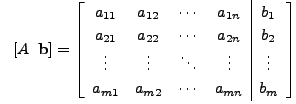
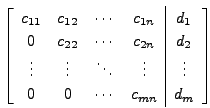
## THEORY

1> GAUSS’ ELIMINATION METHOD:**Gaussian elimination is a method of solving a linear system Ax=b (consisting of n equations in n unknowns) by bringing the augmented matrix**

 to an upper triangular form.

2>**REGULA FALSI METHOD :**The convergce process in the bisection method is very slow. It depends only on the choice of end points of the interval [a,b]. The function f(x) does not have any role in finding the point c (which is just the mid-point of a and b). It is used only to decide the next smaller interval [a,c] or [c,b]. A better approximation to c can be obtained by taking the straight line L joining the points (a,f(a)) and (b,f(b)) intersecting the x-axis. To obtain the value of c we can equate the two expressions of the slope m of the line L.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| m = | f(b) - f(a) | = | 0 - f(b) |  |
| (b-a) | (c-b) |  |

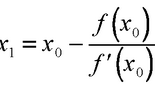
Such that

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| => (c-b) \* (f(b)-f(a)) = -(b-a) \* f(b)   |  |  | | --- | --- | | c = b - | f(b) \* (b-a) | | f(b) - f(a) | | IMG_256 |  |

Now the next smaller interval which brackets the root is obtained by checking f(a) \* f(b) < 0 then b = c   
                > 0 then a = c   
                 = 0 then c is the root.

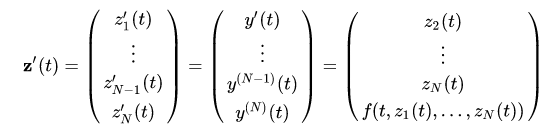
Selecting **c** by the above expression is called Regula-Falsi method or False position method.

3> **NEWTON-RAPHSON METHOD:**Newton-Raphson (N-R) technique requires only one inital value *x*0, which we will refer to as the *initial guess* for the root. To see how the N-R method works, we can rewrite the function *f*(*x*) using a Taylor series expansion in (*x*-*x*0): *f*(*x*) = *f*(*x*0) + *f*'(*x*0)(*x*-*x*0) + 1/2 *f*''(*x*0)(*x*-*x*0)2 + ... = 0

where *f*'(*x*) denotes the first derivative of *f*(*x*) with respect to *x*, *f*''(*x*) is the second derivative, and so forth. Now, suppose the initial guess is pretty close to the real root. Then (*x*-*x*0) is small, and only the first few terms in the series are important to get an accurate estimate of the true root, given *x*0. By truncating the series at the second term (linear in *x*), we obtain the N-R iteration formula for getting a better estimate of the true root:  Thus the N-R method finds the tangent to the function *f*(*x*) at *x*=*x*0 and extrapolates it to intersect the *x* axis to get *x*1. This point of intersection is taken as the new approximation to the root and the procedure is repeated until convergence is obtained whenever possible.

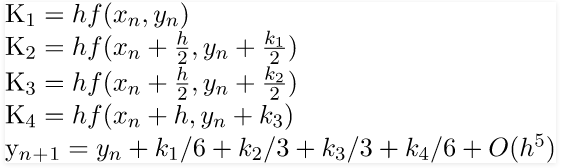
4>**EULER’S METHOD: F**ollowing is a general first order IVP given by equation:

Choose a value ‘h’ h {\displaystyle h} for the size of every step and sett n = t 0 + n h {\displaystyle t\_{n}=t\_{0}+nh} . Now, one step of the Euler method from t n {\displaystyle t\_{n}} to : is .The value of yn y n {\displaystyle y\_{n}} is an approximation of the solution to the ODE at time tn:yn~~y(tn)t n {\displaystyle t\_{n}} :y n ≈ y ( t n ) {\displaystyle y\_{n}\approx y(t\_{n})} . The Euler method is [explicit](https://en.wikipedia.org/wiki/Explicit_and_implicit_methods" \o "Explicit and implicit methods), i.e. the solution yn+1 y n + 1 {\displaystyle y\_{n+1}} is an explicit function of yi y i {\displaystyle y\_{i}} for i<=n.While the Euler method integrates a first-order ODE, any ODE of order *N* can be represented as a first-order ODE: to treat the equation

