Numerical Methods for Mechanical Engineers
Assignment No.1
Solution of Equations and Eigen Value Problems
Submission date : 24-8-2018

1. Fanning friction factor f for turbulent flows in pipe is given by

$$\frac{1}{\sqrt{f}} = 4log_{10}\left(Re\sqrt{f}\right) - 0.4.$$

Write a computer program to take input from the user and determine *f* for water flow through a smooth pipe using fixed iteration method and Newton-Raphson method. For this case, the parameters are ρ = 999 kg/m^3, μ = 1.0 × 10^{-3} Ns/m^2, *D* = 0.075 m and Q = 0.0084 m^3/s. Take 0.001 as initial guess and ε_s = 0.5%.

Algorithm (Fixed iteration method)

1. Take the input from the user: density, viscosity, diameter, flow rate, initial guess, maximum no. of iterations and limit for approximation error.
2. Calculate velocity and Reynolds no.
3. Initialize fold = initial guess and iteration no. = 0
4. Calculate fnew=g(fold).
5. Calculate % error and increase iteration by 1.
6. Check if % error is less than limit or no. of iterations > maximum no. If yes, stop and display value of f, iteration no. and error, else set fold = fnew and goto step 4.

Algorithm (Newton-Raphson method)

1. Take the input from the user: density, viscosity, diameter, flow rate, initial guess, maximum no. of iterations and limit for approximation error.
2. Calculate velocity and Reynolds no.
3. Initialize fold = initial guess and iteration no. = 0
4. Calculate $f_{new} = f_{old} - \frac{f(f_{old})}{f'(f_{old})}$
5. Calculate % error and increase iteration by 1.
6. Check if % error is less than limit or no. of iterations > maximum no. If yes, stop and display value of f, iteration no. and error, else set fold = fnew and goto step 4.

2. Three masses are suspended vertically by a series of identical springs where mass 1 is at the top and mass 3 is at the bottom. If g = 9.81 m/s^2, m_1 = 2 kg, m_2 = 3 kg, m_3 = 2.5 kg, and the *k*'s = 10 kg/s^2, write a program to solve for the displacements *x* using Gauss Elimination/Gauss Jordan method.

Algorithm

1. Take the input from the user: masses, g and stiffness of each spring.
2. Define elements of the coefficients matrix and right side constants.
3. Define n = no. of equations
4. **Forward Elimination**
   for k = 0 to n-2
   for i = k+1 to n-1
   factor = $a_{i,k} / a_{k,k}$
   for j= k+1 to n-1
   ai, j = $a_{i,j}$ *- factor** $a_{k,j}$
   end
   $c_i = c_i - factor * c_k$

end
end

5. Backward Substitution
   $x_{n-1} = b_{n-1} / a_{n-1,\, n-1}$
   for i = n-2 to 0
   $sum = b_i$
   for j = i+1 to n-1
   $sum = sum - a_{i,j} * x_j$
   End
   $x_i = sum / a_{i,i}$
6. Display values of displacements.

3. Consider a two-dimensional rectangular plate of dimension L = 1 m in the x direction and H = 1 m in the y direction. The plate material has constant thermal conductivity. The steady- state temperature distribution within this plate is to be determined for the following imposed boundary conditions: (i) y = 0, T = 200^{0}C, (ii) x = 0, T = 100^{0}C, (iii) y = H, T = 400^{0}C, and (iv) x = L, T = 300^{0}C. Choose a uniform grid size of 0.25 m in both directions. Write a program to solve the problem using finite difference method and point-by-point Gauss-Seidel iterative method. Use W-E direction first and S-N later.

Algorithm

1. Take the input from the user: Boundary condition information, length, height, no. of grid points in x-direction and no. of grid points in y-direction, Initial guess.
2. Calculate x and y coordinates for all points.
3. Implement all boundary conditions.
4. Set T_{old} = guess for all cv's and T_{new} = 0 for all cv's.
5. Calculate T_{new} at all cv's using Gauss-Seidal point by point.
   For i= 2 to m+1
   For j = 2 to n+1
   $$T_{i,j}^{new} = \frac{T_{i-1,j}^{new} + T_{i+1,j}^{new} + T_{i,j-1}^{new} + T_{i,j+1}^{new}}{4}$$
   Calculate Residual = $T_{i,j}^{new} - T_{i,j}^{old}$
   Assign $T_{i,j}^{old} = T_{i,j}^{new}$
6. Check for convergence for iterations (max residual <ε)
7. If converged, stop and display temperatures with x and y coordinates of the points.

4. Polynomial interpolation consists of determining the unique $(n - 1)^{th}$ order polynomial that fits n data points. Such polynomials have the general form,
   $$f(x) = p_1x^{n-1} + p_2x^{n-2} + \cdot \cdot \cdot + p_{n-1}x + p_n \qquad (1)$$
   where the p's are constant coefficients. A straightforward way for computing the coefficients is to generate n linear algebraic equations that we can solve simultaneously for the coefficients. Suppose that we want to determine the coefficients of the fourth-order polynomial $f(x) = p_1x^4 + p_2x^3 + p_3x^2 + p_4x + p_5$ that passes through the following five points: (2, 0.746), (2.5, 0.675), (3, 0.616), (4, 0.525), and (5, 0.457). Each of these pairs can be substituted into Eq. (1) to yield a system of five equations with five unknowns (the p's). Use this approach to write a program to solve for the coefficients.

Algorithm

1. Take the input from the user: Coordinates of points..
2. Define elements of the coefficients matrix and right side constants.

3. Define n = no. of equations.
4. **Forward Elimination**
   for k = 0 to n-2
   for i = k+1 to n-1
   factor = $a_{i, k} / a_{k, k}$
   for j= k+1 to n-1
   ai, j = $a_{i, j}$ *- factor** $a_{k, j}$
   end
   $c_i = c_i - factor * c_k$
   end
   end
   Backward Substitution
   $x_{n-1} = b_{n-1} / a_{n-1, n-1}$
   for i = n-2 to 0
   sum = b_i
   for j = i+1 to n-1
   sum = sum – $a_{i,j} * x_j$
   End
   $x_i = sum / a_{i,i}$
5. Display values of coefficients.