1 4.2.a+b

The following set of plots correspond to emulation of system E,ES,S,P as specified in 4.1. Code can be found at github.com/cfmcginn/SystBio/HW1. C++ and ROOT needed to run. A directory containing the set of pdfs used in this document is available at the git repository, along with the .tex used to create this, if wanted individually. Time steps of simulation are .0001, sampling done such that 1000 bins appear on the plot. Simulations show little deviation with variation of time steps up and down an order of magnitude.
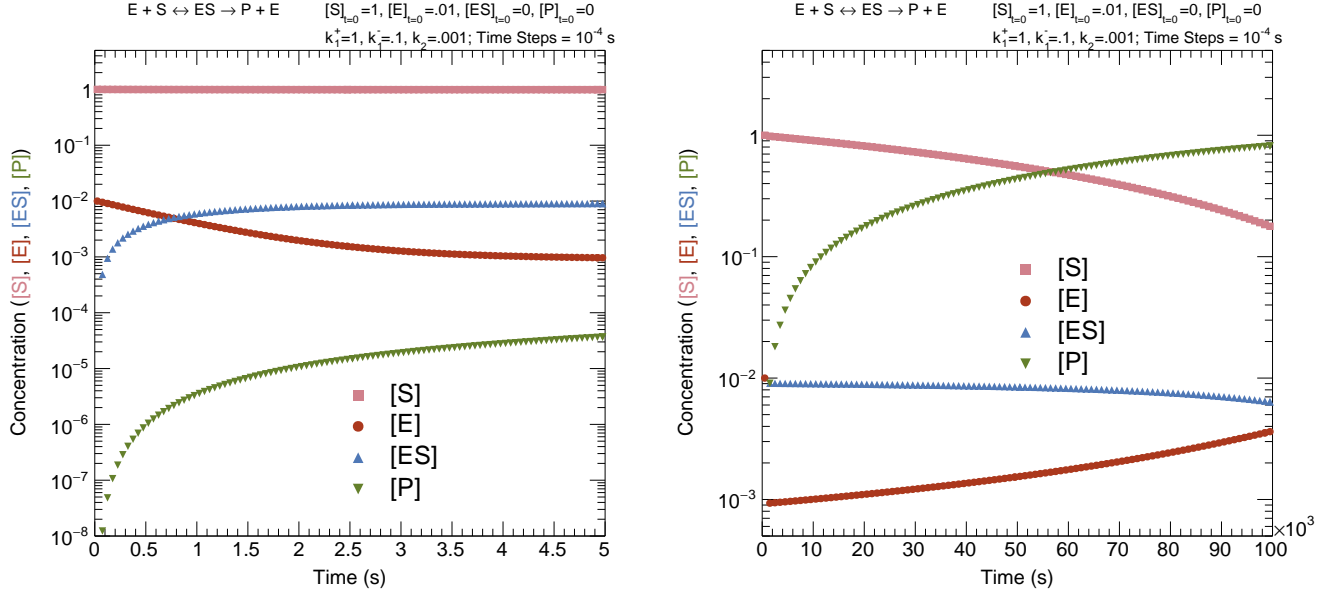

