

# Detailed Overview on POLYMATH Software for Chemical Engineering Analysis

Abdulhalim Musa Abubakar <sup>\*1</sup>, Bello Iliyasu <sup>2</sup>, Zakiyyu Muhammad Sarkinbaka <sup>3</sup>

<sup>1</sup> Department of Chemical Engineering, Modibbo Adama University (MAU), P.M.B 2076, Yola, Adamawa State-Nigeria

<sup>2</sup> Department of Mathematics, Modibbo Adama University (MAU), P.M.B 2076, Yola, Adamawa State-Nigeria

<sup>3</sup> Department of Chemical Engineering, Federal University Wukari, Faculty of Engineering, P.M.B 1020, Wukari, Taraba State, Nigeria

\*Corresponding author: Abdulhalim Musa Abubakar, +2347050244277 & [abdulhalim@mautech.edu.ng](mailto:abdulhalim@mautech.edu.ng)

**ABSTRACT:** It is pertinent to highlight areas POLYMATH software is useful for chemical engineering analysis. Its applications had been demonstrated in this paper using 10 Problem Set, in areas that includes transport phenomena, heat transfer, reaction, and bioreaction kinetics to solve differential equations, nonlinear equations, simultaneous linear equations, graphical representation and regression problems arising in these fields using Licensed POLYMATH Software Version 6.10.261. POLYMATH is mostly used by students, teachers and researchers for educational purposes, but however limited in application by engineers in process industries, as more sophisticated softwares are preferred. Incorporating many solution approaches there in the literature for solving mathematical method problems, especially the newly proposed novel Extended Runge-Kutta Method of Order 6 into POLYMATH and the enabling of log-log plots is hereby recommended.

**KEYWORDS:** POLYMATH, Problem Set, Chemical Engineering Analysis, Runge-Kutta-Fehlberg, ODE

## 1. Introduction

Tiring engineering task, including calculations and analysis, today are carried out using computerized systems or specific softwares designed for specific type or large variety of functions. Computers are therefore, powerful tools employed to collect, store, process and analyze data for simulation, explaining mathematical models and analysis of basic scientific phenomena [1]. Chemical engineers employ the services of softwares such as POLYMATH, ASIM, MathCad, AQUASIM, MAPLE, Mathematica, FORTRAN, Excel, MATLAB and ASPEN among others for purposeful analysis. So, the act of finding the root causes of observed phenomenon utilizing both statistical, graphical and numerical analysis softwares is termed chemical engineering analysis. Chemical engineering analysis is thus broadly applied in chemical reaction engineering (CRE), transport

phenomena (e.g. fluid mechanics, heat and mass transfer), process dynamics and control, thermodynamics, phase equilibria and distillation and biochemical engineering among others [2]. In CRE, problems solved with POLYMATH are co-current and countercurrent heat exchange, nonlinear regression of rate equations, nonlinear algebraic balances on continuous stirred tank reactors (CSTRs), growth kinetics in bioreactors and membrane reactors, multiple reactions with heat effects, to mention a few [3].

POLYMATH is an educational software recommended for mathematicians, engineers [4] and financial analysts, created by Shacham and Cutlip in January 1, 2003. The challenge of using FORTRAN and C++ is that, a very lengthy time is spent developing and testing codes [3] in addition to hours spent analyzing the result based on different possible scenarios. Unlike

ASPEN and COMSOL where the equations behind the computations are hidden, POLYMATH allow the user to provide their mathematical model where necessary [5] and are solved immediately. Errors in the typed mathematical models, entry of wrong initial estimates or the selection of an inappropriate solution algorithm would prevent the software package from given the correct solutions to problems entered [6, 7]. A table of results containing initial estimates, problem statement and calculated unknown parameters are presented by POLYMATH together with graphs relating some of the parameters in the expression [8]. A hardcopy of this result can sometimes be printed by pressing the function key (F3) on the keyboard [9].

Famously, a solution to ten (10) chemical engineering numerical problems that cut across different aspects of chemical engineering, titled (i) molar volume and compressibility factor from Van Der Waal's equation, (ii) steady state material balances on a separation train, (iii) vapor pressure data representation by polynomials and equations, (iv) reaction equilibrium by multiple gas phase reactions, (v) terminal velocity of falling particles, (vi) unsteady state heat exchange in a series of agitated tanks, (vii) diffusion with chemical reaction in a one dimensional slab, (viii) binary batch distillation, (ix) reversible, exothermic, gas phase reaction in a catalytic reactor, and (x) dynamics of a heated tank with PI temperature control, had been solved using six numerical software packages including Excel, MAPLE, MATHCAD, MATLAB, MATHEMATICA, and POLYMATH [4] for undergraduate education [1]. Statistical analysis of adsorption isotherm models and estimation of growth kinetic parameters can be done using POLYMATH [10, 11]. It is a user friendly software for numerical computation with short learning time, requiring minimal user intervention in technical details of the solution process and had been incorporated into several text, such as Kyle's textbook of "Chemical and Process Thermodynamics" and Fogler's textbook of "Elements of Chemical Reaction Engineering" among others [1]. An android application that can run on a minimum of Android 2.2 version called PolyMath Lite had been developed by Cutlip in 2016 [12].

## 2. Features of POLYMATH Software Package

POLYMATH is a simple software package developed to help students, academicians and engineers

solve numerical analysis problems. In a book written by [2], 170 chemical and biochemical engineering problems requiring numerical solutions had been highlighted. The package consists of a calculator, units converter, records of scientific constants and numerical method icons. Numerical methods in chemical engineering deals broadly with subjects like (i) quantum mechanical calculations of atoms and molecules, (ii) numerical treatment of chemical reaction kinetics, (iii) transport processes, (iv) mathematical description of unit operations, and (v) stationary and instationary simulation and optimization of chemical plants among others [13]. Thus, POLYMATH solves four type of numerical problems arising in those subjects, namely, system of nonlinear equations, system of ordinary differential equations (ODEs), data regression and system of linear equations.

POLYMATH can solve up to 300 set of nonlinear equations with 300 explicit equations using four types of solution approaches, including 'safenewt', 'fastnewt', 'safebroydn' and constrained nonlinear equations [5]. Same number of ODEs can be solved by POLYMATH by either Runge-Kutta-Fehlberg (RKF45), RKF56, Burlirsch-Stoer (BS), STIFF or STIFFBS method. The data regression window is divided into regression, analysis and graph tabs. Multiple linear, nonlinear and linear and polynomial regression with room for finding polynomial derivative and integration are found under the regression tab which is capable of regressing 200 variables with up to 1000 data points for each. Under the analysis tab, numerical differentiation, integration and interpolation can be done using 3 solution methods each. Graphs are plotted entering the 'x' and 'y' data in cell box sheet provided by POLYMATH under the graph tab. Lastly, POLYMATH can accommodate 264 simultaneous linear equations as stated by [5]. Table 1 shows selected chemical engineering problems that can be solved using the four numerical methods discussed above:

Table 1: Chemical Engineering Problem Set with POLYMATH Solutions [14–17]

