BT5240 Computational Systems Biology

Computational Systems Biology - Assignment 3

N Sowmya Manojna | BE17B007
Department of Biotechnology,
Indian Institute of Technology, Madras

Parameter Estimation

Parameter estimation was done by using multiple initializations and `fminsearch`.

1 Approach

Initially a grid based search was implemented with the range for each of the unknown parameters as [0, 10]. But, as the ODEs weren't integrable at several time instances with integer parameter values, the values from the grid search weren't used.

The parameter space was then scanned using random initializations ranging from (0, 10). The program was run for about 1000 initializations in order to sample the space uniformly. As most of the minimum values reported from the 1000 random initialization fell in this range, the search space was then reduced to (0, 2) for all the three parameters.

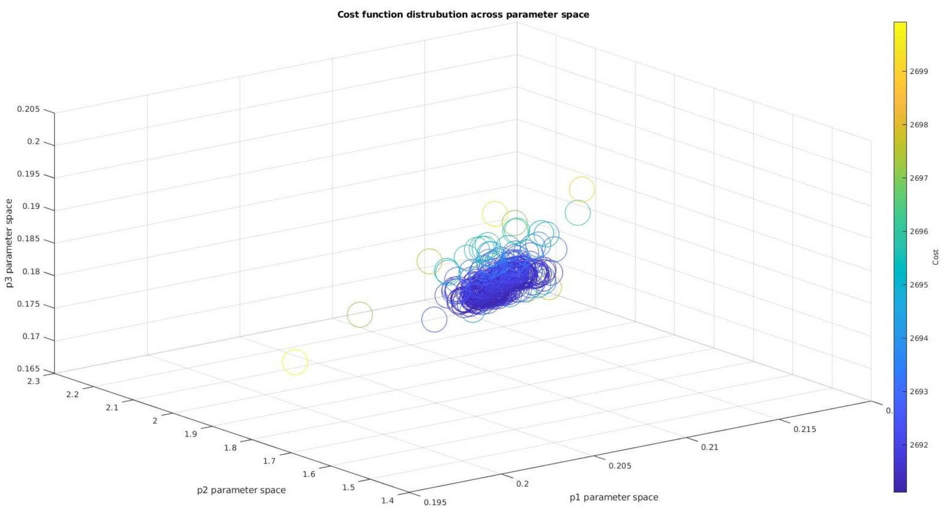

Figure 1: The parameter set that has the lowest cost values from the 1000 iterations (dataset - D1 ∪ D2)

After restricting the search space to (0,2) for all the three parameters, the space was scanned again using 250 random initializations.

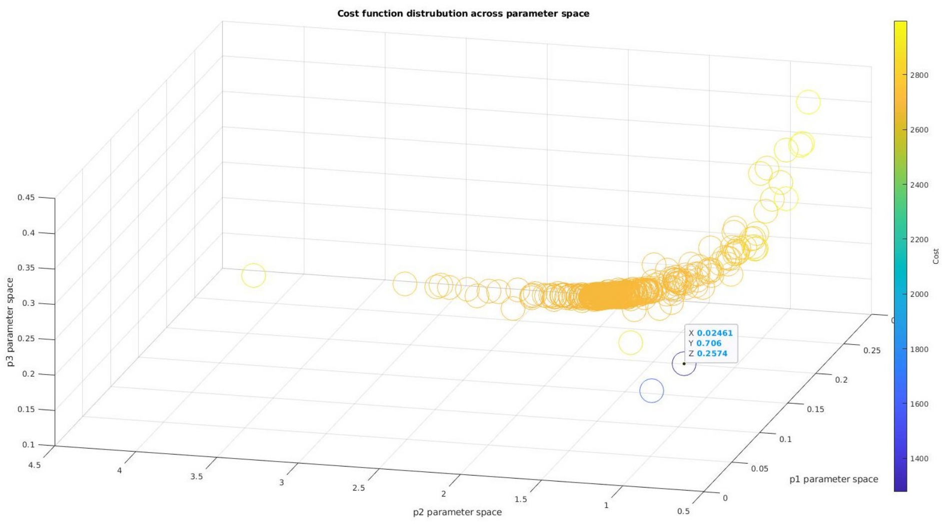

Figure 2: Cost function distribution after restricting the parameter space. The minima obtained in the previous run is now found to be a local minima (indicated by the yellow band) (dataset - D1 ∪ D2)

Upon analyzing this result, another minima around the following range was obtained:

$$p_1 \approx 0.025$$
$$p_2 \approx [0.70, 0.75]$$
$$p_3 \approx [0.25]$$

The parameter set close to (0.2, 1.8, 0.18) was found to be a local minima as shown in the figure above.

