NUFEB MATLAB model - Version 2.0

Rebeca González-Cabaleiro & Matthew Wade, Newcastle University

October 10, 2016

1 Introduction

This document describes the implementation of the NUFEB Mathematical Model V2.0 comprising individual-based cell growth in a 2D plane. The code is written using MATLAB as this allows for rapid and efficient testing of the biological components of the model and their integration with the physical phenomenon (e.g., diffusion and cell shoving). For more efficient simulation in 3D space and a better description of the *particle* physics, the code is ported to LAMMPS.

2 Model initialisation

An Excel file (*NUFEB.xls*) has been created to store the model state parameters and initial conditions:

- **Reaction Matrix** - Catabolic and anabolic reaction constants (multiplication factors) for each chemical species per cell type (i);
- **SpecParam** - Growth kinetic parameters per cell type ($K_{S,i}$, $\mu_{max,i}$, Y_i);
- **States** - Initial conditions for chemical and biological species and their state of matter. For liquids the value describes the Dirichlet (fixed) boundary condition;
- **ThermoParam** - Thermodynamic parameters for each chemical species and biomass component in the system;
- **Bacteria** - Bacterial cell parameters describing its physical properties and neighbourhood constraints (IB model);
- **Parameters** - Constants describing the environmental and thermodynamic conditions of the system;
- **Discretization** - Finite difference parameters defining Δx and Δt, as well as solver step sizes;
- **Diffusion** - Diffusion coefficients per liquid chemical (substrate).

Running the command `call` will initiate the program and the values described above are loaded into the workspace and assigned to a variable structure (**R**). General system parameters are assigned to the substructure (**pOp**) and the state variables are stored in (**St**). The stoichiometric/reaction (Petersen) matrix is modified to remove the unconserved metabolites (water and protons), which don't contribute significantly to the biochemical reactions. The matrix is assigned to the (**rm**) substructure. The growth function parameters are assigned to (**pTh**). Thermodynamics and pH are stored in the (**DGR**) and (**pH**) substructures, respectively. Diffusion coefficients are assigned to the (**Dif**) substructure. A full list of all input parameters is given in Table 1.

Table 1: Description of model parameters defined in the Excel file

Function	Parameter	Value	Unit	Description
Reaction Matrix				
	Cat	Var_S	-	Catabolic stoichiometry
	Anab	Var_S	-	Anabolic stoichiometry
	Decay	Var_S	-	Decay stoichiometry
SpecParam				
	K_S	Var_X	M	Half-saturation constant
	mu_max	Var_X	$\text{mol}_s.\text{mol}_x^{-1}.\text{h}^{-1}$	Maximum specific growth rate
	Kdec	Var_X	h^{-1}	Decay constant
	DGdis	Var_X	kJ.mol^{-1}	Gibbs free energy of dissipation
	Yield	Var_X	$\text{mol}_x.\text{mol}_s^{-1}$	Growth yield coefficient
	Maintenance	Var_X	kJ.mol^{-1}	Anabolic maintenance energy
States				
	L	Var_L	M	Initial concentrations of liquid components
	G	Var_G	bar	Initial partial pressure of gas components
	S	Var_X	M	Initial biomass concentrations
ThermoParam				
	$L^c_{\Delta G}$	Var^c_L	kJ.mol^{-1}	Gibbs free energy per liquid species (c)
	$G_{\Delta G}$	Var_G	kJ.mol^{-1}	Gibbs free energy per gas component
	$S_{\Delta G}$	Var_X	kJ.mol^{-1}	Gibbs free energy per biomass component
Bacteria				
	bac_mmax	$1.57e^{-16}$	g	Maximum cell mass
	bac_mmin	$1.57e^{-18}$	g	Minimum cell mass
	bac_rmax	$5e^{-7}$	m	Maximum cell radius
	bac_h	$2e^{-6}$	m	Cell height
	bac_rho	100	g/L	Cell density
	bac_MW	24.6	g/mol	Molar mass
	bac_nmax	Var	-	Maximum number of cells
	k	1.1	-	Shoving factor
	s_dist	$5e^{-6}$	m	Shoving distance
	overlap	$1e^{-8}$	m	Overlap distance
Parameters				
	T	298.15	K	Temperature
	Rth	0.0083144	$\text{kJ.mol}^{-1}.\text{K}^{-1}$	Universal gas constant (thermodynamics)
	Rg	0.08205746	$\text{atm.L.mol}^{-1}.\text{K}^{-1}$	Universal gas constant (gas transfer)
	P	1	bar	Pressure
	Vgas	$8e^{-11}$	L	Gas volume
	Vr	$8e^{-11}$	L	Liquid volume
	pH	7	-	pH
Discretization				
	nx & ny	Var	-	Number of grid nodes
	maxx & maxy	Var	m	Maximum grid size
	maxT	Var	h	Maximum simulation time
	dT	Var	h	Diffusion step size
	dT_bac	Var	h	Growth step size
	dT_print	Var	h	Print to screen step size
	Tol	$1e^{-3}$	-	Steady-state tolerance
Diffusion				
	Diff_L,G	$\text{Var}_{L,G}$	$\text{m}^2.\text{h}^{-1}$	Liquid and gas diffusion coefficients
	kLa_G	Var	h^{-1}	Mass transfer coefficients for gases