No .	Mathematical Method	Problem Statement	Equation, Nomenclature & POLYMATH Result
1.	Nonlinear with Constants	Given the equation of saturation concentration of dissolved oxygen in freshwater, $o_{sf} = 14.621$ mg/L, find the absolute temperature, $T_a$ , taking initial guess of 273.15K.	$\ln o_{sf} = -139.34411 + \frac{1.575701 \times 10^5}{T_a} - \frac{6.642308 \times 10^7}{T_a^2} + \frac{1.243800 \times 10^{10}}{T_a^3} - \frac{8.621949 \times 10^{11}}{T_a^4}$ <p>where, <math>o_{sf}</math> = the saturation concentration of dissolved oxygen in fresh-water at 1 atm (mg/L) and <math>T_a</math> = absolute temperature (K). From Problem Set 1, it could be concluded that <math>o_{sf}</math> ranges from 14.621 mg/L at 0°C to 6.413 mg/L at 40°C.</p>
2.	Nonlinear with Constants	Solve the Manning equation for a rectangular open channel for H given $Q = 5$ , $S = 0.0002$ , $B = 20$ , and $n = 0.03$ taking initial guess $\geq 0$ . Prove that the scheme converges for all initial guesses of $H \geq 0$ .	$Q = \frac{\sqrt{S} (BH)^{5/3}}{n(B + 2H)^{2/3}}$ <p>where, <math>Q</math> = flow (m<sup>3</sup>/s), <math>S</math> = slope (m/m), <math>H</math> = depth (m) and <math>n</math> = the Manning roughness coefficient. Results of Problem Set 2 is <math>H = 0.7022933</math>m for <math>H \geq 0</math>.</p>
3.	Nonlinear with Constants	Von Karman equation can be used to predict the fanning friction factor, $f$ , for pipes. For the range and initial value of $f = 0.001(0.0001)0.01$ , estimate $f$ by numerical analysis taking typical Reynold's number, $Re$ for turbulent flow from 10,000 to 500,000	$\frac{1}{\sqrt{f}} = 4 \log(Re\sqrt{f}) - 0.4$ <p>where, <math>f</math> = fanning friction factor and <math>Re</math> = Reynold's number. POLYMATH gives <math>f = 0.007727</math> when <math>Re = 10,000</math> and <math>f = 0.003291</math> when <math>Re = 500,000</math>.</p>
4.	Nonlinear with Explicit Function	Reynolds' dimensionless expression, Colebrook equation and pressure drop ( $\Delta P$ ) can be solved	$Re = \frac{\rho V D}{\mu}$ $\frac{1}{\sqrt{f}} = -2.0 \log \left( \frac{\epsilon}{3.7D} + \frac{2.51}{Re\sqrt{f}} \right)$ $\Delta P = \frac{f L \rho V^2}{2D}$

		simultaneously for $f$ and $\Delta P$ (Pa). Take $L = 0.2\text{m}$ , $\rho = 1.23 \text{ kg/m}^3$ , $\mu = 1.79 \times 10^{-5} \text{ N.s/m}^2$ , $D = 0.005\text{m}$ , $V = 40 \text{ m/s}$ and $\varepsilon = 1.5 \times 10^{-6}\text{m}$ using the Blasius formula, $f = 0.316/Re^{0.25}$ as good initial guess of $f$ for a smooth tubing.	where, $f$ = friction factor, $L$ = length of pipe, $\rho$ = density, $V$ = velocity, $\mu$ = dynamic viscosity, $D$ = diameter, $\varepsilon$ = roughness, and $\Delta P$ = pressure drop. The results window presents $f = 0.02844$ for $\Delta P = 1119.459 \text{ Pa}$ .
5.	Simultaneous Nonlinear	Fig. 2a. displays heat transfer in the Thermos, $q_1$ , $q_2$ and $q_3$ in the inner compartment, middle layer and outside shell which are by radiation, convection in a small space and natural convection respectively. Find the temperatures, $T_1$ and $T_2$ at steady state when $T_0 = 450^\circ\text{C}$ and $T_3 = 25^\circ\text{C}$ , considering that $q_1 = q_2 = q_3$ from each region are equal.	$q_1 = 10^{-9}[(T_0 + 273)^4 - (T_1 - 273)^4]$ $q_2 = 4(T_1 - T_2)$ $q_3 = 1.3(T_2 - T_3)^{4/3}$ where, $q_1$ = heat flux from inner compartment to the next layer, $q_2$ = heat flux between middle layer and outside shell, $q_3$ = heat flux from the outside shell, and $T$ = temperature in those regions ( $^\circ\text{C}$ ). Constrained nonlinear POLYMATH solution approach gives, $q = 273.0047$ , $T_1 = 148.417^\circ\text{C}$ and $T_2 = 80.16582^\circ\text{C}$ .
6.	Differential Equation	A half full storage tank in which liquid is supplied at a sinusoidal rate, $3Q\sin^2(t)$ , and withdrawn at constant flowrate to meet demand is shown in Fig. 2b. Use the continuity equation to solve for $y$ which is initially zero, taking area, $A = 1200 \text{ m}^2$ , $Q = 500$	$\frac{dy}{dt} = 3\frac{Q}{A}\sin^2(t) - \frac{Q}{A}$ where, $Q$ = liquid flowrate ( $\text{m}^3/\text{day}$ ), $A$ = area ( $\text{m}^2$ ), $y$ = liquid depth (m) and $t$ = time (days). The algorithm of Runge-Kutta-Fehlberg (RKF45) method gives a depth, $y = 1.798038\text{m}$ at $t = 10$ days. Though Excel solution of this problem is a tedious step, results are however the same.

		m <sup>3</sup> /day, at time from t = 0 to 10 days, selecting a suitable step size.											
7.	Simultaneous Nonlinear Differential Equations	Given the Monod model for bioreaction kinetics, find the substrate concentration, $s$ , and the biomass concentration, $x$ for values of $y = 0.05$ , $k_s = 20$ , $k = 5$ , and $b = 0.01$ , starting with $s_0 = 1000$ and $x_0 = 100$ at time ranging from t = 0 to 1.	$\frac{ds}{dt} = -\frac{ksx}{k_s + S}$ $\frac{dx}{dt} = y \frac{ksx}{k_s + S} - bx$ <p>where, <math>s</math> = growth limiting substrate concentration (ML<sup>-3</sup>), <math>x</math> = biomass concentration (ML<sup>-3</sup>), <math>k</math> = maximum specific uptake rate of the substrate (T<sup>-1</sup>), <math>k_s</math> = half saturation constant for growth (ML<sup>-3</sup>), <math>y</math> = yield coefficient (MM<sup>-1</sup>), and <math>b</math> = decay coefficient (T<sup>-1</sup>). Both RKF45 and RKF56 solution approaches gives <math>s = 452.5393</math> and <math>x = 126.2464</math>.</p>										
8.	Multiple linear Regression	Fit the data of Figure 3 to Antoine equation and estimate the values of the constant parameters, A, B & C. Take 10, 1000 and 100 as initial values of A, B and C respectively. The expected fitted data curve is also displayed in column 3 of the table in Figure 3.	$\log P = A - \frac{B}{T + C}$ <p>where, <math>P</math> = vapor pressure (mmHg) and <math>T</math> = temperature (°C). Hint: Rewrite the above equation in form of <math>y = a_0 + a_1x_1 + a_2x_2</math>; a multiple linear regression format, i.e.</p> $\log P = A + \frac{AC - B}{T} - \frac{C \log P}{T}$ <p>and compare the two for constants, <math>a_0</math>, <math>a_1</math> and <math>a_2</math> which in turn gives A, B and C. POLYMATH gives A = 17.49696, B = 4515.673 and C = 241.6434 with coefficient of variation, R<sup>2</sup> = 0.99998</p>										
9.	Linear Regression	Find the constants, 'a' and 'b' in the given formula relating thermal conductivity of iron (k) with temperature (T) by regression with POLYMATH. Hint: Linearize the given function.	$Tk^a = b$ <p>where, T = temperature (K) and k = thermal conductivity (W/cm-K)</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>T/K</th> <th>200</th> <th>600</th> <th>1000</th> <th>1400</th> </tr> </thead> <tbody> <tr> <td>k(W/cm-K)</td> <td>1.0</td> <td>0.4</td> <td>0.3</td> <td>0.25</td> </tr> </tbody> </table> <p>Using the above data, it was found that, <math>a = 1.3739</math>, <math>b = 191.97</math> and R<sup>2</sup> = 0.9895.</p>	T/K	200	600	1000	1400	k(W/cm-K)	1.0	0.4	0.3	0.25
T/K	200	600	1000	1400									
k(W/cm-K)	1.0	0.4	0.3	0.25									
10.	Numerical Interpolation, Differentiation	Day and night variation of temperature of a	Measured Data:										

n & Integration.	furnace wall with time is given in the table of the side column. Interpolate for T at t = 12hrs. Differentiate, $\frac{dT}{dt}$ at t = 15 hrs. Carryout numerical integration to determine, $\int_2^{24} f(t)$ .	t/(hr)	2	3	5	8	10	15	18	22	24
		T/(°C)	86.5	97.7	104.0	101.7	92.5	62.3	55.0	67.5	80.0

POLYMATH gives T = 80.086, 79.89355 and 80.42°C selecting RATINT, SPLINT and Linear solution type respectively, interpolated at t = 12 hrs. At t = 15 hrs,  $\frac{dT}{dt} = -4.5996$  °C/hr by STANDA and SPLINE solution method and  $-4.8155$  °C/hr by LAGRANGE's numerical differentiation method. GAULEG and SPLINE integration gives approximately 1745 while SIMPSON numerical integration technique gives 1742.8

