

```

MainFunction[F_, J_, xinit_, opti_List, vars_] :=
Module[{nm, inm, thnm, xNM, xiNM, xthNM, oNM, oiNM, othNM, summrray, ss, s, len, Fk,
summ},

(* Find root of F *)
n = Length[vars];
x00 = Table[1.25, {n}];
v = {vars, x00} // Transpose;
s = FindRoot[F, v];
ss = {x1, x2, x3, x4} /. s;

(* Running all the methods: NewtonN, ImprovedNewtonN, and ThirdOrderNewtonN *)
nm = NewtonN[F, J, xinit, opti, vars][[1]];
inm = ImprovedNewtonN[F, J, xinit, opti, vars][[1]];
thnm = ThirdOrderNewtonN[F, J, xinit, opti, vars][[1]];

(* Extracting solutions from each method *)
xNM = nm[[4]];
xiNM = inm[[4]];
xthNM = thnm[[4]];

(* Calculating convergence rates *)
len = Length[xNM];
oNM = Table[
Log[Norm[xNM[[k + 1]] - ss]/Norm[xNM[[k]] - ss]]/
Log[Norm[xNM[[k]] - ss]/Norm[xNM[[k - 1]] - ss]], {k, 2, Length[xNM] - 1}];

oiNM = Table[
Log[Norm[xiNM[[k + 1]] - ss]/Norm[xiNM[[k]] - ss]]/
Log[Norm[xiNM[[k]] - ss]/Norm[xiNM[[k - 1]] - ss]], {k, 2, Length[xiNM] - 1}];

othNM = Table[
Log[Norm[xthNM[[k + 1]] - ss]/Norm[xthNM[[k]] - ss]]/
Log[Norm[xthNM[[k]] - ss]/Norm[xthNM[[k - 1]] - ss]], {k, 2, Length[xthNM] - 1}];

(* Combine results into one array *)
summrray = Join[
nm[[1 ;; 3]], {oNM[[-1]]},
inm[[1 ;; 3]], {oiNM[[-1]]},
thnm[[1 ;; 3]], {othNM[[-1]]}
];

(* Return summary as table *)

```

```
    summrray // TableForm  
];
```

```
(* Example Usage *)
```

```
opt = {100, 10^-6};
```

```
MainFunction[F, J, x00, opt, vars]
```