BIO2020: BIOMASS-NUTRIENT-FLOW SIMULATION CODE

CHOAH SHIN

1. Introduction

This document describes the MATLAB code used to simulate the biomass-nutrient dynamics with flow in 2d [5]. We use the MAC method to solve for the flow, upwind method for advection, and CCFD in space and backward Euler in time for the diffusion-reaction. We also use the operator splitting method to handle advection and diffusion-reaction separately.

1.1. **Credits and use.** The code and documentation are made publicly available through GitHub platform at [4]. If you use this code, we ask you to acknowledge it according to the rules of Creative Commons CC BY-NC-ND 4.0.

1.2. **Acknowledgement.** This work was partially supported by the National Science Foundation DMS-1912938 and DMS-1522734, and by the NSF IRD plan 2019-21 for M. Peszynska. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

1.3. **Contents.** The main driver for BN_Flow simulation is BN_Flow.m. The +supp folder includes following dependent mfiles:

(1) BN_Brinkman: biomass-nutrient model coupled to heterogeneous Brinkman flow
(2) BN_Stokes: biomass-nutrient model coupled to Stokes flow
(3) Brinkman2D: solves 2d heterogeneous Brinkman flow
(4) DR2D: solves constrained biomass-nutrient model using semismooth Newton method
(5) figGen: generate figures
(6) input_data: read input scenario/case and convert units for simulations
(7) record: save data files
(8) Stokes2D: solves 2d Stokes flow
(9) Upwind2D_B: 2d advection solver by first-order upwind method for biomass
(10) Upwind2D_N: 2d advection solver by first-order upwind method for nutrient
(11) Upwind1D: 1d first-order upwind method called in 2d upwind method for x- and y-sweeps.

The input parameters can be located in +Scenario folder with available porous geometries in +Porous_Medium and +Obstacles.

2. BN_Flow.m

2.1. **Notations.** Let $\Omega = (x_a, x_b) \times (y_a, y_b)[\text{mm}^2]$ be a complex domain with the boundary Γ. We use Ω_r and Ω_n to denote the rock and non-rock regions in Ω, respectively. The interface between Ω_r and Ω_n is denoted by Γ_{rn}.

Let (B, N) represent the mass density of biomass and nutrient, respectively. The maximum density of biomass is denoted by B^* and the threshold for biofilm phase is denoted by B_*.

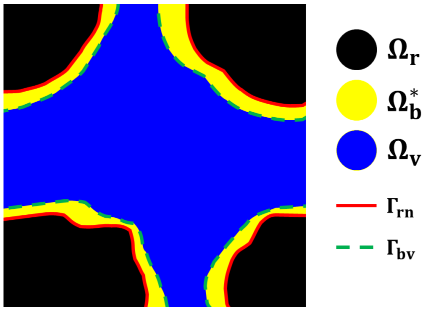


FIGURE 1. An illustration of pore-scale Ω with different regions as labelled.

Now we define the different regions in Ω_n as follows

$$\begin{aligned}
\Omega_0(t) &= \{x : B(x,t) = 0\},\\
\Omega_b(t) &= \{x : B(x,t) > 0\},\\
\Omega_b^*(t) &= \{x : B_* \leq B(x,t) \leq B^*\}, \qquad \text{biofilm phase.}
\end{aligned}$$

An illustration of Ω with biofilm phase is shown in Fig. 1. We use the subscripts D and N for Γ to indicate the Dirichlet and Neumann boundaries, respectively. We also denote the inflow and outflow boundaries of Γ by Γ_{in} and Γ_{out}.

2.2. **Problem.** We consider the biomass-nutrient-flow model from [5]

$$\partial_t B + \nabla\cdot(uB) - \nabla\cdot(d_B(B)\nabla B) \quad +\partial I_{(-\infty,B^*]}(B) \quad = \kappa_B B m(N), \quad x \in \Omega, \tag{1a}$$

$$\partial_t N + \nabla\cdot(uN) - \nabla\cdot(d_N(B)\nabla N) \qquad = B m(N), \qquad x \in \Omega, \tag{1b}$$

with initial and boundary conditions:

$$d_B \nabla B \cdot \nu|_\Gamma = 0, \qquad B(x,0) = B_0 \chi_{\Omega_b(0)}(x), \quad x \in \Omega, \tag{1c}$$

$$d_N \nabla N \cdot \nu|_{\Gamma_N} = 0, \quad N|_{\Gamma_D} = N_{bd}, \qquad N(x,0) = N_{init}(x), \qquad x \in \Omega, \tag{1d}$$

where ν is the normal vector to Γ.

