**CSCI 2150L**

**Session 13 Topics:**

**10/27/2014**

**Fall 2014**

1. The secant method, another root finding method was discussed. Previously, the newton Raphson method for finding roots approximated the root at st iteration as follows:

Though this method doesn’t need to have two initial points as the previous methods (bisection and method of false position) needed Newton Raphson depends on two functions i.e. functions *f* and *f’* to evaluate the approximate root at any stage. The secant method replaces the derivative in the equation above by the following expression that depends only on function *f* evaluated at two different points and.

This is a close approximation of the derivative as suggested by the use of sign. This approximation is more formally referred to as finite difference approximation to the first derivative. Now replacing the expression for in the original equation found in Newton Raphson method we get a new equation for approximate root as follows.

The new root found in st iteration is shown graphically as follows. The root approximated is the point where the secant line connecting points and on the curve intersects x-axis.



Exact root

Fig. 1: Secant method. [Courtesy: Numerical Mathematics and Computing, Kincaid and Cheney, Sixth edition.]

[N.B. Secant method also has to be provided with two initial points to start iteration. However, unlike bisection and method of false position these initial points don’t have to satisfy any condition i.e. they don’t have to lie on the either side of the exact root of the function.]

1. From root finding techniques we moved to interpolation techniques. The problems belonging to this class of problems deal with correctly estimating the value of a function at a given point. Let’s suppose we are given the following table which holds the values of a function evaluated at points and the respective values of the function are

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  | … |  |
|  |  |  |  | … |  |

Now if we don’t know the exact form of the function it’ll be non-trivial to evaluate the approximate value of the function at an arbitrary point which is not included in the table above. Therefore the problem of interpolation is to estimate an approximate form of the unknown function based on the values given by the function at discreet points of the table above.

We discussed in class a technique that interpolates by creating a Lagrange polynomial form for the function. This method first create a cardinal polynomial indicated as for each of the input points (see the table above). The form of is as follows.

Notice that the term doesn’t appear in the numerator whereas all the input points except are being subtracted from. In this way the term doesn’t appear in the denominator. Formally can be represented as follows.

Thus each cardinal polynomial that excludes from numerator should have n roots. It is depicted in the following picture where few Lagrange polynomials created from five points are shown. Each of these polynomials created from five initial points (i.e. the size of the initial table was five containing five points). Therefore each one of them should have four roots (one less than the size of the table). In the figure each cardinal polynomial covers four of the five points i.e. for each the point is not covered. Please verify this fact from the figure below. The covered points for each cardinal polynomial are the roots.



Fig. 2: Lagrange Polynomials [Courtesy: Numerical Mathematics and Computing, Kincaid and Cheney, Sixth edition.]

Subsequently the Lagrange form of the interpolation polynomial is represented as

Notice that the values are available from the original table. As the table contained *n+*1 points the polynomial would have a degree at most *n*. This is indicated by the subscript of. We can thus evaluate the approximate value given by the unknown function at an arbitrary point *a* by evaluating at