2.1 IB Model parameters

The cell specific properties are stored in the (**bac**) substructure. A number of transformations are performed for use later in the model.

- Cell mass (g): $bac_{mm,i} = 0.95 \cdot bac_{mmax,i}$;
- Cell mass (mol): $bac_m = bac_{mm,i}/bac_{mw,i}$;
- Cell radius (m): $bac_r = \sqrt{bac_{mm,i}/(\pi \rho h)}$, assuming a cylindrical cell with density (ρ) and height (h). *N.B.* In the 2D implementation, it is assumed that a *top down* view of the system is characterised, i.e., cell width $\times$ cell length.

2.2 Space discretisation

The two-dimensional Eulerian space (x, y) is used to determine diffusion and calculate daughter cell positions as new biomass is created. The space information is stored in the (**Sxy**) substructure. The following properties are defined for the grid elements:

- Element width (m): $dx = maxx/(nx - 1)$, $dy = maxy/(ny - 1)$;
- If $dx < 2 \cdot bac_{rmax,i}$ (bacterial cell diameter is larger than grid element) then increase grid size: $nx_{new} = 1 + \lfloor (maxx/2 \cdot bac_{rmax,i}) \rfloor$, $dx = maxx/(nx_{new} - 1)$, $ny_{new} = nx_{new}$;
- Number of grid elements: $nT = nx_{new} ny_{new}$;
- Cell volume (L): $V_g = dx \cdot dy \cdot h \cdot 1000$;

Concentrations are described for each of the nT individual grid elements in a single column vector, which greatly reduces the computational complexity. Thus, column vectors for initial state concentrations have the size $[nT * nS^*, 1]$, where nS is the number of species assigned to gas, liquid or biomass (solid) state. The initial bacterial cell position is calculated as follows:

- $ps = maxx/(2-(k \cdot bac_{dmax}((nX-1)/2))$, where k is the shoving factor, bac_{dmax} is the maximum cell diameter, and nX is the number of bacterial species;
- The cells can then be assigned coordinates along (x, y) axes defined by this starting position (ps), the step size ($k \cdot bac_{dmax}$), and the end position ($ps + k \cdot bac_{dmax} \cdot (nX - 1)$).

The grid space is then defined using the sparse matrix form to significantly reduce memory requirements for matrix calculations for highly sparse matrices.

- Create two sparse identity matrices, I_x and I_y of size $(nx - 2, nx - 2)$ and $(ny - 2, ny - 2)$;
- Create two sparse Laplacian operator matrices, Ls_x and Ls_y, where $Ls = E + E' - 2I$, and E is the identity matrix at position $(n - 3)$. This determines the degree and adjacency of each cell in the grid in matrix form. The degree component contains the number of vertices (or cell connections) per cell in the diagonal, whilst adjacency is represented by a vector of zeros (no connection) and ones (connection) along the cell axes (i_x, i_y);
- Create Kronecker tensor products of the identity and operator matrices and sum: $Ls_{xy} = I_x \otimes Ls_x/dx^2 + I_y \otimes Ls_y/dy^2$;
- Perform a similar procedure for the boundary conditions: $bc_x = \vec{1} \otimes I_{y,1} + \vec{1} \otimes I_{y,end}$, $bc_y = I_{x,1} \otimes \vec{1} + I_{x,end} \otimes \vec{1}$, and $bc_{xy} = bc_x/dx^2 + bc_y/dy^2$.