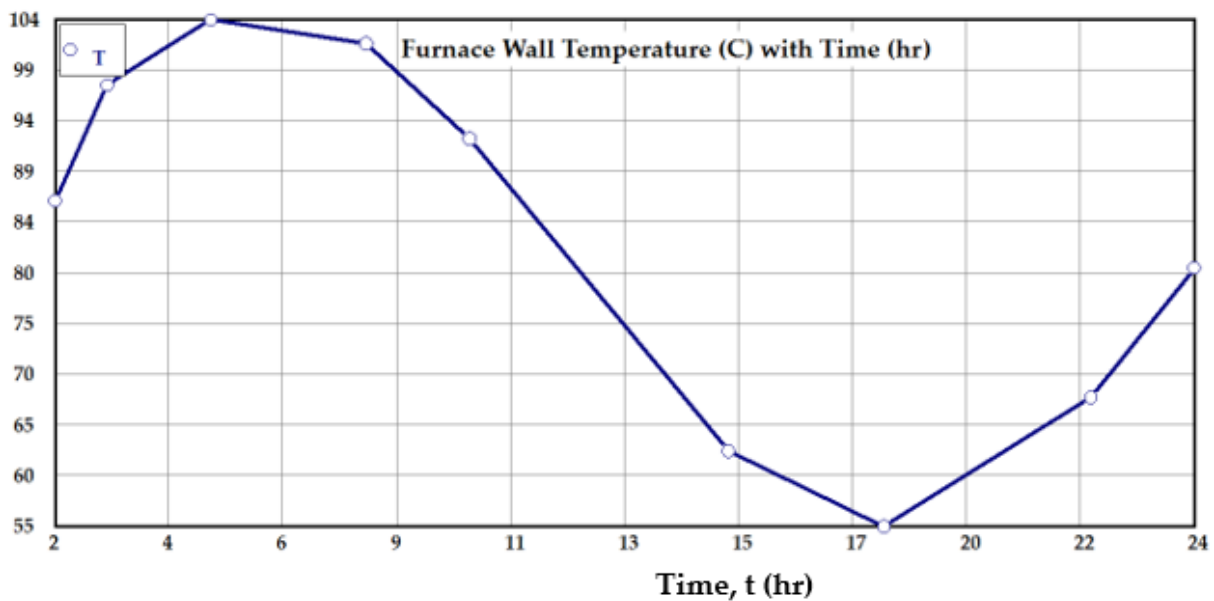


Figure 1: Plot of T(°C) versus t(hrs) for Problem Set 10

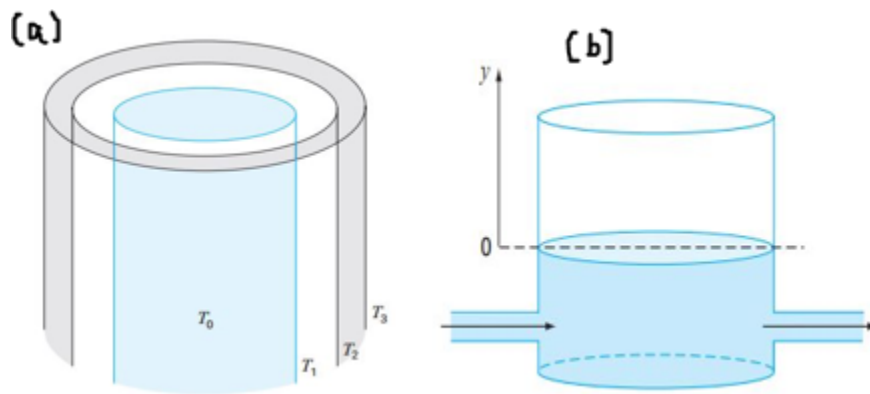


Figure. 2: (a) Heat Transfer in Thermos and (b) Storage Tank Liquid Flow [15]



### 3. Nonlinear Equations

Chemical engineering problems can possibly result in a single or sets of nonlinear equations solvable by numerical methods, including Secant Method, Regula-Falsi, Iterative Method, Bisection, and Newton's Method. Equations of state (EOS) in Thermodynamic, flow equations in Transport Phenomena (e.g. Manning, Von Karman and Colebrook equations as defined in Problem Set 2-4 of Table 1), equilibrium equation in Separation Processes, and Heat Transfer problems seen in Problem Set 5, to mention a few are solved using any of these methods. In [18], Newton's algorithm had been applied to formulate a programming codes capable of solving cubic EOS. Among these EOS, [6] demonstrated how Redlich-kwong equation of state is solved using successive substitution method, a method similar to how Problem Set 5 in this work was solved using POLYMATH. In [19], the modified Raoult's law had been solved using simultaneously with Wilson equation to determine the composition of a non-ideal binary mixture.

In POLYMATH, nonlinear equations are solved with basically two methods: (a) Newton's, and (b) Broyden's method. The "safenewt" algorithm uses the Newton's technique with line search to guarantee reduction of the norm of the function values in all iteration [20]. Broyden is a quasi-Newton method (or the generalization of the Secant method to nonlinear systems) that is superlinearly convergent for finding roots in  $k$  variables, as it reduces the amount of calculation at each iteration without significantly degrading the speed of convergence [21]. Therefore, both 'safenewt' and 'safebroyd' includes function minimization along the Newton and Broyden directions.

### 4. Differential Equations

POLYMATH would request the typing of a function to be solved by numerical methods or user-defined functions to be entered for regression purposes. Functions generate one unique output for each input [8] which are termed as arguments, often enclosed in parenthesis [9]. Some of the functions recognized by POLYMATH are 'sin', 'sinh', 'cos', 'cosh', 'tan', 'tanh', 'ln', 'log', 'arcsin', 'arccos', 'arctan', 'exp', 'frac', 'int', 'abs',

'sign', and 'round'. For example, absolute value of say 'x' can be written as "abs(x)". Also, factorial of an integer, say '5' is written as "5!" in POLYMATH. Differential functions which are implicit type of functions as well as explicit functions must be specified in POLYMATH.

Differential equations are classified into ordinary or partial differential equations and linear or nonlinear equations [22]. A mathematical equation that relates some functions with their derivatives are known as differential equations. Practically, the functions represent physical quantities, the derivatives stand for their rates of change, and the equation defines a relationship between the two [8]. Modeling of the variation of a physical quantity, such as temperature, pressure, displacement, velocity, stress, strain, current, voltage, or concentration of a pollutant, with the change of time or location, or both would result in differential equations [23]. Differential equation govern almost all physical processes of interest to chemical engineers, namely [24]; description of thermodynamic laws, description of flow processes (e.g. using Darcy's law, Bernoulli's equation, Navier-Stokes and mass conservation equations), microbial growth kinetic constant estimation, representation of reaction rates in CRE, Fick's law of diffusion in mass transfer and many more. Elsewhere, [25] highlighted some of the applications of ODEs in reaction engineering. Partial differential equations (PDEs) on the other hand, finds applications in CRE to model complex reactors and reacting systems [3]. Important examples of PDEs are, Tricomi equation, Laplace equation, Helmholtz equation, Poisson-Boltzmann equation, Lorenz equations, Cauchy-Riemann equations, Poisson's equation, Plate bending, Schrödinger's equation, Geodesic equation, Navier-Stokes, Wave equation (1D-3D) and Fourier equation [22, 23]. Softwares that can deal with PDEs are COMSOL (formerly FEMLAB) and MATLAB, especially for reactions taking place in tubular reactors [3]. ODEs in Problem Set 6 & 7 where solved by POLYMATH which is also utilizing the six parameters ( $k_1, k_2, k_3, k_4, k_5$  and  $k_6$ ) of the RKF method in equations (1) [26, 27].

$$k_1 = h f(t_k, y_k)$$

$$k_2 = h f\left(t_k + \frac{1}{4}h, y_k + \frac{1}{4}k_1\right)$$

$$\begin{aligned}
 k_3 &= h f \left( t_k + \frac{3}{8}h, y_k + \frac{3}{32}k_1 + \frac{9}{32}k_2 \right) \\
 k_4 &= h f \left( t_k + \frac{12}{13}h, y_k + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3 \right) \\
 k_5 &= h f \left( t_k + h, y_k + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4 \right) \\
 k_6 &= h f \left( t_k + \frac{1}{2}h, y_k - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5 \right)
 \end{aligned} \tag{1}$$