2 Results

The minima found and the corresponding datasets, from the above runs are as follows:

2.1 Dataset 1

Optimal Cost $\approx$ 700 units
Optimal parameter sets:

p_1	p_2	p_3	Cost
0.0250	0.7359	0.2492	702.7844
0.0250	0.7370	0.2489	701.1474
0.0251	0.7430	0.2467	702.5539
0.0254	0.7648	0.2419	700.3625
0.0255	0.7781	0.2395	698.6413

The fit provided by the parameter set was then tested by using the `check_using_plot` function. The plots are as follows:

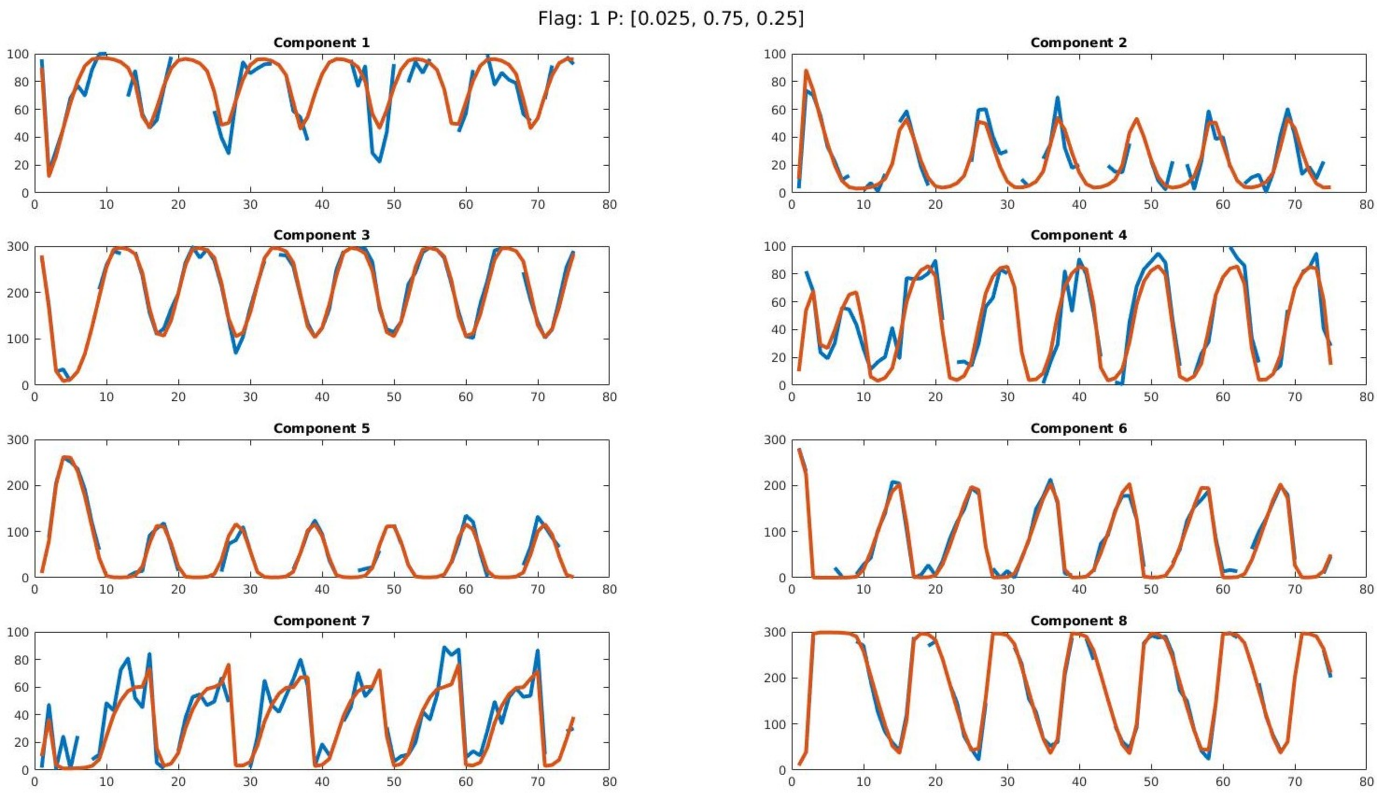

Figure 3: Best fit obtained after optimization on dataset 1 (D1). Blue represents the measured values and

orange represents the computed values.