3 Model solver

The model is solved for the microbial growth and diffusion equations over a predefined run time ($maxT$). However, solutions are found at different time steps in each case, defined by dT for diffusion and dT_{bac} for growth. A third time step, dT_{Print}, determines when results from the solver are printed to the screen.

3.1 Individual Based Model

The IB model (`bacteria.m`) is used to calculate the propagation (through division) and position of biomass cells through time and Euclidean space. A vector of initial conditions for each state is created ($\vec{U}$).

3.1.1 Cell division

Cell division only occurs when the current cell mass is greater than a predefined threshold mass value ($bac_m > \max(bac_m)$). The following routine is performed whilst this condition holds true for the parent cell.

- If mass of cell i is greater than the threshold mass then start division calculation;
- Generate a random angle for division: $\sphericalangle = \mathrm{U}([0,1])\pi$;
- Increase number of cells and cell type (species) by one: $bac_n = bac_n + 1$, $bac_{ns} = bac_{ns} + 1$;
- Determine coordinates of daughter cell: $bac_{x,i+1} = bac_{x,i} + bac_{r_i}\cos(\sphericalangle)$, $bac_{y,i+1} = bac_{y,i} + bac_{r_i}\sin(\sphericalangle)$;
- Assume daughter cell inherits biological properties of parent cell ($\mu_i, K_{S,i}$);
- Calculate daughter cell mass: $bac_{m,i+1} = 0.45 \cdot bac_{m,i}$;
- Calculate daughter cell radius: $bac_{r,i+1} = \sqrt{bac_{m,i+1}/(\pi\rho h)}$;
- Determine new coordinates of parent cell: $bac_{x,i} = bac_{x,i} - bac_{r_i}\cos(\sphericalangle)$, $bac_{y,i} = bac_{y,i} - bac_{r_i}\sin(\sphericalangle)$;
- Calculate remaining mass of parent cell: $bac_{m,i} = bac_{m,i} - bac_{m,i+1}$;
- Calculate new radius of parent cell: $bac_{r,i} = \sqrt{bac_{m,i}/(\pi\rho h)}$.

3.1.2 Cell decay

The updated model includes biomass decay, where cells are removed from the community at a rate k_{dec}. This is implemented by assuming a lower limit of cell mass, such that decay occurs if $bac_m < \min(bac_m)$. In this case, the number of cells are reduced by one and its coordinates, radius and species parameters are deleted.

3.1.3 Cell shoving

As new biomass is generated the cells in the grid are shifted to new positions so the following process is implemented to prevent overlap of cells and to determine the effect of the new cells on their neighbouring cells.

