Bristol iGEM 2017 Modelling Supplement

Jérémie Joannès & Albert Wigmore

August 2017

This document has the purpose of describing the mathematical and technical aspects of Bristol's modelling as part of iGEM 2017. Two main models were created and run: an atmospheric model, to predict the location and concentration of NOx gases and a single-cell model to predict gene expression and impact of this on NOx reduction. The document should be read as a supplement to the overview and results reported in the wiki.

1 Atmospheric Modelling

The goal of the atmospheric model is to estimate concentrations of NOx throughout Bristol as they are emitted, spread and moved by the wind. Although NO and NO_2 are unstable, switching from one to the other, together they can be considered as a stable gas as they do not react significantly with other parts of the atmosphere. Sources are assumed to be roads emitting concentration at a particular rate, whilst diffusion and advection are to be modelled. The highly diffusive effects of turbulence are ignored—thus this model will over-predict NOx concentrations.

An appropriate equation for this model is therefore the diffusion-advection equation, shown below in its most general form:

$$\frac{\partial f}{\partial t} = \nabla \cdot (\mathbf{C} \nabla f) - \nabla \cdot (\underline{u} f) + S(\underline{r}, t). \tag{1}$$

It implies that the rate of change of a property (here f would be a concentration) is equal to a diffusion related to the second spacial gradient of concentration minus an advection related to the first gradient of the concentration and a velocity field $\underline{u}$. A time- and space-varying source S is also included to model roads emitting NOx. As with $\frac{\partial f}{\partial t}$, this has units of concentration/time. Assuming the diffusivity is homogeneous, i.e. $\mathbf{C} = c$, that the velocity field is also homogeneous, i.e. $\frac{\partial \underline{u}}{\partial \underline{r}} = 0$, and finally that the source vector is time-invariant, then the above equation reduces to

$$\frac{\partial f}{\partial t} = c\nabla^2 f - \underline{u} \cdot \nabla f + S(\underline{r}). \tag{2}$$

Expanded into two cartesian dimensions, the advection-diffusion equations become

$$\frac{\partial f}{\partial t} = c\left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right) - \left(u\frac{\partial f}{\partial x} + v\frac{\partial f}{\partial y}\right) + S(x, y). \tag{3}$$

From this it appears that the changes in the quantity f in one direction is independent from that in the other direction, so directions can be treated separately. To solve this equation, a "Forward Time, Central Space" (FTCS) finite difference scheme is derived using Taylor expansions [2]. With such a scheme, derivatives can be approximated as follows for the x-dimension and time (the y-dimension follows similarly):

$$
\begin{aligned}
\left.\frac{\partial f}{\partial t}\right|_i^n &= \frac{f_i^{n+1} - f_i^n}{\Delta t} \\
\left.\frac{\partial f}{\partial x}\right|_i^n &= \frac{f_{i+1}^n - f_{i-1}^n}{2\Delta x} \\
\left.\frac{\partial^2 f}{\partial x^2}\right|_i^n &= \frac{f_{i+1}^n + f_{i-1}^n - 2f_i^n}{\Delta x^2}
\end{aligned}
\tag{4}
$$

where superscripts indicate time and subscripts indicate space. Equations (4) can be substituted into equation (3), then rearranged to yield an expression for the value of f at time $n+1$ as a function of the current time n:

$$
\begin{aligned}
f_{i,j}^{n+1} = f_{i,j}^n + c\Delta t &\left(\frac{f_{i+1,j}^n + f_{i-1,j}^n - 2f_{i,j}^n}{\Delta x^2} + \frac{f_{i,j+1}^n + f_{i,j-1}^n - 2f_{i,j}^n}{\Delta y^2} \right) \\
-\Delta t &\left(u\frac{f_{i+1,j}^n - f_{i-1,j}^n}{2\Delta x} + v\frac{f_{i,j+1}^n - f_{i,j-1}^n}{2\Delta y} \right) + S_{i,j}\Delta t
\end{aligned}
\tag{5}
$$

An initial solution (e.g. a NOx concentration of 0 throughout all the x-y) can be guessed at, then equation (5) predicts the state of concentration at the next timestep and the simulation can be run for as long as needed. This was programmed in Matlab for fast development and troublshooting as well as availability of rich plotting features, although this limits the runtime of the code.