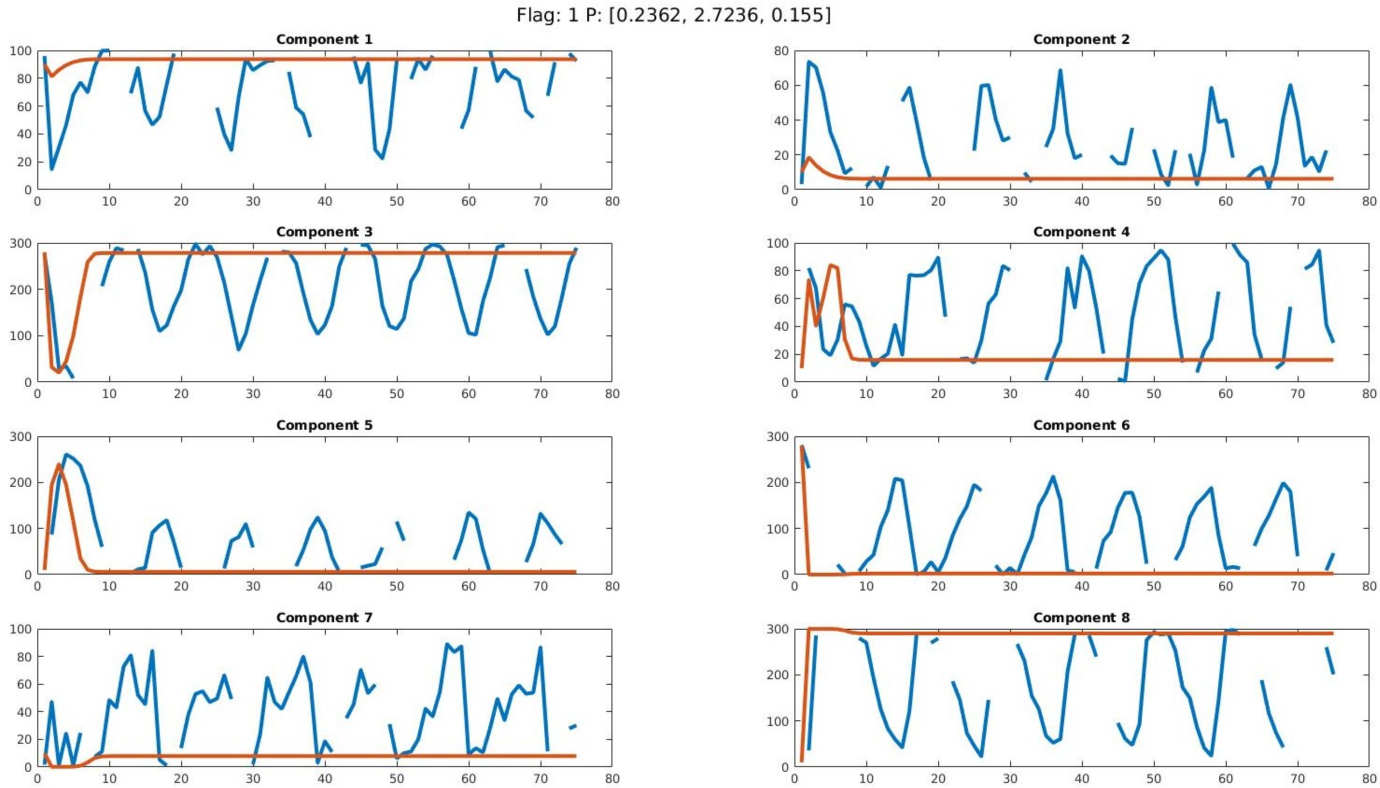

Figure 4: Suboptimal solution obtained after optimization on dataset 1 (D1). Blue represents the measured values and orange represents the computed values.

After rounding the above parameter, the optimal parameter set for D1 is found to be $p_1 = 0.025, p_2 = 0.75, p_3 = 0.25$.

2.2 Dataset 2

Optimal Cost $\approx$ 450 units
Optimal parameter sets:

p_1	p_2	p_3	Cost
0.0708	0.3927	0.7163	451.4106
0.0708	0.3925	0.7157	451.4162
0.0711	0.3959	0.7125	451.4246
0.0257	0.6752	0.2617	515.5303

After rounding the above parameters, the optimal parameter set for D2 is found to be $p_1 = 0.07, p_2 = 0.40, p_3 = 0.70$.

