# Ethanol Production

**Simulating ethanol production through glucose fermentation using matlab and differential equation**

## Contributors


Abhijit Dalai(220030)
Anunay Minj(220183)
Kushal Gupta(220576)
Manisha Balot(220622)
Mitesh Wandhare(220650)
Nikant Yadav(220707)
Aniket Shivhare(221022)
Sujal Jain(221100)
Tanzeela Akhtar(221134)


## Overview

In the processing of ethanol fermentation, microorganisms such as yeasts play an essential role. They are used in industrial plants due to valuable properties in ethanol yield, ethanol tolerance, ethanol productivity, growth in simple, inexpensive media and undiluted fermentation. As the main component in fermentation, yeasts affect the amount of ethanol yield.

An advantage of continuous fermentation processes is that it is likely to lead to reduced process costs during ethanol production. The productivity of a fermentation system is the main measurement for evaluating the performance of a fermentation system. This productivity is the amount of produced product per unit of time and reactor volume.

## Problem Statement

Model of microbial kinetics for growth and fermentation processes is described by nonlinear differential equations in which the change of fermented product rate, substrate consumption and biomass are related to ethanol, glucose and biomass concentrations by the means of suitable functional representation of some kinetic growth rate models. Due to the concentration of the preventive substrate, the growth rate of the microorganism can be described by the Monod equation.

$\mu = f(S)$ $\{S \rightarrow \text{Glucose}\}$

Here, the main focus is on ethanol effective production with the rate constant for ethanol production.

# Method for Solving

### Simulation parameters

The simulation parameters provided are as follows:

- Reactor volume (V): 1000 liters
- Initial glucose concentration (C_glucose_initial): 100 g/L
- Initial yeast concentration (C_yeast_initial): 5 g/L
- Initial ethanol concentration (C_ethanol_initial): 0 g/L
- Maximum specific growth rate (mu_max): 0.1 h^-1
- Monod constant (K_s): 25 g/L

- Ethanol production rate constant (k_ethanol_production): 0.5 1/h
- Maximum yeast concentration (C_yeast_max): 20 g/L

These parameters are used in the simulation code to calculate the changes in glucose, yeast, and ethanol concentrations over time.

## Simulation time and time step

The simulation time and time step are specified as follows:

* Total simulation time (Simulation_time): 6 hours
* Time step (time_step): 0.05 hours

These values determine the duration of the simulation and the granularity at which the concentrations are updated. In this case, the simulation will run for 6 hours with a time step of 0.05 hours, meaning that the concentrations will be calculated and updated every 0.05 hours (or every 3 minutes).

## Number of time steps

The number of time steps (num_steps) is calculated by dividing the total simulation time by the time step. In this case, with a simulation time of 6 hours and a time step of 0.05 hours, the number of time steps will be (6/0.05) Therefore, there will be 120 time steps in the simulation.

## Initialize arrays to store concentrations

In this part of the code, arrays are initialized to store the concentrations of glucose, yeast, and ethanol at each time step during the simulation. The size of these arrays is determined by the number of time steps (num_steps) calculated previously.

* The 'time' array is used to store the time values at each time step. It has a size of num_steps x1, meaning it is a column vector with num_steps elements.
* The 'C_glucose' array is used to store the glucose concentrations at each time step. It also has a size of num_steps x1.
* The 'C_yeast' array is used to store the yeast concentrations at each time step. It also has a size of num_steps x1. * The 'C_ethanol' array is used to store the ethanol concentrations at each time step. It also has a size of num_steps x1.

By initializing these arrays with zeros, the code ensures that they are ready to store the concentration values during the simulation. As the simulation progresses, these arrays will be updated with the calculated concentrations at each time step.

## Set initial conditions

In this part of the code, the initial conditions for the simulation are set. The initial values are assigned to the corresponding arrays at the first index (index 1) to represent the starting concentrations at time zero.

* The 'time' array is set to 0 at the first index: `time(1) = 0`.
* The ‘C_glucose’ array is set to the initial glucose concentration (C_glucose_initial) at the first index: ‘C_glucose(1) = C_glucose_initial’.
* The ‘C_yeast’ array is set to the initial yeast concentration (C_yeast_initial) at the first index: ‘C_yeast(1) = C_yeast_initial’.
* The ‘C_ethanol’ array is set to the initial ethanol concentration (C_ethanol_initial) at the first index: ‘C_ethanol(1) = C_ethanol_initial’.

These initial conditions provide the starting point for the simulation, allowing the concentration values to be updated and calculated from this initial state as the simulation progresses.

## Perform simulation

In this part of the code, the simulation of ethanol production is performed. It iterates over the time steps from 2 to num_steps and calculates the changes in glucose, yeast, and ethanol concentrations at each time step using the created function “getRatesofChange”.