An approximate solution,  $y_{k+1}$ , to given initial value problems (IVPs) can then be executed using the RKF45 approximation as in equation (2):

$$y_{k+1} = y_k + \frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4101}k_4 - \frac{1}{5}k_5 \tag{2}$$

An improved value of the solution,  $z_{k+1}$ , is determined using another approximation. Check equation (3):

$$z_{k+1} = y_k + \frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56460}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6 \tag{3}$$

Note that the approximations in (2) and (3) of the RKF45 method [26] ignores,  $k_2$ . This makes equation (2) of order 4 and (3), an order 5 equation; basically, making use of the classical Runge-Kutta methods of order 4 and 5 together, therefore defining RKF45 algorithm holistically. The selection of appropriate step sizes,  $h$ , guarantees the accuracy of most IVPs. The advantage of the RKF45 algorithm is that, it ensures that the right,  $h$  is used on its own and not allowing the user to specify any step size. Problem Set 6 had been coded to give,  $y_{k+1}$  and  $z_{k+1}$  using a step size of  $h = 0.1$  for the RKF45 technique using C++ as seen in the Appendix. RKF56 on the other hand is termed the 6th order Runge-Kutta-Fehlberg method and is presented by [28]. The Runge-Kutta-Fehlberg method is generally thought of as one of the best methods existing for solving non-stiff systems of equations [27].

Starting from order 1 to 6, differential equations are solved using (i) Euler's method of order 1, (ii) Heun's method (or modified Euler or leap-frog formula) of order 2, (iii) Bogacki-Shampine method of order 3, (iv) Runge-Kutta method of order 4, (iv) R-K-Fehlberg method of order 4 with an error estimator of order 5 (RKF45), and (v) RKF method of order 5 with an error estimator of order 6 (RKF56). Beyond this, not long ago, [29] proposed a novel Extended RK 6th order method to bring the

number of the solution approaches to six. A differential equation of the form,  $y' = f(t, y)$  is said to be stiff if its exact solution  $y(t)$  includes a term that decays exponentially to zero as  $t$  increases (i.e.  $e^{-ct}$ , where  $c$  is a large positive constant), but whose derivatives are much greater in magnitude than the term itself [30]. A stiff differential equation is further characterized as being numerically unstable unless the step size is extremely small [30]. Stiff and Burlirsch-Stoer (BS) are alternative solution techniques similar to Runge-Kutta.

## 5. Data Regression

The most extensively used statistical technique is regression analysis, which involves detecting, evaluating, and analyzing the connection between the dependent and independent variables [31]. As mentioned earlier, the regression window in POLYMATH is divided into 3 tabs, namely, 'Regression', 'Analysis' and the 'Graph' tab. User-defined linear and multiple linear regression was carried out in Problem Set 8 and 9 to determine unknown constants, however, using the nonlinear function definition provided by POLYMATH. Regressed graph of the calculated and experimental data of Problem 8 is shown in Figure 3.

In chemical engineering, regression tools are widely used to determine unknown constant parameters in Arrhenius equation, Antoine equation, microbial growth models, EOS parameters, rate equations, biogas kinetic models and many other proposed models in the literature. KaleidaGraph program (Synergy software), PAST, SigmaPlot, SAS, EViews, GraphPad Prism, SPSS, Matrixer, SHAZAM, Statistica, ORIGINPro, Datafit, JASP, PSPP, Xlstat Excel Add-in, Excel Solver, Jamovi, Minitab, Stata, KyPlot, Scilab, nonlinear curve fitting toolbox of MATLAB and NCSS are competitive softwares for regression analysis apart from POLYMATH [32, 33].



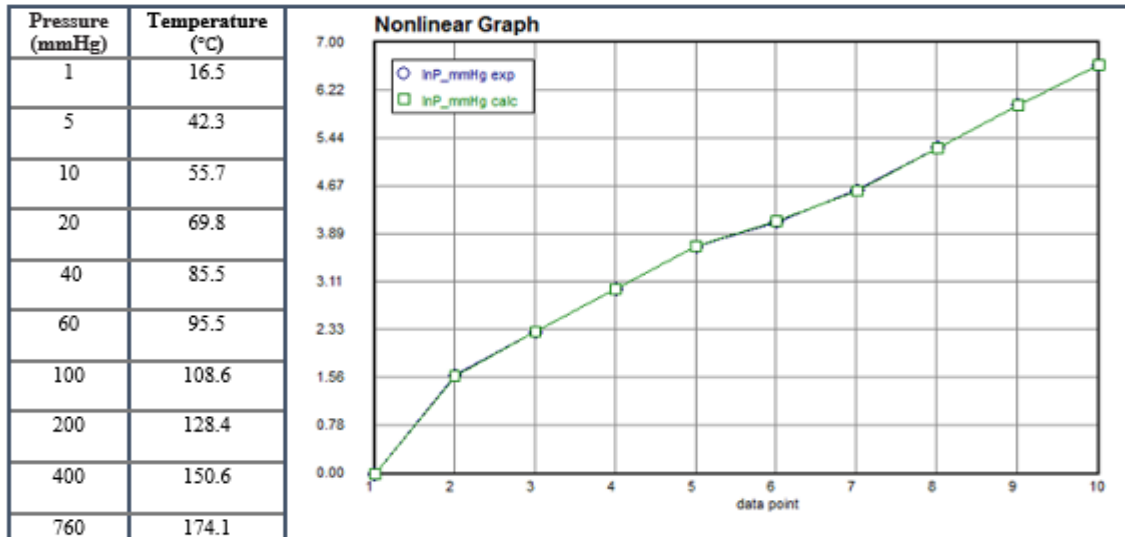


Figure 3: Experimental Vapor Pressure for N-Decane with Regressed Curve [8]

We have demonstrated how to use POLYMATH to carryout numerical differentiation, interpolation, integration as well as data plots for Problem Set 10. Interpolation, formerly can be described as a method of estimating missing data values that are in-between a given set of field data [34]. Typical example is approximating of function values (e.g. temperature) in heat transfer and/or thermodynamics at points where there is no data reading and when computing derivatives from empirical data values.

In chemical engineering, regression tools are widely used to determine unknown constant parameters in Arrhenius equation, Antoine equation, microbial growth models, EOS parameters, rate equations, biogas kinetic models and many other proposed models in the literature. KaleidaGraph program (Synergy software), PAST, SigmaPlot, SAS, EViews, GraphPad Prism, SPSS, Matrixer, SHAZAM, Statistica, ORIGINPro, Datafit, JASP, PSPP, Xlstat Excel Add-in, Excel Solver, Jamovi, Minitab, Stata, KyPlot, Scilab, nonlinear curve fitting toolbox of MATLAB and NCSS are competitive softwares for regression analysis apart from POLYMATH [32, 33]. We have demonstrated how to use POLYMATH to carryout numerical differentiation, interpolation, integration as well as data plots for Problem Set 10. Interpolation, formerly can be described as a method of estimating missing data values that are in-between a given set of field data [34]. Typical example is

approximating of function values (e.g. temperature) in heat transfer and/or thermodynamics at points where there is no data reading and when computing derivatives from empirical data values.

To show how one quantity vary with another related quantity, graphs are readily used to convey information to readers than if same information are explained in words [35]. Graphs can also be plotted in POLYMATH just like Excel as depicted in Figure 1. However, POLYMATH cannot be used to plot a log-log scale. Log-log graphs is a scatterplot in which both axes have a logarithmic scale [36]. A log-log plot displays the relationship, such as one in Problem Set 9 ( $Tk^a = b$ ) as a straight-line equivalent to " $\ln T = \ln b - a \ln k$ ". The Bode plot, a graphical depiction of the frequency responses of a system in control and electrical engineering is a log-log plot [37]. The idea behind this type of plot according to [36] is to use the axes so we can more easily see details for small and large values of the dependent variable, and to show % change or multiplicative factors. Log-log scales are plotted very easily in Excel for an existing scatter plot by clicking on the "Logarithmic Scale" check box under Format Axis after right-clicking on the values along the x-axis. This is then repeated for the y-axis.

