Unit-1

1> Bisection Method:

→ Two initial guessess that brackets root. (say a & b).

→ We use those guessess in given function ~~al value~~ [i.e. calculate $f(a)$ & $f(b)$]

→ Then we calculate mid-point as; $x_m = \dfrac{a+b}{2}$ and calculated $f(x_m)$.

→ $f(a) \times f(x_m) < 0$
then,
$b = x_m$; | else $a = x_m$;

→ Error $= \left|\dfrac{b-a}{b}\right|$ or $\left|\dfrac{a-b}{b}\right|$

cos 1
cos 2
value calculate गर्दा calculator लाई Radian mode मा राख्नु पर्छ
mode + DRG

Iteration 2^{nd}

$a =$ ____ $b =$ ____

Simelarly we proceed until we get desired approximation.

2> Newton Raphson Method

→ Initial guess. (let x_0)

→ Calculate $f(x_0)$ and $f'(x_0)$

→ Calculate x_1 as; $\boxed{x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}}$.

→ Error $= \left|\dfrac{(x_1 - x_0)}{x_1}\right|$

→ Proceed until desired approximation.

e.g. $f(x) = \log x - \cos x$.

$f'(x) = \dfrac{1}{x} + \sin x$.

3> Secant Method;

→ Two initial guesses that brackets root (say x_1 & x_2).
with given error precision E.

→ We calculate $f(x_1)$ and $f(x_2)$

→ Then we calculate x_3 as; $\boxed{x_3 = x_2 - \dfrac{f(x_2)(x_2 - x_1)}{f(x_2) - f(x_1)}}$

x_3 $f(x_3)$

Scanned by TapScanner

4) Fixed-Point Iteration Method:

→ We write given $f(x)$ in the form $x = g(x)$, by adding x on both sides.

→ Then we assume initial guess. (say x_0)

→ We just put value (initial guess) in $x = g(x_0)$ to get x_1.

→ $\text{Error} = \left| \frac{x_1 - x_0}{x_1} \right|$

5) Finding Multiple roots by using Newton-Raphson Method:

→ Newton's Method and Synthetic division is applied.

→ Analyse the degree of polynomial and assume initial guess (x_0).

→ Find the root using Newton Raphson method $x_r = x_0 - \frac{f(x_0)}{f'(x_0)}$

→ Now use synthetic division by $(x - x_r)$, to get the polynomial with degree one less than original polynomial. Again use Newton Raphson method to get next root.

→ Continue process until polynomial of degree one is achieved as ; $a_1 x + a_0 = 0$.

& Calculate final root as $x_r = -\frac{a_0}{a_1}$.

6) Horner's Method:-

→ We write coeff. of given polynomial as $a_3 = --$, $a_2 = --$, $a_1 = --$, $a_0 = --$

→ Given to find at $x = ------$ in question.

→ Now,

$b_n = a_n =$

$$\boxed{b_{n-1} = a_{n-1} + b_n * x} \quad \text{while } n \geq 0$$

→ finally put value of x in eqn $b_0 = --$

Unit-2 [INTERPOLATION]

1) Lagrange Interpolation:

n order of polynomial interpolation is given by; $\boxed{P_n(x) = \sum_{i=0}^{n} f_i \cdot l_i(x)}$.

i.e., $P_n(x) = f_0 \cdot l_0(x) + f_1 \cdot l_1(x) + \ldots + f_n \cdot l_n(x)$ — (i)

Now we choose n-data points closest to given value to find and also bracket that value.

where, $\boxed{l_i(x) = \prod_{j=0,\ j\neq i}^{n} \frac{x - x_j}{x_i - x_j}}$

Finally we substitute values of f_i & $l_i(x)$ in eqn (i) to get solution $P_n(x)$.

Derivation: A second order polynomial can be written in the form;

$$\boxed{P_2(x) = b_0(x-x_0)(x-x_1) + b_1(x-x_1)(x-x_2) + b_2(x-x_2)(x-x_0)}$$

Then we calculate $P_2(x_0) = f_0$, $P_2(x_1)$, $P_2(x_2)$ and substitute.