The costs of the current parameter set and the one obtained as the best parameter set for D1 were analyzed and the the cost distribution, in the same order, is as follows:

cost(x1)	cost(x2)	cost(x3)	cost(x4)	cost(x5)	cost(x6)	cost(x7)	cost(x8)	Total cost
26.6097	110.6925	71.9419	10.2650	28.5658	76.9884	18.7135	107.6338	451.4106
1.3191	562.7325	2.7817	1.4801	4.8092	4.7517	5.6855	3.0402	586.6001

It was found that the best parameter set for D2 (i.e. P2) has a lot lesser error in the second component than the best parameter set of D1. This could be explained by the extremely small value of $x(2)$ when P2 is the parameter set. Thus the summation in turn yields a very small error. The fit provided by the parameter set was then tested by using the `check_using_plot` function. The plots are as follows:

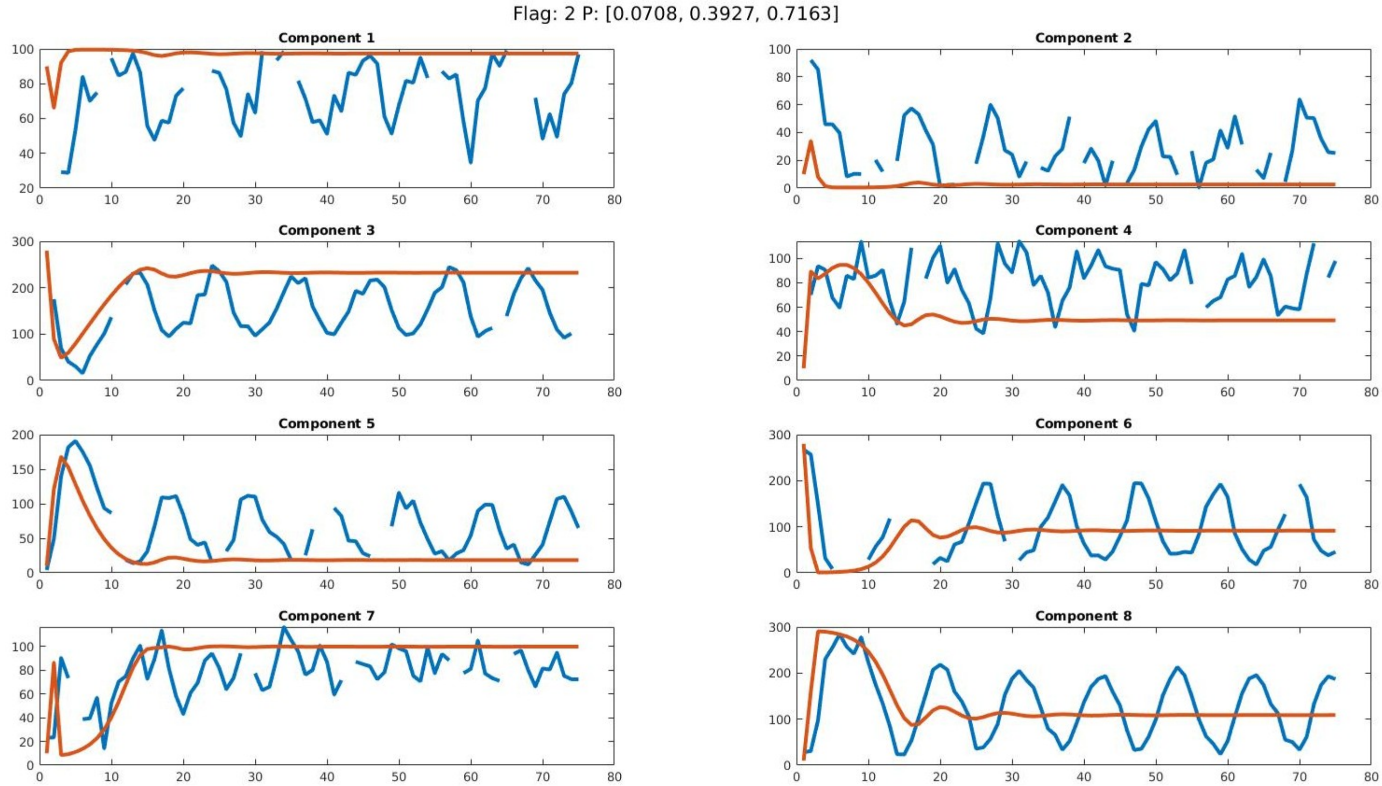

Figure 5: Best fit obtained after optimization on dataset 2 (D2). Blue represents the measured values and orange represents the computed values.