## 6. Linear Equations

Not one, but sets of simultaneous linear equations are solved using numerical analysis softwares. Methods

of solution are usually, Cramer's rule, substitution method, elimination method [35], graphical, and adjoint method which requires painstaking steps if the number of equations is more than 3. The fastest tool of solving a system of simultaneous linear equations is POLYMATH. In chemical engineering, simultaneous equations are used to balance chemical equations, finding unknowns in material balances, constrained optimization problems of differential equations after applying Laplace transform, solving certain functional equations among others. In real life, a smartphone user can use system of linear equations to select, best cell phone data plan knowing the charge rate of different companies per minute [38]. Also, to determine if one will make more money at one job or another, taking multiple variables (e.g. salary, benefits and commissions) into account [38]. POLYMATH cannot solve sets of linear inequality constraint equations used in cost optimization in chemical plants [39].

## 7. Conclusion

POLYMATH provisions for solutions to numerical problems makes it a useful chemical engineering analysis tool. An Android application called PolyMathLite had been created to further go into user's smart phones. The software package is limited to not possessing the features for solving PDEs in engineering. Unlike Excel, Tora, Lingo and sometimes C++ programs, it cannot solve linear programming problems, involving cost minimization or maximization in chemical plants. POLYMATH has the advantage of solving ODEs by selecting the best step size necessary while the use of programming codes would take much time and lengthy iterations before arriving at a solution and sometimes crash a system if 'h' is very negligible.

## Conflict of Interest

The authors declare no conflict of interest.

## Acknowledgement

The authors acknowledge the support given by Federal University Wukari as well as the University of Maiduguri.

## References

- [1] A. Brenner, M. Shacham, M. B. Cutlip, "Applications of mathematical software packages for modelling and simulations in environmental engineering education," *Environmental Modelling & Software*, vol. 20, pp. 1307-1313, 2005, doi:10.1016/j.envsoft.2004.09.007.
- [2] M. B. Cutlip, M. Shacham, *Problem solving in chemical and biochemical engineering with POLYMATH, Excel, and MATLAB*, 2nd ed. (Prentice Hall International Series, 2008).
- [3] H. S. Fogler, M. B. Cutlip, "Chemical Reaction Engineering (CRE) education: From the era of slide rule to the digital age." [www.fdcocuments/in](http://www.fdcocuments/in) . (accessed: 30-Sep-2021).
- [4] M. B. Cutlip, M. Shacham, "Polymath solutions to the chemical engineering demonstration problem set." .
- [5] R. Hesketh, "Application of numerical problem solving in chemical engineering coursework," *ASEE 2017 Chemical Engineering Summer School*, pp. 1-6, 2017.
- [6] M. Shacham, N. Brauner, "Solving a system of nonlinear algebraic equations: You only get error messages-What to do next?," *Chemical Engineering Education*, vol. 15, no. 2, pp. 75-82, 2017.
- [7] M. Shacham et al., "Can I trust this software package? An exercise in validation of computational results," *Chemical Engineering Education*, vol. 42, no. 1, pp. 53-59, 2008.
- [8] Hashim Khan, "Polymath tutorial for Chemical Engineers." : 1-4, 2005.
- [9] M. Shacham, M. B. Cutlip, "User-Friendly numerical analysis programs." <http://www.polymath-software.com> . (accessed: 27-Sep-2021).
- [10] G. K. Rajahmundry et al., *Statistical analysis of adsorption isotherm models and its appropriate selection*, vol. 276, (2021).
- [11] A. M. Abubakar, Z. Soltanifar, M. M. Kida, *Sensitivity analysis of kinetic growth model data: Monod equation* (Revista Rebibio, 2022).
- [12] M. B. Cutlip, "Enabling extensive numerical problem solving on smartphones and tablets," *ASEE International Forum*, pp. 1-10, 2014.
- [13] F. J. Keil, "Application of numerical methods in chemical process engineering," *Scientific Computing in Chemical Engineering*, pp. 1-19, 1996, doi:10.1007/978-3-642-80149-5\_11.
- [14] V. J. Law, *Numerical methods for Chemical Engineers using Excel, VBA, and MATLAB* (New York: CRC Press, 2013).
- [15] S. C. Chapra, R. P. Canale, *Numerical methods for engineers*, 6th ed. (New York, United States: McGraw Hill Companies, 2020).
- [16] CheGuide, "Curve fitting data to an Antoine equation-A guide for Chemical Engineers working in process industry," in *Chemical Engineer's Guide (CheGuide)*, (2015).
- [17] UMD, "Regression practice problems," in *Practical Numerical Methods*, (University of Minnesota Duluth, 2021), 30-36.
- [18] A. M. Abubakar, A. A. Mustapha, "Newton's method cubic equation of state C++ source code for iterative volume computation," *SSRG International Journal of Recent Engineering Science*, vol. 8, no. 3, pp. 12-22, 2021, doi:10.14445/23497157/IJRES-V8I3P103.
- [19] J. G. Eberhart, "Solving nonlinear simultaneous equations by the method of successive substitution: Applications to equations of state," *Journal of Chemistry Education*, vol. 71, no. 12, pp. 1038-1040, 1994, doi:https://doi.org/10.1021/ed071p1038.
- [20] W H Press et al., "Attempting to solve the systems of equations," in *Numerical Recipes*, 2nd ed. , (Cambridge, United States: Cambridge University Press, 1992), 40.

- [21] A. Ramli, M. L. Abdullah, M. Mamat, *Broyden's method for solving Fuzzy nonlinear equations* (Hindawi, 2010).
- [22] Cheng Yung Ming, "Solution of differential equations with applications to engineering problems," (IntechOpen, 2017), 33, doi:http://dx.doi.org/10.5772/67539.
- [23] B. Sumithra, "Engineering applications of differential equations," *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, vol. 6, no. 7, pp. 110–114, 2017.
- [24] P Rees, O Kryvchenkova, and P D Ledger, "Engineering Analysis 2: Ordinary Differential Equations." College of Engineering, Swansea University, United Kingdom, : 1–37, 2017.
- [25] C. B. Mohod, "Application of Ordinary Differential Equations (ODEs) in reaction engineering: Review," *International Journal of Chemical and Molecular Engineering*, vol. 2, no. 1, pp. 1–9, 2016.
- [26] John H Mathews and Kurtis K Fink, "Runge-Kutta Methods," in *Numerical Methods Using Matlab*, 4th ed. , (New Jersey: Prentice-Hall Inc, 2004), 497–499.
- [27] C. F. Mayo, "Implementation of the Runge-Kutta-Fehlberg method for solution of ordinary differential equations on a parallel processor," (Naval Postgraduate School (NPS)-Dudley Knox Library, 1987).
- [28] L. Derr, C. Outlaw, D. Sarafyan, "A new method for derivation of continuous Runge-Kutta formulas," *Computers Math. Application*, vol. 26, no. 3, pp. 7–13, 1993.
- [29] W. Suryaningrat, R. Ashgi, S. Purwani, "Order Runge-Kutta with extended formulation for solving ordinary differential equations," *International Journal of Global Operations Research (IJGOR)*, vol. 1, no. 4, pp. 160–167, 2020.
- [30] Jim Lambers, "Stiff differential equations: Lecture 9 notes," in *MAT 461/561*, (2010), 1–4.
- [31] J. O. Rawlings, S. G. Pantula, D. A. Dickey, *Applied regression analysis: A research tool*, 2nd ed. (Carolina, USA: Springer, 1998).
- [32] J. Tellinghuisen, "Using least squares to solve systems of equations." Tennessee, USA, 2015.
- [33] P. Rani et al., *Experimental and kinetic studies on co-digestion of agrifood and sewage sludge for biogas production*, vol. 16, no. 1, (Taylor and Francis Group, 2022).
- [34] C. R. Chikwendu, H. K. Oduwole, S. I. Okoro, *An application of spline and piecewise interpolation to heat transfer (cubic case)*, vol. 5, no. 6, (IISTE Knowledge Sharing Partners, 2015).
- [35] J. Bird, *Basic engineering mathematics*, 5th ed. (United Kingdom: Elsevier Ltd, 2010).
- [36] M. Bourne, "Graphs on logarithmic and semi-logarithmic axes." <https://www.intmath.com/exponential-logarithmic-functions/7-graphs-log-semilog.php> . (accessed: 06-Oct-2021).
- [37] Y. Bavafa-Toosi, "Introduction to linear control systems," *SIAM Review*, vol. 61, no. 1, pp. 209–225, 2019, doi:<https://doi.org/10.1137/19N974750>.
- [38] M. H. Snyder, "10 ways simultaneous equations can be used in everyday life." [www.sciencing.com/10-can-used-everyday-life-8710568.html](http://www.sciencing.com/10-can-used-everyday-life-8710568.html) . (accessed: 04-Oct-2021).
- [39] A. M. Abubakar et al., "Simplex C++ syntax for solving chemical engineering cost optimization problems," *Research Inventy: International Journal of Engineering And Science*, vol. 11, no. 7, pp. 39–47, 2021, doi:10.5281/zenodo.5146856.