Figure 1: Left: $0 < t < 5$ seconds simulation, all concentrations. Right: $0 < t < 100{,}000$ seconds simulation, all concentrations.
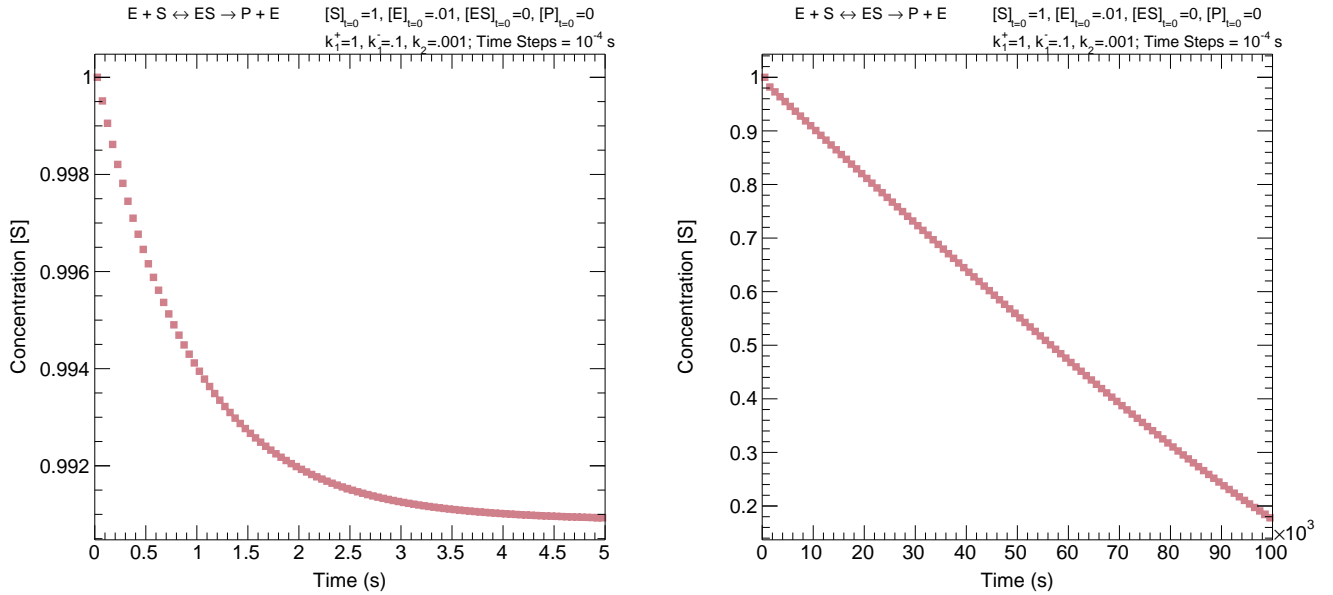

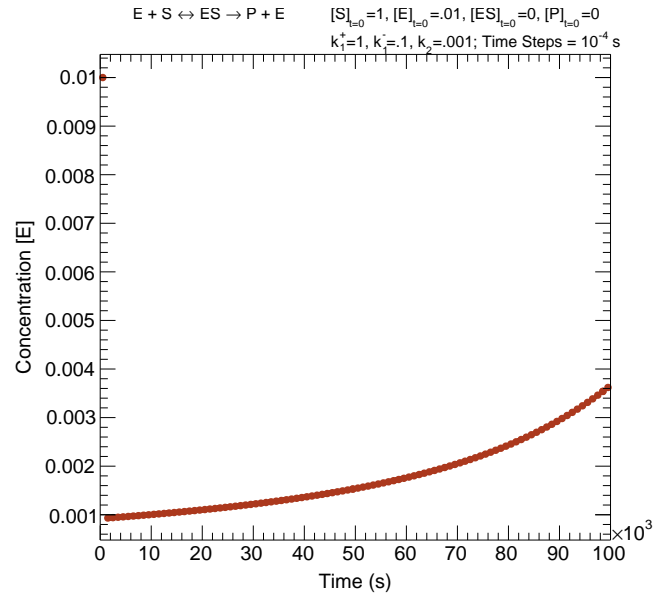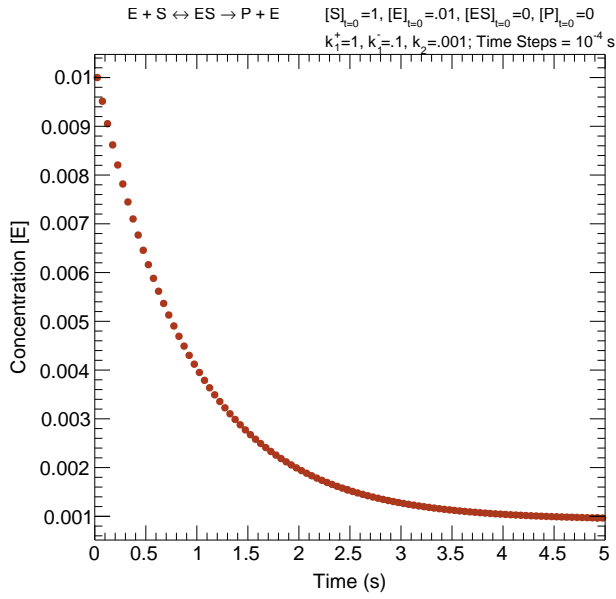Figure 2: Left: $0 < t < 5$ seconds simulation, [S]. Right: $0 < t < 100{,}000$ seconds simulation, [S].