The diffusivities $d_B(B)$ and $d_N(B)$ are defined as

$$d_B(B) \quad = \quad d_{B,0}\left(\frac{B}{B^* - B}\right)^\alpha, \tag{1e}$$

$$d_N(B) \quad = \quad d_{N,w}(B^* - B) + d_{N,b}B. \tag{1f}$$

For (partially) permeable biofilm, we use the heterogeneous Brinkman equations

$$-\mu\Delta u + \mu k_{bx}^{-1}(x)\chi_{\Omega_b^*} + \nabla p \quad = \quad 0, \quad x \in \Omega_n \tag{1g}$$

$$\nabla \cdot u \quad = \quad 0 \quad x \in \Omega_n \tag{1h}$$

with appropriate boundary conditions:

$$u|_{\Gamma rn} = 0, u|_{\Gamma_{in}} = u_D(x), \text{ and } \mu\nabla u \cdot \nu - p\nu = 0 \text{ on } \Gamma_{out}. \tag{1i}$$

2.3. **Data.** The input values are defined in a script file inside the +scenario folder. The script file has the name SaCb.m which stands for scenario a and the case b in each scenario. For example, we have S0C0.m as our sample file. The user must provide the values as in Tab. 1.

2.3.1. *Available porous media and initial biomass assignments.*

Parameter	Description	Default value
	flag.< parameter >	
restart	1 = restart triggered	0
adv	1 = turn on advection	1
DR	1 = turn on diffusion-reaction	1
flow	1 = solve for velocity and pressure solutions	0
	Simulation constraints: num.< parameter >	
T	Final time [h]	100
dtmax	Maximum time-step size [h]	10^{-2}
tnmax	Maximum number of time steps	100
flow	How often we solve for the flow	1
tau	Number of minor time steps within each major time steps Δt	10
rec	How often we record data	10
plot	How often we generate figures	10
	Unit conversion factors from SI units $[\mathrm{m}, \mathrm{s}, \mathrm{kg}]$ to ...: unit.< parameter >	
str_len	Unit for length	‘mm’
length	Length conversion factor from [m] to [str_len]	10^3
str_time	Unit for time	‘h’
time	Time conversion factor from [s] to [str_time]	1/3600
str_mass	Unit for mass	‘g’
mass	Mass conversion factor from [kg] to [str_mass]	10^3
	Spatial data: x.< parameter >, y.< parameter >	
lo, hi	Lower and upper bounds [m]	vary
n	Number of mesh in x or y	vary
	Geometry	
pm	Porous geometry selected from +Porous_Medium folder	0
obst	Initial biomass distribution; see available patterns in +Obstacles folder	0
kb	Permeability of biofilm phase	vary
	Parameters for flow: flow.< parameter >	
um	Mean flow velocity $\overline{u}_D$[m/s]	vary
uf	Fully developed flow velocity between parallel plates based on $\overline{u}_D$	fixed
dir	Flow direction; 1. B→T, 2. T→B, 3. L→R, 4. R→L	3
cfl	Maximum CFL number for advection step	0.95
mu	Viscosity of water $[\mathrm{Pa} \cdot \mathrm{s}]$	8.9×10^{-4}
	Parameters for biomass: b.< parameter >	
kappa	Biomass growth coefficient	0.5
star	Maximum biomass density B^*	1
nu	Coefficient for biofilm phase threshold $B_* = \nu B^*$	0.9
init	Initial biomass density B_0	0.5
eberla	Parameter for singular biomass diffusivity α	2
dB0	Motility coefficient $d_{B,0}[\mathrm{mm}^2/\mathrm{h}]$	10^{-4}
eberlstar	$\overline{B}^*$ for singular diffusivity	1.01
	Parameters for nutrient: n.< parameter >	
kappa	Nutrient uptake rate [1/h]	2
inlet	Inlet nutrient density; for nutrient rich system, $N(x,0) = n.inlet$.	1
N0	Monod constant k_N in $m(N)$	$1.18 \times 10-3$
df	Nutrient diffusivity in bulk fluid, $d_{N,w}[\mathrm{mm}^2/\mathrm{h}]$	6
db	Nutrient diffusivity in biofilm phase $d_{N,b}[\mathrm{mm}^2/\mathrm{h}]$	0.6