Creative Commons Attribution (CC BY-SA) license (<https://creativecommons.org/licenses/by-sa/4.0/>).



**ABDULHALIM MUSA ABUBAKAR** bagged his B.Eng. (2018) Chemical Engineering from University of Maiduguri-Nigeria.

He works as research assistant with Modibbo Adama University-Nigeria. He has carried out researches in the areas of Linear Optimization, Petroleum and Reservoir Engineering. He is also interested in programming application in chemical engineering. He is a member of International Association of Engineers (IAENG), Graduate Member of the Nigerian Society of Engineers (NSE) and an affiliate member of Onshore Offshore Oil and Gas Professionals (O3GP). He had several other qualifications in areas of soft skill, data science with POLYMATH, ORIGIN software, SPSS and Excel, C++ Programming, Project Management, Webb Development, Information Security and had undergone a Professional Course in Safety Management and the use of AutoCAD 2D. He had 6 publications, two of which is related to the application of C++.



**BELLO ILIYASU** has bagged a Bachelor of Science (B.Sc.) Degree in Mathematics at a reputed Nigerian University. He had studied at Federal College of Education, Yola, Adamawa State.

He works as research fellow at Modibbo Adama University (MAU) being an experienced researcher. He is currently working on various research papers.



**ZAKIYU MUHAMMAD SARKINBAKA** is a research assistant at Federal University Wukari (FUWukari), Taraba-Nigeria. He is a graduate of Chemical Engineering from the University of Maiduguri.

He is an experienced high school teacher before his current job at FUWukari. He enrolled for a Masters Program in Chemical Engineering in 2019 in the same university. He had published three articles in reputed journals. Two among his publications are titled: "Neural Network Based Performance Evaluation of a Waterflooded Oil Reservoir" and "Multivariate Optimization of a Jacketed Heating System: A Generic Algorithm Approach".

**Copyright:** This article is an open access article distributed under the terms and conditions of the

**Appendix**

Figure 4 is a Dev C++ 5.11 coding for the Runge-Kutta-Fehlberg (RKF45) Method of order 4 and 5.

```

RKF45 Method.cpp  Untitled1.cpp
1  /* RUNGE-KUTTA-FEHLBERG (RKF45) METHOD */
2  /* PROBLEM SET 6 */
3  #include <iostream>
4  #include <math.h>
5  #include <iomanip>      // For setw, setprecision & setfill
6  using namespace std;
7
8  #define f(t,y) (3*Q/A)*pow(sin(t),2)-0.4167 // Defining ODE to be solved "Problem Set 6"
9  int main()
10 {
11     float t0, y0, tn, h, k1, k2, k3, k4, k5, k6, y_4, y_5;
12     int i, n;
13     int Q=500, A=1200;      // Defining Flowrate (Q) and Area (A)
14     cout<<"RUNGE-KUTTA-FEHLBERG (RKF45) METHOD"<<endl<<endl;
15     cout<<"Enter Initial Condition: t = ";
16     cin>>t0;
17     cout<<setw(29)<<"y = ";
18     cin>>y0;
19     cout<<"\n"<<"Calculation point, tn = ";
20     cin>>tn;      // Stopping value of t
21     cout<<"Suitable Step Size, h = ";
22     cin>>h;
23     cout<<endl;
24     n=(tn-t0)/h;      // Number of Iterations, n
25
26     cout<<setw(3)<<"i"<<setw(10)<<"t(i)"<<setw(11)<<"k1"<<setw(10)<<"k2"<<setw(13)<<"k3"
27     <<setw(15)<<"k4"<<setw(16)<<"k5"<<setw(17)<<"k6"<<setw(18)<<"y(i)4"<<setw(20)<<"y(i)5"<<endl;
28
29     for (i=1; i<=n; i++)
30     {
31         k1=h*(f(t0,y0));      // RKF45 parameters: k1, k2, k3, k4, k5 & k6
32         k2=h*(f((t0+h/4),(y0+k1/4)));
33         k3=h*(f((t0+3*h/8),(y0+3*k1/32+9*k2/32)));
34         k4=h*(f((t0+12*h/13),(y0+1932*k1/2197-7200*k2/2197+7296*k3/2197)));
35         k5=h*(f((t0+h),(y0+439*k1/216-8*k2+3680*k3/513-845*k4/4104)));
36         k6=h*(f((t0+h/2),(y0-8*k1/27+2*k2-3544*k3/2565+1859*k4/4104-11*k5/40)));
37         y_4=y0+(25*k1/216)+(1408*k3/2565)+(2197*k4/4104)-k5/5;      // RKF45 METHOD of ORDER 4
38         y0=y_4;
39
40         y_5=y0+(16*k1/135)+(6656*k3/12825)+(28561*k4/56460)-(9*k5/50)+(2*k6/55); // RKF45 METHOD of ORDER 5
41         tn=t0+h;      // Time interval equation
42         t0=tn;
43         cout<<setw(3)<<i<<setw(10)<<t0<<setw(11)<<k1<<setw(10)<<k2<<setw(13)<<k3<<setw(15)
44         <<k4<<setw(16)<<k5<<setw(17)<<k6<<setw(18)<<y0<<setw(20)<<y_5<<endl;
45     }
46     cout<<"\n"<<"Value of y RKF45 of ORDER 4 after "<<n<<" iterations, at t = "<<tn<<" is "<<y_4<<endl;
47     cout<<"\n"<<"Value of y RKF45 of ORDER 5 after "<<n<<" iterations, at t = "<<tn<<" is "<<y_5;
48     return 0;
    }
    
```

Figure 4: Problem Set 6 Coded C++ Solution

As explained earlier the RKF45 method as used in POLYMATH selects the appropriate step size on its own for the solution of a given problem. Entering a very

minute step size in C++ execution of Problem Set 6 will display a lengthy table of calculation before given the result, simply because, the smaller the value of h, the



higher the number of iterations. This is the reason why, the execution of the code in Figure 4, for a step size of  $h = 0.1$  gave  $y = 0.604765m$  and  $0.58828m$  (for RKF45 of order 4 & 5 respectively) in Figure 5, which is not the same with the value obtained by POLYMATH and Excel, which are  $y = 1.798038m$ . The step size used by POLYMATH for Problem Set 6 is  $h = 0.000001$ . Despite that, the C++ result cannot be discarded because it falls between the minimal and maximal value given by POLYMATH in Figure 6.

Note that varying the step size,  $h$ , during C++ execution of the problem will definitely affect the result. A step size of  $h = 0.001, 0.01$  and  $0.5$  gives  $0.604776$  &  $0.604656, 0.604736$  &  $0.603485$  and  $0.604765$  &  $0.455666m$  for RKF45 of order 4 and 5 respectively. Therefore, in chemical engineering where time will not permit the use of C++ codes due to the 'h' myths, POLYMATH have an edge in this case, over C++ and other coding software programs for solving ODEs by Runge-Kutta-Fehlberg method.

```

RUNGE-KUTTA-FEHLBERG (RKF45) METHOD
Enter Initial Condition: t = 0
                      y = 0
Calculation point, tn = 10
Suitable Step Size, h = 0.1
    
```