2) Newton's Divided Difference Interpolation:-

Polynomial of degree n is of form;

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

Choose n+1 nearest data points that bracket given value as;

$x_0 = \text{---}$ $\quad f(x_0) = \text{--}$

$x_1 = \text{---}$ $\quad f(x_1) = \text{--}$

$\vdots$

$x_{n-1} = \text{---}$ $\quad f(x_{n-1}) = \text{--}$

Now we calculate $a_0, a_1, a_2 \text{---} a_{n-1}$ as;

$$a_0 = f[x_0]$$
$$a_1 = f[x_1, x_0]$$
$$\vdots$$
$$a_{n-1} = f[x_{n-1}, x_{n-2}, \ldots, x_0]$$

Putting all values in eqn (i) we get general solution.

Finally putting value of x we get final solution.

Divided difference Table

x_0	$f(x_0)$ ← a_0	a_1	a_2	a_3
x_1	$f(x_1)$	$f[x_1, x_0]$	$f[x_2, x_1, x_0]$	
x_2	$f(x_2)$	$f[x_2, x_1]$	$f[x_3, x_2, x_1]$	$f[x_3, x_2, x_1, x_0]$
x_3	$f(x_3)$	$f[x_3, x_2]$		

Newton's Forward Difference Interpolation:

$h = x_{i+1} - x_i$

$x = x_0 + sh$

i.e, $s = \frac{x - x_0}{h}$ where, x is given value in question.

$$\boxed{P_n(x) = P_n(x_0 + sh) = f[x_0] + \Delta f(x_0)\frac{s}{1!} + \Delta^2 f(x_0) \cdot \frac{s(s-1)}{2!} + \dots + \Delta^n f(x_0) \cdot \frac{s(s-1)(s-2)\dots(s-n+1)}{n!}}$$

Newton's Backward Difference Interpolation:

all same as above only replace x_0 by x_n, Δ by ∇

$h = x_{i+1} - x_i$

$x = x_n + sh$

i.e, $s = \frac{x - x_n}{h}$ where, x is given value in question.

$$\boxed{P_n(x) = P_n(x_n + sh) = f[x_n] + \nabla f(x_n)\frac{s}{1!} + \nabla^2 f(x_n) \cdot \frac{s(s+1)}{2!} + \dots + \nabla^k f(x_n) \cdot \frac{s(s+1)\dots(s+n-2)(s+n-1)}{k!}}$$

also $(s-1)$ by $s+1$

(3)

[REGRESSION]

1) Linear Regression [least square regression line $y = ax+b$] :-

Given $x = x_i$

$Y = f(x) = y_i$

Now we calculate x_i^2 and $x_i y_i$ from given data. i.e. [Σx_i^2 & $\Sigma x_i y_i$]

We know that,

$y = ax+b$ —— (i)

So, we find values of a and b as follows;

$$a = \frac{\sum_{i=1}^{n} y_i}{n} - b.\frac{\sum_{i=1}^{n} x_i}{n}$$

$$\& \quad b = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

Finally putting values of a and b in eq^n (i) we get solution.

2) Non-linear Regression :- [i.e. Exponential $y = ae^{bx}$]

a) By fitting Exponential Model:

Form: $y = ae^{bx}$

Values of a & f(b) are evaluated as;

$$a = \frac{\sum_{i=1}^{n} y_i e^{bx_i}}{\sum_{i=1}^{n} e^{2bx_i}}$$

left

$$f(b) = \sum_{i=1}^{n} y_i x_i e^{bx_i} - \frac{\sum_{i=1}^{n} y_i e^{bx_i}}{\sum_{i=1}^{n} e^{2bx_i}} \sum_{i=1}^{n} x_i e^{2bx_i} = 0$$

f(b) can be solved for b using bisection method. We assume b = ... and b = ... [two values] that bracket root of f(b)=0.

from copy

b) Fitting Exponential model by Linearization.

Form: $y = ae^{bx}$

$\Rightarrow \log y = \log a + bx$

This is similar to linear equation,

$y = a + bx$.

Thus we evaluate a and b as;

$$\log a = \frac{\sum_{i=1}^{n} \log y_i}{n} - b.\frac{\sum_{i=1}^{n} x_i}{n}$$

$$\& \quad b = \frac{n\sum_{i=1}^{n} x_i \log y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} \log y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

log y_i, $x_i \log y_i$ ...