TABLE 1. Input data for model in BN_Flow

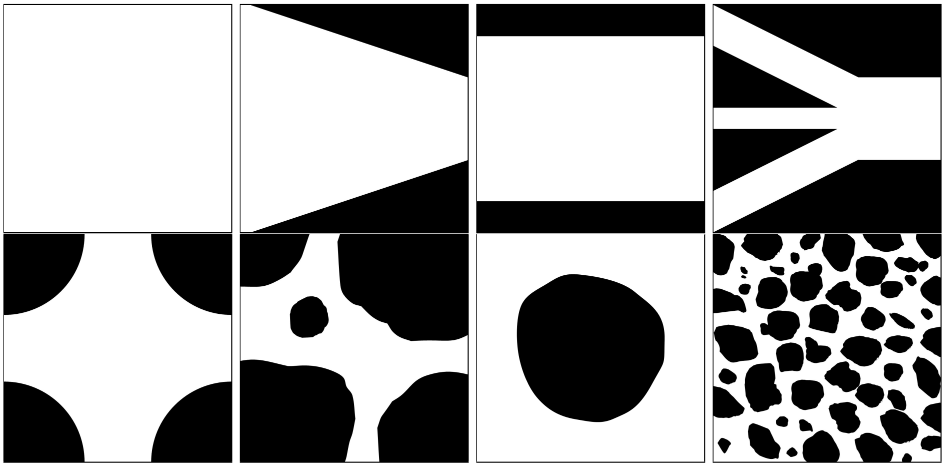


FIGURE 2. Selected available porous media. See more options in side the Porous_Medium folder. Select pm in +Scenario/S1C1.m. Similarly, choose the initial biomass distribution in the Obstacle folder and assign obst in +Scenario/S1C1.m.

2.4. **Discretization.** The transport part of (1) is handled by the operator splitting method [1]. We use the first-order Upwind method [1] for the advection term, and the Cell-Centered Finite Difference (CCFD) method [3] for the diffusion and reaction steps together. For the heterogeneous Brinkman flow, we use the Marker-And-Cell (MAC) method [2, 6].

2.5. **Solver.** We solve the saddle point problem obtained from the flow model using the MATLAB "backslash" \ operator. The nonlinear system from the diffusion and reaction steps is resolved by using the semismooth Newton method.