i	t(i)	k1	k2	k3	k4	k5	k6	y(i)4	y(i)5
1	0.1	-0.04167	-0.0416075	-0.0415294	-0.0408203	-0.0406733	-0.0414202	-0.0413373	-0.0826637
2	0.2	-0.0406733	-0.0401156	-0.0397913	-0.0380171	-0.037723	-0.0394368	-0.0806946	-0.120042
3	0.3	-0.037723	-0.0366924	-0.0361346	-0.0333662	-0.0329368	-0.0335491	-0.116171	-0.151638
4	0.4	-0.0329368	-0.0314742	-0.0307053	-0.027053	-0.0265053	-0.0299121	-0.146019	-0.17586
5	0.5	-0.0265053	-0.0246692	-0.0237198	-0.0193292	-0.0186851	-0.0227505	-0.168718	-0.191411
6	0.6	-0.0186851	-0.0165486	-0.0154566	-0.0105027	-0.00978789	-0.0143498	-0.18303	-0.197339
7	0.7	-0.00978789	-0.00743612	-0.00624508	-0.000925358	-0.000168352	-0.00504494	-0.188052	-0.193075
8	0.8	-0.000168352	-0.00230486	0.00354751	0.00902093	0.00978998	0.00479315	-0.183253	-0.178457
9	0.9	0.00978998	0.012286	0.0135308	0.0189397	0.0196901	0.0147722	-0.168492	-0.153735
10	1	0.0196901	0.0221095	0.0233067	0.0284355	0.0291374	0.0244945	-0.144024	-0.119564
11	1.1	0.0291374	0.0313836	0.0324855	0.0371297	0.0377551	0.0335723	-0.110494	-0.0769738
12	1.2	0.0377551	0.0397387	0.0407014	0.0446758	0.0451997	0.0416438	-0.0689058	-0.0273295
13	1.3	0.0451997	0.0468416	0.0476267	0.0507729	0.0511744	0.0483872	-0.0205853	0.0277215
14	1.4	0.0511744	0.0524091	0.0529853	0.0551779	0.0554411	0.0535336	0.032873	0.0863165
15	1.5	0.0554411	0.0562194	0.0565637	0.0577152	0.0578296	0.0568779	0.08967	0.146452
16	1.6	0.0578296	0.0581204	0.0582192	0.0582837	0.0582447	0.0582868	0.147874	0.206061
17	1.7	0.0582447	0.0580365	0.0578857	0.0568607	0.0566699	0.057704	0.205495	0.263102
18	1.8	0.0566699	0.0559709	0.0555766	0.053503	0.0531679	0.0551528	0.26057	0.315631
19	1.9	0.0531679	0.052006	0.051384	0.0483443	0.0478784	0.050735	0.311235	0.361886
20	2	0.0478784	0.0462999	0.0454749	0.0415904	0.0410122	0.0446266	0.355801	0.400356
21	2.1	0.0410122	0.0390801	0.0380849	0.0335105	0.032843	0.0370712	0.392824	0.429838
22	2.2	0.032843	0.0306344	0.0295088	0.0244268	0.0236966	0.02837	0.42116	0.44949
23	2.3	0.0236966	0.0212994	0.0200883	0.0147013	0.0139076	0.0188698	0.440013	0.458861
24	2.4	0.0139076	0.0114474	0.010199	0.00472181	0.00395507	0.00894944	0.448961	0.457908
25	2.5	0.00395507	0.00147115	0.000235257	-0.00511382	-0.00585309	-0.0009956	0.447981	0.447002
26	2.6	-0.00585309	-0.00823168	-0.0094058	-0.0144135	-0.0150958	-0.0105689	0.437444	0.42691
27	2.7	-0.0150958	-0.0172743	-0.0183398	-0.0228065	-0.0234046	-0.0193887	0.418101	0.398765
28	2.8	-0.0234046	-0.0252961	-0.0262105	-0.0299581	-0.0304483	-0.0271035	0.391057	0.36402

29	2.9	-0.0304483	-0.0319773	-0.0327043	-0.0355833	-0.0359459	-0.0334056	0.357721	0.324394
30	3	-0.0359459	-0.0370516	-0.0375621	-0.0394579	-0.0396785	-0.0380439	0.319754	0.281798
31	3.1	-0.0396785	-0.0403167	-0.0405904	-0.0414273	-0.0414971	-0.0408334	0.279003	0.238262
32	3.2	-0.0414971	-0.0416425	-0.0416683	-0.041413	-0.0413293	-0.0416629	0.237423	0.195854
33	3.3	-0.0413293	-0.0409759	-0.040753	-0.0394157	-0.0391817	-0.0404994	0.197805	0.156597
34	3.4	-0.0391817	-0.0383438	-0.0378809	-0.0355148	-0.0351399	-0.0373892	0.159692	0.122388
35	3.5	-0.0351399	-0.0338508	-0.0331666	-0.0298661	-0.0293652	-0.0324563	0.127303	0.09423
36	3.6	-0.0293652	-0.0276763	-0.0267978	-0.0226945	-0.0220877	-0.0258974	0.101463	0.0756287
37	3.7	-0.0220877	-0.0200663	-0.0190287	-0.0142862	-0.0135975	-0.017974	0.0835328	0.0656065
38	3.8	-0.0135975	-0.0113242	-0.0101689	-0.00497615	-0.00423314	-0.0090019	0.0745598	0.0655881
39	3.9	-0.00423314	-0.00179864	-0.000571533	-0.00486433	-0.00563207	0.000661156	0.0752337	0.0759064
40	4	0.00563207	0.00813077	0.00938067	0.014843	0.0156048	0.0106299	0.0858599	0.096482
41	4.1	0.0156048	0.0180681	0.019291	0.024562	0.0252876	0.020507	0.106347	0.126827
42	4.2	0.0252876	0.0276173	0.0287644	0.0336339	0.0342943	0.0298987	0.136209	0.166063
43	4.3	0.0342943	0.0363975	0.037423	0.041697	0.0422659	0.0384305	0.17459	0.212959
44	4.4	0.0422659	0.0440587	0.0449219	0.0484298	0.0488845	0.0457622	0.22029	0.265976
45	4.5	0.0488845	0.0502956	0.0509619	0.053564	0.0538864	0.0516017	0.271819	0.323334
46	4.6	0.0538864	0.0548594	0.0553024	0.0568949	0.0570721	0.055716	0.327456	0.383078
47	4.7	0.0570721	0.0575682	0.0577702	0.0582897	0.0583146	0.0579412	0.385315	0.443158
48	4.8	0.0583146	0.0583141	0.058267	0.0576927	0.0575644	0.0581886	0.44342	0.50151
49	4.9	0.0575644	0.0570673	0.0567729	0.0551278	0.0548514	0.0564483	0.499789	0.556142
50	5	0.0548514	0.0538775	0.0533476	0.0506972	0.0502837	0.0527897	0.552504	0.605206
51	5.1	0.0502837	0.0488718	0.0481276	0.0445776	0.0440435	0.0473585	0.599798	0.647079
52	5.2	0.0440435	0.0422499	0.041321	0.0370129	0.0363794	0.0403715	0.640116	0.680424
53	5.3	0.0363794	0.0342757	0.0331992	0.0283047	0.0275972	0.0321071	0.672183	0.704243
54	5.4	0.0275972	0.0252671	0.0240858	0.0188002	0.0180468	0.0228947	0.695054	0.717919
55	5.5	0.0180468	0.0155833	0.0143444	0.00887822	0.008109	0.0131017	0.708148	0.721239
56	5.6	0.008109	0.00561033	0.00436318	-0.00106556	-0.00181995	0.00311846	0.711275	0.714402
57	5.7	-0.00181995	-0.00425421	-0.00545989	-0.0106348	-0.0113443	-0.006657	0.704643	0.698014
58	5.8	-0.0113443	-0.0136171	-0.0147332	-0.0194479	-0.0200842	-0.015835	0.688848	0.673059
59	5.9	-0.0200842	-0.0221049	-0.0230871	-0.0271537	-0.0276914	-0.0240496	0.664852	0.640864
60	6	-0.0276914	-0.0293795	-0.0301884	-0.0334448	-0.0338625	-0.0309733	0.633945	0.603046
61	6.1	-0.0338625	-0.0351507	-0.0357542	-0.0380705	-0.0383516	-0.0363302	0.597689	0.561443
62	6.2	-0.0383516	-0.0391884	-0.0395625	-0.0408463	-0.0409796	-0.0399065	0.557863	0.518047
63	6.3	-0.0409796	-0.0413218	-0.0414614	-0.0416617	-0.0416617	-0.0415599	0.516386	0.47492
64	6.4	-0.0416617	-0.0414953	-0.0413753	-0.040484	-0.0403117	-0.0412243	0.475244	0.434113
65	6.5	-0.0403117	-0.0396724	-0.0393077	-0.0373603	-0.0370425	-0.0389131	0.43641	0.397585