Matlab has an inbuilt Gaussian noise function, allowing the possibility to add noise to both the wind and the source strenghts. The model was then run several times to observe the effect of noise, as well as changing the nominal wind direction.

Final considerations:
It can be shown with Von Neumann stability analysis that the FTCS scheme derived above is unconditionally unstable for the advection part. This means that whatever the choice of Δx or Δy or Δt, the solution will grow exponentially as a result not of the physics, but of the method used to solve the problem. However, it is common in the field of computational fluid dynamics to add an artificial dissipation term which damps out this explosion of the solution. Given that dissipation is part of the physical problem we wish to model, it is already being modelled without having to be added artificially. Thus, the model with both dissipation and advection is found to be conditionally stable (i.e. subject to appropriate Δx, Δy and Δt).

2 Single Cell Modelling

This model is described graphically below; the inputs are nitrite and nitrate in solution as well as the inducer IPTG, the output is ammonium. An alternative representation of this denitrification pathway is given in our Parts page.

Within the cytoplasm, the IPTG induces expression of the Nap and Nrf operons through a gene regulatory network (GRN), which has yet to be designed. This network yields the concentrations of enzymes expulsed into the the periplasm. Here, Nap then reduces nitrate to nitrite, which can be converted to ammonium by the Nrf enzyme.

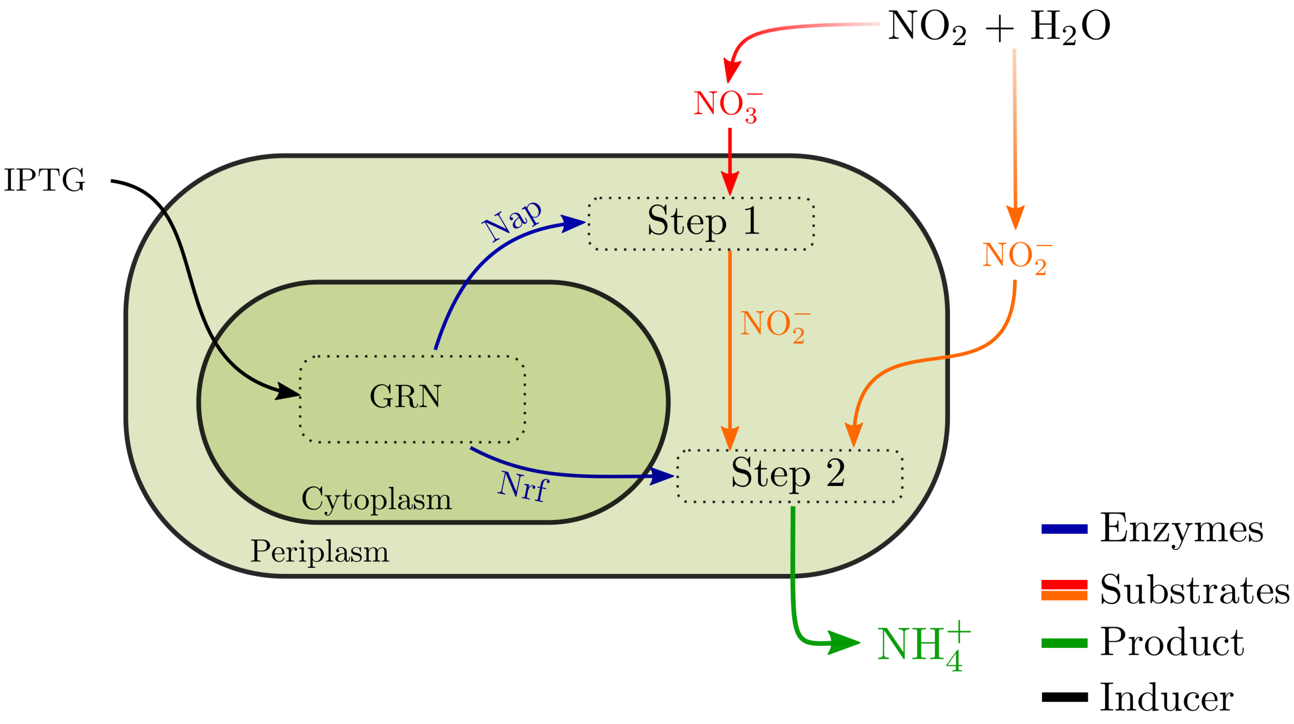

Figure 1: Overview of the single-cell model

2.1 Gene Expression Model

To model gene expression, it is important to understand the high-level reactions occurring within the cytoplasm. These are found from [3] and can be reiterated as:

$$\begin{aligned}
&\text{DNA} \xrightarrow[\text{Transcr.}]{k_1} \text{mRNA} \xrightarrow[\text{Transl.}]{k_2} \text{Protein}\\
&\text{mRNA} \xrightarrow[\text{Degrad.}]{d_1} \emptyset\\
&\text{Protein} \xrightarrow[\text{Degrad.}]{d_2} \emptyset
\end{aligned} \tag{6}$$

The gene expression model created in this project is simple and free of any known feedback loops; two operons, both controlled by IPTG-inducible promoters, contain the genes that encode our enzymes. The rate of transcription is controlled by this inducer, so in terms of equation (6), repression and induction kinematics will be captured by the transcription rate k_1.

Being unsure of whether IPTG should be introduced continuously or in impulses, and in which concentrations, the model to link enzyme concentration (output) to IPTG concentration (input) is created as an ordinary differential equation (ODE), which can capture time-varying effects. This ODE can be created using mass-action kinetics as follows:

$$\frac{d[\text{mRNA}]}{dt} = k_1[\text{DNA}] - d_1[\text{mRNA}] \tag{7}$$

$$\frac{d[\text{Protein}]}{dt} = k_2[\text{mRNA}] - d_2[\text{Protein}] \tag{8}$$

Whilst initial DNA concentrations are known (assuming a known plasmid copy number per cell), the constants are not established but can be derived as below.

1. **Calculation of the transcription rate k_1**
   This calculation is based on the derivation in [1], bearing in mind that transcription rate is

identical for both operons as they use the same promoter. The resulting transcription rate k_1 is derived as

$$k_1 = \frac{\beta}{1 + \dfrac{[\mathrm{X_T}]}{K_\mathrm{d}} \dfrac{1}{1 + [\mathrm{S_x}]/K_\mathrm{x}}}, \tag{9}$$

where $\mathrm{X_T}$ is the total repressor protein LacI (both bound and free), $\mathrm{S_x}$ is the inducer IPTG that binds X, K_d is the dissociation constant of DNA site and repressor protein and K_x is the dissociation constant of repressor and inducer. Finally, β is the maximum transcription rate; equation (9) therefore represents a maximum transcription rate moderated by the probability of an unrepressed DNA site. This maximum rate β is simply the the polymerase rate per nucleotide divided by the number of nucleotides in the operons.

2. **Translation rate k_2 from the literature**

3. **mRNA degradation rate d_1 from the literature**

4. **Protein degradation rate d_2 from the literature**

Given the constants as well as k_1, which varies as a function of the IPTG and LacI, the differential equations (7) and (8) can be simulated in MATLAB's in-built ODE solver. Unfortunately, it was found late in the process that with values from the literature, the model becomes unstable, such that it could not be reliably integrated with the following models.

2.2 Enzyme Kinetics

Having calculated the concentrations of Nrf and Nap enzymes produced within the cytoplasm by means of a GRN or by direct measurement, it is now of interest to model how they catalyse the reduction of nitrate to nitrite to ammonia in the periplasm. A high-level view of these interactions is given in Figure 1. As is typical for enzymatic reactions, we model the two reductions with the individual Michaelis-Menten chemical equations (10) and (11).

$$\text{Step 1: Nitrate to nitrite} \quad \mathrm{Nap} + \mathrm{NO_3^-} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{Nap} \cdot \mathrm{NO_3^-} \xrightarrow{k_2} \mathrm{Nap} + \mathrm{NO_2^-} \tag{10}$$

$$\text{Step 2: Nitrite to ammonium} \quad \mathrm{Nrf} + \mathrm{NO_2^-} \underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}} \mathrm{Nrf} \cdot \mathrm{NO_2^-} \xrightarrow{k_4} \mathrm{Nrf} + \mathrm{NH_4^+} \tag{11}$$