- Set shoving flag ON to enable shoving. Perform the following steps until the system is stable and then turn shoving flag OFF;
- For each cell in the grid search for overlapping cells in 2D plane;
- Calculate distance between cells in the x-plane: $dis_x = bac_{x,i} - bac_{x,j}$, where $j = i + 1$ is the next cell position in the x-plane;
- If $|dis_x| < dis_{x,min}$, calculate distance between cells in y-plane: $dis_y = bac_{y,i} - bac_{y,j}$;
- If $|dis_y| < dis_{y,min}$, calculate distance between cell centres: $dis_{xy} = \sqrt{dis_x^2 + dis_y^2 + 1e^{-20}}$;
- Calculate overlap: $r0 = \frac{k(bac_{r,i}+bac_{r,j})-dis_{xy}}{dis_{xy}}$;
- Multiply intercellular distance by overlap distance: $dis_{x,s} = dis_x \cdot r0$, $dis_{x,s} = dis_x \cdot r0$;
- If cell i overlaps with cell j, i.e. $r0 > 0.1$ then proceed to shoving step to calculate new positional shifts for both cells: $xr_i = xr_i - \frac{dis_{x,s}bac_{m,i}}{(bac_{m,i}+bac_{m,j})}$, $xr_j = xr_j + \frac{dis_{x,s}bac_{m,j}}{(bac_{m,i}+bac_{m,j})}$, $yr_i = yr_i - \frac{dis_{y,s}bac_{m,i}}{(bac_{m,i}+bac_{m,j})}$, $yr_j = yr_j + \frac{dis_{y,s}bac_{m,j}}{(bac_{m,i}+bac_{m,j})}$;
- Calculate new coordinates of cell i: $bac_{x,i} = bac_{x,i} - xr_i$, $bac_{y,i} = bac_{y,i} - yr_i$;
- Reset positional coordinate of cell i: $xr_i = yr_i = 0$.

3.1.4 Kinetics

Currently microbial growth is described by simple Monod kinetics. The following procedure details the solver steps for determining the biomass growth based on the kinetic parameters. The updated model has the option to include thermodynamics (variable yield) and pH to the calculation of the growth kinetics, and more details of these can be found in their respective subsections.

- Calculate the metabolic matrix for all microbial species: $\mathbf{M_i} = \frac{\mathbf{C}_i}{Y_i} + \mathbf{A}_i$, where $\mathbf{C_i}$ and $\mathbf{A_i}$ signify the catabolic and anabolic reaction matrices described earlier. Y_i are the growth yield parameters per species;
- Remove endergonic components from the metabolic matrix: $\mathbf{M_i} = \mathbf{M_i}(\mathbf{M_i} < 0)$;
- Define grid node edges in x- and y-planes: e_x, e_y;
- For each node (n), check if centre of bacterial cell is present:
  - If $bac_x > e_{x,n}$, $bac_x - e_{x,n+1} < 1e^{-14}$, $bac_y > e_{y,n}$ and $bac_y - e_{y,n+1} < 1e^{-14}$;
  - Get indices of nodes containing cells. The index position is calculated by assuming a vector of length equal to the number of elements in the boundary free grid multiplied by the number of chemical species active in the system: $\hat{nT} = (nx - 2)(ny - 2)nS$. Denoting ix as cell position in x-plane, and iy as cell position in y-plane, then the indices for the cell in the nS solution spaces is given by: $Ind = (ix - 1) + (nx - 2)(iy - 2) + (\vec{nS} - 1)\hat{nT}$;
  - Determine chemical concentrations at Ind and remove those with zero magnitude;
  - Calculate growth rate dependent on limiting substrate using defined growth function (e.g. Monod): $\mu_i = \mu_{max,i} \min(\frac{\vec{S_s}}{K_{S,i}+\vec{S_s}})$, where $_i$ denotes cell type (species) and $\vec{S_s}$ denotes vector of all chemical species (substrates) in the system;
  - Calculate amount of biomass formed [mol]: $X_i = \mu_i \cdot bac_{m,i}$;

- Calculate amount of substrate consumed [mol]: $S_i = \mathbf{M_i} \cdot X_i$;
- Calculate liquid concentrations [M]: $S_{i,liq} = S_{i,liq}/V_g$;
- Calculate gas partial pressures [bar]: $P_{i,gas} = S_{i,gas} R_g T/V_h$, where R is the universal gas constant [L.atm.K^{-1}.mol^{-1}], T is the temperature [K], and V_h is the gas volume [L];
- Update chemical species concentrations in populated node locations.

