
if metal_reduction.true == 1
    kred = [5e-5; % reduction outside
            5e-5; % reduction periplasmic
            0];% reduction cytoplasmic
    nred = [1;
            0;
            0];
end


% ---------------- NUCLEATION PARAMETERS ---------------
hetero_nucleation.true = 1;
```

```
if hetero_nucleation.true == 1
    nuc_sites = [0; 1e4; 0];% number of nucleating sites
    theta = 0.2; % amount by which gamma reduced = gamma/gamma_0
else
    nuc_sites = [0; 0; 0];
    theta = 1.0;
end
initial_nuc_sites = nuc_sites;
nucleation.enforce_number = 1; % this turns ON (1) or OFF (0) a finite number
of sites
% note that if = 1, heteronucleation stops when sites fill up

i = 1; % time (step) variable, increases each step by +1

% --------------- WARNING SWITCHES ---------------
ww = [0;0;0]; % array for different volumes (extra, peri, cyto) incase any
molar concentrations become negative, which is unphysical

% print some basic inputs about simulation
fprintf('Some key parameters: \n')
fprintf('k1=%.1g  gamma=%.3g  kgr(1)=%.1g  kgr(2)=%.1g  C0_sat=%.1g  \n
\n',[k1 gamma kgr(1) kgr(2) C0_sat])

fprintf('Initial concentration of chemicals in extracellular space: \n')
fprintf('[As(III)] = %.2fmM  S2- = %.2fmM  [As(V)] = %.2fmM \n',
[y(1,1,1)*1e3  y(1,2,1)*1e3  y(1,4,1)*1e3])


% ---------------------------- Start of Calculation ------------------------
--
while  i < Nsteps
    % --------------- START CALCULATING THINGS ---------------

    % Recall y(time, type, compartment) is the concentration (in moles/L) of
chemical species where time is the time step of the simulation, type is
    % the type of chemical species underscrutinity, and compartment is the
extracellular space, the periplasm, or the cytoplasm.

    % Update moles
    y(i+1,1,:) = y(i,1,:) + h*dydx(i,1,:); % e.g. As(III)
    y(i+1,2,:) = y(i,2,:) + h*dydx(i,2,:); % e.g. S2-
    y(i+1,3,:) = y(i,3,:) + h*dydx(i,3,:); % e.g. As2S3
    y(i+1,4,:) = y(i,4,:) + h*dydx(i,4,:); % e.g. As(V)
    y(i+1,5,:) = y(i,5,:) + h*dydx(i,5,:); % e.g. thiosulfate

    % Chemical Reaction & Transport
    % Precursor 1, e.g. As(III)
    dydx(i+1,1,:) = [-k1*y(i,1,1)*y(i,2,1) - (N_trans(1,1) +
leaky(1,1))*Vmax(1)*(y(i,1,1)-y(i,1,2))/(vol(1)*(Km(1)+y(i,1,1)+y(i,1,2))) +
nred_As(1)*kred_As(1)*y(i,4,1);
                    -k1*y(i,1,2)*y(i,2,2) + (N_trans(1,1) +
leaky(1,1))*Vmax(1)*(y(i,1,1)-y(i,1,2))/(vol(2)*(Km(1)+y(i,1,1)+y(i,1,2))) -
(N_trans(1,2) + leaky(1,2))*Vmax(1)*(y(i,1,2)-
y(i,1,3))/(vol(3)*(Km(1)+y(i,1,3)+y(i,1,2))) +
```

```
nred_As(2)*kred_As(2)*y(i,4,2); % rxn + diffusion into external + diffsion in
cytoplasm
                    -k1*y(i,1,3)*y(i,2,3) + (N_trans(1,2) +
leaky(1,2))*Vmax(1)*(y(i,1,2)-y(i,1,3))/(vol(3)*(Km(1)+y(i,1,2)+y(i,1,3))) +
nred_As(3)*kred_As(3)*y(i,4,3)];