we introduce auxiliary variables z 1 ( t ) = y ( t ) , z 2 ( t ) = y ′ ( t ) , … , z N ( t ) = y ( N − 1 ) ( t ) {\displaystyle z\_{1}(t)=y(t),z\_{2}(t)=y'(t),\ldots ,z\_{N}(t)=y^{(N-1)}(t)} and obtain the equivalent equation:

This is a first-order system in the variable z(t)z ( t ) {\displaystyle \mathbf {z} (t)} z(t)and can be handled by Euler's method or, in fact, by any other scheme for first-order system.

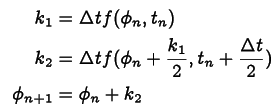
5> **RUNGE-KUTTA 4th ORDER METHOD:**Given following inputs: An ordinary differential equation that defines value of dy/dx in the form x and y. & Initial value of y, i.e., y(0).Thus we are given below .  
  
  
We have to find value of unknown function y at a given point x.The Runge-Kutta method finds approximate value of y for a given x. Only first order ordinary differential equations can be solved by using the Runge Kutta 4th order method.

Below is the formula used to compute next value yn+1 from previous value yn. The value of n are 0, 1, 2, 3, ….(x – x0)/h. Here **h is step height** and **xn+1 = x0 + h**. Lower step size means more accuracy.  
The formula basically computes next value yn+1 using current yn plus weighted average of four increments.

* k1 is the increment based on the slope at the beginning of the interval, using y
* k2 is the increment based on the slope at the midpoint of the interval, using y + hk1/2.
* k3 is again the increment based on the slope at the midpoint, using using y + hk2/2.
* k4 is the increment based on the slope at the end of the interval, using y + hk3.

The method is a fourth-order method, meaning that the local truncation error is on the order of O(h5), while the total accumulated error is order O(h4).

6>**RUNGE-KUTTA 2ND ORDER:**Given a vector of unknowns at time tn, and the first order differential equation, the second order Runge-Kutta estimate for is given by

where The error on each step is of order .

7>**BISECTION METHOD:** The method is applicable for numerically solving the equation *f*(*x*) = 0 for the real variable *x*, where *f* is a continuous function defined on an interval [*a*, *b*] and where *f*(*a*) and *f*(*b*) have opposite signs. In this case *a* and *b* are said to bracket a root since, by the intermediate value theorem, the continuous function *f* must have at least one root in the interval (*a*, *b*). At each step the method divides the interval in two by computing the midpoint *c* = (*a*+*b*) / 2 of the interval and the value of the function *f*(*c*) at that point. Unless *c* is itself a root (which is very unlikely, but possible) there are now only two possibilities: either *f*(*a*) and *f*(*c*) have opposite signs and bracket a root, or *f*(*c*) and *f*(*b*) have opposite signs and bracket a root. The method selects the subinterval that is guaranteed to be a bracket as the new interval to be used in the next step. In this way an interval that contains a zero of *f* is reduced in width by 50% at each step. The process is continued until the interval is sufficiently small.

Explicitly, if *f*(*a*) and *f*(*c*) have opposite signs, then the method sets *c* as the new value for *b*, and if *f*(*b*) and *f*(*c*) have opposite signs then the method sets *c* as the new *a*. (If *f*(*c*)=0 then *c* may be taken as the solution and the process stops.) In both cases, the new *f*(*a*) and *f*(*b*) have opposite signs, so the method is applicable to this smaller interval.

8> **WEDDLE’S RULE:**

Formula:

Let the values of a function IMG_256be tabulated at points IMG_257equally spaced by IMG_258, so IMG_259, IMG_260, .... Then Weddle's rule approximating the integral of IMG_261is given by the formula

|  |
| --- |
| IMG_262 |

i ≤ n {\displaystyle i\leq n} iiii

------------------------CORRECTIONS---------------------