**CSE218-Numerical Methods**

1905072-Mahir Labib Dihan

1. **SOLVING NONLINEAR EQUATIONS**
   1. **Bisection:**
      1. **Steps:**
         1. Choose and as two guesses for the root such that [It’s better to chooses two x values on the same side where the root is]
         2. Estimate the root, of the equation as the mid-point between and as,
         3. Now check the following

If , then the root lies between and ; then ; .

If , then the root lies between and ; then ; .

If , then the root is ; Stop the algorithm.

* + - 1. Find the new estimate of the root, . Find the absolute relative approximate error,
      2. Compare the absolute relative approximate error with the pre-specified relative error tolerance . Also, check if the number of iterations has exceeded the maximum number of iterations allowed. If so, one needs to terminate the algorithm and notify the user.
    1. **Pros:**
       1. Always convergent
       2. The root bracket gets halved with each iteration - guaranteed.
    2. **Cons:**
       1. This method will work only when f(x) changes sign. If a function f(x) is such that it just touches the x-axis it will be unable to find the lower and upper guesses.
          1. If f(x) changes sign there will be odd (1, 3, 5..) number of roots
          2. If f(x) doesn’t change sign there will 0/1/2/4/6... roots
       2. Slow convergence
       3. If one of the initial guesses is close to the root, the convergence is slower
       4. Have to guess two points
       5. When function changes sign but root doesn’t exist
  1. **Newton-Raphson:**
     1. **Steps:** 
        1. Evaluate symbolically
        2. Use an initial guess of the root, , to estimate the new value of the root, [It’s better to choose the point on the side of x axis where the root is]
        3. Find the absolute relative approximate error,
        4. Compare the absolute relative approximate error with the pre-specified relative error tolerance . Also, check if the number of iterations has exceeded the maximum number of iterations allowed. If so, one needs to terminate the algorithm and notify the user.
     2. **Pros:**
        1. Requires only one guess
        2. Converges fast
     3. **Cons:** 
        1. Division by zero: if we guess the point, where the slope is 0 (Root can’t be found)
        2. Divergence at inflection points
        3. Oscillations near local maximum and minimum (Program won’t stop until max iteration limit exceeded)
        4. Root Jumping

1. **SOLVING LINEAR EQUATIONS**
   1. **Naïve Gaussian Elimination:**
      1. **Steps:**
         1. Forward Elimination
            1. Transform coefficient matrix into upper triangular matrix
            2. (n-1) steps of forward elimination
         2. Back Substitution
            1. Solve each equation starting from the last equation
      2. **Cons:**
         1. Division by zero
         2. Large round off error
   2. **Gaussian Elimination with Partial Pivoting:**
      1. **Steps:**
         1. Forward Elimination
            1. Transform coefficient matrix into upper triangular matrix
            2. (n-1) steps of forward elimination
            3. At the beginning of the step of forward elimination, swap row with the row which have maximum absolute value at column. (This will avoid division by zero, as long as the maximum absolute value at column is not zero itself.)
         2. Back Substitution
            1. Solve each equation starting from the last equation
2. **INTERPOLATION**

Note: Higher order polynomial doesn’t guarantee more accurate result.

* 1. **Newton’s Divided Difference Polynomial Method**
     1. **Steps**
        1. x=The point that needs to be interpolate [Must be in the range. If not in the range, then ExtrapolationRegression]
     2. **Pros:**
        1. Just a new term is added with the change in degree
     3. **Cons:**
        1. It’s hard to find the constants
  2. **Lagrange**
     1. **Steps**
        1. x=The point that needs to interpolate [Must be in the range. If not in the range, then ExtrapolationRegression]
     2. **Pros:**
        1. Coefficient can be found easily
     3. **Cons:**
        1. All the terms changes with the change in degree

1. **Integration**

Note: More subsegments, less error.

* 1. **Trapezoidal Rule**
     1. **Steps**
     2. **True error**
     3. **Pros**
        1. Can work with both odd and even number of sub segments
     4. **Cons**
        1. Converges slower
  2. **Simpsons 1/3rd rule**
     1. **Steps**
     2. **True error**
     3. **Pros**
        1. Converges faster
     4. **Cons**
        1. Can’t work with odd number of sub segments

1. **Regression**

Note: If specific model is not given, we have to guess the model from the graph of given data.

* 1. **Linear**
     1. **Steps**
  2. **Non-Linear**
     1. **Exponential**
        1. **Steps**
           1. [Solve this using numerical methods for solving nonlinear equation like **Bisection**]
        2. **Converting to Linear**
     2. **Polynomial**
        1. **Steps**
           1. Find using numerical method of solving linear equation like **Gaussian Elimination.**
     3. **Saturation Growth Model**
        1. **Steps**
        2. **Convert to Linear**
     4. **Power Model**
        1. **Steps**
        2. **Convert to Linear**