    % Precursor 2, e.g. S2-
    dydx(i+1,2,:) = [-k1*y(i,1,1)*y(i,2,1) - (N_trans(2,1) +
leaky(2,1))*Vmax(2)*(y(i,2,1)-y(i,2,2))/(vol(1)*(Km(2)+y(i,2,1)+y(i,2,2)));
                    -k1*y(i,1,2)*y(i,2,2) + (N_trans(2,1) +
leaky(2,1))*Vmax(2)*(y(i,2,1)-y(i,2,2))/(vol(2)*(Km(2)+y(i,2,1)+y(i,2,2))) -
(N_trans(2,2) + leaky(2,2))*Vmax(2)*(y(i,2,2)-
y(i,2,3))/(vol(2)*(Km(2)+y(i,2,3)+y(i,2,2))) +  nred_S(2)*kred_S(2)*y(i,5,2);
% rxn + diffusion into external + diffsion in cytoplasm
                    -k1*y(i,1,3)*y(i,2,3) + (N_trans(2,2) +
leaky(2,2))*Vmax(2)*(y(i,2,2)-y(i,2,3))/(vol(3)*(Km(2)+y(i,2,2)+y(i,2,3)))];

    % Monomer 1, e.g. As2S3
    dydx(i+1,3,:) =  [k1*y(i,1,1)*y(i,2,1);
                      k1*y(i,1,2)*y(i,2,2);
                      k1*y(i,1,3)*y(i,2,3)];

    % pre-precursor 1, turns into y1, e.g. As(V) -> As(III)
    dydx(i+1,4,:) =  [-(N_trans(4,1) + leaky(4,1))*Vmax(4)*(y(i,4,1)-
y(i,4,2))/(vol(1)*(Km(4)+y(i,4,1)+y(i,4,2))) -
nred_As(1)*kred_As(1)*y(i,4,1);
                     (N_trans(4,1) + leaky(4,1))*Vmax(4)*(y(i,4,1)-
y(i,4,2))/(vol(2)*(Km(4)+y(i,4,1)+y(i,4,2))) - (N_trans(4,2) +
leaky(4,2))*Vmax(4)*(y(i,4,2)-y(i,4,3))/(vol(3)*(Km(4)+y(i,4,2)+y(i,4,3))) -
nred_As(2)*kred_As(2)*y(i,4,2); % rxn + diffusion into external + diffsion in
cytoplasm
                      (N_trans(4,2) + leaky(4,2))*Vmax(4)*(y(i,4,2)-
y(i,4,3))/(vol(3)*(Km(4)+y(i,4,2)+y(i,4,3))) -
nred_As(3)*kred_As(3)*y(i,4,3)];

    % pre-precursor 1, turns into y2, e.g. thiosulfate -> S(2-)
    dydx(i+1,5,:) = [-(N_trans(5,1) + leaky(5,1))*Vmax(5)*(y(i,5,1)-
y(i,5,2))/(vol(1)*(Km(5)+y(i,5,1)+y(i,5,2)));
                    (N_trans(5,1) + leaky(5,1))*Vmax(5)*(y(i,5,1)-
y(i,5,2))/(vol(2)*(Km(5)+y(i,5,1)+y(i,5,2))) - (N_trans(5,2) +
leaky(5,2))*Vmax(5)*(y(i,5,2)-y(i,5,3))/(vol(3)*(Km(4)+y(i,5,2)+y(i,5,3))) -
nred_S(2)*kred_S(2)*y(i,5,2); % rxn + diffusion into external + diffsion in
cytoplasm
                    (N_trans(5,2) + leaky(5,2))*Vmax(5)*(y(i,5,2)-
y(i,5,3))/(vol(3)*(Km(5)+y(i,5,2)+y(i,5,3)))];

    % Supersaturation
    S(i,:) = y(i,3,:)/C0_sat;

    % ==================== Nucleation ====================
    for ll = 1:3 % loop through regions
        if real(-log10(y(i,3,ll)))>9 % error check
            y(i,3,ll) = 0.0;
        end
        if S(i,ll) > 1 % check if anything can happen
            % homogenous nucleation always occurs if S > 1
```

