

Numerical Methods in Chemical Engineering Lab (Practical)



Department of Chemical Engineering
S.B.S STATE UNIVERSITY GURDASPUR



Sardar Beant Singh State University Gurdaspur

(Established by the Government of Punjab –
Recognised under UGC Act, 1956)

Department of Chemical Engineering



Numerical Methods in Chemical Engineering Lab

Course Outcomes

CO1: Learn to obtain numerical solutions of nonlinear equations using bisection, secant, Newton and fixed-point iteration methods using a programming language.

CO2: Solve systems of linear equations numerically using direct and iterative methods using a programming language.

CO3: Apply basic concepts of ordinary differential equations and integration and their applications in chemical engineering.

Experiment-wise Mapping

Experiment	Content	CO	Bloom's Level
1	Introduction to computers for numerical calculations	CO1, CO2, CO3	L2 – Understand
2	Gauss elimination, Gauss-Seidel	CO2	L3 – Apply
3	Nonlinear equations using bracketing/open methods	CO1	L3 – Apply
4	Interpolation and approximation	CO3	L3 – Apply
5	Numerical integration	CO3	L3 – Apply
6	Euler method	CO3	L3 – Apply
7	Runge-Kutta methods for ODEs	CO3	L3 – Apply
8	System of ODEs	CO3	L4 – Analyze
9	Solution of simple PDEs	CO3	L4 – Analyze

CO-Bloom Mapping

CO	Bloom's Level	Justification
CO1	L3 – Apply	Programming and application of nonlinear equation algorithms
CO2	L3 – Apply	Numerical implementation of direct and iterative methods
CO3	L3-L4 – Apply/Analyze	Application and analysis of ODE, integration and PDE problems



Sardar Beant Singh State University, Gurdaspur
Department of Chemical Engineering

Bisection Method Algorithm

1. Start
2. Define function $f(x)$
3. Choose initial guesses x_0 and x_1 such that $f(x_0)f(x_1) < 0$
4. Choose pre-specified tolerable error e .
5. Calculate new approximated root as $x_2 = (x_0 + x_1)/2$
6. Calculate $f(x_0)f(x_2)$
 - a. if $f(x_0)f(x_2) < 0$ then $x_0 = x_0$ and $x_1 = x_2$
 - b. if $f(x_0)f(x_2) > 0$ then $x_0 = x_2$ and $x_1 = x_1$
 - c. if $f(x_0)f(x_2) = 0$ then goto (8)
7. if $|f(x_2)| > e$ then goto (5) otherwise goto (8)
8. Display x_2 as root.
9. Stop



Sardar Beant Singh State University, Gurdaspur
Department of Chemical Engineering

Algorithm for False Position Method

1. start
2. Define function $f(x)$
3. Choose initial guesses x_0 and x_1 such that $f(x_0)f(x_1) < 0$
4. Choose pre-specified tolerable error e .
5. Calculate new approximated root as:
$$x_2 = x_0 - ((x_0 - x_1) * f(x_0)) / (f(x_0) - f(x_1))$$
6. Calculate $f(x_0)f(x_2)$
 - a. if $f(x_0)f(x_2) < 0$ then $x_0 = x_0$ and $x_1 = x_2$
 - b. if $f(x_0)f(x_2) > 0$ then $x_0 = x_2$ and $x_1 = x_1$
 - c. if $f(x_0)f(x_2) = 0$ then goto (8)
7. if $|f(x_2)| > e$ then goto (5) otherwise goto (8)
8. Display x_2 as root.
9. Stop



Sardar Beant Singh State University, Gurdaspur
Department of Chemical Engineering

Algorithm: Secant Method

1. Start
2. Define function as $f(x)$
3. Input initial guesses (x_0 and x_1),
tolerable error (e) and maximum iteration (N)
4. Initialize iteration counter $i = 1$
5. If $f(x_0) = f(x_1)$ then print "Mathematical Error"
and goto (11) otherwise goto (6)
6. Calculate $x_2 = x_1 - (x_1 - x_0) * f(x_1) / (f(x_1) - f(x_0))$
7. Increment iteration counter $i = i + 1$
8. If $i \geq N$ then print "Not Convergent"
and goto (11) otherwise goto (9)
9. If $|f(x_2)| > e$ then set $x_0 = x_1$, $x_1 = x_2$
and goto (5) otherwise goto (10)
10. Print root as x_2
11. Stop