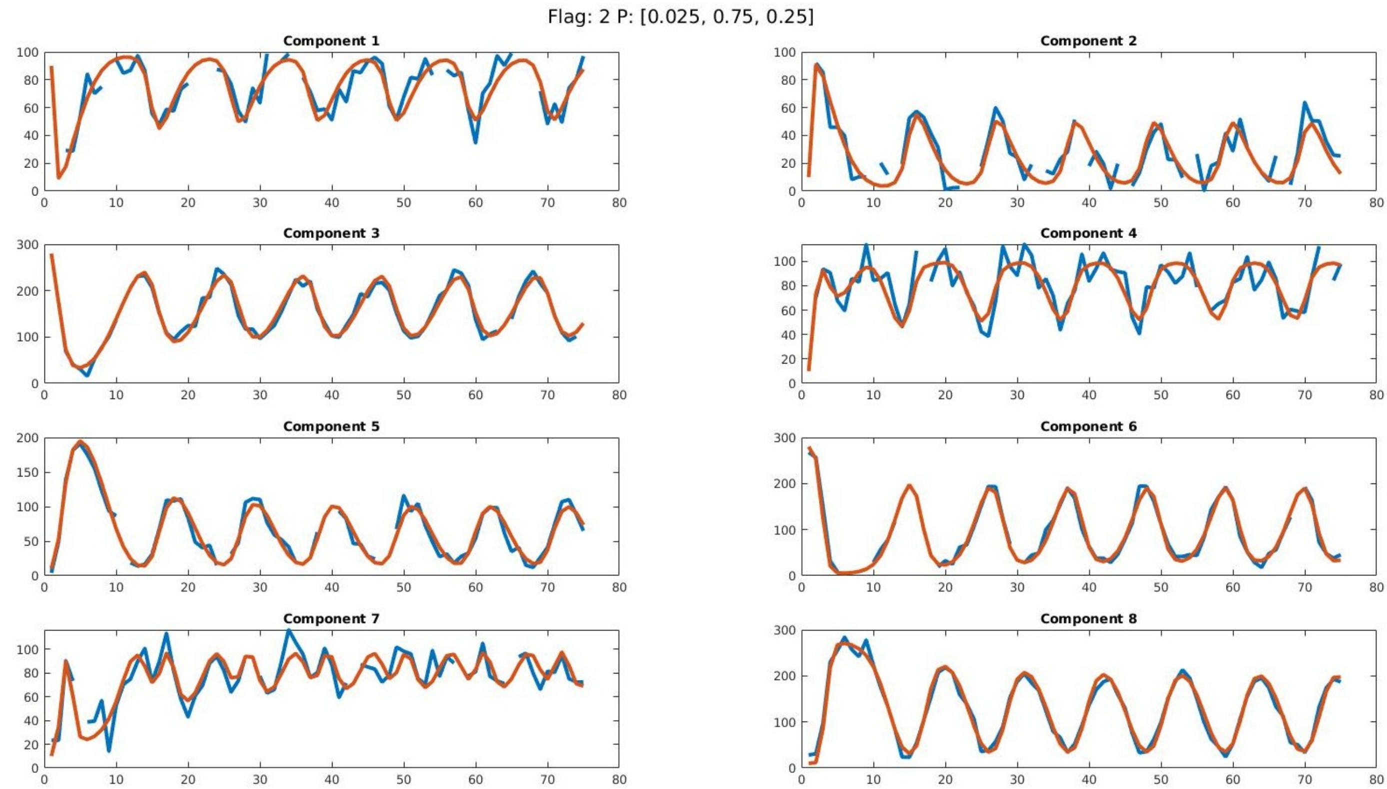

Figure 6: Fit obtained from using the best parameter set reported for dataset 1 (D1) on dataset 2 (D2). Blue represents the measured values and orange represents the computed values.

Points that accounted for large increase in the cost function with $p_1 = 0.025, p_2 = 0.75, p_3 = 0.25$ were analyzed and the plot obtained was as below. These points were found to largely deviate negatively from the actual oscillation, thereby contributing largely to the cost function.