2.6. **Code preamble and parameters.** Following code requires an inputcard.

```
function [end_clock] = BN_Flow(ns,nc)
%=========================================================================%
% 2d coupled flow and biomass-nutrient dynamics simulator
%-------------------------------------------------------------------------%
% Inputs:
% ns: number for scenario
% nc: number for case
% user must first complete the input data script in the +Scenario folder.
% Example:
% To import input data from +Scenario/S1C1.m,
% run BN_Flow(1,1) with ns = 1 and nc = 1.
% Simulation results are saved in Simulation/Scenario1/Case1 folder
%            under different version number starting V1.
%            If we run same example for multiple times, we will have
%            multiple folders with names V1,V2,....
%-------------------------------------------------------------------------%
% See, e.g., an inputcard +Scenario/S1C1.m for details about input data
%-------------------------------------------------------------------------%
% Outputs:
% end_clock: the total computational time
% This simulator also saves the plots of numerical solutions
%      in the appropriate folder.
%-------------------------------------------------------------------------%
```

```
% supp.input_data reads the chosen example input data and translate into
%                       appropriate units and format
% start_clock & end_clock measures the total computational time
% T is the final time constraint
% flag.restart is an input flag in +Scenario/S1C1.m which allows
%                 to restart the simulation from the last recorded data
%                 when flag.restart = 1
% [B0,N0,L0] are the initial biomass, nutrient, and Lagrange multiplier
% G.b0 is the indicator for biofilm phase where B_*≤ B ≤ Bˆ*
% flag.Gb is empty until we first see the biofilm formation
%            which can skip calculating the flow
%            while there is no change in permeability
% G.kb is the permeability of biofilm defined in +Scenario/S1C1.m
% supp.BN_Brinkman solves the coupled flow and biomass-nutrient dynamics
%                  when G.kb > 0: biofilm is (partially) permeable.
%                  We use the heterogeneous Brinkman flow.
% supp.BN_Stokes solves the coupled flow and biomass-nutrient dynamics
%                when G.kb = 0: biofilm is impermeable.
%                We use the Stokes flow.
% supp.record records data: [B,N,L] and [U,V,P,K] if flow is enabled
% supp.figGen saves figures: B, N, and velocity, P if flow is enabled
%=========================================================================%
```

```
+Scenario/S1C1.m
%% Input parameters for Biomass-nutrient-flow solver
%% flags
flag.restart = 0;   % restart trigger
flag.adv = 0;       % advection trigger
flag.DR = 0;        % diffusion-reaction trigger
flag.flow = 1;      % flow only

%% constraints
num.T = 1e2;        % max time in [h]
num.dtmax = 1e-3;   % max time step size [h]
num.tnmax = 1e3;    % max number of time steps
num.flow = 1;       % how often we solve for flow
num.tau = 1;        % number of minor time steps within each major time steps
num.rec = 10;       % how often we record data
num.plot = 10;      % how often we generate figure

%% unit conversion factors from [m,s,kg] to ...
unit.str_len = 'mm'; unit.length = 1e3;
unit.str_time = 'h'; unit.time = 1./(3600);
unit.str_mass = 'g'; unit.mass = 1e3;

%% space in [m]
x.lo = 0;       x.hi = 1000e-6;     x.n = 50;
y.lo = 0;       y.hi = 1000e-6;     y.n = 50;

%% geometry
pm = 3;
switch pm
    case 0; porous_medium = 'channel';
    case 1; porous_medium = 'rock1';
    case 2; porous_medium = 'double_channel';
    case 3; porous_medium = 'corner_nonsymmetric';
    case 4; porous_medium = 'Symmetric';
    case 5; porous_medium = 'Nonsymmetric';
    case 6; porous_medium = 'porous1';
end

obst = 3;
switch obst
    case 0; obstacle = 'none';
    case 1; obstacle = 'obstacle1';
    case 2; obstacle = 'channel_biofilm2';
    case 3; obstacle = 'corner_nonsymmetric_biofilm';
    case 4; obstacle = 'Symmetric_biofilm';
    case 5; obstacle = 'Nonsymmetric_biofilm';
    case 6; obstacle = 'biofilm2';
end

kb = 1e-9; % permeability of obstacle

%% parameters for flow and advection
% Dirichlet inlet, natural outflow, no-slip walls
flow.um = 1e-9;    % mean flow velocity
flow.uf = @(LB,UB,val,Um) -6.*Um.*(LB-val).*(UB-val)./(LB-UB).^2; % fully ...
    developed parabolic velocity
flow.dir = 3; % flow direction; 1: B->T, 2: T->B, 3: L->R, 4: R->L
flow.cfl = 0.95;
flow.mu = 8.9e-4; % dynamic viscosity of water [Pa-s]

%% parameters for diffusion-reaction
% biomass
b.kappa = .5;
b.star = 1;
b.nu = 0.9;
b.init = 0.9.*b.star;
b.eberla = 2;
b.dB0 = 1e-4.*unit.time./unit.length.^2;
b.eberlstar = 1.01;
% nutrient
n.kappa = 2.*unit.time;
n.inlet = 1;
if flag.adv == 1; n.init = 0; else; n.init = n.inlet; end
n.N0 = 1.18e-3;
n.df = 6.*unit.time./unit.length.^2;
n.db = 0.1.*n.df;
```

2.7. **Hard-coded parameters.** User can hard-code the diffusion coefficients d_B and d_N in supp.DR2D.m.