```
            Rcap = 2*gamma*v0/kT; % Capillary radius
            rcrit(i,ll,1) = Rcap/log(S(i,ll)); % radius of new nuclei
            % Prevent non-physical nuclei forming
            if rcrit(i,ll,1) <= r0
                rcrit(i,ll,1) = r0; % if rcrit < radius of single atoms,
force nuclei to be a single atom
                %dN(i,ll,1) = 0;
            end
            vcrit_homo = (4*pi/3)*rcrit(i,ll,1)^3; % volume of new nuclei
            dG = 16*pi*gamma^3*v0^2/(3*(kT)^2*log(S(i,ll))^2);

            % NUCLEATION RATE (HOMOGENOUS)
            knuc_homo =
1000*(y(i,3,ll)*Na*vcrit_homo)*(kT/(3*pi*(2*r0)^3*viscosity))*exp(-dG/kT);
            dN(i,ll,1) = h*(knuc_homo);
            % check if heterogenous nucleation can occur
            % You can allow unbounded nucleation to occur (assuming NP
detatch and new NP can form)
            % or you can stop nucleation after all sites fill-up (assume NP
do not detach from nucleating site).
            if nucleation.enforce_number == 1
                nuc_sites(ll) = initial_nuc_sites(ll) - totNumber(i,ll,2);
            end

            if nuc_sites(ll) > 0
                    Rcap = 2*theta*gamma*v0/kT; %
                    rcrit(i,ll,2) = Rcap/log(S(i,ll)); % radius of new nuclei
                    if rcrit(i,ll,2) <= r0
                        rcrit(i,ll,2) = r0;% if rcrit < radius of single
atoms, force nuclei to be a single atom
                    end
                    vcrit_hetero = (4*pi/3)*rcrit(i,ll,2)^3; % volume of new
nuclei
                    dG_hetero =
16*pi*(theta*gamma)^3*v0^2/(3*(kT)^2*log(S(i,ll))^2);

                    % NUCLEATION RATE (HETEROGENEOUS)
                    knuc_hetero =
1000*(y(i,3,ll)*Na*vcrit_hetero)*(kT/(3*pi*(2*r0)^3*viscosity))*exp(-
dG_hetero/kT);
                    dN(i,ll,2) = h*(knuc_hetero);
                    if nucleation.enforce_number == 1
                        if dN(i,ll,2) > nuc_sites(ll)
                            fprintf('Nucleation Site Number Enforced i=%d dN
= %.2f \n', [i,dN(i,ll,2)])
                            dN(i,ll,2) = nuc_sites(ll);
                        end
                    end
            else
                   nuc_sites(ll) = 0;
                   rcrit(i,ll,2) = 0;
                   vcrit_hetero = 0;
                   dN(i,ll,2) = 0;
            end % end if nuc sites
        else % if S < 1 set values to zero, as no nucleation occured
            rcrit(i,ll,1) = 0;
```

```
            rcrit(i,ll,2) = 0;
            vcrit_homo = 0;
            vcrit_hetero = 0;
            dN(i,ll,1) = 0;
            dN(i,ll,2) = 0;
        end % end of if S > 1

        % == Update moles of available monomer
        y(i+1,3,ll) = y(i+1,3,ll) - ((dN(i,ll,1)*vcrit_homo +
dN(i,ll,2)*vcrit_hetero)/(v0*Na))/vol(ll); % gain in free monomer from
chemistry - loss due to nucleation

        % Warnings
        if y(i+1,3,ll) < 0  && ww(ll) == 0
            warning('\n\n WARNING: (@Nucleation) Negative Molar Concentration
\n i=%d ll=%d Conc=%fuM [Nucl]=%fuM
\n\n',[i,ll,y(i+1,3,ll)*1e6,((dN(i,ll,1)*vcrit_homo +
dN(i,ll,2)*vcrit_hetero)/(v0*Na))*1e6])
            if real(-log10(y(i+1,3,ll))) > 9
                y(i+1,3,ll) = 0;
            else
                break
            end
            ww(ll) = 1;
        end % end of if conc < 0