66	6.6	-0.0370425	-0.0359358	-0.0353408	-0.0324151	-0.0319644	-0.0347185	0.401763	0.367125
67	6.7	-0.0359358	-0.0340345	-0.0329633	-0.0258455	-0.02528	-0.0288077	0.373917	0.344278
68	6.8	-0.02528	-0.0233878	-0.0224117	-0.0179134	-0.0172556	-0.0214163	0.35165	0.330288
69	6.9	-0.0172556	-0.0150766	-0.0139648	-0.00893507	-0.00821114	-0.0128391	0.338846	0.326045
70	7	-0.00821114	-0.00583221	-0.00462912	-0.000731546	-0.00149271	-0.00341788	0.335448	0.332095
71	7.1	0.00149271	0.00397675	0.00522322	0.0107011	0.0114691	0.00647167	0.341923	0.348393
72	7.2	0.0114691	0.0139593	0.0151994	0.0205761	0.0213204	0.0164353	0.358345	0.374761
73	7.3	0.0213204	0.0237174	0.0249017	0.0299629	0.0306538	0.0260758	0.384391	0.410429
74	7.4	0.0306538	0.032862	0.0339434	0.0384873	0.0390973	0.0350088	0.419355	0.454309
75	7.5	0.0390973	0.0410286	0.041964	0.0458094	0.0463141	0.0428783	0.462176	0.504984
76	7.6	0.0463141	0.0478917	0.0486436	0.0516373	0.0520166	0.0493704	0.511478	0.560766
77	7.7	0.0520166	0.0531775	0.0537161	0.0557387	0.0559775	0.0542264	0.565628	0.619763
78	7.79999	0.0559775	0.0566755	0.0569792	0.0579501	0.0580388	0.0572526	0.622799	0.679954
79	7.89999	0.0580388	0.058246	0.0583028	0.0581832	0.0581184	0.0583284	0.681044	0.739273
80	7.99999	0.0581184	0.0578266	0.0576342	0.0564289	0.0562131	0.057411	0.738373	0.795687
81	8.09999	0.0562131	0.0554338	0.0549999	0.052757	0.0523989	0.0545369	0.792833	0.847279
82	8.2	0.0523989	0.0511633	0.0505051	0.047314	0.0468277	0.0498206	0.842585	0.892323
83	8.3	0.0468277	0.045185	0.0443288	0.0403167	0.0397218	0.0434502	0.885976	0.929357
84	8.4	0.0397218	0.0377375	0.0367174	0.0320443	0.0313643	0.0356797	0.92161	0.957236
85	8.5	0.0313643	0.0291176	0.0279743	0.0228264	0.0220885	0.0268188	0.948398	0.97518
86	8.6	0.0220885	0.0196689	0.018448	0.0130306	0.0122643	0.0172208	0.965604	0.982807
87	8.7	0.0122643	0.00976816	0.00851828	0.0030474	0.00228312	0.00726838	0.972875	0.980144
88	8.8	0.00228312	-0.000189899	-0.0014189	-0.00672521	-0.00745693	-0.00264172	0.970251	0.967629
89	8.9	-0.00745693	-0.0098083	-0.0109674	-0.0158976	-0.0165676	-0.0121144	0.958171	0.946095
90	9	-0.0165676	-0.0187036	-0.0197466	-0.0241041	-0.0246857	-0.020772	0.937447	0.91673
91	9.1	-0.0246857	-0.0265211	-0.0274065	-0.0310176	-0.0314875	-0.0282695	0.909239	0.881039
92	9.2	-0.0314875	-0.0329492	-0.0336416	-0.0363624	-0.036702	-0.0343078	0.875002	0.840775
93	9.3	-0.036702	-0.0377316	-0.0382035	-0.0399254	-0.0401211	-0.0386462	0.836434	0.797877
94	9.4	-0.0401211	-0.0406777	-0.0409102	-0.0415646	-0.0416086	-0.0411119	0.795404	0.754386
95	9.5	-0.0416086	-0.04167	-0.0416538	-0.0412147	-0.0411052	-0.0416064	0.753881	0.712369
96	9.6	-0.0411052	-0.0406689	-0.0404047	-0.0388895	-0.038631	-0.0401101	0.713852	0.673833
97	9.7	-0.038631	-0.0377144	-0.0372127	-0.0346819	-0.0342846	-0.0366827	0.677244	0.640646
98	9.8	-0.0342846	-0.0329242	-0.0322051	-0.0287595	-0.0282392	-0.0314607	0.64585	0.614463
99	9.9	-0.0282392	-0.0264893	-0.0255814	-0.0213584	-0.020736	-0.0246524	0.621252	0.596661
100	10	-0.0264893	-0.0186663	-0.0176058	-0.0127738	-0.0120739	-0.0165291	0.604765	0.58828

Value of y RKF45 of ORDER 4 after 100 iterations, at t = 10 is 0.604765

Value of y RKF45 of ORDER 5 after 100 iterations, at t = 10 is 0.58828

Figure 5: RKF45 C++ Output Results of Problem Set 6

The continuity equation in Problem Set 6 is a single first order differential equation. Column 2 of Table 2 gives the solution of the Monod Model Kinetic equation, often used to explain growth and decay rate of microbes in bioreaction engineering. Problem 7, though solved by RKF56 method in POLYMATH, differs from Problem 6, being a multiple first order differential equation solved simultaneously. Table 2: POLYMATH Result of Problem Set 6 and 7



### Problem Set 6

d/dt x= ini-finl i x →

RKF45

Differential Equations: 1 Auxiliary Equations: 2 ✓ Ready for solution

# CONTINUITY EQUATION OF LIQUID FLOW  
 $d(y) / d(t) = 3*(Q/A)*sin(t)*sin(t)-(Q/A)$

# Explicit Equation  
 $Q = 500$  # Flowrate (m3/day)  
 $A = 1200$  # Area (m2)

# Time Range (0 to 10 days)  
 $t(0) = 0$  # Initial Time (days)  
 $t(f) = 10$  # Final Time (days)

# Initial Condition  
 $y(0) = 0$  # Depth of the liquid (m)

**POLYMATH Report**  
 Ordinary Differential Equations

**Calculated values of DEQ variables**

Variable	Initial value	Minimal value	Maximal value	Final value
1 A	1200.	1200.	1200.	1200.
2 Q	500.	500.	500.	500.
3 t	0	0	10.	10.
4 y	0	-0.1645208	2.12963	1.798038

**Differential equations**

1  $d(y)/d(t) = 3*(Q/A)*sin(t)*sin(t)-(Q/A)$

**Explicit equations**

1  $Q = 500$   
Flowrate (m3/day)

2  $A = 1200$   
Area (m2)

### Problem Set 7

d/dt x= ini-finl i x →

RKF56

Table Graph ✓ Report

Differential Equations: 2 Auxiliary Equations: 4 ✓ Ready for solution

# MONOD MODEL: BIOREACTION KINETICS  
 $d(s) / d(t) = -(k*s*x)/(ks+s)$   
 $d(x) / d(t) = y*(k*s*x)/(ks+s)-b*x$

# Explicit Equation  
 $k = 5$  # Maximum specific uptake rate of the substrate  
 $b = 0.01$  # Decay coefficient  
 $y = 0.05$  # Yield coefficient  
 $ks = 20$  # Half saturation constant for growth

# Initial Conditions  
 $s(0) = 1000$  # Growth limiting substrate concentration  
 $x(0) = 100$  # Biomass concentration  
 $t(0) = 0$  # Initial time  
 $t(f) = 1$  # Final time

**POLYMATH Report**  
 Ordinary Differential Equations

**Calculated values of DEQ variables**

Variable	Initial value	Minimal value	Maximal value	Final value
1 b	0.01	0.01	0.01	0.01
2 k	5.	5.	5.	5.
3 ks	20.	20.	20.	20.
4 s	1000.	452.5393	1000.	452.5393
5 t	0	0	1.	1.
6 x	100.	100.	126.2464	126.2464
7 y	0.05	0.05	0.05	0.05

**Differential equations**

1  $d(s)/d(t) = -(k*s*x)/(ks+s)$

2  $d(x)/d(t) = y*(k*s*x)/(ks+s)-b*x$