Figure 3: Left: $0 < t < 5$ seconds simulation, [E]. Right: $0 < t < 100,000$ seconds simulation, [E].
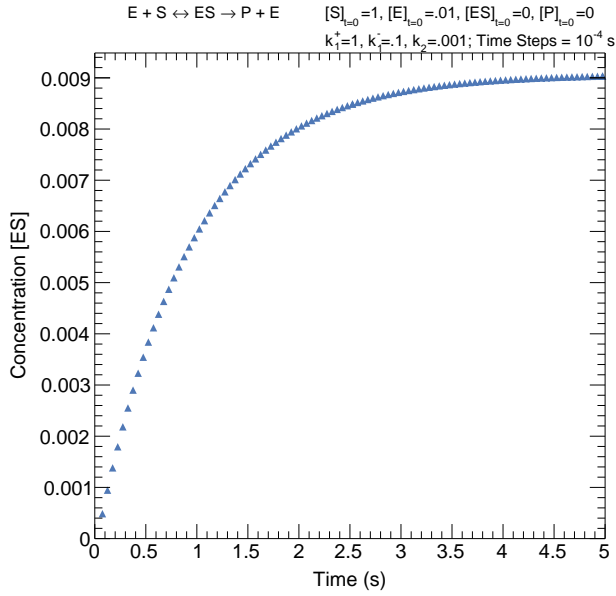
Figure 4: Left: $0 < t < 5$ seconds simulation, [ES]. Right: $0 < t < 100,000$ seconds simulation, [ES].