        % === update numbers of NP
        for k = 1:2




            % ----- add nuclei with radius Rcrit
            if dN(i,ll,k) > 0
                Radius(i+1,i+1,ll,k) = rcrit(i,ll,k); % add new nuclei of
radius Rcrit (typically ~0.5nm (S=1000) - 1 nm (S=5))
                Number(i+1,i+1,ll,k) = dN(i,ll,k); % add new nuclei
            else
                Radius(i+1,i+1,ll,k) = 0; %
                Number(i+1,i+1,ll,k) = 0; %
            end % end of if dN > 0
            % ---- carry over values from previous time step
            Radius((1:i),i+1,ll,k) = Radius((1:i),i,ll,k);% % NP radius
            Number((1:i),i+1,ll,k) = Number((1:i),i,ll,k); % number conserved
step to step, does not change
            totNumber(i+1,ll,k) = sum(Number((1:i),i+1,ll,k)); % we should
account for new ones yes?
        end % end of for k = 1:2
    end % end of for ll  = 1:3

    % ====================== GROWTH ======================
    for ll = 1:3 % loop through regions
        monomer_to_growth = 0; % local variable to account for reduction in
monomer conc due to growth
```

```
        monomer_from_dissolve = 0; % local variable to account for increase
in monomer conc due to dissolution
        if S(i,ll) > 1 % can only grow if S > 1 (available monomer)
            for j = 1:i % loop through each family "j" of NPs that exist in
solution at time "i"
                for k = 1:2 % homo and hetero growth
                    if Number(j,i+1,ll,k) > 0 % there must be a particle
existing for growth

                        % define Rcap for homo or hetero
                        if k == 1 % if particle nucleated on site, change
Rcap (avoids premature dissolving)
                            Rcap = 2*gamma*v0/kT; % capillary radius
                        else
                            Rcap = 2*theta*gamma*v0/kT; % capillary radius
                        end

                        % ---- GROWTH RATE EQUATION -----
                        drdt(j,i,ll,k) = 1000*(v0*Na)*C0_sat*D*(S(i,ll)-
exp(Rcap./Radius(j,i,ll,k))) /
(Radius(j,i,ll,k)+(D/kgr(k))*exp(alpha*Rcap/Radius(j,i,ll,k))); % Eqn 14 from
Talapin

                        % if dR/dt, dissolve and free monomers from
dissolving
                        if drdt(j,i,ll,k) < 0 %
                            monomer_from_dissolve = monomer_from_dissolve +
Number(j,i+1,ll,k)*((4*pi)*(Radius(j,i,ll,k)^3)/3)/(v0*Na*vol(ll)); % moles
of C0 freed from disintegration
                            Number(j,i+1,ll,k) = 0.0; % disintegrate
particles in that family
                            Radius(j,i+1,ll,k) = 0.0; % set radius of that
disintegrated family to zero
                        else
                            Radius(j,i+1,ll,k) = Radius(j,i,ll,k) +
h*drdt(j,i,ll,k); % update radius
                        end % end of if drdt < 0

                        % Calulcate increase in volume to each NP
                        dVol = (4*pi/3)*((Radius(j,i+1,ll,k))^3 -
(Radius(j,i,ll,k))^3); % additional new volume per particle in m^3

                        % Calculate moles of monomer dedicated to growth
                        monomer_to_growth = monomer_to_growth +
(Number(j,i+1,ll,k)*dVol/(v0*Na*vol(ll))); % new

                    end % end of if Number(i,j,k)>0
                end % end of k = 1:2
            end % end of j = 1:i-1

          else % if S(i) < 1
            for k = 1:2
                for j = 1:i % used to be i-1
                    drdt(j,i,ll,k) = 0; % no growth if S < 1
```

```
                    Radius(j,i+1,ll,k) = Radius(j,i,ll,k); % Radius is
unchanged if S < 1
                    monomer_to_growth = 0;
                end % end of j = 1:i
            end % end of k = 1:2
        end % end of if S(i) > 1, end of GROWTH