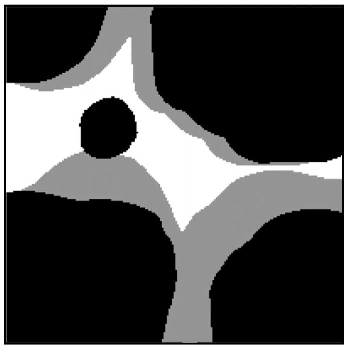
```
%% diffusion coefficients
% biomass (Eberl)
Bbar = B0./b.star;
G.dB = b.dB0.*(Bbar./(b.eberlstar -Bbar)).^b.eberla;
```

```
% linear nutrient diffusivity by [Pirt]
G.dN = n.df.*(b.star-Bbar) + n.db.*Bbar;
% Piecewise constant nutrient diffusivity used in [PTISW]
% G.dN = n.df.*G.rock;
% G.dN(G.b0' == 1) = n.db;

G.dN(G.rock_id) = 0;
```

2.8. **Examples.**

Example 1 (Biofilm growth in the nutrient-rich system in a micro-pore)**.** *Consider the scenario 1 case 1 (+scenario/S1C1.m);* $ns = 1, nc = 1$. *We simulate the biofilm plugging the single-pore geometry:*



with Ω_r *in black,* Ω_b *in gray, and* Ω_0 *in white. We run following line:*

```
>> [end_clock] = BN_Flow(1,1);
```

The evolution results are saved in Simulation/Scenario1/Case1/V1/ under plt0 folders with the numeric representing time steps. Fig. 3 shows the evolution of biomass and nutrient dynamics.

Example 2 (Biofilm growth in the nutrient-rich system in a macro-pore)**.** *Consider the scenario 1 case 2 (+scenario/S1C2.m);* $ns = 1, nc = 2$. *All input parameters are the same as before in Ex. 1 except* $x.hi = y.hi = 1000e6[\mathrm{m}]$ *We run following line:*

```
>> [end_clock] = BN_Flow(1,2);
```

The evolution results are saved in Simulation/Scenario1/Case2/V1/ under plt0 folders with the numeric representing time steps. Fig. 4 shows the evolution of biomass and nutrient dynamics.

Example 3 (Coupled biomass-nutrient-flow dynamics in a macro-pore)**.** *Consider the scenario 2 case 1 (+scenario/S2C1.m);* $ns = 2, nc = 1$. *All input parameters are the same as before in Ex. 2, but now we turn on the advection: set* $flag.adv = 1$. *Fig. 4 shows the evolution of biomass and nutrient dynamics with flow. We run following line:*

```
>> [end_clock] = BN_Flow(2,1);
```

The evolution results are saved in Simulation/Scenario2/Case1/V1/ under plt0 folders with the numeric representing time steps.

References

[1] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge texts in applied mathematics. Cambridge University Press, Cambridge ; New York, 2002.

[2] S.V. Patankar. *Numerical heat transfer and fluid flow.* Series in Computational Methods in Mechanics and Thermal Sciences. Routledge, 1980.