Sardar Beant Singh State University, Gurdaspur
Department of Chemical Engineering

Algorithm for Newton Raphson Method

1. Start
2. Define function as $f(x)$
3. Define first derivative of $f(x)$ as $g(x)$
4. Input initial guess (x_0), tolerable error (e)
and maximum iteration (N)
5. Initialize iteration counter $i = 1$
6. If $g(x_0) = 0$ then print "Mathematical Error"
and goto (12) otherwise goto (7)
7. Calculate $x_1 = x_0 - f(x_0) / g(x_0)$
8. Increment iteration counter $i = i + 1$
9. If $i \geq N$ then print "Not Convergent"
and goto (12) otherwise goto (10)
10. If $|f(x_1)| > e$ then set $x_0 = x_1$
and goto (6) otherwise goto (11)
11. Print root as x_1
12. Stop



Sardar Beant Singh State University, Gurdaspur
Department of Chemical Engineering

Algorithm for Gauss Elimination Method

1. Start
2. Read Number of Unknowns: n
3. Read Augmented Matrix (A) of n by $n+1$ Size
4. Transform Augmented Matrix (A)
to Upper Triangular Matrix by Row Operations.
5. Obtain Solution by Back Substitution.
6. Display Result.
7. Stop



Sardar Beant Singh State University, Gurdaspur
Department of Chemical Engineering

Algorithm for Gauss Jordan Method

1. Start
2. Read Number of Unknowns: n
3. Read Augmented Matrix (A) of n by $n+1$ Size
4. Transform Augmented Matrix (A)
to Diagonal Matrix by Row Operations.
5. Obtain Solution by Making All Diagonal Elements to 1.
6. Display Result.
7. Stop



Sardar Beant Singh State University, Gurdaspur
Department of Chemical Engineering

Algorithm for Power Method

1. Start
2. Read Order of Matrix (n) and Tolerable Error (e)
3. Read Matrix A of Size n x n
4. Read Initial Guess Vector X of Size n x 1
5. Initialize: $\text{Lambda_Old} = 1$
6. Multiply: $X_NEW = A * X$
7. Replace X by X_NEW
8. Find Largest Element (Lamda_New) by Magnitude from X_NEW
9. Normalize or Divide X by Lamda_New
10. Display Lamda_New and X
11. If $|\text{Lambda_Old} - \text{Lamda_New}| > e$ then
 set $\text{Lambda_Old} = \text{Lamda_New}$ and goto
 step (6) otherwise goto step (12)
12. Stop



Sardar Beant Singh State University, Gurdaspur
Department of Chemical Engineering

Gauss Seidel Iterative Method Algorithm

1. Start
2. Arrange given system of linear equations in diagonally dominant form
3. Read tolerable error (ϵ)
4. Convert the first equation in terms of first variable, second equation in terms of second variable and so on.
5. Set initial guesses for x_0 , y_0 , z_0 and so on
6. Substitute value of y_0 , z_0 ... from step 5 in first equation obtained from step 4 to calculate new value of x_1 . Use x_1 , z_0 , u_0 in second equation obtained from step 4 to calculate new value of y_1 . Similarly, use x_1 , y_1 , u_0 ... to find new z_1 and so on.
7. If $|x_0 - x_1| > \epsilon$ and $|y_0 - y_1| > \epsilon$ and $|z_0 - z_1| > \epsilon$ and so on then goto step 9
8. Set $x_0 = x_1$, $y_0 = y_1$, $z_0 = z_1$ and so on and goto step 6
9. Print value of x_1 , y_1 , z_1 and so on
10. Stop



Sardar Beant Singh State University, Gurdaspur
Department of Chemical Engineering

Algorithm: Lagrange Interpolation Method

1. Start
2. Read number of data (n)
3. Read data X_i and Y_i for $i=1$ to n
4. Read value of independent variables say x_p
whose corresponding value of dependent say y_p is to be determined.
5. Initialize: $y_p = 0$
6. For $i = 1$ to n
Set $p = 1$
For $j = 1$ to n
If $i \neq j$ then
Calculate $p = p * (x_p - X_j) / (X_i - X_j)$
End If
Next j
Calculate $y_p = y_p + p * Y_i$
Next i
6. Display value of y_p as interpolated value.
7. Stop



Sardar Beant Singh State University, Gurdaspur
Department of Chemical Engineering