- Calculate the overall liquid mass balance in grid: $S^*_{i,liq} = (\sum S_{i,liq} V_g)/V_r$, where V_r is total volume in grid;
- Distribute total gas concentration across entire grid: $S^*_{i,gas} = (\sum S_{i,gas})/nT$;

3.1.5 pH

Proton concentration is calculated implicitly using a numerical approximation of the roots of the charge balance equation:

$$[H^+] - [OH^-] = Ch(pH) = \sum_{i=1}^{j} z[S_i^{z-}] \tag{1}$$

where z is the charge number and j is the number of chemical species contributing to the pH. Using the general form of Newton-Raphson, where ϵ_n is the error of the initial proton estimation:

$$x_{n+1} = x_n + \epsilon_n, \;\; n = 0 \ldots \infty \tag{2}$$

and by Taylor approximation, then the proton concentration is given by:

$$[H^+] = [H^+]_0 - \frac{\Delta_{ch}(H^+)}{\frac{d\Delta_{ch}(H^+)}{d[H^+]}} \tag{3}$$

As this is an approximation then, defining τ as the charge balance tolerance (the minimum gap in the charge balance to satisfy equilibrium), ν as the maximum number of iterations to consider (to avoid continuation if the solution does not converge), we first define expressions describing the fraction of each chemical species in the solution from:

$$f_{S,i} = \frac{S_i}{\sum_{i=1}^{n} S_i} \tag{4}$$

We can also get a general expression for the dissociation constants of each species in the equilibrium:

$$K_i = \frac{\sum_j^{nP} P_j^{m_j}}{\sum_i^{nS} S_i^{m_i}} \tag{5}$$

where S and P are reaction substrates and products, respectively, and m is the number of moles of the reaction component. We can substitute this into a material balance for each equilibrium reaction and, from Eq. 4, rearrange to generate the species fractions as a function of proton concentration and the relevant dissociation constants; $\alpha_{i,z} = f([H^+], K_i)$. Defining $C_{i,z}$ as the charge weighting for chemical species i, then the following algorithm is performed to converge to true proton concentration per grid unit (x, y) and solver step:

While $n < \nu$

If $|\epsilon_n| > \ 1e^{-14}$ & $|\mathbf{F_n}| > \ \tau$

$\mathbf{F_n} = [H^+]_n + \sum_{i=1}^{j} (S_i \cdot \alpha_{i,z} \cdot C_{i,z})$

$\mathbf{F'_n} = \frac{\mathrm{d}F_n}{\mathrm{d}[H^+]}$

$\epsilon_n = \frac{\mathbf{F_n}}{\mathbf{F'_n}}$

$[H^+]_{n+1} = [H^+]_n - \epsilon_n$

else

$[H^+]_{n+1} = [H^+]_n$

end

$\mathrm{pH}_{n+1} = -log_{10}([H^+]_{n+1})$

end

The current model implementation includes an option to keep the pH fixed as this step slows down the simulation speed considerably.

3.1.6 Thermodynamics

A generalised approach to calculate the Gibbs free energy of the chemical reactions is implemented in the model as follows:

- Defining the Gibbs free energy of the reaction:

  $\Delta G_{rxn} = \Delta G^0 + R_{th} T \ln \left(\frac{\prod_j^{nP} P_j^{m_j}}{\prod_i^{nS} S_i^{m_i}} \right)$;

- In the model, the quotient is calculate from the stoichiometric reaction matrix ($\mathbf{M}$) and the matrix of dissociated chemical species involved in the reaction ($\mathbf{Q}$) :

  $\Delta G_{rxn} = \Delta G^0 + R_{th} T \left(\mathbf{M^T} ln(\mathbf{Q}) \right)$;

- The Gibbs free energy of catabolism (ΔG_{cat}) and anabolism (ΔG_{ana}) can be extracted per organism from the resulting vector;

- In the case that a variable yield is selected by the user, the catabolic and anabolic energy values can be used to derive the catabolic reaction equation:

  $\lambda_{cat} = \Delta G_{cat}^{-} \left(-\frac{\Delta G_{ana} + \Delta G_{dis}}{\Delta G_{cat}} \right)$;

- This is equivalent to the inverse yield: $Y_i^* = \frac{1}{\lambda_{cat}}$;

- The procedure described in Section 3.1.4 is then followed with Y_i^* replacing the yield term in the expression of the metabolic matrix calculation.

