Chapter 1 : Review of Calculus, Errors and Rounding

Review of Calculus

Continuity

A function f(x) is continuous on [a,b] if f(x) ∈ C[a,b] and f(x) ∈ C`[a,b], that is, f(x) and its derivative are continous on [a,b].

In laymans terms, a function is continous over [a,b] if there are no points where there are problems, such as division by zero, or the function is not a piecewise function that does not cover every point in [a,b]

Increasing and Decreasing Functions

A function is classified as increasing or decreasing depending on its derivative.

- if f`(x) > 0 then the function is increasing
- if f`(x) < 0 then the function is decreasing

if f`(x) = 0, then that is called an inflection or critical point.

if f(x) is continuous on [a,b] then f has an absolute **upper bound** = M and an absolute **lower bound** = m, such that :

$$m \leq f(x) \leq M \forall x \in [a, b]$$

Initial Value Theorem

Lets say that f ∈ [a,b], L is between f(a),f(b) ∃ c ∈ [a,b] such that f(c) = L

Bolzano Theorem

let us assume that :

1. f ∈ C[a,b]
2. f(a).f(b) < 0

Then ∃ c in (a,b) such that f(c) = 0.

x = c is a root(a zero) for f(x), and x = c is a solution for the equasion.

Mean Value Theorem

if f ∈ C[a,b] and f`∈ C(a,b), then ∃ c ∈ (a,b) such that :

$$f‘(c) = \frac{f(b) - f(a)}{b - a}$$

alternatively, this can be expressed as

$$|f(b) - f(a)| = |f`(c)||(b - a)|$$

Recall that :

$$|x \pm y| \leq |x| + |y|$$

Rolle's Theorem

Rolle's Theorem is a special case of the MVT, which states :

> If f is continuos and differentiable on (a,b) and if f(a) ≤ f(b) , then ∃ c ∈ (a,b) such that f`(c) = 0 .

In simple terms, if f is generally decreasing on an open interval, then there is a critical point in that interval.

Taylor Series

The Taylor Series/ Taylor Expansion is a way to express a function that isn't very "nice" as a series of polynomials that can be used to express the function. In more technical terms, the taylor series can be formally defined as follows :

> Given f(x) and a center x = a, then the taylor expantion series for f(x) about x = a is given by :

$$f(x) = f(a) + f`(a)(x - a) + \frac{f``(a)(x - a)}{2!} + ... + \frac{f^n(a)(x - a)^n}{n!}$$

Which is used to approximate f(x) when x is close to a.

The error of this approximation is given by :

$$E(x) = \frac{f^{n+1}(c)(x - a)^{n+1}}{(n + 1)!}$$

where c is a number between x and a.

Errors and Rounding

Significant Digits

A digit is said to be significant when a change in its value changes the number. To find out the number of significant digits in a number, start counting from the the first non-zero number on the left. For example,

```
0.0012341
```

has 5 significant digits, and

```
1200
```

has 4 significant digits.

Sources of Errors

There are 2 sources of errors :

1. Round-off Error : an error when a number is approximated by another number.
2. Truncation Error : an error when a function is approximated by another function.

The reason we have round-off error is because computers use **finite digit arithematic.** There are only a finite number of digits a computer can handle at a time. Numbers in computers are represented in **floating point representation.** Normalized floating point representation is a special form of this, where all the **significant digits** are to the right of the decimal point.

Floating Point Representation

FPR is a way of representing numbers such that we reduce the number of uneeded digits/zeros. Basically, lets say we have the number

> 0.0023141

then we can represent this as

> $2.3141 * 10^{-3}$

It is called floating point because we are moving the decimal point around. Normalized floating point is simply rewriting the number as

> $0.\text{dddddddd} * 10^{n}$

where the first digit is non-zero.

Round-off

There are 2 types of round-off :

1. Chopping --> simple cut off after a certain amount of significant digits.

2. Rounding --> estimating using the last significant digit, such that if it is ≥ 5, then we add one, otherwise, we add zero.

Propagation of Error

given that

$$p = \hat{p} + E_p$$

and

$$q = \hat{q} + E_q$$

then

$$p + q = \hat{p} + \hat{q} + E_p + E_q$$

That is to say, error may be increased in operations.

Order Of Approximation

Let us say that we are approximating e^h using the taylor expantion :

$$e^h = 1 + h + \frac{h^2}{2}$$

Then we know that the error is given by

$$E = \frac{f^3(c)h^3}{6} \rightarrow \frac{e^c h^3}{6}$$

so we can say

$$E \approx ch^3$$

we can express this error by saying :

$$O(h^3)$$

That is to say, the error is about some constant between `0` and h times h^3.

Formally, we say :

$$f(h) = p(h) + O(h^n)$$

then

$$f(h) \approx p(h)$$

with

$$E \approx C * h^n$$

Determening The Order Of Approximation In Operations.

Say that we have

$$f(h) = p(h) + O(h^n)$$

$$g(h) = k(h) + O(h^m)$$

Then

$$f(h) + g(h) = p(h) + k(h) + O(h^r)$$

where

1. r = min [n,m]
2. $h^l + O(h^r) = O(h^r)$, if l ≥ r

Chapter 2 : How To Solve An Equation Numerically

There are six(actually five, but one of them is a special case of another) main methods we will be using to solve equations numerically :

1. Bisection Method
2. False-Position(Regula-Falsi) Method
3. Fixed Point Iteration
4. Newtons(Newton-Raphson) Method
5. The Seacant Method
6. Accelarated Newton Method

Bisection Method

Recall

> if f ∈ c[a,b] and f(a).f(b) < 0, ∃ at least one r ∈ c(a,b) such that
> f(r) = 0.

By this principle, we know that if the sign of f changes over the domain, then we know that there is a root. We can find this root if we keep halving the interval, until we eventually zero in on the root.

$$f \in [a, b]; [a, b] = [a_0, b_0]$$

$$f(a_0) * f(b_0) < 0$$

Then the first iteration is $C_0 = \frac{a_0+b_0}{2} \to f(C_0)$

if $f(C_0) < 0 \to [a_1, b_1] = [a_0, C_0]$

and the second iteration is $C_1 = \frac{a_1, b_1}{2}$

The general formula for calculating the nth iteration using the bisection method is

$$C_n = \frac{a_n, b_n}{2}$$

And the **upper bound of error** for the bisection method is given by :

$$\frac{b-a}{2^{n+1}}$$

There are 4 ways we stop when using the bisection method :

1. We reach the desired number of iterations.

2. We have reached a certain **accuracy**.

   The accuracy in the bisection method is given by :

   $$|C_n - C_{n_1}| < \epsilon$$

   That is to say, the accuracy is the difference between the last 2 successive iterations.

3. Stop when $f(C_n) < \epsilon$.

4. Stop when $\frac{|C_n - C_{n-1}|}{C_n} \leq \epsilon$.

The main advantage of the Bisection method is that C always converges, and it always converges to the true root. The main disadvantage is that it is really slow, ie, it takes a lot of time and iterations to get within a reasonable degree of error of the root.