ODEs are typically derived from the above chemical equations in order to simulate their interactions. An example derivation of an ODE given a chemical equation is given below, and forms the basis for the derivations of the system of ODEs for the nitrate-nitrite-ammonium chemical equations:

$$\mathrm{A} + \mathrm{B} \underset{k_{off}}{\overset{k_{on}}{\rightleftharpoons}} \mathrm{C} \quad \Rightarrow \quad \frac{d[\mathrm{C}]}{dt} = k_{on}[\mathrm{A}][\mathrm{B}] - k_{off}[\mathrm{C}]$$

Based on this, individual rates of production of each species can be derived accordingly from the chemical equations (10) and (11) to form a system of nonlinear ODEs:

$$\begin{aligned}
\frac{d[\mathrm{NO}_3^-]}{dt} &= k_{-1}[\mathrm{Nap}\cdot\mathrm{NO}_3^-] - k_1[\mathrm{Nap}][\mathrm{NO}_3^-] \\
\frac{d[\mathrm{Nap}]}{dt} &= k_{-1}[\mathrm{Nap}\cdot\mathrm{NO}_3^-] - k_1[\mathrm{Nap}][\mathrm{NO}_3^-] + k_2[\mathrm{Nap}\cdot\mathrm{NO}_3^-] \\
\frac{d[\mathrm{Nap}\cdot\mathrm{NO}_3^-]}{dt} &= k_1[\mathrm{Nap}][\mathrm{NO}_3^-] - k_{-1}[\mathrm{Nap}\cdot\mathrm{NO}_3^-] - k_2[\mathrm{Nap}\cdot\mathrm{NO}_3^-] \\
\frac{d[\mathrm{NO}_2^-]}{dt} &= k_2[\mathrm{Nap}\cdot\mathrm{NO}_3^-] + k_{-3}[\mathrm{Nrf}\cdot\mathrm{NO}_2^-] - k_3[\mathrm{Nrf}][\mathrm{NO}_2^-] \\
\frac{d[\mathrm{Nrf}]}{dt} &= k_{-3}[\mathrm{Nrf}\cdot\mathrm{NO}_2^-] - k_3[\mathrm{Nrf}][\mathrm{NO}_2^-] + k_4[\mathrm{Nrf}\cdot\mathrm{NO}_2^-] \\
\frac{d[\mathrm{Nrf}\cdot\mathrm{NO}_2^-]}{dt} &= k_3[\mathrm{Nrf}][\mathrm{NO}_2^-] - k_{-3}[\mathrm{Nrf}\cdot\mathrm{NO}_2^-] - k_4[\mathrm{Nrf}\cdot\mathrm{NO}_2^-] \\
\frac{d[\mathrm{NH}_4^+]}{dt} &= k_4[\mathrm{Nrf}\cdot\mathrm{NO}_2^-]
\end{aligned}$$

Rate constants are not typically given in the form k_1, k_{-1} etc., rather in the specific case of a Michaelis-Menten reaction, they are quoted in the literature as k_{cat}, k_{d} and K_{m}, which can be related to the earlier nomenclature by

$$\begin{aligned}
k_{\mathrm{cat}} &= k_2 \\
k_{\mathrm{d}} &= \frac{k_{-1}}{k_1} \\
K_{\mathrm{m}} &= \frac{k_{-1}+k_2}{k_1}.
\end{aligned}$$

Matlab's in-built ODE solver is then used to simulate the reactions, given initial concentrations of Nap and Nrf free enzyme, as well as of nitrite and nitrate (assuming no initial ammonia or enzyme-subrstrate complexes). Based on equations (10) and (11), the reaction will always go to completion for a finite amount of each reagant; this model yields the additional information of time behaviour and allows to interface the model with other time-dependent modules.

References

[1] Yuri Allon. *An Introduction to Systems Biology: Design Principles of Biological Circuits.* Chapman & Hall/CRC, 2007.

[2] Dale Anderson Richard H. Pletcher, John C. Tannehill. *Computational Fluid Mechanics and Heat Transfer.* CRC Press, third edition, 2012.

[3] Guy-Bart Stan. Modelling in Biology. Lecture notes, 2017. `http://www.bg.ic.ac.uk/research/g.stan/2010_Course_MiB_article.pdf`, Accessed 04/08/2017.