3.1.7 Gas transfer

The general form of the liquid-gas transfer rate coefficient is given by:

$$\rho_{T,i} = kLa_i \left(\frac{S_{liq,i}}{K_{H,i}} - S_{gas,i} \right) \tag{6}$$

where $K_{H,i}$ is Henry's constant for each gas, calculated by:

$$K_{H,i} = \exp^{\left(\frac{\Delta G^{0,c}_{liq,i} - \Delta G^{0}_{gas,i}}{RT} \right)} \tag{7}$$

For carbon dioxide, c denotes the form it takes in the liquid phase (CO_2 or HCO_3^-). For oxygen, c is 0.

3.2 2D Finite-difference calculation

A number of finite-difference methods may be employed to solve diffusion across a boundary, and these may either be explicit (e.g., forward Euler) where $y(t+\Delta t) = f(y(t))$ or implicit (e.g., backward Euler) where $g(y(t), y(t+\Delta t)) = 0$, solved with respect to $y(t+\Delta t)$. Whilst explicit methods are the easiest to implement, when the requirements for Δt becomes impractically small (i.e., a stiff system), implicit methods are generally employed, although they are more difficult to use than explicit methods. The Crank-Nicolson (C-N) method is a compromise between the two methods and introduces the concept of central difference in the time domain ($t_{n+1/2}$) and a second-order central difference in the space derivative. Although more accurate and numerically stable for small Δt than either explicit or implicit methods alone, C-N is more computationally expensive.

The following procedure is used to implement the C-N method to calculate the diffusion of substrates and products across a pre-defined boundary.

- Defining:
  - Υ: All components;
  - u: Liquid components;
  - Φ: Gas components;
  - Ψ: Biomass components.
- Using the predictor-corrector method, start by approximating the diffusion components (*Predictor step*) with k_i being the diffusion coefficients per (liquid) chemical species:
  - $\beta = ((dT \cdot k_i/2)Ls_{xy} + I_{xy})\,\Upsilon_i + dT \cdot k_i \cdot S^0_{i,liq} \cdot bc_{xy} + dT \cdot S_{i,liq}$;
  - $\alpha = I_{xy} - (dT \cdot k_i/2)Ls_{xy}$;
  - $\Upsilon'_i = \alpha \backslash \beta$.
- Repeat the microbial kinetics procedure with predicted values: $\mathbf{R}' = f(\Upsilon', \Phi, \Psi, \mathbf{R})$;
- Move to the *Corrector step* with the new values of the kinetics:
  - $\beta' = ((dT \cdot k_i/2)Ls_{xy} + I_{xy})\,\Upsilon'_i + dT \cdot k_i \cdot S^0_{i,liq} \cdot bc_{xy} + dT/2(S_{i,liq} + S'_{i,liq})$;
  - $\alpha' = I_{xy} - (dT \cdot k_i/2)Ls_{xy}$;

  - $\Upsilon_i'' = \alpha \backslash \beta$.

- Check for negative values in Υ'' and remove;
- Iterative update of gas components: $\Phi'' = \Phi + dT_{bac} S^*_{i,gas}$, where dT_{bac} is the time step for solving the growth function;
- Iterative update of biomass components: $\Psi'' = \Psi + dT_{bac} X_i$;
- Repeat the microbial kinetics procedure with corrected values: $\mathbf{R}'' = f(\Upsilon'', \Phi'', \Psi'', \mathbf{R}')$;
- Update state components with average of predicted and corrected values:
  - $u^\dagger = u + dT_{bac} \frac{S^*_{i,liq} + S^{**}_{i,liq}}{2}$;
  - $\Phi^\dagger = \Phi + dT_{bac} \frac{S^*_{i,gas} + S^{**}_{i,gas}}{2}$;
  - $\Psi^\dagger = \Psi + dT_{bac} \frac{X^*_i + X^{**}_i}{2}$.
- Calculate errors:
  - Absolute error: $\epsilon_a = \max\left(|(\Upsilon' - \Upsilon'')|, |(\Phi - \Phi'')|, |(\Psi - \Psi'')|\right)$
  - Relative error: $\epsilon_r = \max\left(\frac{(|(\Upsilon' - \Upsilon'')|}{\Upsilon''}, \frac{|(\Phi - \Phi'')|}{\Phi''}, \frac{|(\Psi - \Psi'')|}{\Psi''}\right)$
- Run new inputs through IB model and advance iteration by 1.