Bisection Method Proof

Prove that $\lim_{n \to \infty} C_n = r$

$b_1 - a_1 = \frac{b-a}{2}$ --> their length is equal when we bisect

$b_2 - a_2 = \frac{b_1 - a_1}{2} = \frac{b-a}{2^2}$

$b_3 - a_3 = \frac{b_2 - a_2}{2} = \frac{b-a}{2^3}$

Then we can say that

$b_n - a_n = \frac{b-a}{2^n}$

and

$|r - C_n| < \frac{b_n - a_n}{2}$ --> the difference between the real root and the iteration is less than the length.

and

$0 < |r - C_n| < \frac{b-a}{2^{n+1}}$ --> the diffrence is less than the upper bound of error and bigger than `0`.

then by the sandwich theorem, since $\lim_{n \to \infty} \frac{b-a}{2^{n+1}} = 0$

and $\lim_{n \to \infty} = 0$,

then $\lim_{n \to \infty} r - C_n = 0$,

therefore, $r = \lim_{n \to \infty} C_n$.

Calculating The Number Of Iterations

since we know that the upper bound for the error is given by :

$$\frac{b-a}{2^{n+1}}$$

and that the upper bound of the error is higher than the real error, then we can say $\frac{b-a}{2^{n+1}} < \frac{\epsilon}{1}$.

$\frac{2^{n+1}}{b-a} > \frac{1}{\epsilon} = 2^{n+1} > \frac{b-a}{\epsilon}$.

take ln of both sides $ln(2^{n+1}) > ln(\frac{b-a}{\epsilon})$

--> $(n+1)ln(2) > ln(\frac{b-a}{\epsilon})$ --> $n + 1 > \frac{ln(\frac{b-a}{\epsilon})}{\ln(2)}$ --> $n > \frac{ln(\frac{b-a}{\epsilon})}{\ln(2)} - 1$

Generally then, to find the **minimum** number of iterations for the bisection method to reach a certain accuracy, we use the inequality

$$n > \frac{ln(\frac{b-a}{\epsilon})}{\ln(2)} - 1$$

or

$$n > \log_2(\frac{b-a}{\epsilon}) - 1$$

Converting To f(x) = 0

Eg Estimate $\sqrt[4]{7}$ in the range `[1,2]`

let $x = \sqrt[4]{7}$

$x^4 = 7$

$x^4 - 7 = 0$

$f(x) = x^4 - 7$

Always convert to the form f(x) **= 0** when attempting to solve numerical methods.

Eg Find the intersection point between $y = e^x, y = x^2 + 5$

let $y = y$

$e^x = x^2 + 5$

$e^x - x^2 - 5 = 0$

$f(x) = e^x - x^2 - 5$

Eg $p(x) = e^x + x^2 - \sqrt{x} + 1$ is a profit function. Estimate the number of units that leads to maximum profit in [a,b]

$p'(x) = e^x + 2x - \frac{1}{(2)(\sqrt{x})}$

and find $p'(x) = 0$

False Position Method

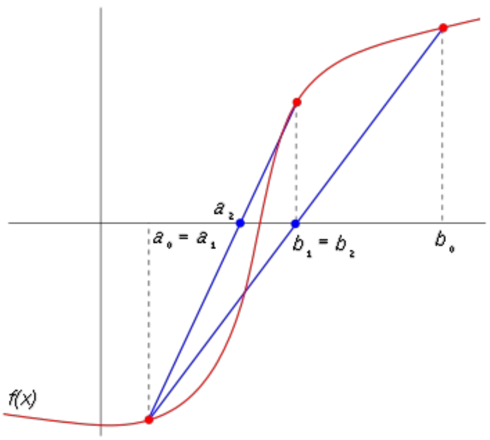
It is similar to bisection, and needs

$f(x) = 0, [a, b]$

where $f(a) * f(b) < 0$

But the value of C_n is not of $f(\frac{b_n - a_n}{2})$

It relies on a geometric method using the **seacant line** between a_n, b_n



Mathemtically, this is expressed as :

$$C_n = b_n - \frac{f(b_n)(b_n - a_n)}{f(b_n) - f(a_n)}$$

then $C_0 = b_0 - \frac{f(b_0)(b_0 - a_0)}{f(b_0) - f(a_0)}, f(C_0)$

and $C_1 = b_1 - \frac{f(b_1)(b_1 - a_1)}{f(b_1) - f(a_1)}, f(C_1)$

so on and so forth.

The slope of the resultant line is given by

$$S = \frac{f(b_0) - f(a_0)}{b_0 - a_0}$$

then $S = \frac{f(b_0)-f(a_0)}{b_0-a_0} = \frac{0-f(b_0)}{C_0-b_0}$

$\frac{C_0-b_0}{-f(b_0)} = \frac{b_0-a_0}{f(b_0)-f(a_0)}$

$C_0 - b_0 = \frac{f(b_0)(b_0-a_0)}{f(b_0)-f(a_0)}$

$C_0 = b_0 - \frac{f(b_0)(b_0-a_0)}{f(b_0)-f(a_0)}$

We determine the next [a,b] depending on the sign of C_n

if $C_n < 0$, then $[a_{n+1}, b_{n+1}] = [C_0, b_n]$, given that $f(a) < 0$and $f(b) > 0$

This method always converges to the true root, but it is very slow as well.

Fixed Point Iteration

Say we have a function $f(x)$ with root p. The general idea is that we have some function taken from $f(x)$, called $g(x)$. We used the **fixed point** of $g(x)$, which will be the roots of $f(x)$. Not all functions derived are suitable however.

What is a Fixed Point?

$x = p$ is a fixed point of $g(x)$ if $g(p) = p$

For example, given the function $g(x) = x^2$, then $0, 1$ are fixed points of $g(x)$. For more complex functions, we solve $g(x) = x$, since we want all images of x that are equal to x. so in this case :

$x^2 = x$

$x(1-x) = 0$

$x = 0, 1$

Geomertrically speaking, the fixed points of $g(x)$ are the intersection points between $g(x)$ and $y = x$, for example,for $g(x) = 4x - x^2$:

In this case, $0, 3$ are fixed points of $4x - x^2$

Where do we get g(x) from?

We need to bring $f(x) = 0$ to the form $x - g(x) = 0$, for example $f(x) = x^2 - 5x + 6 = 0$ becomes $x^2 = 5x - 6 \rightarrow x = \sqrt{5x-6} \rightarrow x - \sqrt{5x-6}$, so in this case $g(x) = \sqrt{5x-6}$. We can extract more than one $g(x)$ from $f(x)$, for example, we can extract $x = \frac{x^2+6}{5}$ by reearanging as well.