Similar as only log added where there is y_i

$\log a = 1.388$ (let)

$\Rightarrow a = e^{1.388} = 4.006$

Finally putting value of a & b we get soln.

3> Fitting Polynomial Models [i.e, Quadratic : $y = a_0 + a_1 x + a_2 x^2$.]

For finding coefficients a_0, a_1, a_2;

$$\begin{bmatrix} n & \left(\sum_{i=1}^{n} x_i\right) & \left(\sum_{i=1}^{n} x_i^2\right) \\ \left(\sum_{i=1}^{n} x_i\right) & \left(\sum_{i=1}^{n} x_i^2\right) & \left(\sum_{i=1}^{n} x_i^3\right) \\ \left(\sum_{i=1}^{n} x_i^2\right) & \left(\sum_{i=1}^{n} x_i^3\right) & \left(\sum_{i=1}^{n} x_i^4\right) \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} x_i^2 y_i \end{bmatrix}$$

We make table to calculate all these values and solving matrix we get a_0, a_1, a_2 and putting in eqn we get soln.

Unit-3

A) Numerical Differentiation:-

(a) Differentiating Continious Functions

Two Point forward difference formula

$$f'(x) = \frac{f(x+h) - f(x)}{h}$$

Two point Backward Difference formula

$$f'(x) = \frac{f(x) - f(x-h)}{h}$$

Three Point formula

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

$$\text{Error} = \left| \frac{\text{true value} - \text{approx value}}{\text{true value}} \right| \times 100\%$$

(b) Differentiating discrete (tabulated) functions:-

i) Derivatives Using Newton's Divided Difference Formula.

→ Used to find derivative of tabulated data when given arguments are <u>unequally spaced</u>.

→ Constructing divided difference table we use Newton's divided difference formula as follows:-

Similar to newtons divided diff. interpolation formula of 2nd chapter

$$P_n(x) = f(x) = f[x_0] + f[x_1, x_0](x-x_0) + f[x_2, x_1, x_0](x-x_0)(x-x_1)$$
$$+ f[x_3, x_2, x_1, x_0](x-x_0)(x-x_1)(x-x_2) + \cdots$$

Now, first derivative of eqn (i) w.r.t x, we get, (given value of x in ques x_0, x_1, x_2, x_3)

$$\boxed{f'(x) = f[x_1, x_0] + f[x_2, x_1, x_0]\{(x-x_1) + (x-x_0)\} + f[x_3, x_2, x_1, x_0]\{(x-x_1)(x-x_2) + (x-x_0)(x-x_2) + (x-x_0)(x-x_1)\} + \cdots}$$

ii) Derivatives Using Newton's Forward Difference formula:- (unit 2 के formula)

→ Used to find derivative of tabulated data when given values are <u>equally spaced</u> and where we want to find the function at a point <u>near to beginning</u>. (formulas refer from copy)

iii) Derivatives Using Newton's Backward Difference formula:-

Same to number (ii) difference is it is used point <u>near to end</u> instead of Beginning

Better to study formulas and questions of these last two from copy.

(c) Maxima and Minima of Tabulated Functions:-

→ First we create forward difference table.

→ we know that;

$$as^2+bs+c=0 \quad \text{--- (i)}$$

where,

$$a=\frac{1}{2}\Delta^3 f(x_0)$$

$$b=\Delta^2 f(x_0)-\Delta^3 f(x_0)$$

$$c=\Delta f(x_0)-\frac{1}{2}\Delta^2 f(x_0)+\frac{1}{3}\Delta^3 f(x_0).$$

→ Now we put values of a, b & c in eqn (i) to get two values of s. [for eg. $s=\pm 1$]

→ Then we compute $x=x_0\pm sh$ to find value of x.

→ Then we compare x with s (e.g. $s=x$).

→ Then putting $s=x$ in Newton's forward difference formula and putting values we get some function $f(x)$.

→ Finally from $f(x)$ we can eaisly calculate $f'(x)$ and $f''(x)$. Putting two values of x in $f''(x)$ we can eaisly determine maximum and minimmum.

Finding $f'(x)$ for equally spaced data near begining: (Forward).