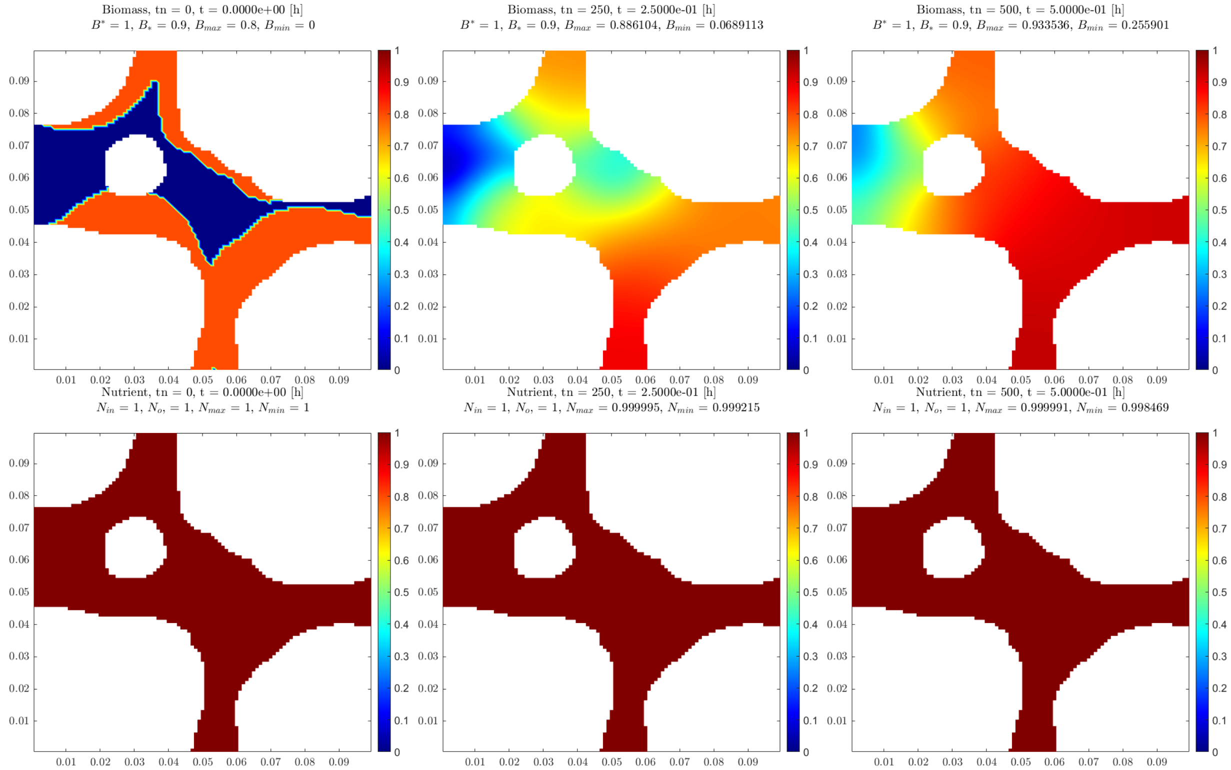


FIGURE 3. Selected figures for Ex. 1. Top: Biomass and bottom: nutrient at $t \in \{0, 0.25, 0.5\}$[h] from left to right.

[3] T. F. Russell and M. F. Wheeler. Finite element and finite difference methods for continuous flows in porous media. In R. E. Ewing, editor, *The Mathematics of Reservoir Simulation*, pages 35–106. SIAM, Philadelphia, 1983.

[4] Choah Shin. Coupled biomass-nutrient-flow simulation code for bio2020 project.

[5] Choah Shin, , Azhar Alhammali, Lisa Bigler, Naren Vohra, and Malgorzata Peszynska and. Coupled flow and biomass-nutrient growth at pore-scale with permeable biofilm, adaptive singularity and multiple species. *Mathematical Biosciences and Engineering*, 18(3):2097–2149, 2021.

[6] Stephen Whitaker. Flow in porous media i: A theoretical derivation of darcy's law. *Transport in Porous Media*, 1(1):3–25, 1986.