We can do this reerangement because we assumed that $f(x) = 0$

Why are Fixed Points of g(x) Roots of f(x)?

let $g(p) = p$

then $p - g(p) = 0$

but $p - g(p) = f(p)$

and since $p - g(p) = 0$, then $p - g(p) = f(p) = 0$ and p is a root of $f(x)$

FPI is a method where we find approximations of the fixed points of $g(x)$. we need :

1. $g(x)$.
2. p_0, or the initial value/guess value.

The formula of FPI is given by :

$$p_{n+1} = g(p_n)$$

then, $p_1 = g(p_0)$, $p_2 = g(p_1)$, so on and so forth.

If the sequence $p_0, p_1, p_2, \ldots p_n$ convrges to some p, then p is a fixed point of $g(x)$, and a root of $f(x)$

This can be proven as follows :

$\lim_{n\to\infty} p_n = g(p_{n-1})$

$p_\infty = g(p_{\infty-1}) = g(p_\infty)$

Existance, Uniqueness, and Convergance of Fixed Points

Given a function $g(x)$ and an interval $[a, b]$, then if

1. $g(x)$ continious on $[a, b]$
2. $a \leq g(x) \leq b \forall x \in [a, b]$

   then $g(x)$ has **at least one** fixed point in $[a, b]$, that is $\exists p \in [a, b]$ where $g(p) = p$

   furethermore, if

3. $|g'(x)| \leq k \leq 1 \forall x \in [a, b]$, then

   1. The fixed point is unique
   2. The FPI of $g(x)$ will converge to p for any $p_0 \in [a, b]$, where k is the maximum of $|g'(x)|$.

Proof

If 1 and 2 are satisifed, then we need to to show that g has a fixed point in $[a, b]$

Let us assume 3 cases :

1. if $g(a) = a$, then we are done.
2. if $g(b) = b$, then we are done.
3. if $g(a) \neq a$ and $g(b) \neq b$:

   We know that $a \leq g(x) \leq b$, and since we know that $g(a) \neq a$ and $g(b) \neq b$, then we know that $g(a) > a$ and $g(b) < b$, then we apply bolazano on $h(x) = g(x) - x$.

   We know that h is continous and also $h(a) = g(a) - a > 0$ and $h(b) = g(b) - b < 0$, Then $\exists p \in [a, b]$ such that $h(p) = 0$. Since $h(x) = g(x) - x$, and $h(x) = 0$, then $g(x) - x = 0$ so $g(x) = x$ exists.

Secondly, we need to prove uniquness and convergence

1. Uniquness

   From the previous proof, we have proved that p exists. Let us assume that there are other fixed points for g in $[a, b]$, called q, then

   $g(p) = p$

   $g(q) = q$

   lets apply the mean value theorem on $[p, q] \leq [a, b]$. We know that g is continous on $[p, q]$ because it is continous on $[a, b]$, Therefore, $\exists c \in (p, q)$ such that $g'(c) = \frac{g(q)-g(p)}{q-p} = 1$, therefore $|g'(c)| = 1$ which is a contradiction, since $max(g'(x)) < 1$, so no other point exists.

2. Convergence

Apply MVT on $[p_0, p]$ therefore $\exists C \in (p_0, p)$ such that

$g'(c) = \frac{g(p)-g(p_0)}{p-p_0}$

$g'(c) = \frac{p-p_1)}{p-p_0}$

$|p - p_1| = |g'(c)||p - p_0|$

$|p - p_1| \leq k. |p - p_0|$

This means that p_1 is closer to p that p_0.

Apply the MVT on $[p_1, p]$ --> $|p - p_2| \leq k. |p - p_1|$ in a similar way.

So, $|p - p_2| \leq k^2. |p - p_0|$

$|p - p_3| \leq k^3. |p - p_0|$

Generally,

$$|p - p_n| \leq k^n. |p - p_0|$$

Which is also an upper bound for the error.

Error in Fixed Point Iteration

Error in FPI is given by the difference between each **2** successive iterations. So we express this as

$$\epsilon \leq |p_{n+1} - pn|$$

Another way to express error is

$$\frac{k^n |p_1 - p_0|}{1 - k} < \epsilon$$

So we can theoretically calculate the number of iterations using

$$n > ln(\frac{\epsilon(1 - k)}{p_1 - p_0})$$

Convergence To Fixed Points

Let p be the fixed point of $g(x)$

Can i prove that the FPI of g will go to p before solving**?**

1. if $|g'(p)| < 1$, then the FPI of $g(p)$ will converge to p for any p_0 close to p. We call this an **attractive fixed point.**
2. if $|g'(p)| > 1$, then the FPI of $g(p)$ will not converge to p. We call this a **repulsive fixed point.**
3. if $|g'(p)| = 1$, then we cannot guarantee any outcome.

It is not nesecarry for p_0 approaches p from one side. It could be approached from both sides, which is called **oscilating convergence.** If it is however, only approached from one side, then that is called **monotonic convergence.**

In fact, if $-1 < g'(p) < 1$, Then, if $-1 < g'(p)$, then it is oscilating, if $g'(p) > 1$, then it is monotonic. Otherwise, we can make no guarantees.