* Inside the loop, the specific growth rate (mu) is calculated using the Monod equation with inhibition. * The rates of change for glucose, yeast, and ethanol concentrations are computed based on the specific growth rate and the given equations using the function getRatesofChange .

*The function is called each time the loop runs and the value is transferred to the function explicitEuler .

* In explicitEuler function, The concentrations at the current time step (i) are updated using Euler's method, where the rates of change are multiplied by the time step and added to the concentrations at the previous time step. * To ensure that

concentrations remain non-negative, the concentrations are clamped to a minimum value of 0. * The time value at the current time step is updated.

* By iterating over the time steps and updating the concentrations using the specific growth rate and rate equations, the simulation calculates the changes in glucose, yeast, and ethanol concentrations over time.

## Plot the concentrations

In this part of the code, the concentrations of glucose, yeast, and ethanol over time are plotted using the plot function.

* A new figure is created using the 'figure' function to display the plot.

* The concentrations of glucose, yeast, and ethanol are plotted on the same graph using the 'plot' function. The time values are plotted on the x-axis, and the corresponding concentration values are plotted on the y-axis. The concentration values are represented by different colors and line widths to distinguish them.

* The x-axis label is set to 'Time (hours)' using the 'xlabel' function, and the y-axis label is set to 'Concentration (g/L)' using the 'ylabel' function.

* A legend is added to the plot using the 'legend' function to indicate which line represents each concentration. * The title of the plot is set to 'Ethanol Production Simulation' using the 'title' function.

By plotting the concentrations, the code provides a visual representation of how the glucose, yeast, and ethanol concentrations change over time during the simulation.

## Display the final concentrations

In this part of the code, the final concentrations of glucose, yeast, and ethanol after the simulation are displayed in the command window using the fprintf function.

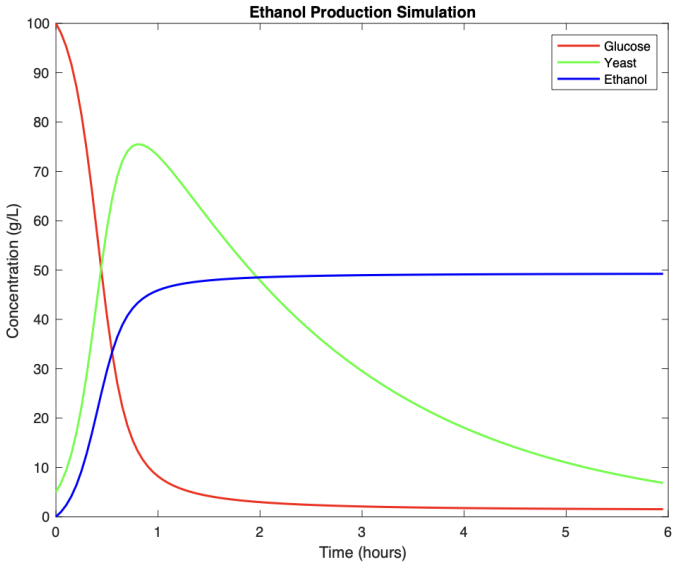
* The fprintf function is used to display text in the command window.

* The lines starting with 'fprintf' display the final concentrations of glucose, yeast, and ethanol. The '%f' format specifier is used to display the concentrations.

* The 'C_glucose(end)', 'C_yeast(end)', and 'C_ethanol(end)' expressions are used to access the final elements of the respective concentration arrays, which represent the concentrations at the last time step of the simulation.

By using fprintf, the code prints the final concentrations of glucose, yeast, and ethanol in the command window for the user to see.

# GRAPH



# Conclusion

The given code represents a simulation of ethanol production in a reactor using a numerical approach. The numerical aspect of this simulation is the use of Euler's method to approximate the rates of change and update the concentrations over time.

Euler's method is a basic numerical integration technique that approximates the solution to a differential equation by dividing the continuous time interval into discrete steps. In this simulation, the time interval is divided into small time steps (defined by the variable "time_step"), and the concentrations are updated at each step using the rates of change calculated based on the given equations.

The rates of change (dC_glucose_dt, dC_yeast_dt, and dC_ethanol_dt) are calculated based on the specific growth rate (mu) and the given rate constants. These rates are then multiplied by the time step and added to the previous concentrations to obtain the updated concentrations for the next time step.

It's important to note that Euler's method is a first-order approximation and may introduce errors, especially when the time step is relatively large. Other more advanced numerical integration methods, such as the Runge-Kutta methods, can provide better accuracy by taking into account higher-order derivatives.

Overall, the numerical aspect of this simulation involves using Euler's method to numerically solve the system of differential equations and track the concentrations of glucose, yeast, and ethanol over time.