4 List of symbols

$\otimes$	Tensor product operator
$\sphericalangle$	Angle of cell division (°)
α	Upper tridiagonal matrix for Crank-Nicolson
α_z	Dissociated chemical species fraction
β	Function of lower tridiagonal matrix for Crank-Nicolson
Δ_{ch}	Charge balance
ΔG^0	Gibbs free energy under standard conditions (kJ.mol^{-1})
ΔG_{ana}	Gibbs free energy of anabolism (kJ.mol^{-1})
ΔG_{cat}	Gibbs free energy of catabolism (kJ.mol^{-1})
ΔG_{dis}	Gibbs free energy of dissipation (kJ.mol^{-1})
ΔG_{rxn}	Gibbs free energy of reaction (kJ.mol^{-1})
ϵ	Error
λ_{cat}	Catabolic reaction term
phi	Finite difference matrix of gas components
psi	Finite difference matrix of biomass components
ρ	Cell density (mol.L^{-1})
ρ_T	Liquid-gas transfer rate coefficient
μ	Specific growth rate (mol.mol^{-1}.h^{-1})
μ_{max}	Maximum specific growth rate (mol.mol^{-1}.h^{-1})
ν	Maximum number of Newton-Raphson iterations
τ	Charge balance tolerance
Υ	Finite difference matrix of all components
$\mathbf{A}$	Anabolic reaction matrix
bac$_{dmax}$	Maximum celll diameter (m)
bac$_m$	Cell mass (mol)
bac$_{mm}$	Cell mass (g)
bac$_{mmax}$	Maximum cell mass (g)
bac$_{mw}$	Cell molecular weight (mol.g^{-1})
bac$_r$	Cell radius (m)
bacr_{max}	Maximum cell radius (m)
bac$_x$	x-coordinate of cell
bac$_y$	y-coordinate of cell
bc	Dirichlet boundary conditions
$\mathbf{C}$	Catabolic reaction matrix
C$_z$	Charge weight

dis	Distance between cells (m)
dx	Grid element width, x plane (m)
dy	Grid element width, y plane (m)
ex	Grid edges, x plane
ey	Grid edges, y plane
E	Edge identity matrix
F	Matrix of ionic species concentrations
$[H^+]$	Proton concentration (mol)
h	Cell height (m)
I	Sparse identity matrix
k	Shoving factor
k_{dec}	Cell decay rate (h^{-1})
k_i	Diffusion coefficients per chemical species i ($m^2.h^{-1}$
kLa	Mass transfer coefficient (h^{-1})
K_H	Henry's constant ($L.atm.mol^{-1}$)
K_i	Dissociation constant for species i
K_S	Half saturation constant (M)
Ls	Laplacian operator matrix
m	Number of moles of reaction component
M	Metabolic matrix
maxx	Grid length, x plane (m)
maxy	Grid length, y plane (m)
nP	Number of reaction products
nS	Number of reaction substrates
nS^*	Number of state components
nT	Total number of grid elements
nx	Number of grid elements. x plane
ny	Number of grid elements, y plane
P_{gas}	Gas partial pressure (bar)
P_j	Reaction products (mol)
ps	Initial cell position
Q	Reaction quotient
R_g	Universal gas constant for gas transfer ($atm.L.mol^{?1}$ $.K^{?1}$)
R_{th}	Universal gas constant for thermodynamics ($kJ.mol^{-1}.K^{-1}$)
r0	Overlap distance (m)
S	Substrate consumed / Reaction substrates (mol)
S_{gas}	Gas concentration (mol)
S_{liq}	Concentration of liquid species (M)
T	Temperature (K)
u	Finite difference matrix of liquid components
U[(0,1)]	Standard uniform distirbution
V_g	Bacterial cell volume (L)
V_h	Gas volume (L)
X	Biomass formed (mol)
xr	Cell position after shoving, x plane
Y	Growth yield ($mol.mol^{-1}$)
yr	Cell position after shoving, y plane