Newton-Raphson(or Newtons) Method

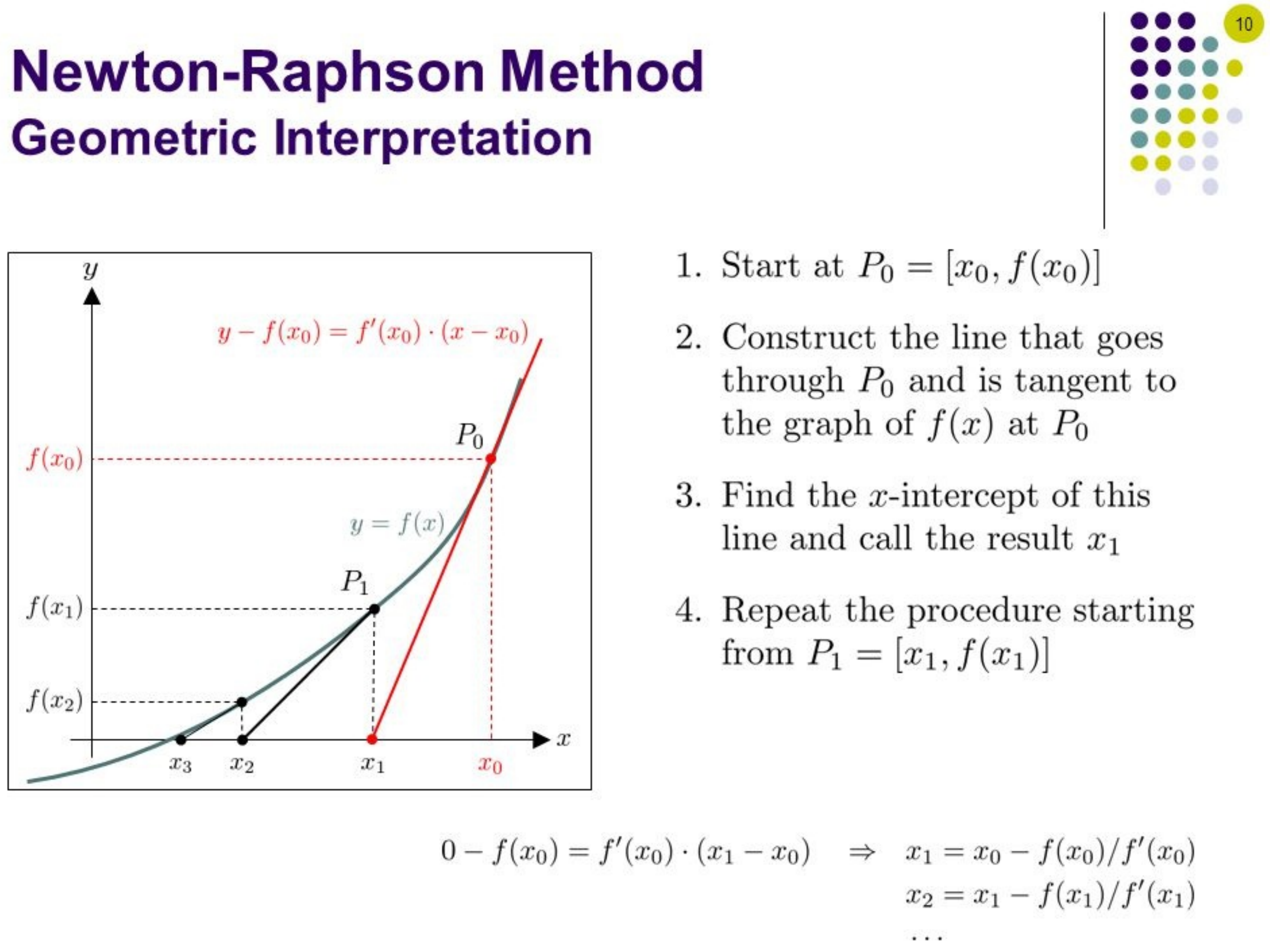
Say we have $f(x) = 0$, and some P_0

Then we can say

$$P_{n+1} = P_n - \frac{f(P_n)}{f'(P_n)}$$

Newtons-Raphson is a special case/example of Fixed Point Iteration. Threfore, any theorem that applies to FPI applies to the Newton-Raphson method(including error, convergence, and attractiveness)

Geometrically, We can express it as follows

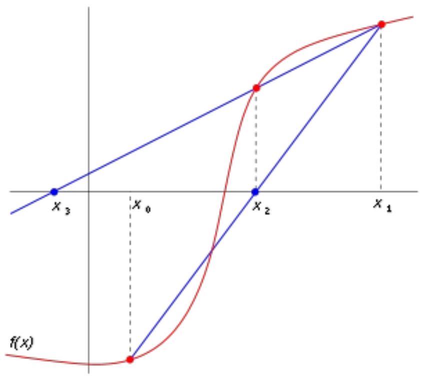


Secant Method

Say we have $f(x) = 0$, some P_0, and some P_1, then

$$P_{n+1} = P_n - \frac{f(P_n)(P_n - P_{n-1})}{f(P_n) - f(P_{n-1})}$$

Geometrically, this can be expressed as



Where P_{n+1} is the point where the secant line of P_n, P_{n-1} intersects the x-axis.

Measuring Speed (Order Of Convergence)

The order of convergence R, is a measure of how fast a method converges to the root. It is a positive number, the higher it is, the faster the method is, that is, the error between subsequent interations decreases faster.

The value of R depends on the type of root.

Multiplicty of Roots

Multiplicity is the number of times a root is repeated. That is, if we have a function that has the roots $(1, -2, 1)$, then 1 has a multiplicity of 2, and 2 has a multiplicity of 1.

Formally, let p be a root of $f(x)$. if

$$f(p) = f'(p) = f"(p) \ldots f^{(M-1)}(p) = 0$$

but

$$f^{(M)}(p) \neq 0$$

, then we say that the root p has multiplicity M. The smallest value of M is 0. A root with $M = 1$ is called a **simple root.** if $M > 1$, then the root is called a **multiple root.** if $M = 2$, it is called a **double root**, so on, and so forth.

Another defenition, that not always works but is useful sometimes, is that let p be a root of $f(x)$. This root has multiplicity M if we can write

$$f(x) = (x - p)^M . h(x); h(p) \neq 0$$

where $h(x)$ is some arbitriary function.

The secant and newton methods are fast for roots with $M = 1$

Given a sequece of iterations $[P_n]_{n=0}^{\infty}$ that converges to p. And $|E_n| = |p - p_n|$. If the convergence is fast, then E decreases faster.

Now, if $\exists$ two positve real numbers A, R such that

$$\lim_{n \to \infty} \frac{|E_{n+1}|}{|E_n|^R} = A$$

Then we say that the sequence converges to p with **order of convergence** R and A is called the **asymptotic error constant.** Usually, $A < 1$.

The above limit means that when n is large, then the value of $\frac{|E_{n+1}|}{|E_n|^R} \approx A$. Rearranging, we get

$$|E_{n+1}| \approx A|E_n|^R$$

That is, when R increases, then it converges faster (error decreases faster).

Note that :

1. If $R = 1$, then the convergence is called linear.
2. If $R > 1$, then the convergence is called quadratic, cubic, etc.
3. if $1 < R < 2$, then the convergence is superlinear.

Secant Method Convergence

Remember that

$$P_{n+1} = P_n - \frac{f(P_n)(P_n - P_{n-1})}{f(P_n) - f(P_{n-1})}$$

If it converges to p, then we have 2 cases :

1. If p is a simple root, then $R = 1.618$, and $A = |\frac{f"(p)}{2f'(p)}|^{0.618}$
2. If p is a multiple root, then $R = 1$, and we cant find A theoretically, only numerically. This is usually done by finding the real root, and finding $|E_n| = |p - p_n|$, and using $A = \frac{|E_{n+1}|}{|E_n|^1}$

Bisection Method Convergence

$R = 1$ in all cases, and $A = 0.5$.

False Positon Method Convergence

$R = 1$ in all cases, and A has no theoretical value.