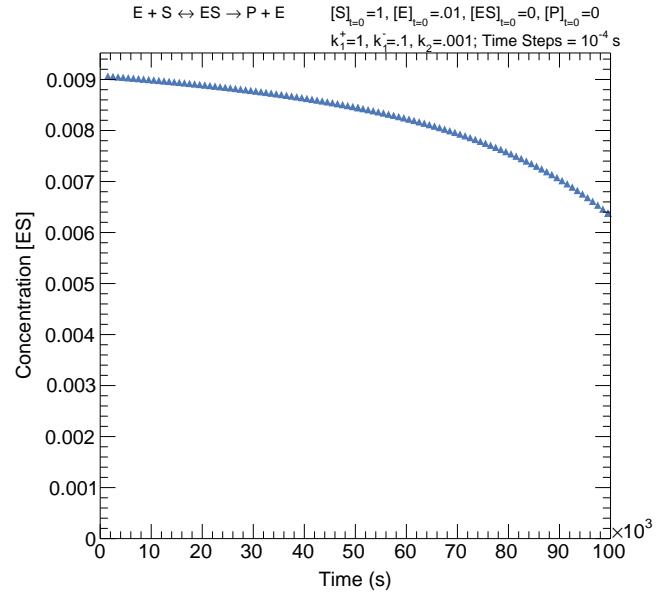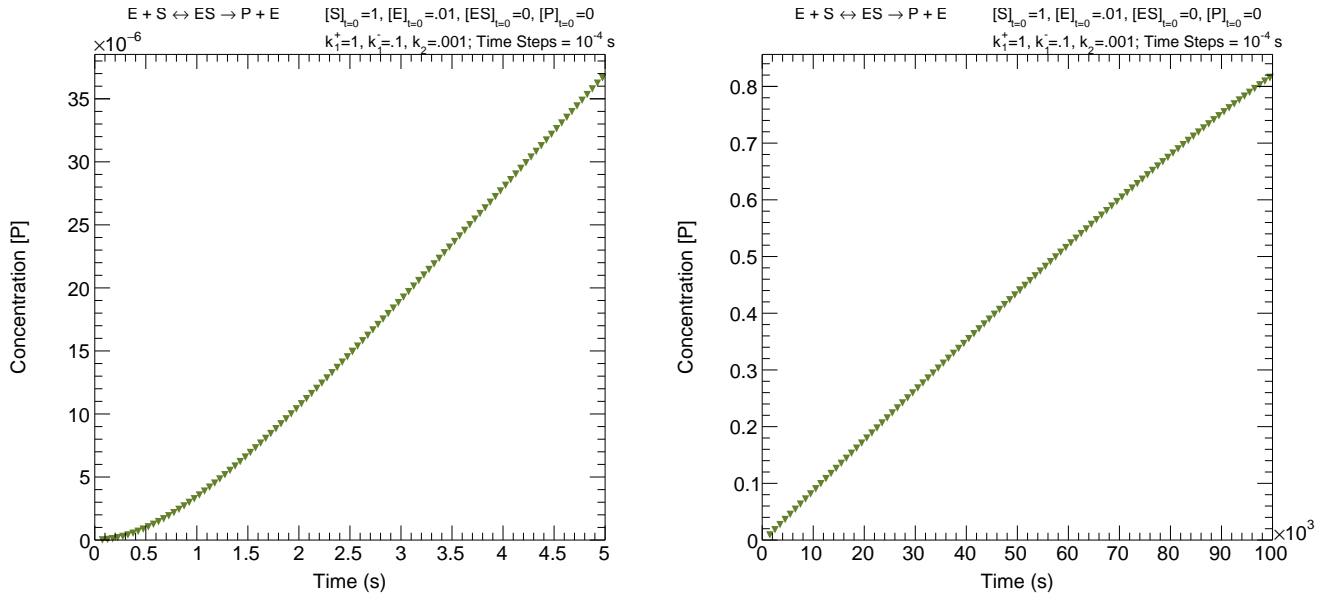Figure 5: Left: $0 < t < 5$ seconds simulation, [P]. Right: $0 < t < 100{,}000$ seconds simulation, [P].
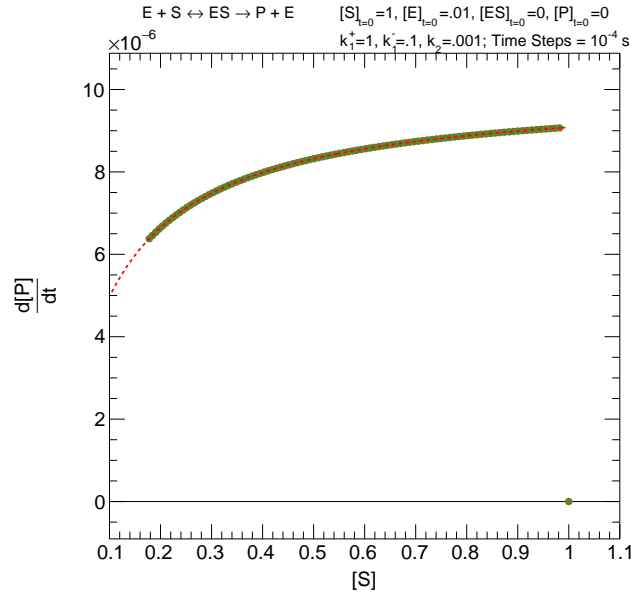
2   4.2.c

Figure 6: Production rate of [P] vs. concentration [S]. Note that the deviation from overlaid Michaelis-Menten (red-dashed curve) at [S] equal to 1 is because the system at that point has not yet reached quasi-equilibrium (see the corresponding set of $t < 5$ concentration plots, rapid change on that epoch).