1st Derivative: $f'(x)=\frac{1}{h}\left\{\Delta f(x_0)+\Delta^2 f(x_0)\cdot\frac{(2s-1)}{2!}+\Delta^3 f(x_0)\cdot\frac{(3s^2-6s+2)}{3!}\right.$

2nd Derivative: $f''(x)=\frac{1}{h^2}\left\{\Delta^2 f(x_0)+\frac{1}{3!}(6s-6)\Delta^3 f(x_0)+\frac{1}{4!}(12s^2-36s+22)\Delta^4 f(x_0)\right\}$

Finding $f'(x)$ for equally spaced data near end. (Backward)

~~1st Derivative $f'(x)=\frac{1}{h}\{\nabla$~~ Same as forward just replace. $\Delta\rightarrow\nabla$

$x_0\rightarrow x_n$

$-ve\rightarrow +ve$

B> Numerical Integration

1> Trapezodial Rule:

ⓐ Two-point trapezodial rule (Simply trapezodial rule).

$$\int_{x_0}^{x_1} f(x).dx = (x_1 - x_0)\left[\frac{f(x_1)+f(x_0)}{2}\right]$$

ⓑ. Composite trapezodial rule (OR Multiple-segment trapezodial rule).

$$\int_{x_0}^{x_n} f(x).dx = \frac{h}{2}\left[f(x_0) + 2\left\{\sum_{i=1}^{k-1} f(x_0+ih)\right\} + f(x_n)\right]$$

where, $h = \frac{x_n - x_0}{k}$

x_0, h

$x_1 = x_0 + h$

$x_2 = x_0 + 2h$

$x_3 = x_0 + 3h$

$\vdots$

~~first term~~ [illegible] total k segment [illegible] $\frac{h}{2}$ times first term + 2 times sum of all middle terms + last term

(let k=4 then last term will be $f(x_3)$ since we start from 0)

2> Simpson's 1/3 Rule:-

ⓐ

$$I = \int_{x_0}^{x_2} f(x).dx = \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right]$$

where, $h = \frac{x_2 - x_0}{n}$

(We take n=2 in this rule)

ⓑ. Composite Simpsons 1/3 Rule:-

$$\int_{x_0}^{x_n} f(x).dx = \frac{h}{3}\left[f(x_0) + 4\sum_{\substack{i=1\\ i=odd}}^{k-1} f(x_i) + 2\sum_{\substack{i=2\\ i=even}}^{k-2} f(x_i) + f(x_n)\right]$$

where, $h = \frac{x_n - x_0}{k}$

(i.e, no. of segments)

ⓒ Simpson's 3/8 Rule:-

$$I = \int_{x_0}^{x_3} f(x).dx = \frac{3}{8}h\left[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right]$$

(We take n=3 in this rule)

where, $h = \frac{x_n - x_0}{n}$

ⓓ Composite Simpsons 3/8 Rule:-

$$\int_{x_0}^{x_n} f(x).dx = \frac{3}{8}h\left[f(x_0) + 3\sum_{\substack{i=1\\ i \bmod 3 \neq 0}}^{k-1} f(x_i) + 2\sum_{\substack{i=2\\ i \bmod 3 = 0}}^{k-1} f(x_i) + f(x_n)\right]$$

where, $h = \frac{x_n - x_0}{n}$

(other than multiples of 3) (multiples of 3)

c> Simp-

Gaussian Integration:

Step-1

$$\int_a^b f(x)\,dx = \frac{b-a}{2}\int_{-1}^{1} g(z)\cdot dz$$

then, $x = \frac{b-a}{2}z + \frac{a+b}{2} = 2z$

Step-2 Formulas:

One-point: $\int_{-1}^{1} f(x)\,dx = 2\cdot f(0).$

Two-point: $\int_{-1}^{1} f(x)\cdot dx = f\left(\frac{-1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right).$

Three-[redacted]point: $\int_{-1}^{1} f(x)\cdot dx = \frac{5}{9}f\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9}f(0) + \frac{5}{9}f\left(\sqrt{\frac{3}{5}}\right)$

Romberg Integration:- [Escaped]