Fixed Point Iteration Convergence

Let P be a fixed point of $g(x)$. If $g'(p) = g''(p) = ... = g^{(k-1)}(p) = 0$, but $g^{(k)} \neq 0$ then the fixed point iteration will converge to p with $R = k$, $A = \frac{g^{(k)}(p)}{k!}$

Proof

Based on what is given , we need to show that

$$lim_{n\to\infty} \frac{E_{n+1}}{E_n^k} = |\frac{g^{(k)}(p)}{k!}|$$

Now, take taylor expansion of $g(x)$ about P, which is

$$g(x) = g(p) + g'(p)(x-p) + \frac{g''(p)(x-p)^2}{2!}....\frac{g^{(k)}(p)(x-p)^k}{k!}$$

We know however, that $g'(p) = g''(p) = ... = g^{(k-1)}(p) = 0$, so we end up with

$$g(x) = p + \frac{g^{(k)}(c)(x-p)^k}{k!}$$

let $x = P_n$

$$g(x) = p + \frac{g^{(c)}(p)(P_n - p)^k}{k!}$$

and so

$$|P_{n+1} - p| = \frac{|g^{(c)}(p)||(P_n - p)^k|}{k!}$$

which results in

$$|E_{n+1}| = \frac{|g^{(c)}(p)|}{k!}|E_n|^k$$

Therefore,

$$\frac{|E_{n+1}|}{|E_n|^k} = \frac{|g^{(c)}(p)|}{k!}$$

Where $P_n < c < P$

Newton's Method Convergence

Remember that

$$P_{n+1} = P_n - \frac{f(P_n)}{f'(P_n)}$$

If it converges to p, then we have **2** cases :

1. If p is a simple root($M = 1$), then $R = 2$ and $A = |\frac{f"(p)}{2f'(p)}|$. So,

$$\lim_{n\to\infty} \frac{|E_{n+1}|}{|E_n|^R} = |\frac{f"(p)}{2f'(p)}|$$

1. If p is a multiple root($M > 1$), then $R = 1$ and $A = \frac{M-1}{M}$

Proof

We need to prove that

$$lim_{n\to\infty} \frac{E_{n+1}}{E_n^k} = |\frac{f''(p)}{2f'(p)}|$$

Method 1

We can consider newton as a special case of FPI where $g(x) = x - \frac{f(x)}{f'(x)}$

Now, since it is a special case of FPI, use the proof of FPI.

$$g'(x) = 1 - \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2}$$

Simplifying, this results in

$$g'(x) = \frac{f(x) * f''(x)}{(f'(x))^2} = 0$$

Deriving again

$$g''(x) = \frac{(f'(x))^2 f(x)f'''(x) - f''(x)f'(x))}{f'(x))^4} THISEQUATIONMIGHTNOTBECOMPLETE$$

Substituting $x = p$, we get $g''(p) = \frac{f''(p)}{f'(p)}$

so by FPI theorems, $R = 2$, A=$|\frac{g''(p)}{2!}| = |\frac{f"(p)}{2f'(p)}|$, Which is what we want to demonstrate.

Method 2

If p is a simple root of $f(x)$ we want to prove that

$$\lim_{n\to\infty} \frac{|E_{n+1}|}{|E_n|^R} = |\frac{f"(p)}{2f'(p)}|$$

Apply taylor series to $f(x)$ about $x = p_n$ $f(x) = f(p_n) + f'(p_n)(x - p_n) + f''(c)\frac{(x-p_n)^2}{2!}$

let $x = p$

$f(p) = f(p_n) + f'(p_n)(p - p_n) + f''(c)\frac{(p-p_n)^2}{2!}$

$0 = f(p_n) + f'(p_n)(p - p_n) + f''(c)\frac{(p-p_n)^2}{2!}$

$0 = \frac{f(p_n)}{f'(p_n)} + p - p_n + \frac{f''(c)}{f'(p_n)}(p - p_n)^2$

$0 = p - (p_n - \frac{f(p_n)}{f'(p_n)}) + \frac{f''(c)}{2f'(p_n)}(p - p_n)^2$

$|p - p_{n+1}| = |\frac{f''(c)}{2f'(p_n)}|(p - p_n)^2$

$\frac{|E_{n+1}|}{|E_n|^2} = |\frac{f''(c)}{2f'(p_n)}|$

Accelarated Newton Method

For multiple roots, newton's method is slow. Given that $M > 1$,then we can change $R = 1$ to $R = 2$ by using the formula

$$P_{n+1} = P_n - \frac{Mf(P_n)}{f'(P_n)}$$

Summary Table

Method Name	Requirments	Iteration	Convergence
Newton	$p_0, f(p), f'(p)$	$p_{n+1} = p_n - \frac{f(p_n)}{f'(p_n)}$	$if \quad M = 1, R = 2, A = \frac{f''(p)}{2f'(p)}$,$if \quad M > 1, R = 1, A = \frac{M-1}{M}$
Seacant	$p_0, p_1, f(p)$	$p_{n+1} = p_n - \frac{f(p_n)(p_n - p_{n-1})}{f(p) - f(p_{n-1})}$	$if \quad M = 1, R = 1.618, A = \|\frac{f''(p)}{2f'(p)}\|^{0.618}$, $if \quad M > 1, R = 1,$ calculate p numerically and use $\frac{E_{n+1}}{E_n}$ to find A
Accelarated Newton	$p_0, f(p), f'(p), M$	$p_{n+1} = p_n - \frac{Mf(p_n)}{f'(p_n)}$	$R = 2,$ A can only be calculated numerically
Bisection	$f(p), (a_0, b_0)$	$c_n = \frac{a_n + b_n}{2}$, choose (a_{n+1}, b_{n+1}) based on sign of $f(c_n)$	$R = 1, A = 0.5$
False Position	$f(p), (a_0, b_0)$	$c_n = b_n - \frac{f(b_n)(b_n - a_n)}{f(b_n) - f(a_n)}$, choose (a_{n+1}, b_{n+1}) based on sign of $f(c_n)$	$R = 1$,A can only be found numerically(as in seacant method)
Fixed Point Iteration	$p_0, g(x) = x$	$p_{n+1} = g(p_n)$	$R = k$, where k is the order of first nonzero derivative of p, $A = \|\frac{g^{(k)}(p)}{k!}\|$

Chapter 3 : Solving (Square) Systems of Equations

There are 2 kinds of systems : linear and nonlinear systems.

Non-Linear Systems

Non linear systems are systems of equations where there is an exponent to the variables. A solution to a system satisfies all the equations.

In general, a 2x2 system looks like

$f_1(x, y) = 0$

$f_2(x, y) = 0$

with exact solution

(p, q)

a 3x3 system looks like

$f_1(x, y, z) = 0$

$f_2(x, y, z) = 0$

$f_3(x, y, z) = 0$

with exact solution

(p, q, r)

There are three methods we are going to be using to solve nonlinear systems.