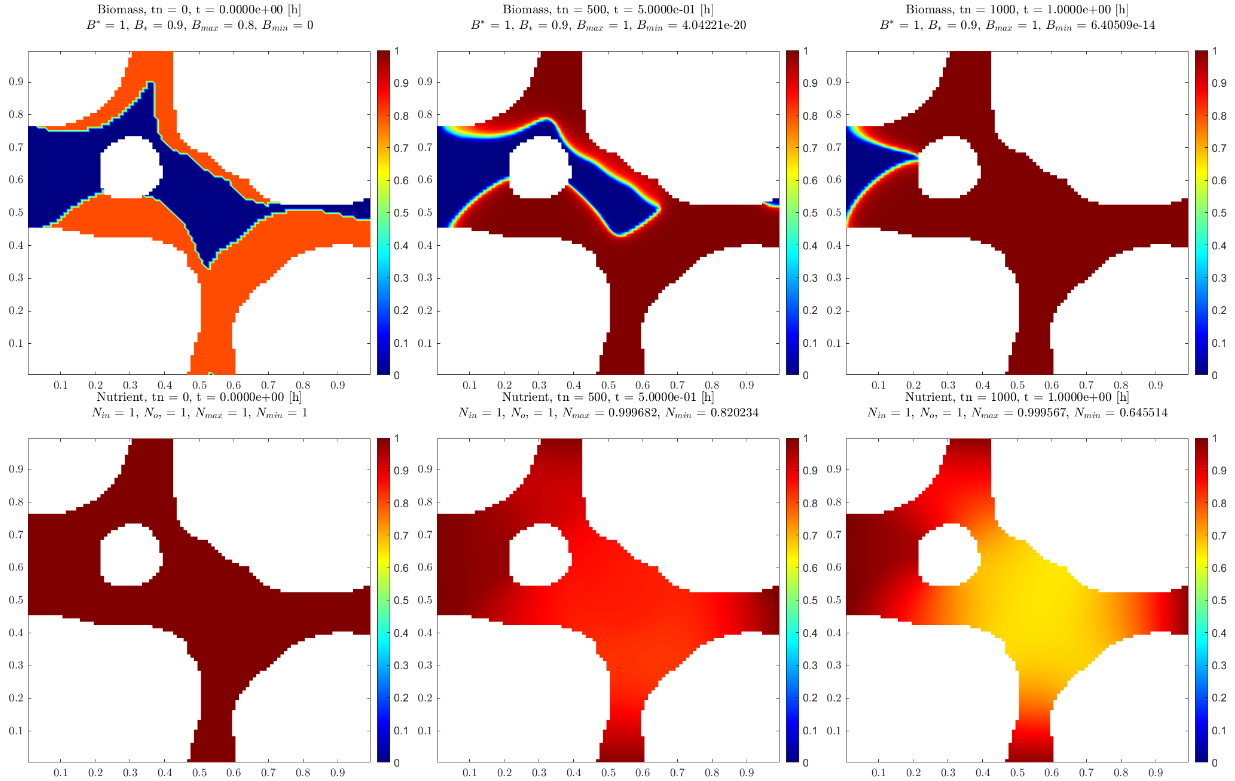
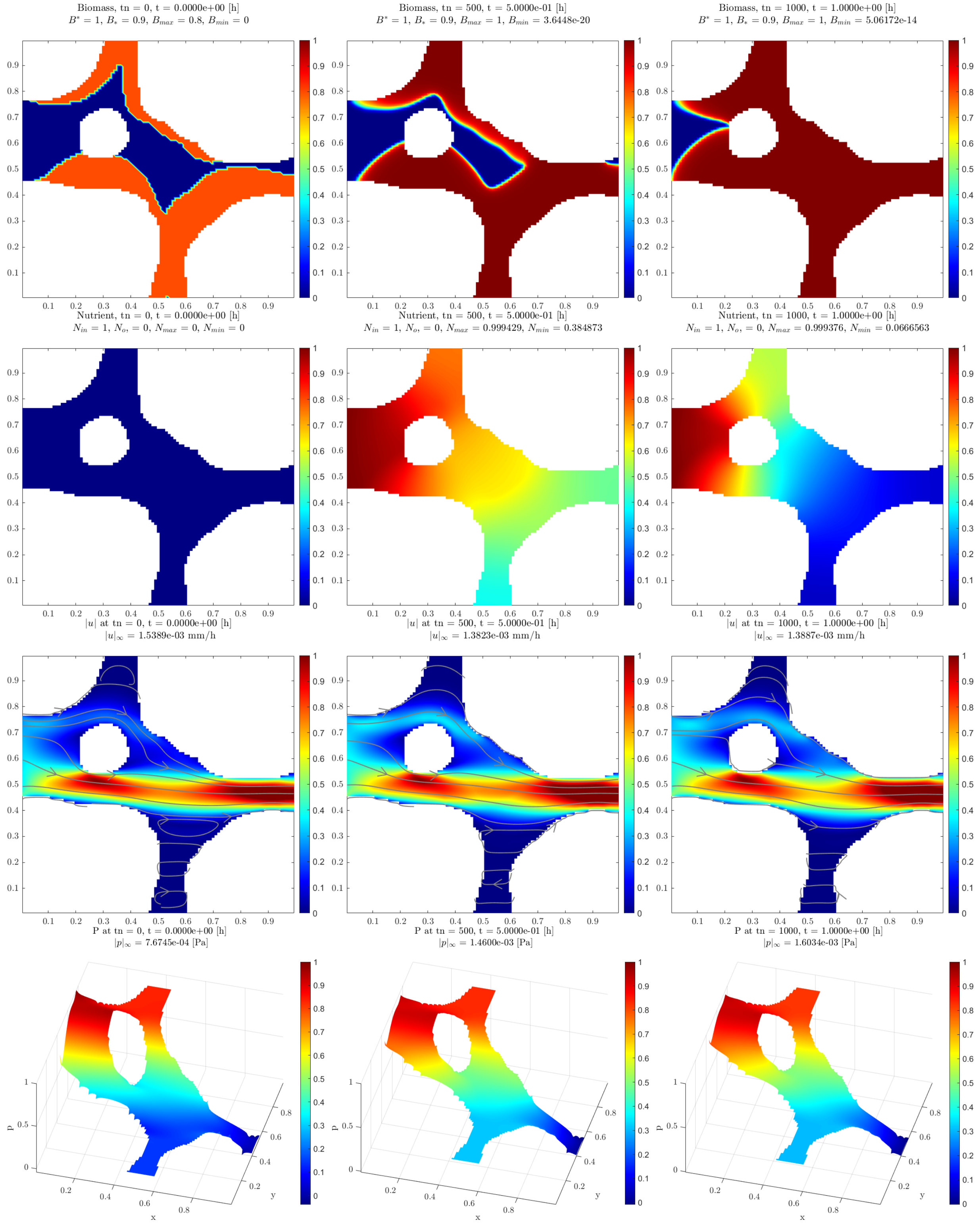