        % Warnings
        if y(i+1,3,ll) < 0 && ww(ll) == 0
            fprintf('\n WARNING: (@Growth) Negative Molar Concentration \n
i=%d ll=%d Conc=%.1f ff=%.f \n\n',[i,ll,y(i+1,3,ll),monomer_to_growth])
            ww(ll) = 1;
            if real(-log10(y(i+1,3,ll))) > 9
                y(i+1,3,ll) = 0;
                fprintf(' Reset y(i+1,3,ll) to 0 %f \n\n',y(i+1,3,ll))
            else
                break
            end
        end

        % == Update moles of available monomer after growth
        y(i+1,3,ll) = y(i+1,3,ll) - monomer_to_growth +
monomer_from_dissolve; % change in moles of free C0 due to growth + disolve
    end % end of for ll = 1:3

    % Increase time step by +1
    i = i+1;

    % local variables for warning switches
    % if "break" uncommented, the simulation will stop when there is an error
    if ww(1) == 1 || ww(2) == 1 || ww(3) == 1
        %break
    end

end %
% ---------------------------- End of Calculation --------------------------
---



% Plotting
% Colormap for plotting
c = [0,  113/255,  200/255; % Extracellular, homogenous (blue)
     0.44,  0.71,  0.89; % Extracellular, heterogenous (blue)
     1,  0.06,  0; % Periplasm, homogenous (red)
     0.93,  0.68,  0.58; % Periplasm, heterogenous (red)
     0/255,  102/255,  0/255; % Cytoplasm, homogenous (green)
     119/255,  172/255,  48/255];% Cytoplasm, heterogenous (green)

% Find the probability distribution function (PDF) for the NP radii using
Number and Radius arrays.
nbins = 101; % number of bins into which radii are placed
PDF = zeros(nbins,3,2); % empty array for PDF
ed = zeros(nbins,3,2); % edges
for l = [1 2 3]
    for k = [1 2]
```

```
        ed(:,l,k) = linspace(1.0, round(1.5*max(max(max(Radius(:,end-
2,l,k))))*1e9),nbins); % define the bin edges to add histograms
        for i = 1:nbins-1
            % homo and hetero indipendent normalized
            PDF(i,l,k) = sum(Number(find(ed(i,l,k) < Radius(:,end-2,l,k)*1e9
& Radius(:,end-2,l,k)*1e9 < ed(i+1,l,k)),end-2,l,k))/
sum(Number(find(ed(1,l,k) < Radius(:,end-2,l,k)*1e9 & Radius(:,end-2,l,k)*1e9
< ed(end,l,k)),end-2,l,k));
        end
        PDF(isinf(PDF(:,l,k)),l,k)=0; % set NaNs equal to zero
        % ignore nuclei when Radius < lower_cutoff
        radius_lower_cutoff = 1e-9; % smallest NP considered
        totNumber(i,l,k) =
sum(Number(Radius(:,i,l,k)>radius_lower_cutoff,i,l,k));
    end
end

% calculate the value of and index of the maximum probability
for l = [1 2 3]
    for k = [1 2]
         [m.MaxPDF(l,k), m.MaxIndex(l,k)] = max(PDF(:,l,k));
    end
end

% change bin edges to bin centers
for l = [1 2 3]
    for k = [1 2]
        for i = 1:length(ed)
            X(i,l,k) = ed(i,l,k) + (ed(2,l,k)-ed(1,l,k))/2; % must match x
var in gausFcn
        end
    end
end



% radius distribution
figure(1)
set(gcf,'Position', [500, 10, 700, 900]);
jj = 1;
vv = {'Extracellular Homo','Extracelluar Hetero','Periplasm Homo','Periplasm
Hetero','Cytoplasm Homo','Cytoplasm Hetero'};
for l = [1 2 3]
    G = @(a1,b1,c1,x)a1*exp(-((x-b1)/c1).^2);
    for k = [1 2]
            xData = X(:,l,k);
            yData = PDF(:,l,k);
            if totNumber(end-2,l,k) > 1
                    % Set up fittype and options.
                    ft = fittype( 'gauss1' );
                    opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
                    opts.Display = 'Off';
                    opts.Lower = [0 ed(1,l,k) 0];
                    opts.Upper = [1 max(max(max(ed))) max(max(max(ed)))];
                    opts.StartPoint = [m.MaxPDF(l,k) ed(m.MaxIndex(l,k),l,k)
ed(m.MaxIndex(l,k),l,k)/5];
                    [fitresult, gof] = fit( xData, yData, ft, opts );
```

