

1. The enzyme luciferase is often used as a “reporter” to study delivery of transgenes into cells, for instance in determining how effective delivery vectors (such as different polymer formulations) might be in doing so. This is accomplished by incorporating the gene for luciferase along with your actual transgene into the DNA delivered by the vector; when this gene is transcribed into mRNA and subsequently translated into protein, the expression level of luciferase can be monitored by the amount of light generated (using a luminometer) when the substrate luciferin is added to the cell medium. As a gene delivery vector researcher, you would like to develop a quantitative model for salient aspects of the transgene expression process. You write the following set of mass-action kinetic equations for the levels of luciferase mRNA, x [# / cell], and luciferase protein, y [# / cell]:

$$\frac{dx}{dt} = k_1 \sigma g - k_2 x$$

$$\frac{dy}{dt} = k_3 x - k_4 y$$

where k_1 is the transcription rate constant, g is the luciferase gene copy number, σ is the promoter strength coefficient, k_2 is the message degradation rate constant, k_3 is the translation rate constant, and k_4 is the protein degradation rate constant, relevant for the luciferase transgene. (We assume here that the ribosomal activity coefficient is unity.)

You wish to determine as many of the key model parameters as is feasible from the following battery of experimental data; in particular, you wish to gain an estimate for the “promoter strength coefficient”, σ , since this will presumably govern the transcription of your actual transgene similarly. After allowing your cells in culture to take up the vector for a sufficiently long period of time that they contain on the average g copies of the luciferase transgene per cell, you induce expression by adding a chemical factor to the medium that activates a critical transcription factor for the corresponding promoter. You monitor light generation as a function of time, and find that a steady-state is reached in approximately 8 hours. Having calibrated the amount of light generated on a per-molecule basis, you can ascertain that there are roughly 10,000 luciferase molecules per cell at this steady-state. And doing quantitative PCR and RT-PCR measurements on the luciferase DNA mRNA in a small sample from this steady-state culture, you learn that there are approximately 8 gene copies per cell and about 100 message copies per cell. Now, you shut off synthesis by removing the chemical inducer of transcription, and watch the light generation gradually decrease to essentially zero; it takes about 24 hours for this to occur. Using this set of measurements, see if you can determine all the model parameters – especially the “promoter strength coefficient”, σ . If you cannot, propose additional experiment(s) that would enable you to do so.

2. You are analyzing the prospective operation of a microbial-based process for remediating a hydrocarbon spill. Your microbial population can be characterized empirically as the organic compound $C_{4.4}H_{7.3}N_{0.86}O_{2.2}$, and growing on this hydrocarbon a respiratory coefficient of 0.63 [moles CO_2 formed per moles O_2 consumed] is measured. Use NH_3 as a nitrogen source. If the hydrocarbon is primarily hexadecane ($C_{16}H_{34}$), determine the stoichiometric coefficients for the reaction equation describing its cell-mediated conversion to cell mass. What is the cell-to-substrate mass yield coefficient for this conversion? If the heat of combustion of your microbial cells is approximately 6 kcal per g dry cell mass, estimate the heat yield coefficient, Y_H , for this process. (You can find carbohydrate heats of combustion on this website: <http://members.nuvox.net/~on.jwclymer/rq/>) How much heat will be generated for each ton of hydrocarbon remediated? How much cooling water would need to be evaporated to dissipate this heat?

3. 1. In this MATLAB problem you will learn how to solve algebraic equations and systems of algebraic equations using the `fsolve` function. As you can see by typing “help `fsolve`” at the MATLAB prompt (please do this), `fsolve` solves nonlinear equations by a least squares method and solves equations of the form: $F(x)=0$ where F and X may be vectors or matrices.

To learn about how this method works, please download the files `fsolve_demo.m` and `myfun.m` to see a demo where we solve the system of equations:

$$x^2 + 2*y^2 = 3$$

$$y = 5*x^2 + 6*x + 1$$

Note that the solution depends strongly on the chosen initial guess! I get the correct (colored green) solutions when I choose initial guesses of $[1,1]$ or $[-1,1]$ but I get an incorrect solution when I choose an initial guess of $[-1,1]$. This is because, as the output says: “Optimizer is stuck at a minimum that is not a root” – the optimizer just minimizes the sum of squares of the residuals to find a minimum, and this wrong solution is at a minimum. Please read about Newton’s method (the Wikipedia entry has a pretty good picture of how it works).

Now, let's look at a real system and see if we can use `fsolve` to find the steady-state solution. In class, we wrote down a detailed set of equations on 4/24/06 describing endocytic trafficking and receptor-ligand dynamics. We then proceeded to make a series of assumptions to gain a conceptual understanding of the factors influencing trafficking and dynamics. On 4/26/06 we looked at receptor downregulation but neglected recycling and assumed that internalized receptor and ligand are degraded essentially immediately. Now, we will try to relax these assumptions and solve the full set of equations for the steady-state equations, using MATLAB.

a. The invasive migration of tumor cells is driven by EGF/EGFR complex signaling from the cell plasma membrane, so we would like you to calculate the steady-state concentration of receptor-ligand (EGFR/EGF) complexes on the surface of migration tumor cells (all dynamic equations will be at steady-state). The ligand (EGF) levels can be assumed to be constant in this analysis since there is an ongoing source from the surrounding tissue. The equations that describe this system are the same as we wrote in class, and the equations and necessary parameters are included in the file `pset8_matlab.m`. Please solve this system of equations for their steady-state values, using the `fsolve` function in MATLAB, and try using as an initial guess (all in units of #/cell): $2e3$ for R_s , $3e3$ for C_s , $3e3$ for R_i , $3e3$ for C_i , and $3e-9$ for L_i .

Please give the best solution that you get. Personally, I was unable to find a good solution using `fsolve` here, (Optimizer is stuck at a minimum that is not a root). However, we do know that the solution found by `fsolve` is one where the residuals are small, even if it has not converged. Please comment on why you think MATLAB may not be able to find the steady-state solution using `fsolve`. (Or, if you manage to get MATLAB to converge, let us know and you can get an extra-credit point! There probably is a way to do it, and it probably involves changing the `optimset` variables.)

b. Please solve the system from part a again, this time using an initial guess of 1 (#/cell) for each species. Which of the two solutions do you think is more realistic and why? (Please argue based on realistic parameters and our in-class analysis.)