FIGURE 4. Selected figures for Ex. 2. Top: Biomass and bottom: nutrient at $t \in \{0, 0.5, 1\}$[h] from left to right.


Biomass, tn = 0, t = 0.0000e+00 [h]
$B^* = 1$, $B_* = 0.9$, $B_{max} = 0.8$, $B_{min} = 0$

Biomass, tn = 500, t = 5.0000e-01 [h]
$B^* = 1$, $B_* = 0.9$, $B_{max} = 1$, $B_{min} = 3.6448\text{e-}20$

Biomass, tn = 1000, t = 1.0000e+00 [h]
$B^* = 1$, $B_* = 0.9$, $B_{max} = 1$, $B_{min} = 5.06172\text{e-}14$

Nutrient, tn = 0, t = 0.0000e+00 [h]
$N_{in} = 1$, N_o, $= 0$, $N_{max} = 0$, $N_{min} = 0$

Nutrient, tn = 500, t = 5.0000e-01 [h]
$N_{in} = 1$, N_o, $= 0$, $N_{max} = 0.999429$, $N_{min} = 0.384873$

Nutrient, tn = 1000, t = 1.0000e+00 [h]
$N_{in} = 1$, N_o, $= 0$, $N_{max} = 0.999376$, $N_{min} = 0.0666563$

$|u|$ at tn = 0, t = 0.0000e+00 [h]
$|u|_\infty = 1.5389\text{e-}03$ mm/h

$|u|$ at tn = 500, t = 5.0000e-01 [h]
$|u|_\infty = 1.3823\text{e-}03$ mm/h

$|u|$ at tn = 1000, t = 1.0000e+00 [h]
$|u|_\infty = 1.3887\text{e-}03$ mm/h

P at tn = 0, t = 0.0000e+00 [h]
$|p|_\infty = 7.6745\text{e-}04$ [Pa]

P at tn = 500, t = 5.0000e-01 [h]
$|p|_\infty = 1.4600\text{e-}03$ [Pa]

P at tn = 1000, t = 1.0000e+00 [h]
$|p|_\infty = 1.6034\text{e-}03$ [Pa]

FIGURE 5. Selected figures for Ex. 3. From top to bottom, we have biomass, nutrient, velocity magnitude, and pressure profiles. From left to right, the results at $t \in \{0, 0.5, 1\}$[h] are shown.