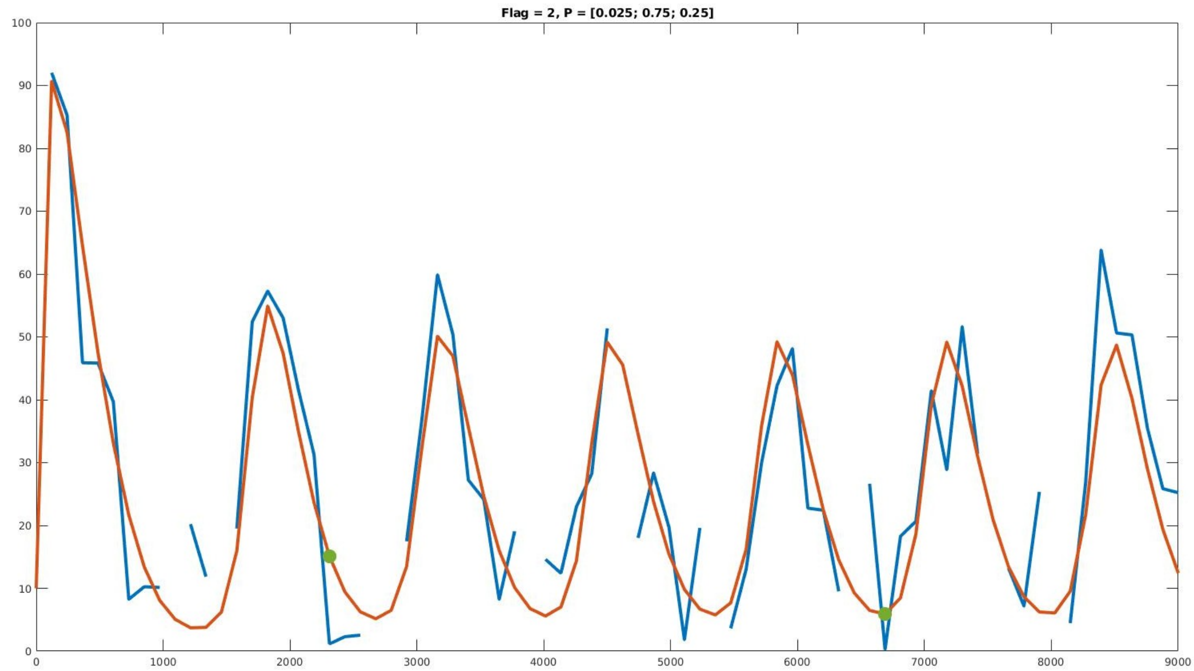

Figure 7: Parameter set [0.025, 0.75, 0.25] against actual data. Green dots represent the positions with maximum contribution towards the cost function. Blue represents the measured values and orange represents the computed values.

2.3 Dataset 3

Optimal Cost ≈ 450 units
Optimal parameter sets:

p_1	p_2	p_3	Cost
0.0245	0.6996	0.2598	1281.318
0.2084	1.8879	0.1772	2691.146
0.2090	1.8455	0.1798	2691.175
0.2091	1.8477	0.1798	2691.214

After rounding the above parameters, the optimal parameter set for D3 is found to be $p_1 = 0.025, p_2 = 0.70, p_3 = 0.25$.

The best parameter values of D3 was found to be closer to the best parameter values of D1. This could be because of the higher cost associated with D1 when compared to D2. Hence, during the SSE calculation for D3, the weight variations would have been adjusted to minimize the D1 error. This would have caused the parameter values to move closer to [0.025; 0.75; 0.25].

The fit provided by the parameter set was then tested by using the `check_using_plot` function. The plots are as follows:

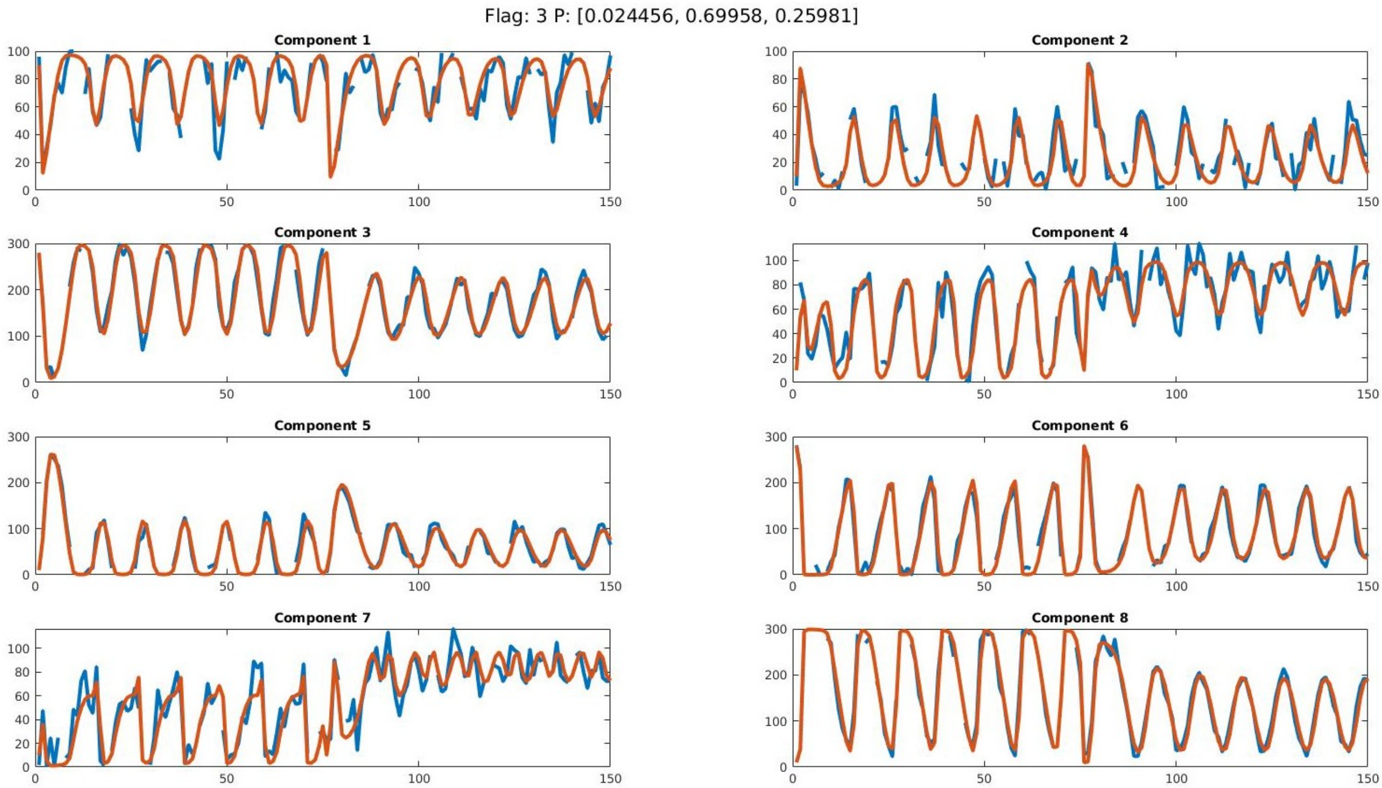

Figure 8: Best fit obtained after optimization on dataset 3 (D3). Blue represents the measured values and orange represents the computed values.

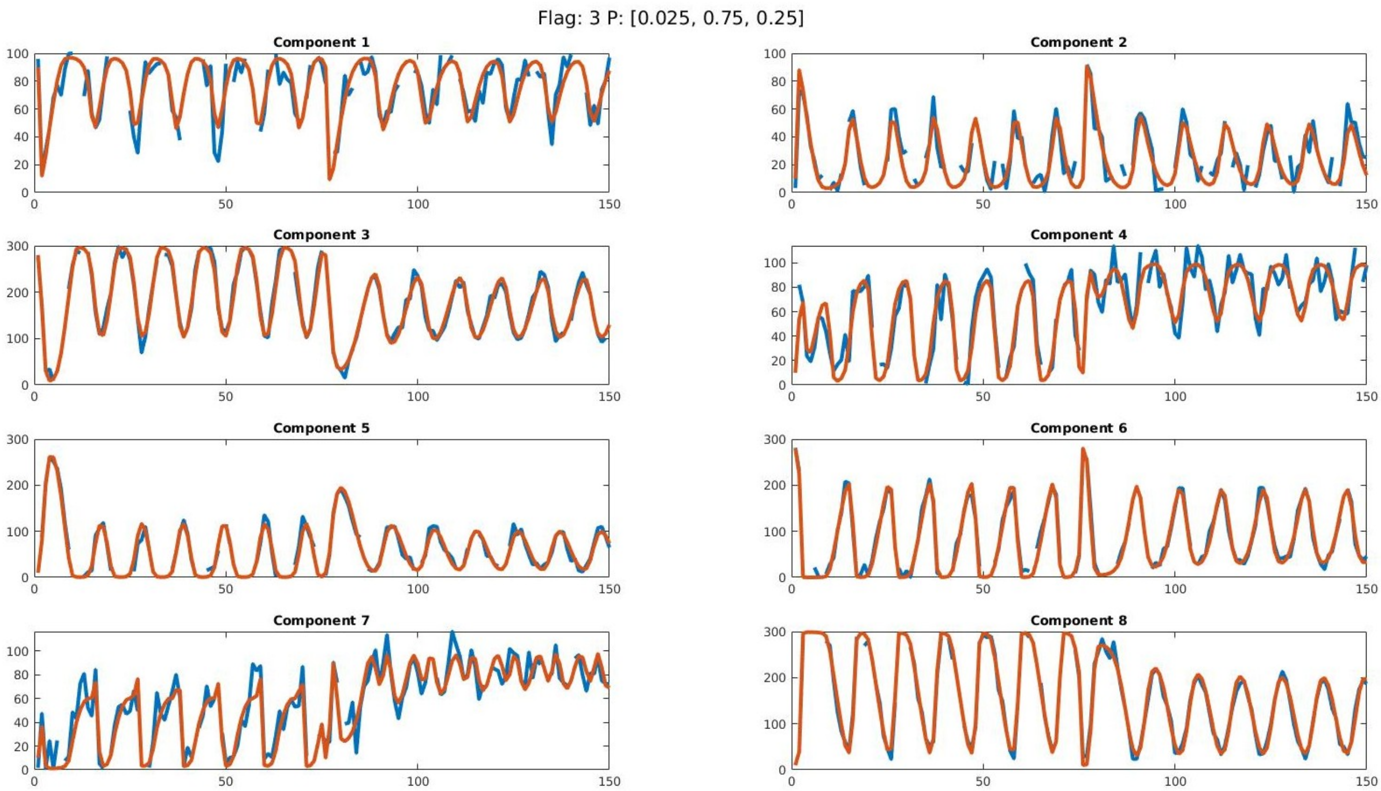

Figure 9: Fit obtained from using the best parameter set reported for dataset 1 (D1) on dataset 3 (D3). Blue represents the measured values and orange represents the computed values.

