An Introduction to Programming though C++

Abhiram G. Ranade

Lecture Sequence 3.3

Ch. 8: Computing Common Mathematical Functions

Goal: Learn methods for performing common mathematical operations

- Evaluating common mathematical functions such as
  - sin(x), log(x), ...
- Integrating functions numerically, i.e. when you do not know the closed form.
  - Intrinsically useful
  - Also useful for evaluating certain functions
- Finding roots of functions, i.e. find x s.t. f(x) = 0.
  - Intrinsically useful
  - Also useful for evaluating certain functions
- All the methods give approximate answers.
  - Exact answers cannot be produced using just finitely many arithmetic operations.
  - Error can be made as small as we want by doing more and more work.
- The programs will be simple, using just a single loop.

Outline

- Taylor Series
- Numerical Integration
- Bisection Method
- Newton-Raphson Method

Taylor series example

How do we calculate sin(x), cos(x), ln(x), ...?

- Can be expressed as sum of infinite terms

Example:

$\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + \ldots$

- Each term is easy to evaluate
- By taking the first few terms we can get fairly good answer
- Need more accuracy? Take more terms.

Additional property: Error using first k terms $\leq$ |k+1th term|.

Let us compute $\sin(x) = x - x^3/3! + x^5/5! - \ldots$

- In k^{th} iteration we will add k^{th} term to a running sum, initialized to 0.
- Let t_k = k^{th} term of the series, k=1, 2, 3…
- $t_1 = +x$, $t_2 = - x^3/3!$, $t_3 = +x^5/5!$, ...
- $t_k = (-1)^{k+1}x^{2k-1}/(2k-1)!$
- $t_1 = x$ is given as input.
- At the beginning of the k^{th} iteration: we will ensure we have t_k in variable T.
- At end of iteration k, we must have S = sum of first k terms, T = $k+1^{th}$ term
- $t_{k+1} = (-1)^{k+2}x^{2k+1}/(2k+1)!$
- $t_{k+1} = t_k (-x^2)/((2k)(2k+1)$
- In iteration k: add T to S. Then multiply T by $(-x^2)/((2k)(2k+1))$.

Program

```
main_program{
  double x; cin >> x;
  cout <<"Calculating sin(x) for x = "<<x<<endl;

  double s=0, t=x;
  for(int k=1; k<=10; k++){
    s = s + t;
    t = - t * x * x/(2*k)/(2*k+1);
    cout <<"No of terms: "<< k <<"  Value: "<< s
         <<"  Error estimate: "<< abs(t) <<endl;
  }
  cout <<endl<<" From library function: "<<sin(x)<<endl;
}
```

Taylor's theorem

Taylor's theorem: Many interesting functions can be written as:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + f''(x_0)(x - x_0)^2 / 2! + f'''(x_0)(x - x_0)^3 / 3! + \ldots$$

if x is close to x_0, i.e. $|x - x_0|$ < "radius of convergence".

We can often choose x_0 such that it is possible to easily evaluate f and its derivatives at x_0

Example: $f(x) = \sin(x)$: Choose $x_0 = 0$.

$f(0) = \sin(0) = 0$, $f'(0) = \cos(0) = 1$,

$f''(0) = -\sin(0) = 0$, $f'''(0) = -\cos(0) = -1$

...

So $\sin(x) = 0 + x + 0 - x^3/3! + 0 \ldots$

Radius of convergence happens to be infinite for sin(x) at $x_0 = 0$.

What we discussed

- Taylor series are available for many many functions.
  - Exercise: construct the series and write the program to calculate cos(x).
- Other infinite expressions have also been used for calculating math functions, e.g. infinite products, continued fractions.
  - See exercises in book.