Given, $\int_{x_0}^{x_1} f(x)\cdot dx$ with estimate $T(i, j)$. Let

$x_1 - x_0$

Step-1: Use trapezodial rule to compute $T(0,0) = \frac{h}{2}\left(f(x_0) + f(x_1)\right)$

Step-2: Use recursive trapezodial rule as; $T(i,0) = \frac{T(i-1,0)}{2} + \frac{h}{2^i}\sum_{k=1}^{2^{i-1}} f\left(x_0 + (2k-1)\frac{h}{2^i}\right)$

~~i.e, T(0,0)~~ i.e, $T(1,0)$ & $T(1,1)$

where, $h = (x_1 - x_0)$

Step-3: Finally use Romberg integration formula as;

to compute $T(1,1)$, $T(2,1)$, $T(2,2)$, ~~$T(\ ,k)$~~

$$\int_a^b f(x)\cdot dx = T_{m+k,k} \cancel{+ O(h^{2(k+1)})}$$

$$= \frac{4^k T_{m+k,k-1} - T_{m+k-1,k-1}}{4^k - 1} \cancel{+ O(h^{2(k+1)})}$$

$T(m+k,k)$ form.

where, m is integer.

Numerical integration is the process of computing ...

Unit-4

Do-Little LU Decomposition:-

given matrix

$$A = LU$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \quad \text{--- (i)}$$

- Naive Gauss elimination Method
  - → Forward elimination of unknowns
  - → Back substitution.
- Gauss elimination with partial pivoting
  - → First we select largest absolute value along selected row then we proceed same as above.
- Gauss Jordan Method
  - → Augmented matrix $[A \vdots C]$
  - → Normalize pivot elements.
  - → Make others zero.
- Matrix Inversion

$$[A \vdots I] \downarrow [I \vdots A]$$

Inverse of A

$u_{11} = a_{11} = \text{value}$; $u_{12} = a_{12} = \text{value}$; $u_{13} = a_{13} = \text{value}$;

~~$l_{i1} = \frac{a_{i1}}{u_{11}}$~~ → $l_{i1} = \dfrac{a_{i1}}{u_{11}}$ (If $i > j$ condition & $j = 1$)

For l_{ij} only these two

row, column

1) If $i \le j$ then,

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}$$

e.g. $u_{22} = a_{22} - l_{21} u_{12}$

Since we have $i = 2$
So, $i-1 = 2-1 = 1$
Hence, loop runs only once

Similarly

$u_{33} = a_{33} - l_{31} u_{13} - l_{32} u_{23}$

Since we have $i = 3$,
Now, $i-1 = 3-1 = 2$
So, loop runs twice

2) If $i > j$, then,

$$l_{ij} = \frac{1}{u_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} \cdot u_{kj} \right)$$

e.g. $l_{31} = \dfrac{a_{31}}{u_{11}}$

Since if $i > j$ & $j = 1$ then we use $l_{i1} = \dfrac{a_{i1}}{u_{11}}$ formula

Similarly

e.g. $l_{32} = \dfrac{1}{u_{22}} \left(a_{32} - l_{31} u_{12} \right)$

Finally put all these values in eqn (i) to get ~~final sol~~ factorization

But for solⁿ we proceed as follows:-

i> Solve $[L][Z] = [C]$ using forward substitution to calculate values of Z.

ii> Again solve $[U][X] = [Z]$ by using backward substitution to get final solutions $x_1, x_2, x_3 \ldots$

Cholesky Method

$$[A] = [L][U] = \begin{bmatrix} u_{11} & 0 & 0 \\ u_{12} & u_{22} & 0 \\ u_{13} & u_{23} & u_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \quad \text{— (i)}$$

Two factors of coeff. matrix A

where,

$$u_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} u_{ki}^2}$$

& if $j > i$,

$$u_{ij} = \frac{1}{u_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj} \right).$$

Now we place these obtained values of u_{ii} and u_{ij} in eqⁿ (i) and this completes our factorization.

⇒ But for solution we proceed same process as we did for Do little LU decomposition as above.

Iterative Methods:-

Last

$AX = \lambda X$ — matrix, Eigen value, Eigen vector

Power Method

$Y = AX$

$X = \frac{1}{k} Y$ — k: element of Y with largest magnitude

$A^{-1}X = \frac{1}{\lambda}X$

i> Jacobi Iteration Method

→ Rewrite largest coeff. variable on left side.