1. Newton's Method --> only for 2x2
2. Fixed Point Iteration --> 2x2 or 3x3
3. Gauss-Seidel Iteration --> 2x2 or 3x3

Newtons Method for Non-Linear Systems

We need p_0, q_0 and $f_1(x, y) = 0, f_2(x, y) = 0$

Note : The solution can be written as a vector

The iteration is given by :

$$\begin{pmatrix} p_1 \\ q_1 \end{pmatrix} = \begin{pmatrix} p_0 \\ q_0 \end{pmatrix} - J^{-1}\big|_{(p_0, q_0)} \begin{pmatrix} f_1(p_0, q_0) \\ f_2(p_0, 1_0) \end{pmatrix}$$

where J is the **jacobian matrix** of the 2x2 system is given by

$$\begin{pmatrix} \frac{\delta f_1}{dx} & \frac{\delta f_1}{dy} \\ \frac{\delta f_2}{dx} & \frac{\delta f_2}{dy} \end{pmatrix}$$

This can be considered the derivative of the system. Therefore, the above iteration equation is equivalent to $p_{n+1} = p_n - \frac{f(p_{)})}{f'(p_n)}$

Fixed Point Iteration for Non-Linear Systems

(For **2x2** systems)We need p_0, q_0 and $g_1(x,y) = x, g_2(x,y) = y$

This means that $g_1(p,q) = p$ and $g_2(p,q) = q$

The iteration is given by :

$$p_{n+1} = g_1(p_n, q_n)$$
$$q_{n+1} = g_2(p_n, q_n)$$

Gauss-Seidel Iteration for Non-Linear Systems

It is almost identical to Fixed Point Iteration, with some improvment.

(For **2x2** systems)We need p_0, q_0 and $g_1(x,y) = x, g_2(x,y) = y$

$p_1 = g_1(p_0, q_0)$

$q_1 = g_2(p_1, q_0)$

notice that we are substituting the found value of p_{n+1} to find q_{n+1}. Generally

$$p_{n+1} = g_1(p_n, q_n)$$
$$q_{n+1} = g_2(p_{n+1}, q_n)$$

Linear Systems

Revision : Row Operations

Directly Solving an NxN Linear System

Any *square* linear system is given by :

$A_{nxn}X_{nx1} = b_{bx1}$

We will only deal with linear systems that have a **unique solution.** That is to say, A is **non-singular**(A^{-1} exists). For example, the system

$2x_1 + x_2 - x_3 = 7$

$5x_1 + 5x_3 = 8$

$-3x_1 + x_2 - 2x_3 = -1$

is expressed as

$$A = \begin{pmatrix} 2 & 1 & -1 \\ 5 & 0 & 5 \\ -3 & 1 & -2 \end{pmatrix}$$

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$b = \begin{pmatrix} 7 \\ 8 \\ -1 \end{pmatrix}$$

There are 5 direct ways to solve a system

1. Cramers rule
2. Gaussian Elimination
3. Gaus-Jordan reduction
4. Inverse Method
5. Lu factorization

The numerical part of what we are learning, we are going to calculate the **cost** of these methods.

The cost, or **complexity** of a method, is the number of operations$(+ - \div x)$ in this method. For example, the cost of solving

$$\frac{5 + 2 * 4}{7 - 3}$$

is 4, because there are 4 operations. Now lets say that we have A_{2x2}. What is the cost of αA **?**.

$$\alpha \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

There are 4 operations here, so the cost is 4. Actually, we can say that for any matrix A_{nxn}, the cost of $\alpha A_{nxn} = n^2$

But what about the cost of $A_{nxn} + B_{nxn}$**?** well, it looks like this :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} + \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a+e & b+f \\ c+g & d+h \end{pmatrix}$$

So the cost is 4, or n^2.

And For $A_{nxn}B_{nxn}$, which looks like :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} * \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a*e+b*g & -- \\ -- & -- \end{pmatrix}$$

Each new member of the matrix costs us 3, multiplied by 4, then the cost is 12, or generally, if both matricies are the same size, then the cost is $2n^3 - n^2$.

However, if we are multiplying $A_{3x3}B_{3x1}$, the cost is $15 = 3(3+2)$. generally, for $A_{nxn}B_{nx1}$, the cost is $2n^2 - n$.

The cost of $det(A_{2x2}) = 3$. The cost of $det(A_{3x3}) = 14$, because we are calculating the determinant of three $3x3$ matricies, doing 3 multiplications, 1 addition, and 1 subtraction. For a $4x4$ matrix, the cost of $det(A_{4x4}) = 4 * 14 + 4 + 3 = 63$ in the same way.

Consider the following :

```
for p = 1 : n

    a = (p+s)/(2p+1)
```

The cost of this segment of code is $4 * n$

Cramers Rule

Let there be a $2x2$ system given by :

$2x_1 + x_2 = 5$

$x_1 - x_2 = 1$

Let us solve this system using cramers rule, and the cost of solving it.

$$A = \begin{pmatrix} 2 & 1 \\ 1 & -1 \end{pmatrix}, b = \begin{pmatrix} 5 \\ 1 \end{pmatrix}$$

Cramers rule :

$$x_n = \frac{det(A_n)}{det(A)}$$

Where A_n is A with column n replaced with b, so

$$A_1 = \begin{pmatrix} 5 & 1 \\ 1 & -1 \end{pmatrix}, A_2 = \begin{pmatrix} 2 & 5 \\ 1 & 1 \end{pmatrix}, det(A_1) = -6, det(A_2) = -3, det(A) = -3$$

The cost so far is $3 * 3 = 9$

$x_1 = \frac{-6}{-3} = 2$

$x_2 = \frac{-3}{-3} = 1$

The total cost is $3 * 3 + 2 = 11$

Generally speaking, the cost of solving using cramers rule is

$$(n + 1) * (cost(det(A_{nxn})) + n$$

Special Case : Strictly Upper Triangular System of NxN

An Upper triangular matrix is a matrix where all members beneath the diagonal are `0` and none of the members of the diagonal are `0`, for example :

$$A = \begin{pmatrix} 2 & 3 & -1 \\ 0 & 6 & 2 \\ 0 & 0 & 3 \end{pmatrix}, b = \begin{pmatrix} 3 \\ 8 \\ 3 \end{pmatrix}$$

The cost of solving such a system (by back substitution) **is always** n^2

The cost of converting a system to upper triangular is $\sum_1^{n-1} 2(n-1)^2 + (n-1)$

Row Operations

1. Row Operation I : Switch two rows.
2. Row Operation II : Multiply a row with a nonzero constante
3. Row Operation III : Replace a row by adding it to a multiple of another row, that is $R_k \to R_k - M_{kp}R_p$

Gaussian Elimination

To do gaussian elimination, we must

1. Convert $[A|b]$ into an upper triangular system $[\mu|c]$.
2. Solve by back substitution.

So the cost of the gaussian elimination

$$\sum_{t=n-1}^{1} (2(t)(t+1) + (t)) + n^2$$

Where $t = n - k$. Which when simplified further

$$\frac{4n^3 + 3n^2 - 7n}{6} + n^2$$

Simplifying again

$$\frac{4n^3 + 9n^2 - 7n}{6}$$

Partial Pivoting

Choose a pivot so that all multipliers have magnitude less than 1.

Switch Rows so that the pivot element is the largest in magnitude

Becuase the magnitude of multipliers is less than 1, That means that the error will decrease and the result is closer to the real value.

LU Factorisation

The cost is

$$\sum_{k=1}^{n-1} (2(n-k)^2 + (n+k)) + (n^2 - n) + n^2$$

Which when simplified, gives us

$$\frac{4n^3 + 9n^2 - 7n}{6}$$

again.