2.4 Conclusion

To summarize, it was noted that there wasn't any visual difference between the two datasets observed in D3, an absence of the characteristic oscillations of the data using the best parameter set values reported for D2 was noted, while the best parameter set reported for D1 fit all the three datasets with tolerable error. Hence, the parameter set (i.e.) $p_1 = 0.025, p_2 = 0.75, p_3 = 0.25$ is taken as the final parameter set.

3 Functions Used

3.1 `ode_model1` and `ode_model2`

These functions return the derivatives of the differential equations involved in the model, given a P vector.
All the parameter values corresponding to dataset 1 and dataset 2 are defined in `ode_model1` and `ode_model2` respectively.

The function takes an input of P a column vector comprising of $[p_1, p_2, p_3]'$ in the same order. The function returns `xdot`, the derivative values of the components in the model. The values in `xdot` correspond to

1 - MKKK	5 - MKK-PP
2 - MKKK-P	6 - MAPK
3 - MKK	7 - MAPK-P
4 - MKK-P	8 - MAPK-PP

The handle of these functions are passed to the `analyse_ode` function, to calculate the values across a time duration.

3.2 `analyse_ode`

The `analyse_ode` function returns the concentration of the components post integration, given a P vector and flag. The ODE integrator function used is `ode15s`.

This function uses `readmatrix` which is available in `MATLAB 2020a` and later. If the program throws an error, kindly comment the lines with `readmatrix` and uncomment the line(s) following it.

The function takes an input of P a column vector comprising of $[p_1, p_2, p_3]'$ in the same order and a flag to determine which dataset parameters should be used. The flag mappings are as follows:

- 1 denotes dataset 1 (D1)
- 2 denotes dataset 2 (D2)
- 3 denotes dataset 3 (D1 $\cup$D2)

This function is called by the function `cost_function` for calculating the cost that will be optimized.

3.3 `cost_function`

The `cost_function` returns a normalized sum of squares cost, given a P vector and a flag.

This function uses `readmatrix` which is available in `MATLAB 2020a` and later. If the program throws an error, kindly comment the lines with `readmatrix` and uncomment the line(s) following it.

The definitions of flag and P are same as in the `analyse_ode` function.
The equation used in the function is as follows:

$$\texttt{cost} = \sum_i \left(\frac{x_{measured,i} - x_{computed,i}}{x_{measured,i}} \right)^2$$

As the ODEs weren't integrable at some time instances for certain initial values, constraints were added to the `cost_function` to calculate cost using the above formula only for cases where:

- Integration was complete at all time points (size of the matrix X = [75, 8])
- All the concentration values returned from the integral is greater that or equal to zero.

3.4 `fit_parameters`

The `fit_parameters` is a script that executes `fminsearch` to optimize the normalized least squares cost function.
The key variables in this script are as follows:

- n - denotes the number of iterations (each starting with a random initial P, `P0 = rand(3,1)`)
- flag - determines the dataset to be used.

The definitions of flag and P are same as in the `analyse_ode` function.

After the execution of this program, the P_hist, P0_hist and val_hist arrays were analyzed for values that correspond to the minimum.
The fit of these P values were then analyzed visually using the `check_using_plot` function.

3.5 `check_using_plot`

The `check_using_plot` function plots the concentrations of the components post integration against the respective datasets, given a P vector and a flag.

This function uses `readmatrix` which is available in `MATLAB 2020a` and later. If the program throws an error, kindly comment the lines with `readmatrix` and uncomment the line(s) following it.

The definitions of flag and P are same as in the `analyse_ode` function.

3.6 scatter_4d_plot

The scatter_4d_plot function plots a 4D graph of the variation of the cost function with respect to the three parameters, given a threshold, list of cost values (val_hist) and the corresponding matrix of parameter values (P_hist).

The variation of the cost function is depicted by the color map, while the parameters p_1, p_2, p_3 are represented by X-axis, Y-axis and Z-axis respectively.

3.7 grid_search

The grid_search script was initially used to understand the distribution of the cost function at all integer parameter values in the range [0, 10].

However, as the ODEs weren't integrable at several time instances with integer parameter values, the values from the grid search weren't used.