→ Now we calculate first values of variables with initial guesses zero.

For $x_1 \rightarrow x_2$ & $x_3 = 0$ — This value of x_1 is not used in x_2

for $x_2 \rightarrow x_1$ & $x_3 = 0$

for $x_3 \rightarrow x_2$ & $x_1 = 0$ — Initial guess zero for all x_1, x_2, x_3

→ Now we use these values of x_1, x_2 & x_3 for 2nd iteration and calculated in table

ii> Gauss Seidel Method

→ Same method only difference is that after calculating x_1 we immediately use value of x_1 in eqⁿ of x_2 even in same iteration

Shooting Method is hard should be practiced more and more

Unit-5

+ Integration & Derivative basic rules

Order = highest derivative

degree = power of highest derivative.

A)

1) Taylors Series Method:-

$$y(x) = y(x_0) + (x-x_0)\,y'(x_0) + (x-x_0)^2 \frac{y''(x_0)}{2!} + \dots + (x-x_0)^n \frac{y^n(x_0)}{n!}$$

$$\boxed{\text{i.e. } (x-x_0)^n \cdot \frac{y^n(x_0)}{n!}}$$

y^n is in the form of derivatives e.g. for $n=1$, y', for $n=2$, y'' etc.

Note: $f = f(x,y) = \frac{dy}{dx}$ (i.e. first derivative y')

f_x = partial derivative of $f(x,y)$ w.r.t. x.

f_y = " ... w.r.t. y.

$y' = f(x,y)$

$y'' = f_x + f_y \cdot f$

$y''' = f_{xx} + 2ff_{xy} + f^2 f_{yy} + f_x f_y + f f_y^2$.

2) Picards Method

Step 1 → First we write integral equation of given differential equation in the form $\boxed{y = y_0 + \int_{x_0}^{x} f(x,y)\,dx}$ — (1)

Step 2 → We apply successive approximations on eqn (1) till two values of y becomes same or reaches desired accuracy as; [i.e. we repeat process].

$$y^{(n)} = y_0 + \int_{x_0}^{x} f\left(x, y^{(n-1)}\right) dx$$

use at = .. value in each approx in place of x

3) Eulers Method

New approx given by: $\boxed{y(x_{i+1}) = y(x_i) + h f(x_i, y_i)}$

~~where, n = 0,1,2,3...~~

h = step size

We should continue steps by adding step size each time until it reaches to given approximate value.

Scanned by TapScanner

4) Heun's Method:-

Given differential equation $\frac{dy}{dx} = f(x,y)$ with initial condition $y(x_0) = y_0$.

Step 1: First we find slopes m_1 and m_2 as;

$$m_1 = f(x_i, y_i)$$

$$\& \quad m_2 = f(x_1, y(x_0) + hf(x_0, y_0))$$

Step 2: Then we use Heun's formula $\boxed{y(x_{i+1}) = y(x_i) + \frac{h}{2}(m_1 + m_2)}$

Step 3: Continue step 1 & 2 until we reach to given approximate value by adding step size (h) each time.

h को approx value पुग्ने step

5) Fourth Order Runge-Kutta Method

Given, $\frac{dx}{dy} = f(x,y)$, $y(x_0) = y_0$

Step 1: Calculate m_1, m_2, m_3 & m_4 as;

$$m_1 = f(x_i, y_i)$$

$$m_2 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hm_1\right)$$

$$m_3 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hm_2\right)$$

$$m_4 = f(x_i + h, y_i + hm_3).$$

Step 2: Calculate

step size added in each iteration

$$\boxed{y(x_n + h) = y_n + \frac{1}{6}h(m_1 + 2m_2 + 2m_3 + m_4)}$$

Step 3: Continue step 1 & 2 until ~~given~~ we reach to given approx. h.

B. Solving System of Ordinary Differential Equation:

→ (i.e दुईवटा differential eqn दिएको बेला यसरी गर्ने)

→ यो method बाट बनेर Euler, Heun etc method qestim मा दिएको हुन्छ त्यसै अनुसार formula

→ Read from copy this for more understanding method in clear way