Gauss-Jordan Reduction

the cost is

$$\frac{2n^3 + n^2 + n}{2}$$

Inverse Method

the cost is

$$\frac{6n^3 - n^2 - n}{2}$$

Summary Of Costs

Operation	Solving Cost
Solve by Cramers Rule	$(n+1) * (cost(det(A_{nxn})) + n$
Converting a System to Upper Triangular	$\sum_1^{n-1} 2(n-1)^2 + (n-1)$
Cost of Solving an Upper Triangular System	n^2
Solve by Gaussian Elimination	$\frac{4n^3+9n^2-7n}{6}$
Solve by LU Factorization	$\frac{4n^3+9n^2-7n}{6}$
Solve by Gauss-Jordan Reduction	$\frac{2n^3+n^2+n}{2}$
Solve by Inverse Method	$\frac{6n^3-n^2-n}{2}$

Chapter 4 : Interpolation

part of polinomial approximatin

general idea : given $f(x), x \leq b$, how do you approxmiate $f(x)$ by a polinomial $P_n(x)$ **?**

Answer : You find some points of $f(x)$ and you use these points to find $P_n(x)$ where $P_n(x) \approx f(x)$ for $a \leq x \leq b$

Given the following points $(x_0, y_0), (x_1, y_1), \ldots., (x_n, y_n)$ where $x_0 < x_1 < \ldots. < x_n$, then $\exists!$ Polimomial $P_n(x)$ of degree **at most** n that passes through these points.

Newton Polynomial

$$P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots. + a_n(x - x_0)(x - x_1) \ldots (x - x_{n-1})$$

where a_n is the value of $f[x_0, \ldots. x_n]$

Larange

1. for 2 points : $f(x) = \frac{(x-x_1)}{x_0-x_1} f_0 + \frac{(x-x_0)}{(x_1-x_0)} f_1$
2. for 3 points : $\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} f_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} f_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} f_2$

Divided Difference

The general form of the divided difference is

$$f[x_0, x_1 \ldots., x_n] = \frac{f[x_0, \ldots\ldots x_{n-1}] - f[x_1, \ldots., x_n]}{x_0 - x_n}$$

and

$$f[x] = f(x)$$

so

$$f[x_0, x_1] = \frac{f_0 - f_1}{x_0 - x_1}$$

Uniform Partition(Equally Spaced Nodes)

- divide the range into n points
- Find $h = \frac{x_0 - x_n}{n}$
- Find $x_n = x_0 + n * h$
- Find $y(x)$ for all x
- Solve like before

Interpolation Error

Interpolation Error is a type of **truncation error.** The general idea is that you have a set of points $(x_0, y_0) \ldots (x_n, y_n)$ that you use to approximate $f(x)$. however, it is not the complete function, so

$f(x) = P_n(x) + E_n(x)$ it is similar to the error in taylor series, given by :

$$E_n(x) = \frac{(x - x_0)(x - x_1)\dots(x - x_n)f^{(n+1)}(c)}{(n+1)!}$$

We cannot find the true error, but we can fin the upper bound of the error

Uniform :

1- $|E_1(x)| \leq \frac{h^2 M_2}{8}$

2- $|E_2(x)| \leq \frac{h^3 M_3}{9\sqrt{3}}$

3- $|E_3(x)| \leq \frac{h^4 M_4}{24}$

Where M_k is the max of $f^{n+1}(x)$

Non-Uniform

- Use the general formula, then find the max of $Q(x) = (x - x_0)\dots.(x - x_n)$
- Find the max of $M_k = f^{n+1}(x)$
- calculate $|E(x)| = \frac{MAX(Q(x)) * MAX(f^{(n+1)}(x))}{(n+1)!}$

Proofs

Proof that the upper bound of the uniform of error is given by $|E_1(x)| \leq \frac{h^2 M_2}{8}$

proof

$$E_1(x) \frac{(x - x_0)(x - x_1)f''(x)}{2}$$

Chapter 5 : Cubic Spline and Curve Fitting

Cubic Spline

Cubic Spline is one of the examples of what is called the piecewise polynomial interpolation.

Given $(x_0, y_0)\ldots\ldots(x_n, y_n)$

We want to find a function $S(x)$

But what is the general form of $S(x)$?

well,

$S(x) = (\ S_0(x) =$ نيدعبهلقنبيسفنلل ش$-)$. $-$

We end up with $4n$ unknowns. For $S(x)$ to be a cublic spline, it needs to satisfy

1. $S(x)$ must pass through all of $(x_0, y_0)\ldots\ldots(x_n, y_n)$, that is $S_0(x_0) = y_0, S_1(x_1) = y_1, \ldots S_n(x_n) = y_n$
2. $S(x)$ is connected at the conjunctions, that is $S_0(x_1) = S_1(x_1), S_1(x_2) = S_2(x_2)\ldots\ldots S_{n-2}(x_{n-1}) = S_{n-1}(x_{n-1})$
3. $S(x)'$ is connected at the conjunctions, that is $S_0'(x_1) = S_1'(x_1), S_1'(x_2) = S_2'(x_2)\ldots\ldots S_{n-2}'(x_{n-1}) = S_{n-1}'(x_{n-1})$
4. $S(x)''$ is connected at the conjunctions, that is $S_0''(x_1) = S_1''(x_1), S_1''(x_2) = S_2''(x_2)\ldots\ldots S_{n-2}''(x_{n-1}) = S_{n-1}''(x_{n-1})$

   The sum of the conditions = $(n+1) + 3(n-1) = 4n - 2$

   So we are 2 assumtions short right now !

Now, According to the 2 missing conditions we add, there are 2 types of cubic splines.

1. Natural Cubic Spline

   a. $S_0''(x_0) = 0$

   b. $S_{n-1}''(x_n) = 0$

2. Clamped Cubic Spline

   a. $S_0'(x_0) = f'(x_0)$

   b. $S_{n-1}'(x_n) = f'(x_n)$

Curve Fitting

Say we have points $(x_1, y_1)\ldots(x_n, y_n)$

we need to find a curve $f(x)$ that fits these points the most.

There are 3 types of error norms

1. Max Error = Let error $E = MAX_{1 \leq k \leq n}|e_k|$

2. Average Error = Let error $E = \frac{\sum_{k=1}^{n} |e_k|}{n}$
3. Root Mean Square error = Let error $E = \sqrt{\frac{\sum_{k=1}^{n} (e_k)^2}{n}}$

We will be using the third norm to find the $f(x)$ that fits the data the most. This $f(x)$ is called the **least square curve**

Lineraization

The general idea is to change the shape of the curve to linear. We do tis sby changing

$$y = f(x) \to \gamma = AX + B$$

we use the normal equations

$$A \sum X_k^2 + B \sum X_k = \sum X_k \gamma_k$$

and

$$A \sum X_k + nB = \sum \gamma_k$$

Chapter 6 : Numerical Differentiation

Remember :

$$f'(x_0) = \lim_{h->0} \frac{f(x_0+h)-f(x_0)}{h}$$

where h is the step size or bandwidth.