```
                    GaussFit.amp(l,k) = fitresult.a1;
                    GaussFit.mu(l,k) = fitresult.b1;
                    GaussFit.sig(l,k) = fitresult.c1/sqrt(2);
                    GaussFit.R2(l,k) = gof.rsquare;

                    subplot(3,2,jj)
                    plot(xData,yData,'.k')
                    hold on
                    plot(fitresult,'-r')
                    ll=legend('Data', 'Fit');
                    ll.Box='off';
                    ll.Location = 'northeast';
            else
                    GaussFit.amp(l,k) = NaN;
                    GaussFit.mu(l,k) =  NaN;
                    GaussFit.sig(l,k) = NaN;
                    GaussFit.R2(l,k) = NaN;

                    subplot(3,2,jj)
                    plot(xData,yData,'.k')
                    ylim([0 1])
                    ll=legend('Data');
                    ll.Box='off';
                    ll.Location = 'northeast';
            end
            title({vv{jj},strcat('R =
',num2str(GaussFit.mu(l,k),'%.2f'),'\pm',num2str(GaussFit.sig(l,k),'%.2f'),'n
m , ',' R^2 = ',num2str(GaussFit.R2(l,k),'%.3f'))})
            hold on
            xlabel( 'Radius [nm]', 'Interpreter', 'none' )
            ylabel( 'Probability', 'Interpreter', 'none' );
            grid on
            jj = jj + 1;
    end
end


%Total Number
figure(2)
vv = {'Extracellular Homo','Extracelluar Hetero','Periplasm Homo','Periplasm
Hetero','Cytoplasm Homo','Cytoplasm Hetero'};
set(gcf,'Position', [1, 1, 400, 300]);
jj=1; % local variable to make plot
% Total Number of nuclei per step
for l = [1 2 3]
    for k = [1 2]
        plot(time(1:end-2),totNumber(1:end-2,l,k),'o','Color',c(jj,:))
        hold on
        jj=jj+1;
    end
end
ylabel('Number of Nanoparticles  ')
xlabel('Time [hr]  ')
xlim([0 1.05*max(time)])
if 10^(1.1*round(log10(max(max(max(totNumber(:,:,:))))))) > 1
    ylim([1e0 2*10^(1.2*round(log10(max(max(max(totNumber(:,:,:))))))))])
```

```
else
    if 10^(1.1*round(log10(max(max(max(totNumber(:,:,:)))))) == 1
        ylim([1e0 5e1])
    else
        ylim([1e-10 1e3])
    end
end
ax=gca;
ax.YScale = 'log';
ax.FontName = 'Lucidia Sans';
ax.FontWeight = 'bold';
ax.FontSize = 14;
ll = legend(vv{:});
ll.Box='off';
ll.FontSize = 10;
ll.FontWeight = 'normal';
ll.Location = 'best';
%}


% concentrations of everything
figure(3);
%ylab = {'[Zn^{2+}] [mM]','[Cd^{2+}] [mM]','[Pd^{2+}] [mM]','[As^{5+}]
[mM]','[ZnS^0] [mM]','[CdS^0] [mM]','[Pd^0] [mM]','[As_2S_3^0] [mM]'};
set(gcf,'Position', [1000, 0, 500, 700]);
subplot(3,2,1)
ax=gca;
plot(time(1:end-1),y((1:end-1),4,1)*1000,'Linewidth',2, 'Color', c(1,:))
hold on
plot(time(1:end-1),y((1:end-1),4,2)*1000,'Linewidth',2, 'Color', c(3,:))
hold on
plot(time(1:end-1),y((1:end-1),4,3)*1000,'Linewidth',2, 'Color', c(5,:))
xlabel(u);
ylabel('[As(V)] [mM]')
xlim([0 1.05*max(time)])
legend('External','Periplasm','Cytoplasm')
legend('boxoff')
ax.FontName = 'Lucidia Sans';
ax.FontWeight = 'bold';
ax.FontSize = 12;