Linear Regression Algorithm (Fitting $y = a + bx$)

1. Start
2. Read Number of Data (n)
3. For $i=1$ to n :
 Read X_i and Y_i
Next i
4. Initialize:
 $\text{sumX} = 0$
 $\text{sumX2} = 0$
 $\text{sumY} = 0$
 $\text{sumXY} = 0$
5. Calculate Required Sum
 For $i=1$ to n :
 $\text{sumX} = \text{sumX} + X_i$
 $\text{sumX2} = \text{sumX2} + X_i * X_i$
 $\text{sumY} = \text{sumY} + Y_i$
 $\text{sumXY} = \text{sumXY} + X_i * Y_i$
 Next i
6. Calculate Required Constant a and b of $y = a + bx$:
 $b = (n * \text{sumXY} - \text{sumX} * \text{sumY}) / (n * \text{sumX2} - \text{sumX} * \text{sumX})$
 $a = (\text{sumY} - b * \text{sumX}) / n$
7. Display value of a and b
8. Stop



Sardar Beant Singh State University, Gurdaspur
Department of Chemical Engineering

Trapezoidal Method Algorithm

1. Start
2. Define function $f(x)$
3. Read lower limit of integration, upper limit of integration and number of sub interval
4. Calculate: step size = (upper limit - lower limit) / number of sub interval
5. Set: integration value = $f(\text{lower limit}) + f(\text{upper limit})$
6. Set: $i = 1$
7. If $i > \text{number of sub interval}$ then goto
8. Calculate: $k = \text{lower limit} + i * h$
9. Calculate: Integration value = Integration Value + $2 * f(k)$
10. Increment i by 1 i.e. $i = i + 1$ and go to step 7
11. Calculate: Integration value = Integration value * step size / 2
12. Display Integration value as required answer
13. Stop



Sardar Beant Singh State University, Gurdaspur
Department of Chemical Engineering

Simpson's 1/3 Rule Algorithm

1. Start
2. Define function $f(x)$
3. Read lower limit of integration, upper limit of integration and number of sub interval
4. Calculate: step size = (upper limit - lower limit) / number of sub interval
5. Set: integration value = $f(\text{lower limit}) + f(\text{upper limit})$
6. Set: $i = 1$
7. If $i > \text{number of sub interval}$ then goto
8. Calculate: $k = \text{lower limit} + i * h$
9. If $i \bmod 2 = 0$ then
 Integration value = Integration Value + $2 * f(k)$
 Otherwise
 Integration Value = Integration Value + $4 * f(k)$
 End If
10. Increment i by 1 i.e. $i = i + 1$ and go to step 7
11. Calculate: Integration value = Integration value * step size / 3
12. Display Integration value as required answer
13. Stop



Sardar Beant Singh State University, Gurdaspur
Department of Chemical Engineering

Euler's Method Algorithm (Ordinary Differential Equation)

1. Start
2. Define function $f(x,y)$
3. Read values of initial condition(x_0 and y_0),
number of steps (n) and calculation point (x_n)
4. Calculate step size (h) = $(x_n - x_0) / b$

5. Set $i=0$

6. Loop

$$y_n = y_0 + h * f(x_0 + i*h, y_0)$$

$$y_0 = y_n$$

$$i = i + 1$$

While $i < n$

7. Display y_n as result

8. Stop



Sardar Beant Singh State University, Gurdaspur
Department of Chemical Engineering

Ordinary Differential Equation Using Fourth Order Runge Kutta (RK)

Method Algorithm

1. Start
2. Define function $f(x,y)$
3. Read values of initial condition(x_0 and y_0),
number of steps (n) and calculation point (x_n)
4. Calculate step size (h) = $(x_n - x_0) / n$
5. Set $i=0$
6. Loop
 - $k_1 = h * f(x_0, y_0)$
 - $k_2 = h * f(x_0+h/2, y_0+k_1/2)$
 - $k_3 = h * f(x_0+h/2, y_0+k_2/2)$
 - $k_4 = h * f(x_0+h, y_0+k_3)$
 - $k = (k_1+2*k_2+2*k_3+k_4) / 6$
 - $y_n = y_0 + k$
 - $i = i + 1$
 - $x_0 = x_0 + h$
 - $y_0 = y_n$
- While $i < n$
7. Display y_n as result
8. Stop