But now, we will use numerical methods to find $f'(x_0)$ or higher order derivatives. These formulas are called **difference formulas** or finite difference formulas.

Difeerence Formulas

There are three types of difference formulas :

1. Central Difference Formulas $\rightarrow$ CDF
2. Forward Difference Formulas $\rightarrow$ FDF
3. Backwards Difference Formulas $\rightarrow$ BDF

There are five formulas we are going to be studying

1. CDF of order $O(h^2)$ for $f'(x_0)$

   $f'(x_0) \approx \frac{f(x_0+h)-f(x_0-h)}{2h}$ with error $E = \frac{-h^2 f'''(c)}{6}$.

   Notation : $f(x_0+kh) = f_k, k = \pm1, \pm2, ..$

   so we can express the above as $f'(x_0) = \frac{f_1 - f_{-1}}{2h}$ with error $E = \frac{h^2 f'''(c)}{6}$

2. FDF of order $O(h^2)$ for f'

   $f'(x_0) \approx \frac{-3f_0+4f_1-f_2}{2h}$ Where error $E = \frac{h^2 f'''(c)}{3}$

3. BDF of order $O(h^2)$ for f'

   $f'(x_0) \approx \frac{3f_0+4f_{-1}+f_{-2}}{2h}$ with error $E = \frac{h^2 f'''(c)}{3}$

4. CDF of order $O(h^4)$ flor f'

   $f'(x_0) \approx \frac{-f_2+8f_1-8f_{-1}+f_{-2}}{12}$ with error $E = \frac{h^4 f^{(5)}(c)}{30}$

5. CDF of order $O(h^2)$ for f''

   $f''(x_0) \approx \frac{f_1-2f_0+f_{-1}}{h^2}$ with error $E = \frac{-h^2 f^{(4)}(c)}{12}$

But how do we derive a difference formula?

Deriving Difference Formulas

There are 3 ways to do this

1. Taylor Series

2. Lagrange Interpolation
3. Newton's Polynomial

Deriving Difference Formulas : Taylor Series

Notice that the numerator always contains x_0 somwhere. Say we apply the taylor series of $f(x)$ about x_0. We get $f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{(x-x_0)^2 f''(x_0)}{2!} + \ldots$

Now, lets replace x with $x_0 + kh$. Then we get

$$f(x) = f(x_0) + (x_0 + kh - x_0)f'(x_0) + \frac{(x_0+kh-x_0)^2 f''(x_0)}{2!} + \ldots + \frac{(x_0+kh+x_0)^{n+1} f^(n+1)(c)}{(n+1)!}$$

Simplifying, we get

$$f_k = f_0 + khf'(x_0) + \frac{(kh)^2}{2} f''(x_0) + \ldots. + \frac{(kh)^{n+1} f^{(n+1)}(c)}{(n+1)!}$$

let $k = 1$

$$f_1 = f_0 + hf'(x_0) + \frac{(h)^2}{2} f''(x_0) + \ldots. + \frac{(h)^{n+1} f^{(n+1)}(c)}{(n+1)!}$$

let $k = 2$

$$f_2 = f_0 + 2hf'(x_0) + \frac{(2h)^2}{2} f''(x_0) + \ldots. + \frac{(2h)^{n+1} f^{(n+1)}(c)}{(n+1)!}$$

Simplifying results in our above formulas.

Deriving Difference Formulas : Lagrange and Newton Polynomials.

The general idea is to find $P_n(t)$, so we have

$f(t) = P_n(t) + E_n(t)$

Now, deriving this, we get

$f'(t) = P_n'(t) + E_n'(t)$

Chapter 7 : Numerical Integration

The general idea is to estimate $\int_a^b f(x)dx$

We know that the value of $\int_a^b f(x)dx$ is the area under the curve.

We also know that $\int_a^b f(x)dx = \sum_1^n f(C_k)\triangle x_k$ (Reemans Sum)

The idea is to replace Reemans Sum with other formulas called **Quadrature Formulas** to approximate $\int_a^b f(x)dx$, so

$$\int_a^b f(x)dx = Q[f] + E[f]$$

Quadrature Formulas

Closed Newton-Cotes Quadrature Formulas

1. Trapezoidal Rule --> $\int_a^b f(x)dx = \frac{h}{2}(f_0 + f_1) - \frac{h^3 f''(c)}{12}$, where h is the distance between x_0, x_1
2. Simpsons Rule--> $\int_a^b f(x)dx = \int_{x_0}^{x_2} f(x)dx = \frac{h}{3}[f_0 + 4f_1 + f_2] - \frac{h^5 f^{(4)}(c)}{90}$, where $h = \frac{b-a}{2}$
3. Simpsons $\frac{3}{8}$ rule--> $\int_a^b f(x)dx = \int_{x_0}^{x_3} f(x)dx = \frac{3h}{8}[f_0 + 3f_1 + 3f_2 + f_3] - \frac{3h^5 f^{(4)}(c)}{80}$, where $h = \frac{b-a}{3}$

Degree Of Percision

Defn : given a $Q[f]$, the DOP of $Q[f]$ is the largest positive integer n such that the formula is exact $\forall$ polynomials of degree $\leq n$, In other terms, the DOP of $Q[f]$ is n if $E[1] = E[x] = E[x^2] \ldots = E[x^n] = 0$, but $E[x^{n+1}] \neq 0$

Chapter 9 : Ordinary Differential Equations

How to solve an IVP

1. Eulers Method
2. Taylor's Method of Order 2
3. Taylor's Method of Order 3
4. Taylor's Method of Order 4
5. Heun's Method
6. Runge-Kutta of order 4

Eulers Method

$$y_{k+1} = y_k + hf(t_k, y_k) \quad , E_n = \frac{h^2}{2} y''(c) \quad , E[y(b), h] = \frac{(b-a)h}{2} y''(c) \quad , k = 0, \ldots, n$$

Taylor's Method of Order 2

$$y_{k+1} = y_k + hf(t_k, y_k) \quad , E_n = \frac{h^2}{2} y'(c) \quad , E[y(b), h] = \frac{(b-a)h}{2} y'(c) \quad , k = 0, \ldots, n$$

Heun's Method

$$y_{k+1} = y_k + \frac{h}{2} [f(t_k, y_k) + f(t_{k+1}, y_k + hf(t_k, y_k))]$$

Runge-Kytta of Order 4

$f_1 = f(t_k, y_k)$

$f_2 = f(t_k + \frac{h}{2}, y_k + \frac{h}{2} f_1)$

$f_3 = f(t_k + \frac{h}{2}, y_k + \frac{h}{2} f_2)$

$f_4 = f(t_{k+1}, y_k + hf_3)$

$$y_{k+1} = y_k + frach6[f_1 + 2f_2 + 2_f 3 + f_4]$$