3   4.2.d

For half-time to quasi-equilibrium [ES], simulation gives $t_{1/2} = .6311$ seconds, compared to my calculation of .63158 seconds, or within .07%. Quasi-equilibrium starting value for [ES] is 0.00903606, compared to calculation of .00908, .5% error.

For 100,000 second simulation, time to half [S] in simulation was 56094.1 seconds, compared to my calculation of 57764.7 seconds, for about a 2% relative error. Overall agreement between calculation and simulation is quite good based on this set of comparisons.

4   4c

The following set of plots correspond to fits of data as specified in 4.3. Code can be found at github.com/cfmcginn/SystBio/HW1. C++ and ROOT needed to run. A directory containing the set of pdfs used in this document is available at the git repository, along with the .tex used to create this, if wanted individually. Fitting is done with a simple chi2 (see code for exact options used w/ ROOT TF1 class).
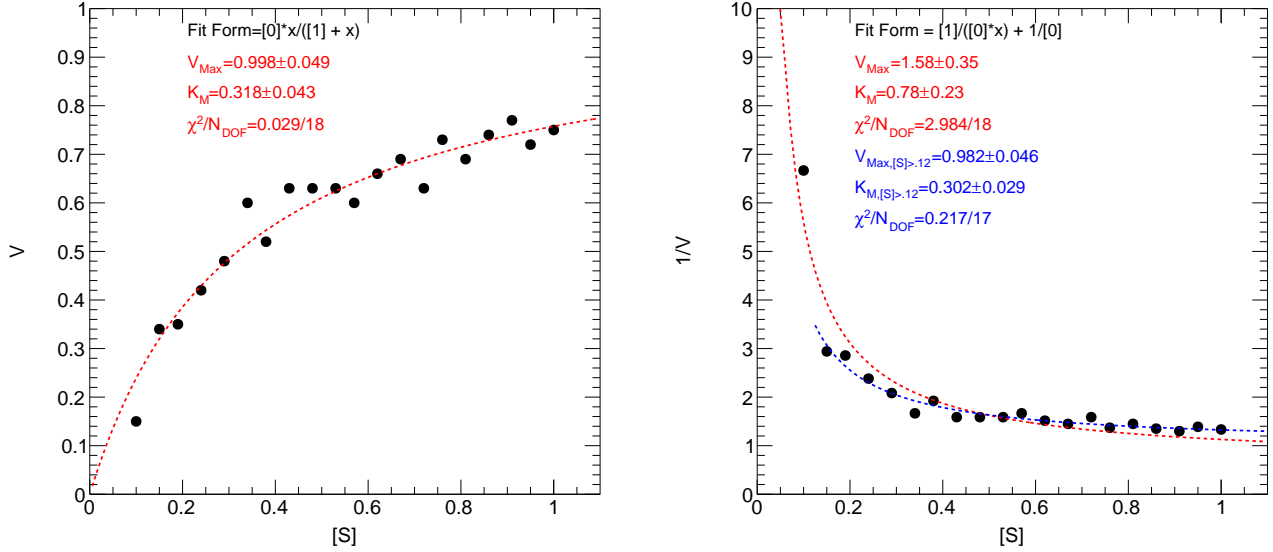

Figure 7: Left: Fit of standard Michaelis-Menten. Right: Fit of inverted Michaelis-Menten. Note: no errors were provided, so we are forced to fit with equal weights. The inverted Michaelis-Menten would return the same fit values if errors were included, as the point causing deviation would have large relative error (low weight) post inversion. Excluding that point in the truncated fit gives perfectly consistent parameters between left and right. Since difference is driven by a single point, which drives parameter errors up significantly, and point exclusion returns previous fit performance, it seems safe to say this is consistent, and inclusion of errors on data points would give consistent results over the full fit range again.