subplot(3,2,2)
ax=gca;
plot(time(1:end-1),y((1:end-1),5,1)*1000,'Linewidth',2, 'Color', c(1,:))
hold on
plot(time(1:end-1),y((1:end-1),5,2)*1000,'Linewidth',2, 'Color', c(3,:))
hold on
plot(time(1:end-1),y((1:end-1),5,3)*1000,'Linewidth',2, 'Color', c(5,:))
xlabel(u);
ylabel('[S_2O_3^{2-}] [mM]')
xlim([0 1.05*max(time)])
ylim([0 1])
legend('External','Periplasm','Cytoplasm')
legend('boxoff')
ax.FontName = 'Lucidia Sans';
ax.FontWeight = 'bold';
```

```
ax.FontSize = 12;

subplot(3,2,3)
ax=gca;
plot(time(1:end-1),y((1:end-1),1,1)*1000,'Linewidth',2, 'Color', c(1,:))
hold on
plot(time(1:end-1),y((1:end-1),1,2)*1000,'Linewidth',2, 'Color', c(3,:))
hold on
plot(time(1:end-1),y((1:end-1),1,3)*1000,'Linewidth',2, 'Color', c(5,:))
xlabel(u);
ylabel('[As(III)] [mM]')
xlim([0 1.05*max(time)])
legend('External','Periplasm','Cytoplasm')
legend('boxoff')
ax.FontName = 'Lucidia Sans';
ax.FontWeight = 'bold';
ax.FontSize = 12;

subplot(3,2,4)
ax=gca;
plot(time(1:end-1),y((1:end-1),2,1)*1000,'Linewidth',2, 'Color', c(1,:))
hold on
plot(time(1:end-1),y((1:end-1),2,2)*1000,'Linewidth',2, 'Color', c(3,:))
hold on
plot(time(1:end-1),y((1:end-1),2,3)*1000,'Linewidth',2, 'Color', c(5,:))
xlabel(u);
ylabel('[S^{2-}] [mM]')
if max(max(y(:,2,:))) ~=0
ylim([0 1.05*max(max(y(:,2,:)))]*1000)
end
xlim([0 1.05*max(time)])
legend('External','Periplasm','Cytoplasm')
legend('boxoff')
ax.FontName = 'Lucidia Sans';
ax.FontWeight = 'bold';
ax.FontSize = 12;

subplot(3,2,5)
ax=gca;
plot(time(1:end-1),y((1:end-1),3,1)*1000,'Linewidth',2, 'Color', c(1,:))
hold on
plot(time(1:end-1),y((1:end-1),3,2)*1000,'Linewidth',2, 'Color', c(3,:))
hold on
plot(time(1:end-1),y((1:end-1),3,3)*1000,'Linewidth',2, 'Color', c(5,:))
xlabel(u);
ylabel('[As_2S_3^{0}] [mM]')
if max(max(y(:,3,:))) ~=0
ylim([0 1.05*max(max(y(:,3,:)))]*1000)
end
xlim([0 1.05*max(time)])
legend('External','Periplasm','Cytoplasm')
legend('boxoff')
ax.FontName = 'Lucidia Sans';
ax.FontWeight = 'bold';
ax.FontSize = 12;
```

```
subplot(3,2,6)
ax=gca;
plot(time(1:end-2),S((1:end-2),1),'Linewidth',2, 'Color', c(1,:))
hold on
plot(time(1:end-2),S((1:end-2),2),'Linewidth',2, 'Color', c(3,:))
hold on
plot(time(1:end-2),S((1:end-2),3),'Linewidth',2, 'Color', c(5,:))
xlabel(u);
xlim([0 1.05*max(time)])
ylabel('Supersaturation')
legend('External','Periplasm','Cytoplasm')
legend('boxoff')
ax.FontName = 'Lucidia Sans';
ax.FontWeight = 'bold';
ax.FontSize = 12;


fprintf('\n'); % this is separate results from runs in the command window
```